A Signaling Flow Based Traffic Model for SIP Messages in IP Multimedia Subsystem (IMS)

by

Jie Xiao

B.A. Sc

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering (OCIECE)

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario, Canada, K1S 5B6

September 2009

Your file Votre référence
ISBN: 978-0-494-60240-9
Our file Notre référence
ISBN: 978-0-494-60240-9

Abstract

This thesis aims to address the problem of how to develop an efficient traffic model that allows the prediction of the Session Initiation Protocol (SIP) server load in IP Multimedia Subsystem (IMS). By utilizing the characteristics of SIP messages, a signaling flow based approach to the traffic model is proposed. This model is based on an in-depth signaling flow analysis of a number of SIP session procedures defined in IMS and on the quantification of the SIP signaling traffic at the signaling flow level. The proposed traffic model allows the load of servers to be predicted with a simple mathematical calculation. Moreover, we develop a measurements architecture based on Event State Publication (ESP) and Event Notification Framework (ENF) in order to collect the network statistics and user behavior. According to the simulations that we carried out using OPNET, the model we proposed is shown to be acceptable. IMS network deployment cost optimization issue is formulated as a linear programming problem with various mapping strategies.

To my parents

献给我最爱的父母

Acknowledgements

I would like to thank my thesis supervisor, Professor C. Huang, and co-supervisor, Professor James Yan for their guidance and encouragement throughout the period of my study. This work was supported in part by NSERC Grant CRDPJ 354729-07, NSERC Grant RGP 261469-2003, OCE Grant CA-ST-150764-8, and Nortel Networks.

Table of Contents

List of Tables

List of Figures

List of Acronyms

3GPP	Third Generation Partnership Project
AAA	Authentication, Authorization and Accounting
CN	Core Network
CSCF	Call/Session Control Function
ENF	Event Notification Framework
EPA	Event Publication Agent
ES	Event Server
ESP	Event State Publication
GPRS	General Packet Radio Service
HSS	Home Subscriber Server
I-CSCF	Interrogating-Call/Session Control Function
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
ISC	IP Multimedia Service Control
MAA	Multimedia-Auth-Answer
MAR	Multimedia-Auth-Request
P-CSCF	Proxy-Call/Session Control Function
QoS	Quality of Service

S-CSCF	Serving-Call/Session Control Function
SAA	Server-Assignment-Answer
SAR	Server-Assignment-Request
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
UAA	User-Authorization-Answer
UAR	User-Authorization-Request
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UA	User Agent
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
VoIP	Voice over IP

Mathematical Notations

Math notations using the English alphabet

A The distribution of session procedures.

B_r The frequency of routing scenario r.

c_y The capacity of physical node y (message per unit time).

c_{vy_v} The capacity of physical node y that hosts logical server v (message per unit time).

c_{gy_g} The capacity of physical node y in group g (message per unit time).

D_v The average queuing delay at logical server v.

f Signaling flow, $f = 1,2,\cdots,17$.

g Group of physical nodes, $g = 1,2,...,G$.

G The number of groups in the physical network.

h_f Signaling flow demand volume for signaling flow f (message per unit time).

H A row vector represents the total traffic volume of the signaling flows for six routing scenarios (message per unit time).

I A matrix identifies the relationship between the signaling flows and the SIP servers.

j The jth time point, $j = 1,2,\cdots,J$.

J The number of points picked from the simulation.

k_y The load of physical node y (message per unit time).

$\mathbf{L_r}$ A row vector represents the load carried by each server for routing scenario r (message per unit time).

$\mathbf{M_r}$ A matrix represents the load carried by each server as generated per session procedure for routing scenario r (message per unit time).

N Average arrival rate of session procedures (session procedure/second).

N_1 Average arrival rate of users (user/second).

N_2 Average number of applications requested by one user (application/user).

N_3 Average number of session procedures triggered by one application (session procedure/application).

p Candidate physical path, $p = 1,2,\cdots,P_f$.

P_f The number of candidate physical paths for signaling flow f.

Q_j The number of applications requested by users at time point j.

r Routing scenario, $r = 1,2,3,4,5,6$.

s Session procedure, $s = 1,2,\cdots,20$.

$\mathbf{T_r}$ A row vector represents the average arrival rate of the session procedures for routing scenario r (session procedure per unit time).

U_{rj} The number of users entered to the system in routing scenario r at time point j.

U_j The number of users entered to the system at time point j.

v Logical server, $v = 1,2,3,4$.

w_{fp} Load allocated to physical path p that is one available physical path of signaling flow f (message per unit time).

w_{fvy_v} The loads allocated to physical node y of signaling flow f for logical server v.

w_{fgy_g} The loads of signaling flow f allocated to physical node y in group g.

$\mathbf{X_r}$ A matrix represents the volumes of the signaling flows per session procedure in routing scenario r (message per unit time).

y Physical node, $y = 1,2,...,Y$.

y_g 1, 2, ... Y_g, physical node located in group g.

Y Total number of physical nodes.

Z_j The number of the session procedures triggered at time point j.

Z_{sj} The number of session procedure s triggered at time point j.

Math notations using Greek alphabet

α_{fv} =1, if signaling flow f traverses logical server v; =0, otherwise.

α_{fg} =1, if signaling flow f traverses group g.

α_{fpyv} =1 if physical node y hosts logical server v along physical path p that is one available physical path of signaling flow f; =0, otherwise.

β_{fv} The number of times that logical server v is involved in signaling flow f.

κ_v The capacity coefficient in time-capacity product unit for logical server v

λ_v The v^{th} element of the row vector $\mathbf{\Pi}$ (message per unit time)

ρ_v The utilization of logical server v.

ε_y The cost coefficient per unit processing capacity for physical node y.

ε_{vy} The cost coefficient per unit processing capacity for physical node y that hosts logical server v.

ε_{gy} The cost coefficient per unit processing capacity for physical node y in group g.

δ_{fvg} =1, if signaling flow f traverses logical server v mapped to group g; =0, otherwise.

μ_v The mean service rate of logical server v (message per unit time).

Л A row vector denotes the total load of each server (message per unit time).

Chapter 1
Introduction

This chapter briefly introduces IP Multimedia Subsystem (IMS), discusses the relevant network entities in IMS, and describes Session Initiation Protocol (SIP). Then, it shows SIP performance is critical to the service's quality of experience. An efficient yet representative model for SIP signaling traffic is needed for conducting a SIP performance evaluation. Since in the public domain, there is no such reference that can provide an efficient traffic model for IMS system. This motivates us to develop an efficient traffic model that allows us to predict server load for IMS system by utilizing the characteristics of SIP messages that we found. We propose a technique to capture such characteristics, so that the traffic model can be built. Moreover, at the end of this chapter, thesis contributions and thesis organization are provided, and our publication is listed.

1.1 Background

IMS is envisioned as the next generation IP-based multimedia communication system that integrates data, speech, and video network technology and covers wireless and wireline networks. The IMS [1][2], as a new core network domain, was first introduced by the Third Generation Partnership Project (3GPP) in two phases (release 5 and release

6) [3] for Universal Mobile Telecommunications System (UMTS) networks. 3GPP2 further defined an IP multimedia framework, which finally harmonized with the IMS.

Figure 1.1 gives a high level view of IMS core network architecture. The main elements are the Call/Session Control Function (CSCF) servers and Home Subscriber Server (HSS). CSCF servers are SIP servers. Based on IP technology, IMS provides a multimedia session control service that allows mobile users to access new multimedia and multisession applications as well as to establish synchronous multimedia sessions across fixed and mobile terminals [1][4][5]. Since the service creation interfaces are standardized by IMS, they allow for the development of new multimedia and multi-session applications. IMS offers this session control to the applications by SIP [6][7].

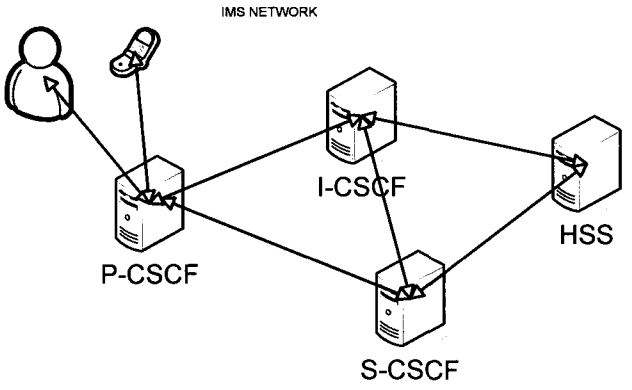


Figure 1.1: A high level view of IMS core network architecture

SIP, as an application layer control protocol, is defined by Internet Engineering Task Force (IETF) [8]. SIP lies at the core of IMS architecture and plays the role of session establishment, modification, and termination between two or more end users. Moreover, SIP is independent of the media being transported and relies on other protocol for

transporting the media data of the session. The SIP messages are the data exchanged between the IMS network elements or between the User Agent (UA) and the network elements [7]. Details of the functionalities and roles of IMS network elements are discussed in Chapter 2.

Additionally, in order to perform various signaling functions, IMS defines a large number of procedures in [9]. The procedures are classified into five different categories, which can be further divided into several cases called session procedures. Each session procedure specifies the sequence of SIP messages passing among the involved servers, how the servers should process the received messages, and what information should be stored for the users.

Hence, in an IMS network, a number of session procedures are triggered by users requesting the available applications or services. It turns out that the SIP messages are traversing among SIP servers as defined, namely, SIP signaling traffic, and SIP servers keep processing the arrived messages for all the users. More details about this are provided in Chapter 2.

However, in IMS network, there exists a problem when SIP servers encounter an overload situation, wherein the traffic load exceeds SIP servers' capacities. Next, we will address this problem in detail.

1.2 Research Problem

In addition to media traffic, signaling traffic is considered to be another important type of network traffic. End users may experience a delay due to signaling network congestion. However, nowadays, network bandwidths are large enough for signaling traffic, which

may be composed of several SIP messages for completing particular signaling purpose; therefore, bandwidth is not considered the bottleneck of performance. By contrast, processing capacity is more likely to be the bottleneck. In IMS network, when the network entities, say SIP servers, have insufficient resources to complete the processing of a SIP message, overload occurs [10]. The resources can include all of the capabilities of the element used to process the messages, including CPU processing, memory, Input or Output (I/O), or disk resources, so called server capacity. Overload can occur for many reasons [10]. Below, we discuss five of these reasons in detail.

1. The first reason is poor capacity planning on SIP servers. SIP servers in IMS network need to be designed with sufficient capacity in order to handle the expected load based on the expected number of users and their behavior in IMS. Capacity planning is the process of determining the expected load with the given user profile. However, if this work is not done properly, the server may have insufficient capacity to handle the actual load, and this causes a server overload. Thus, in order to avoid a server overload, it is essential to have a model that can efficiently predict the load for better server capacity planning.
2. The second reason is component failure. In IMS network, a cluster of SIP servers can be implemented for sharing the load of traffic. When one or more members in the cluster fail, the remaining members undertake the work of the failed member(s). Usually, each member is designed to have enough spare capacity to handle the failure of another member. However, unusual failure condition can cause the cluster to fail simultaneously. In this case, the network system is required to distribute the

load of the failed cluster that can be predicted by a traffic model to other clusters. Meanwhile, network system notifies the network providers to fix the problem.

3. The third reason is one of the most troubling reasons of overload. It is avalanche restart, which happens when a large number of users attempt to connect to IMS network with a registration procedure at the same time. In the case of the failure of a large network connection, the users can detect the failure rapidly. When the network connectivity is recovered and detected, all the uses re-register within a short time period; this then causes the avalanche restart, which subsequently leads to the overload on SIP servers.
4. The forth reason is dependency failures on SIP servers. There are some dependent resources on a SIP server, such as CPU processing, disk resources, and so on, and when one resource has failure, the dependent resource may fail, and in this way, the SIP server becomes overloaded.
5. The last reason is flash crowds, which describes a case of an extremely large number of users attempt to make a call simultaneously. This can cause SIP servers to go into overload.

Any of above reasons can cause SIP server overload, and this overload can cause the failure of other SIP servers, which are trying to process the traffic load. It also brings even more load onto the remaining SIP servers. Moreover, SIP server overload causes SIP messages to be delayed or lost, which causes retransmission to be sent. It then makes more work in the network. This produces the amount of traffic to SIP servers, which makes the network performance even worse.

Furthermore, while the number of users and their demands on new applications are increasing, network providers also need a model to predict the impact of new applications on SIP servers when the new applications are introduced to market. A new application may trigger a different set of session procedures and can change the existing signaling traffic. By having such a traffic model, network providers can predict the network availability as well as maintain the stability of the system, and thereby, increasing their revenue potential. As a result, establishing a SIP signaling traffic model is necessary for performance evaluation.

IMS system is extremely complex due to the combination of various call scenarios and the large number of procedures defined in [11] for various signaling functions. Each combination has to be analyzed individually in order for the server load to be obtained accurately. This requires both complex and time-consuming work. Thus, it is a challenging task to engineer all of the kinds of IMS servers needed in order to satisfy the real-time call performance requirements.

To our best knowledge, in the public domain, there is no such reference that can provide an efficient SIP traffic model for IMS system, which can predict the load of important IMS servers. An obvious way to characterize the load for servers on each session procedure is to simply count the number of messages processed on each server for a given session procedure, and then sum up the total number of messages for the set of session procedures involved. Eventually, the server load can be predicted. However, this approach does not capture the causal relationship among a sequence of messages in the session procedures. In a session procedure, one message typically triggers a sequence

of messages to complete a transaction, so that the load at their corresponding servers is correlated. An increase on server load may bring the increases onto the next set of servers. Therefore, we need to capture this correlation in order to develop a new traffic model that can predict the server load.

Thus, the problem we encounter is to develop a technique or a method that is better at presenting such a causal relationship among the SIP messages, and then utilizing the chosen technique in order to develop an efficient traffic model that allows for the prediction of server load for IMS system. The model will be built based on the signaling traffic analysis performed by the chosen technique. The formulated model not only correctly predict the server load for the identified session procedures, but it will also have the capability to adapt to any unidentified procedures defined, or to be defined, in IMS system. This model must ensure its validity or accuracy as well as its flexibility. Next, we propose a signaling flow based traffic model.

1.3 Our Approach

In order to solve this problem, we introduce the signaling flow approach. A signaling flow is the aggregation of a sequence of messages that follows the same path in a network of IMS servers. This signaling flow concept tries to capture the causal relationship among a sequence of messages so that the load at different servers can be correlated as they occur in the real network. Then, we perform a signaling flow analysis on a number of SIP session procedures and quantify the SIP signaling traffic at signaling flow level. This signaling flow based approach maps various combinations of routing scenarios and session procedures to the limited number of signaling flows. By utilizing the signaling

flow analysis, a model is built so that the load of servers can be predicted with a simple mathematical calculation. The complex correlation structure of the load across different servers is naturally captured by the signaling flow concept we introduce.

Data flow based modeling has been widely used in data plane traffic characterization [12]; however, to our best knowledge, there are no publications on modeling SIP signaling traffic using a flow based approach. Signaling flow based modeling for SIP traffic is quite different from data flow based modeling of data plane traffic, given that it is heavily dependent on signaling procedures. A popular definition for data flow adopted for data plane traffic is defined in [13] as a unidirectional packet stream with a unique *<source-IP-address, source-port, destination-IP-address, destination-port, IP-Protocol>* tuple. The data plane traffic transported by the protocol among the end users can be decomposed based on source, destination, application, and/or transport protocol, depending on the definition of the data flow adopted. On the other hand, the proposed signaling flow is extracted from the session procedures and is defined to be the aggregation of a sequence of messages that follows the same path. The signaling flow path identifies the servers that a specific sequence of messages will traverse as well as the order that these servers will be traversed; therefore, the signaling flow based modeling for signaling traffic is entirely dependent on signaling procedures. Section 4.1.4 provides more details about this.

In this thesis, the five different categories of procedures are selected and each category can be further divided into several cases; in total there are 20 session procedures considered. Then, we define six routing scenarios, based on the concept of call scenarios,

for minimizing the complexity of the IMS signaling traffic. The SIP signaling traffic involved in the selected session procedures is analyzed and quantified by the signaling flow based approach. By utilizing the signaling flow analysis, the load of servers and the mean server queuing delay can be predicted using a simple mathematical calculation. The complex correlation structure of the load across different signaling servers is naturally captured by the signaling flow concept that we introduce. This model also allows for flexibility when expanding the SIP session procedures in IMS network. Lastly, we set up the simulations that we carried out using OPNET, and the model we proposed is proven to be acceptable when the calculated results fall in 95% confidence interval of the simulation results.

In addition, we develop a measurements architecture based on Event State Publication (ESP) and Event Notification Framework (ENF) in order to collect the network statistics and user behavior. This architecture is established upon IMS network and relies on SIP messages in order to collect and subscribe the information that is of interest to the network providers. According to a list of user data collected from the network, the input data of the proposed traffic model can be obtained using a set of mathematical calculations.

Furthermore, we formulate IMS network deployment cost optimization issue as a linear programming problem by utilizing the signaling flow based approach. The cost optimization involves mapping a logical IMS core network topology into a physical network topology. Various mapping strategies are discussed and represented with the proper mathematical formulation so as to take advantage of the signaling flow based

traffic model. One example for each mapping strategy is provided. In the following section, a list of contributions we made is provided.

1.4 Summary of Contributions

The main contributions of this thesis are as follows:

1. Defined a signaling flow based approach that captures the causal relationship among a sequence of SIP messages, so that the corresponding server load can be correlated. This approach maps various combinations of routing scenarios and session procedures to a limited number of signaling flows, so that the load of each IMS server can be easily estimated.
2. Developed a measurements architecture based on ESP and ENF frameworks so as to collect the network statistics and user profile. This architecture is integrated with the IMS network, and it relies on SIP messages in order to collect and subscribe the information that is relevant to the network providers. The input data of our proposed traffic model can be obtained from the collected data by means of a set of mathematical calculations.
3. Formulated IMS network deployment cost optimization issue as a linear programming problem. Created three potential mapping strategies, in which the logical IMS network is mapped into the physical IMS network.

1.5 Thesis Organization

The remaining chapters of this thesis are organized as followed:

Chapter 2: Introduce the background knowledge for the material presented in this thesis. Convey the literature review on the current IMS traffic modeling technology and IMS network implementation.

Chapter 3: Define call scenarios and routing scenarios.

Chapter 4: Analyze SIP signaling traffic by applying the signaling flow based approach and formulate our traffic model. Furthermore, develop a measurements architecture based on ESP and ENF in order to collect the network statistics and user behavior.

Chapter 5: Introduce the implementation of an IMS network in OPNET and collect the results from the simulation. Also, explain the design and implementation of a source model and verify the proposed traffic model.

Chapter 6: Describe the three mapping strategies and formulate IMS network deployment cost optimization problem for each mapping strategy into a linear programming problem.

Chapter 7: Present the conclusion and recommendations for future research.

1.6 Publication

J. Xiao, C. Huang, J. Yan, "A Flow-based Traffic Model for SIP Messages in IMS," *IEEE GLOBECOM*, Hawaii USA, November 2009.

Chapter 2

Background Knowledge

This chapter provides the background information for the material presented in this thesis. It reviews the IP Multimedia Subsystem (IMS) and surveys the existing related work on modeling SIP traffic in IMS. First, the evolution of IMS is introduced, followed by a discussion on the protocols defined in IMS architecture, Session Initiation Protocol (SIP), and Diameter Protocol. Furthermore, a simplified IMS core network architecture is studied, and the involved network entities are discussed individually. Next, the five general categories of procedures defined in IMS are summarized. The diagram for each session procedure is also illustrated and used to demonstrate the causal relationship among the sequence of messages. Lastly, the related work on the SIP signaling traffic model and the IMS network implementation are provided.

2.1 Review of IP Multimedia Subsystem (IMS)

In this section, we will provide an introduction to IMS, including IMS protocols, IMS core network architecture, and IMS session procedures.

2.1.1 Introduction to IMS

In the last decade, the Internet has experienced a dramatic growth due to a rapid increase of services development and the ability of providing these services to millions of users. The services are multimedia and include video, audio, and text. Mobile users are able to browse the web or other Internet activities via data connections. Third Generation (3G) network have now been implemented in order to merge the Internet and the cellular network as well as to support high-speed IP-based data, voice, and multimedia services [1]. The 3G wireless network [2] defined by the 3GPP are developed on the basis of the global system for mobile communication (GSM) and General Packet Radio Service (GPRS).

IMS [1][3] is a key element in the 3G architecture. IMS is designed to act as a common platform in order to develop diverse multimedia services [1][12]. As a standard framework for the deployment of next generation IP-based application services, IMS defines how these services connect and communicate with underlying telecommunications network, how end users are able to set up the multiple services in a single session or multiple synchronized sessions, and how these services integrate with the network provider's end systems. IMS was initially introduced by the 3GPP in two phases (release 5 and release 6) [14] for UMTS network. 3GPP2 further defined an IP multimedia framework, which finally harmonized with current version of IMS.

IMS is an IP-based architecture that provides a multimedia session control service that allows mobile and fixed users to access new multimedia and multisession services as well as to establish synchronous multimedia sessions across fixed and mobile terminals

[1][4][5]. Additionally, with a layered design that separates transport and signaling services, IMS supports the negotiation of next generation multimedia services between end users, and provides the integration of functionalities, such as security, Quality of Service (QoS) control and charging [15]. Since the service creation interfaces are standardized by IMS, this allows for the development of new multimedia and multi-session services. Next, we will introduce IMS protocols.

2.1.2 IMS Protocols

The architecture of IMS utilizes three general categories of protocols [16], which are listed as follows:

1. Protocols used in the signaling or session control plane (e.g. SIP).
2. Protocols for security and authentication (e.g. Diameter)
3. Protocols for carrying the media data in the media plane

In our thesis, we are concerned with the first two of the listed categories.

SIP

The protocol chosen by the 3GPP for session control in IMS is SIP. SIP is defined in [6] [7] [17] as an IETF standard. SIP is an application layer control protocol that lies at the core of the IMS architecture, and it plays the role of session initiation, modification, and termination between two or more end points [18][19]. Session initiation includes the discovery of the user's location and the request of initiating a session by sending an INVITE request. Once the user has initiated a session successfully, the session is said to have been established, thus allowing the media stream to directly travel among two or more end points. After that, the user can modify the parameters regarding an existing

session, such as modifying the current QoS (e.g., the new codec if local policies are applied). Finally, SIP deals with the session termination, which releases the session. Additionally, SIP is independent of the media being transported and relies on other protocols for the transporting of the media data of a session. SIP is based on a request/response model. The requests can be grouped altogether with the corresponding responses into transactions. The SIP messages are the data sent between SIP elements as part of the protocol [7]. The body of a SIP message includes a description of the session, which is encoded in some other protocol format. One of these formats is Session Description Protocol (SDP). SIP messages can be carried by User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) [20]. When SIP chooses TCP as a transport protocol, the transport layer provides reliability. However, when SIP is carried by UDP, reliable delivery procedures are ensured by SIP. Also, UDP is the widely used as SIP transport protocol.

SIP specifies a number of various methods [7]. Appendix A summarizes the methods of SIP that are involved in the analysis of SIP signaling traffic with high frequency.

Diameter

Diameter as security and authentication protocol in the IMS architecture is defined in [21]. It improved an earlier version of the authentication protocol called RADIUS [22]. Diameter messages are carried by reliable transport protocol, such as TCP and Stream Control Transmission Protocol (SCTP).

Now that, we know both SIP and Diameter protocols are used in IMS architecture, we will introduce IMS architecture and IMS network entities.

2.1.3 Overview of IMS Architecture

As illustrated in Figure 2.1, the IMS architecture is composed of three layers [16]. The Application Layer is comprised of applications servers, which provide the end user services, and the Call/Session Control Layer contains the Call/Session Control Function (CSCF) that acts as a central system component of the IMS network infrastructure, and Home Subscriber Server (HSS). This layer mainly provides the routing of the SIP messages to the appropriate application server. HSS database maintains the unique service profile for each user, and other interworking functions. Moreover, the Media Transport and Endpoint Layer allows the end users to initiate and terminate SIP signaling and to set up sessions; it also allows for the exchange of media data among the end users and servers that provide the services.

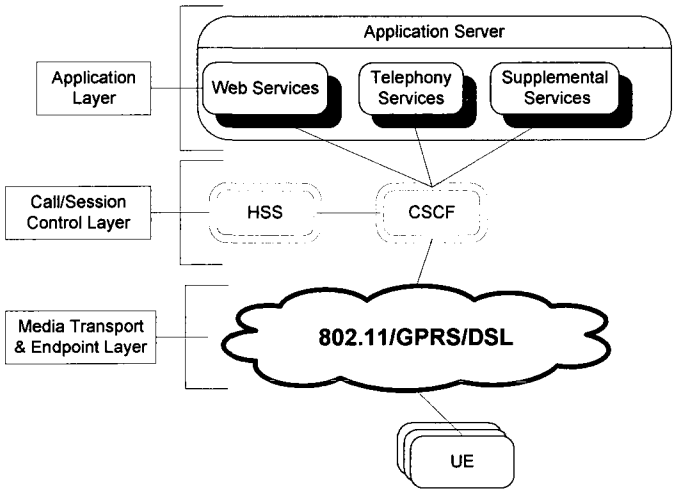


Figure 2.1: IMS layer view: network components and functions per layer

As illustrated in Figure 2.1, both CSCF and HSS are important components in the Call/Session Control layer of the IMS architecture. CSCF provides the session control services including subscription, registration, routing and roaming, and central services, based on charging and QoS control [23]. Once CSCF receives a new SIP call from end users, it contacts the HSS in order to authenticate the user. Once the successful authentication in HSS is completed, the SIP signaling is passed by CSCF over the IP Multimedia Service Control (ISC) interface to the application server. The ISC defines a set of filters that can be obtained from the HSS and assigned to each user. Thus, CSCF decides which application server should provide the requested services by the comparison of each SIP message and the filters associated to the corresponding user in the ISC.

A simplified architecture of IMS network with the network entities is illustrated in Figure 2.2. Depending on the functionalities they provide, CSCF servers are classified into three types: the Proxy-, Interrogating-, and Serving-CSCF servers (P-CSCF, I-CSCF, S-CSCF, respectively [24]). Furthermore, as mentioned previously, SIP, as a signaling control protocol, lies at the core of the IMS architecture and plays the role of session establishment, modification, and termination between two or more end users; this occurs mostly between end user and P-CSCF server or between the two CSCF servers [24]. Besides the SIP, Diameter protocol is used by HSS in order to communicate with the CSCF servers for Authentication, Authorization and Accounting (AAA) dialog. As shown in Figure 2.2, P-CSCF, as a proxy server, is located in the access network, while I-CSCF, S-CSCF, and HSS servers are located in the home network.

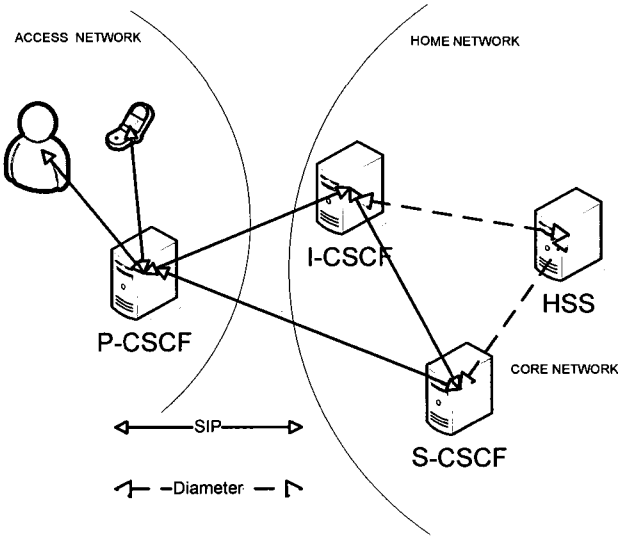


Figure 2.2: A simplified IMS core network architecture

P-CSCF

P-CSCF is placed between the end user and IMS network as an outbound/inbound SIP proxy server for the first core network service access [25][26]. It implies that all the requests initiated by or destined to an IMS end user have to traverse P-CSCF. P-CSCF processes the received SIP messages and forwards them to the next appropriate network node. During the registration, the end user only keeps in contact with one single P-CSCF server. Also, P-CSCF is responsible for authenticating the user, and it asserts the identity of the user to the rest of the nodes within the network. By doing so, the repetitive authentication work is avoided in the rest of the nodes. In addition, P-CSCF supports the verification of SIP requests sent by the IMS end user. This ensures that the SIP requests are guaranteed to follow the SIP rules. P-CSCF also includes a compressor and a decompressor of SIP messages. This helps prevent the long transmission time of SIP messages due to their large size. Usually, an IMS network includes a number of P-CSCFs

for the sake of scalability and redundancy. Each P-CSCF serves a number of IMS end users, depending on the capacity of the node.

P-CSCF Location

Depending on the user's location, P-CSCF is located either in the visited network or in the home network. In other words, P-CSCF is the first contact point for the user and is located in the same network where the user is located.

I-CSCF

I-CSCF, acting as a SIP proxy located at the edge of the administrative domain and performs the routing function, is the contact point for external networks [24][25]. It forwards SIP messages to the network based on the decision of whether access from other networks is granted. Then, I-CSCF is required to hide the network details from the other operators as well as to determine the routing. This facilitates the protection of S-CSCF and HSS servers from unauthorized access from other network. Thus, I-CSCF server behaves like the gateway into each individual network. Both P-CSCF and I-CSCF perform the gateway functionality; however, they are in fact different. P-CSCF is a contact point to reach IMS network and acts as a gateway to non-IMS network, while I-CSCF acts a gateway between two IMS networks. Besides, I-CSCF has another important function, which is the assignment of S-CSCF. In general, S-CSCF is assigned according its capability or service provider policy. Therefore, I-CSCF plays an important role in both security as well as the routing of SIP messages between the different IMS networks. An IMS network usually includes a number of I-CSCFs for the sake of scalability and redundancy.

I-CSCF Location

I-CSCF is usually located in the home network, except in some especial cases. However, in our research, we assume that I-CSCF is always located in the user's home network.

S-CSCF

As a SIP server, S-CSCF server is the central node of the signaling plane [1][24]. It always resides in its user's home network with the responsibilities of both the registration and session control for the users. It acts as a SIP registrar and is responsible for the binding between the public user identity (the user's SIP address) and the user's location (e.g., the current IP address that the user is logged on). The entire SIP messages initiated by (or destined to) IMS users traverse the allocated S-CSCF. S-CSCF inspects the SIP messages in order to decide the next destination based on the contents of the messages. Moreover, one of the main functions of S-CSCF is the provision of the SIP routing services (e.g., translation services) [26]. S-CSCF interacts with the HSS by using the Diameter protocol in order to obtain the authentication information of the user who is trying to access the IMS, to download the user profile, and to inform HSS that this S-CSCF is currently assigned to the user for the duration of the registration [27]. Usually, an IMS network includes a number of S-CSCFs for the sake of scalability and redundancy. Each S-CSCF serves a number of IMS end users depending on the capacity of the node.

S-CSCF Location

S-CSCF is always located in the user's home network.

HSS

HSS is a central database containing user-related information and serves S-CSCF [27]. It stores user profiles, such as the various identities (the private identity and all public user identities), the available applications a user is allowed to access, the network a user is allowed to roam within, and the location of the user device. However, the information stored in HSS is not available to other networks. This means that only S-CSCF within the same network has the permission to access HSS within any network, thus securing the user data within the home network. Besides, as we discussed earlier, P-CSCF and I-CSCF guard S-CSCF and HSS from unauthorized access.

HSS location

In general, the location of HSS is not specified in the references. However, we assume HSS is always located in the user's home network. Next, we will discuss the session procedures that we focus on in this thesis.

2.1.4 IMS Session Procedures and SIP Signaling Traffic

Before introducing IMS session procedures, several definitions provided as follows, for better understanding the following texts.

1. Procedure: Procedure is accomplished by a set of SIP messages that pass back and forth among the involved servers in order to complete a specific signaling purpose. There are five different categories of procedures; each category is further divided into several cases called session procedures; Section 2.1.4 provides more information of session procedures. Figure 2.3 illustrates two session procedures, which are basic session setup procedure and termination procedure (mobile terminal

initiated session release). The basic session setup procedure starts at User Equipment 1 (UE1) by sending the INVITE message to UE2, via P-CSCF and S-CSCF, and it ends at UE2 when UE2 receives the ACK message sent from UE1, via P-CSCF and S-CSCF. When the session is established, the multimedia data passes between two users through a SIP session. The termination procedure is initiated at UE1 by sending the BYE message to UE2, via P-CSCF and S-CSCF, and it ends at UE1 when UE1 receives the 200 message from UE2, via S-CSCF and P-CSCF.

2. Transaction: Transaction presents an interaction between nodes through a series of dependent message exchanges. A transaction can occur between a user and a server, between users, or between servers and it comprises the messages including the request sent from the sender to the receiver and a response sent from the receiver to the sender. A transaction can be established by sending a request and be terminated by receiving the response to previous request. As shown in Figure 2.3, there are five transactions formed and exist in the shown session procedures. The first transaction occurs between UE1 and P-CSCF. UE1 sends an invite message to P-CSCF, then P-CSCF responses with a 100 message. The second transaction involves the same messages, but it happens between P-CSCF and S-CSCF servers. The third transaction happens between S-CSCF and UE2, and uses the same type of messages as transaction 1 and 2. The forth transaction occurs between UE2 and UE1. UE2 sends a 200 message to UE1 via S-CSCF, and P-CSCF, and UE1 responses with an ACK message using the same path. The fifth transaction starts from UE1, with a BYE message. The BYE message travels to UE2 through P-CSCF and S-CSCF.

Once UE2 receives the BYE message, it response with a 200 message via S-CSCF and P-CSCF. The transaction ends when UE1 receives the 200 message.

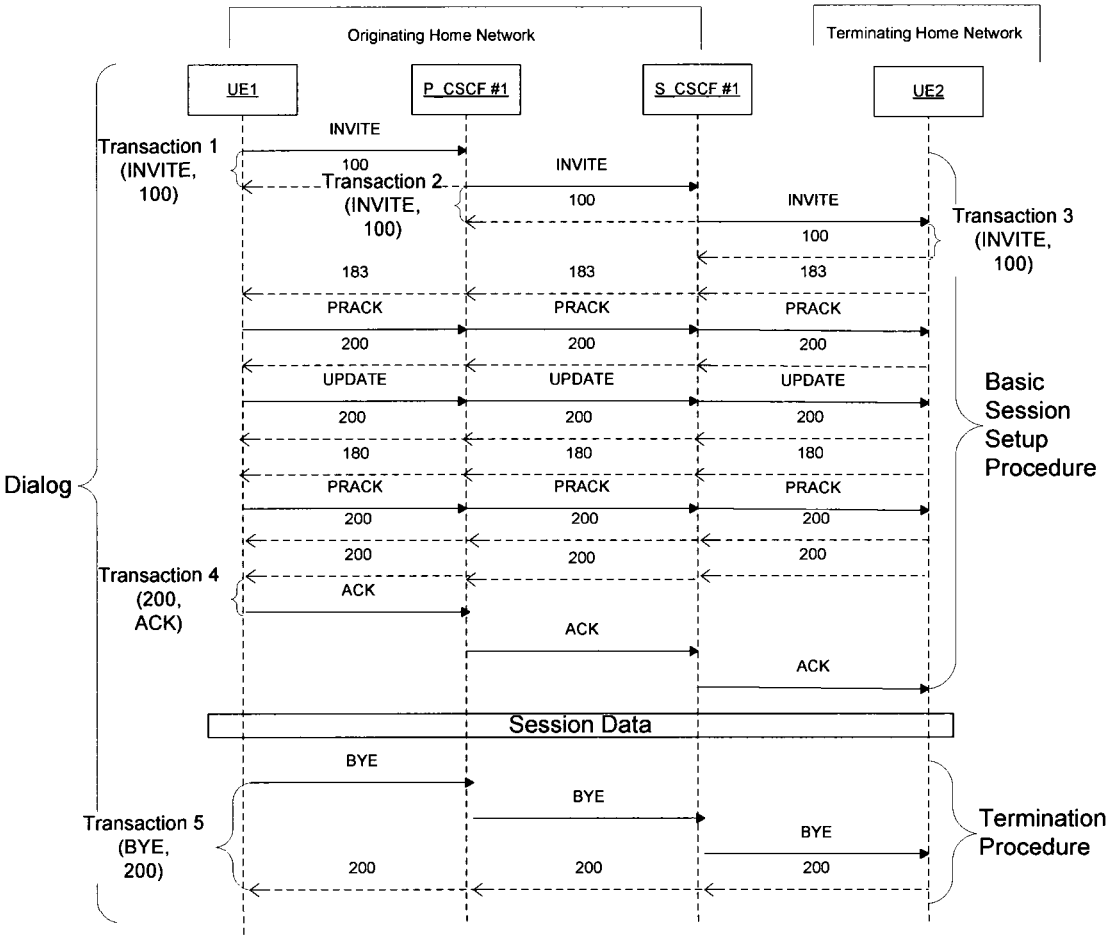


Figure 2.3: Basic session setup procedure and termination procedure (mobile terminal initiated session release), with transaction, dialog analysis

3. Dialog: Dialog represents a peer-to-peer SIP relationship between two users that persists for some time [11]. In a dialog, two users establish a conversation in which

a sequence of messages passes between two users. When the conversation is over, the relationship between the two users is released, and the dialog is then terminated. A dialog can contain one or more transaction(s), since in a dialog there are many requests and the responses to them. The user can initiate new transactions within a dialog. As shown in Figure 2.3, two end users, namely UE1 and UE2, remain in a communication dialog from the time the session is established until the time of termination, including the multimedia data passing between the two end users. There are five transactions in this dialog.

4. Routing Scenario: Routing scenario specifies the status of end users when they enter into an IMS network. For example, as shown in Figure 2.3, the routing scenario of UE1 specifies UE1 as an originator in his home network, and the routing scenario of UE2 specifies UE2 as a terminator in his home network. Section 3.2 provides more information on the routing scenario.

IMS defines a large number of procedures for the various signaling functions in [9]. Generally speaking, there are five different categories of procedures [14], which include registration and de-registration, session initiation, session termination, session failure, and session redirection. As illustrated in Table 2.1, each category is further divided into several cases, which are called session procedures. In the last column of the table, a diagram and brief explanations of some session procedures are provided in the corresponding appendix.

Table 2.1: IMS session procedures

Procedure Category	No.	Session Procedures	Appendix
Session Initiation	1	Basic Session Setup	*Appendix C.1*
	2	Re-invite for new codec, without I-CSCF	*Appendix C.2*
	3	Re-invite for reserved codec	*Appendix C.3*
	4	Re-invite, failure happen	*Appendix C.4*
Registration	5	Registration, user not registered	*Appendix B.1*
	6	Re-registration, user currently registered	*Appendix B.2*
De-registration	7	Mobile initiated	*Appendix B.3*
	8	Network initiated, registration timeout	*Appendix B.4*
	9	Network initiated by HSS, administration	*Appendix B.5*
	10	Network initiated, service platform	*Appendix B.6*
Session Termination	11	Mobile terminal initiated session release	*Appendix D.1*
	12	Network initiated session release P-CSCF initiated	*Appendix D.2*
Session Failure	13	Failure in session abandon, origination procedure	*Appendix E.1*
	14	Failure in obtaining resource, origination procedure	*Appendix E.2*
	15	Failure in termination procedure	*Appendix E.3*
	16	Rejection by termination procedure	*Appendix E.4*
Session Redirection	17	Initiated by S-CSCF to CS-domain	*Appendix F.1*
	18	Initiated by S-CSCF to IM CN subsystem	*Appendix F.2*
	19	Initiated by P-CSCF	*Appendix F.3*
	20	Initiated by UE	*Appendix F.4*

As a starting point, both new and old end users are required to register in IMS network before any session is initiated. After a successful registration, the session initiation can be performed by the users' requests. Once the basic session has been completely set up, the media traffic can travel between the end users. The de-registration procedure may be triggered by mobiles or even the network under some situations. A session initiated by the user may fail due to an error detected in the servers; however, the

decision to redirect a session to a different destination may be made for different reasons in the establishment of a session. At the end of a session, the session termination procedure allows the session to be released.

In session failure, the failure occurs in either the origination procedure or the termination procedure. The origination procedure specifies the signaling path between the UE initiating a session setup and S-CSCF that is assigned to perform the session originating service. By contrast, the session termination procedure specifies the signaling path between S-CSCF assigned to perform the session termination service and the UE. The signaling paths for both origination and termination procedures are determined at the time of UE registration and will remain unvarying for the life of the registration [9]. Accordingly, P-CSCF and S-CSCF along this path are fixed as well for one particular user during his life of registration.

The above presented information is all about the IMS system, SIP, IMS session procedures, and IMS signaling traffic. Next, we will review the literature regarding the current IMS traffic modeling technology and IMS network implementation.

2.2 Related Work

Since IMS network is still evolving, the issues related to IMS are a novel research area. Currently, most of IMS-related research work has concentrated on IMS architecture and IMS service platform [5][16][19][23][25], SIP signaling delay and performance study [20][28][29][30][31][32][33], SIP protocol development [15][34], QoS [8][35], and the IMS scalability issue [24][36]. However, in the literature, only a few studies on SIP

signaling traffic model are published. The related work about modeling the signaling traffic and the implementation of IMS network is presented in the following sections.

2.2.1 SIP Signaling Traffic Model

In this section, we discuss some of the solutions in the current literature and explain why there exists a need for alternative solutions.

Abhayawardhana and Babbage [37] proposed an analytical traffic model that focuses on the signaling traffic created to and from HSS in 3 procedures, which are registration procedure, basic session setup procedure and the procedure of presence service in the home network. This paper initially made the assumption for a number of network parameters, which are treated as a representative of all users in a busy hour. Based on the assumed data values, the number of the procedures triggered by users per hour, can be derived at the end. Furthermore, according to the interfaces of Diameter messages defined in the selected procedures, the load of HSS, which is represented as the number of messages processed in HSS, can be calculated. The calculated results have not been verified with simulation. This model can be used to predict the HSS load based on the number of the procedures triggered by a number of subscribers per hour, which is calculated from the assumed network parameters. For comparison, in our proposed traffic model, we define a signaling flow concept that can capture a causal relationship among a sequence of messages. By using this signaling flow concept, the load of SIP servers is correlated and can be predicted by a simple mathematical calculation. The presence service is a service that could be provided in IMS. The procedure of presence service is

not included in our signaling traffic analysis, since it does not belong to the five general categories of procedures that we have classified.

Urrutia-Valdes [38] examines presence and availability services in the context of IMS and describes its architecture defined in 3GPP/3GPP2. It also provides the signaling analysis on the procedure of presence service, where a presentity, a P-CSCF, a S-CSCF, a presence server and a watcher are involved. A presentity is defined as an entity which provides presence information [39]. After that, a user model associated with presence service is proposed for calculating the signaling traffic on the network. An analytical traffic model is built in order to determine the number of SIP messages sent or received for a presentity in a day. The load of P-CSCF and S-CSCF can be derived by the modification of the previous calculation. This paper proposed a traffic model that focuses on the presence service in IMS. It did not consider which network P-CSCF and S-CSCF are located in. It seems that the authors assume that both of them lie in the home network. Moreover, the server load has not been verified with simulation. The following section provides the related work regarding IMS network implementation.

2.2.2 IMS Network Implementation

In addition to above publications, some IMS researches related to the implementation of the test bed of IMS network by the simulator are listed below.

Rajagopal and Devetsikiotis [12] focused on the formulation of queuing models for IMS network, and attempted to characterize the server load by monitoring actual traffic. It defined a methodology to design an IMS network with optimal performance. IMS network is modeled as a tandem queuing network of K servers of type M/M/1/∞. The

service time distribution of the servers follows the exponential distribution. The users enter to the network as a Poisson process.

The authors, Pandey et al of [4] addressed the problem of measuring the end-to-end delay. This paper discussed the implementation of an IMS signaling prototype test bed that consists of two parts, the UA and the IMS core network nodes. UA is a client that generates the IMS signaling traffic using Seagull, which is an open source, multiple traffic generator test tool. The IMS core network is implemented in NS2, which allows for the defining of new protocols as well as the new features that satisfy the IMS network requirements. Moreover, a sample procedure from UA to IMS core nodes is implemented, and the node delay is evaluated after the simulation.

In [40], the authors studied the presence and instant messaging service defined in [39]. They analyzed the signaling traffic in registration and basic session setup procedures. The structure of a simulation model is developed and implemented using C++. The model consists of four main phases: network input and user profile, network initialization, execution of IMS system, and results collection of delay and signaling load.

Furthermore, Panwar and Singh in [41] presented the IMS SIP core server test bed architecture, which is designed to meet the IMS requirements and to be an exact replica of the real IMS network. This model allows IMS clients to minimize the interoperability with the real IMS network as well as to test all service functionalities. In [42], the authors introduce a simulation tool called Diabelli in order to model IMS network.

Hence, we know that there are various simulation tools used in order to implement IMS network, such NS2, C++, and Diabelli. Usually, the IMS server is implemented as a queuing model, M/M/1, or M/D/1.

In our thesis, an OPNET modeler is chosen as our network simulator. The details of implementation are provided in Chapter 5. In the following chapter, the call scenarios and routing scenarios in IMS are defined in order to offer a better understanding of the IMS signaling traffic in different scenarios.

Chapter 3

Define Call Scenarios and Routing Scenarios in IMS

The analysis of different call scenarios is necessary in order to better understand the IMS signaling traffic in different cases. In this section, we define the call scenarios, which identify the different types of calls between two end users. Following the introduction of call scenarios, the routing scenarios are defined and used to reduce the complexity of the analysis of IMS signaling traffic.

3.1 Call Scenarios

An end-to-end session establishment is clearly defined by IMS. It is achieved by successfully establishing a call between two end users. As two parties, the end users can belong to and lie in two identical/different networks [24]. All possible call scenarios between the two parties are shown in Figure 3.1. This figure illustrates four types of parties, H_1, H_2, V_1, and V_2. Both H_1 and V_2 belong to their home network, Network #1, and they are currently located in Network #1 and Network #2, respectively. By contrast, both V_1 and H_2 belong to their home network, Network #2, and they are located in Network #1 and Network #2, respectively. The so-called home network in IMS refers to an infrastructure provided by the user's network operator. The user uses such an

infrastructure when he requires service in the area where he resides. On the other hand, if the user roams outside the area of coverage of his home network, the infrastructure used by him is provided not by his own operator, but rather, by another operator. In this case, the user is a visitor in the network, and this infrastructure is what we call the visited network.

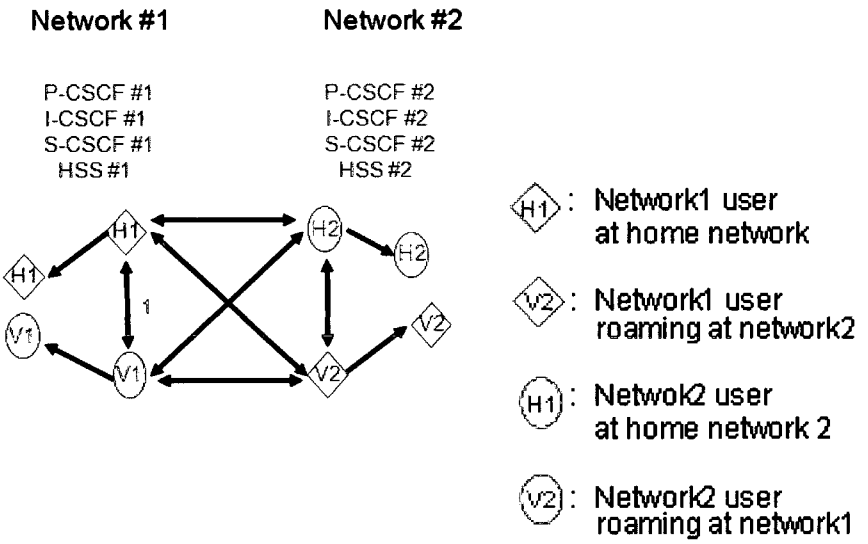


Figure 3.1: End-to-end sessions in 2 networks

Each party, H_1, H_2, V_1, or V_2 acts either as a caller/originator or as a callee/terminator. Any of the parties can call another party in order to establish an end-to-end session. Therefore, as illustrated in Figure 3.1, there are 16 possible call scenarios between these 4 parties. For example, Link 1 represents a complete end-to-end session between a non-roaming user (H_1) and one roaming user (V_1); the two users are subscribers of different network operators and both lie currently in Network #1. Therefore, the possible call scenarios for the session represented by Link 1 are:

1. H_1 user as originator (H_1,O) →Call→ V_1 user as terminator (V_1,T)

2. V_1 user as originator (V_1,O) →Call→ H_1 user as terminator (H_1,T)

The descriptions of 16 call scenario are provided in Appendix G.

Since Network #2 is symmetric, this thesis only focuses on the signaling traffic created to and from Network #1. Moreover, it is noted that the traffic generated from H_2, as originating procedure (H_2, O) and terminating procedure (H_2, T), will be considered unrelated to Network #1 traffic. Then, the 4 parties, which are H_1, V_1, V_2, and H_2, form 15 call scenarios; these scenarios are considered in our analysis as shown in Table 3.1.

Table 3.1: 15 Call scenarios, excluding the case of (H_2, O) calling (H_2, T)

No.	Call Scenarios	Descriptions
1	H1 → H1	(H1,O) calls (H1,T)
2	H1 → V1	(H1,O) calls (V1,T)
3	V1 → H1	(V1,O) calls (H1,T)
4	H1 → H2	(H1,O) calls (H2,T)
5	H2 → H1	(H2,O) calls (H1,T)
6	H1 → V2	(H1,O) calls (V2,T)
7	V2 → H1	(V2,O) calls (H1,T)
8	V1 → V1	(V1,O) calls (V1,T)
9	V1 → H2	(V1,O) calls (H2,T)
10	H2 → V1	(H2,O) calls (V1,T)
11	V1 → V2	(V1,O) calls (V2,T)
12	V2 → V1	(V2,O) calls (V1,T)
13	V2 → V2	(V2,O) calls (V2,T)
14	V2 → H2	(V2,O) calls (H2,T)
15	H2 → V2	(H2,O) calls (V2,T)

Any call scenario can establish an end-to-end session, and we have 20 session procedures identified in Table 2.1. Thus, we have to analyze the signaling traffic involved in the 20 session procedures for the 15 call scenarios individually because they involve

different servers; this is a huge task. As a result, we introduce routing scenarios in an attempt to help us reduce the complexity of the analysis of IMS signaling traffic.

3.2 Routing Scenarios

In this section, we define the routing scenario in IMS network, and determine the server location when a routing scenario and a session procedure are given.

3.2.1 Define Routing Scenario

When a signaling message is routed through the signaling network, the procedures [9], as defined in IMS, can be divided into two parts: the originator part and the terminator part. The procedures related to the originator are called originating procedures while the procedures related to the terminator are called terminating procedures. Therefore, an end-to-end call is a concatenation of the originating procedures and the terminating procedures.

For 15 call scenarios discussed above, originating and terminating procedures consist of (H_1, O), (H_2, O), (V_1, O), (V_2, O), and (H_1, T), (H_2, T), (V_1, T), (V_2, T), respectively. Also, since Network #1 is our concern, the signaling traffic created from (H_2, O) or created to (H_2, T) is considered within Network #2. Therefore, both (H_2, O) and (H_2, T) are excluded from the scope of our research. Consequently, the relevant routing scenarios can be classified into:

- Originating routing scenarios: (H_1, O), (V_1, O), (V_2, O)
- Terminating routing scenarios: (H_1, T), (V_1, T), (V_2, T)

Next, we will determine the server location in the case of a routing scenario that is known.

3.2.2 Determine Servers Locations for a Given Routing Scenario

As mentioned in the previous section, Network #1 performance is our concern. In order to perform the signaling flow analysis within Network #1, we need to identify which servers are logically located in Network #1 and carry the signaling traffic for a given routing scenario. To get this information, we need to analyze each routing scenario separately.

P-CSCF is the first contact point for the user, and it is located in the same network where the user is located. Furthermore, I-CSCF is always assumed to be located in the user's home network, and S-CSCF is always located in the home network. However, since the location of HSS is not specified, we assume HSS is always located in the home network.

Figure 3.2 illustrates, in the case of the Network #1 users as originator (H_1, O) or terminator (H_1, T), all of P-CSCF #1, I-CSCF #1, S-CSCF #1, and HSS #1 servers are logically located in Network #1. The dash lines connecting P-CSCF #1, I-CSCF#1, and S-CSCSF#1 are indicated as the possible links for signaling, depending on the actual session procedure.

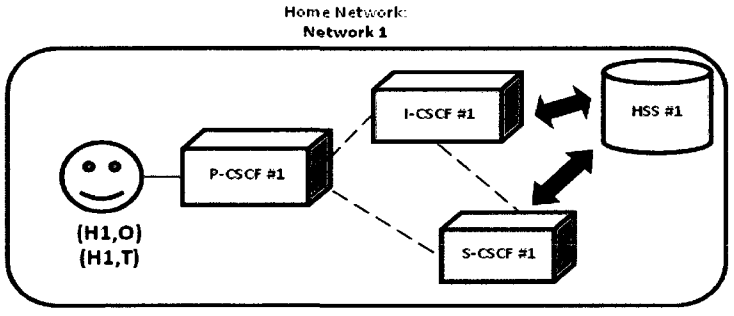


Figure 3.2: IMS servers involved in (H_1, O) and (H_1, T) and servers in orange boxes are logically located in Network #1

The other two examples are shown in Figure 3.3 and Figure 3.4. The first one refers to the cases (V_1, O) and (V_1, T), where only P-CSCF #1 is involved in Network #1. The (V_1, O) or (V_1, T) users are roaming at the Network #1, but they belong to Network #2. In this case, P-CSCF #1, as a proxy server located in Network #1, is responsible for processing the received SIP messages and forwarding them to the next appropriate network node, I-CSCF #2 or S-CSCF #2, which is located in Network #2. Thus, in the case of (V_1, O) or (V_1, T), only P-CSCF #1 is logically located in Network #1. By contrast, the second one is the cases (V_2, O) and (V_2, T), where I-CSCF #1, S-CSCF #1, and HSS #1 are located in Network #1. The users of (V_2, O) and (V_2, T) are roaming in Network #2, yet they belong to Network #1. Also, P-CSCF #2, as a proxy server, is located in Network #2, but at the same time, the rest of servers, including I-CSCF #1, S-CSCF #1, and HSS #1, are located in the home network, Network #1.

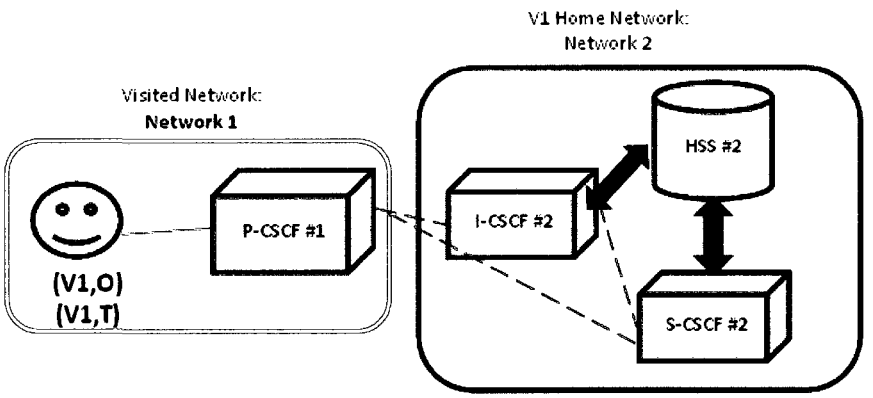


Figure 3.3: IMS servers involved in (V_1, O) and (V_1, T) and servers in orange boxes are logically located in Network #1

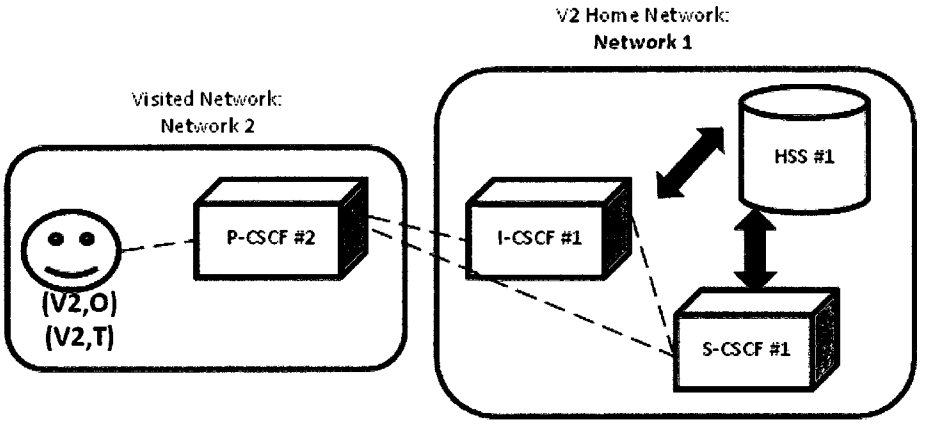


Figure 3.4: IMS servers involved in (V_2, O) and (V_2, T) and servers in orange boxes are logically located in Network #1

In summary, the results indicating which servers logically located in Network #1 are carrying the signaling traffic for a given routing scenario are shown in Table 3.2. It should be noticed that the servers listed in Table 3.2 may not appear in all session procedures for a given routing scenario. In other words, in order to confirm the presence of a server in a given session procedure of a given routing scenario, we have to look at which given session procedure is in discussion. Further details are provided in next section.

Table 3.2: Network #1 server(s) that involved in routing scenarios

Routing Scenario	Involved Server(s) in Network #1
(H_1, O)	P-CSCF #1, I-CSCF #1, S-CSCF #1, HSS #1
(H_1, T)	P-CSCF #1, I-CSCF #1, S-CSCF #1, HSS #1
(V_1, O)	P-CSCF #1
(V_1, T)	P-CSCF #1
(V_2, O)	I-CSCF #1, S-CSCF #1, HSS #1
(V_2, T)	I-CSCF #1, S-CSCF #1, HSS #1

3.2.3 Determine Servers Locations for a Given Routing Scenario with a Given Session Procedure

We examine the involvements of servers in a given session procedure with a known call scenario, and use two examples in order to effectively illustrate this problem. As we mentioned earlier, the routing of an end-to-end call scenario is the concatenation of the corresponding originating, and terminating routing scenarios. One example is shown in Figure 3.5. The example shows the servers that are involved in the call scenario where (V_1, O) calls (V_2, T) in order to establish a basic session. It can be observed that P-CSCF #1 is not involved at all, while I-CSCF #1 is involved once in the path of the terminating routing scenario, and S-CSCF #1 is involved twice in the end-to-end call scenario: once in the path of originating routing scenario and once in the terminating routing scenario. The HSS #1 server is only in the terminating routing scenario.

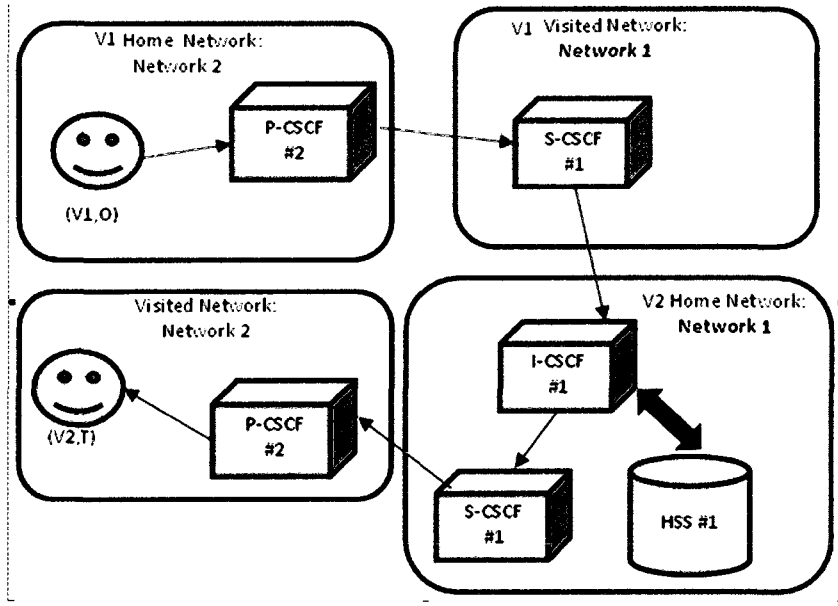


Figure 3.5: IMS end-to-end call scenario where (V_1, O) calls (V_2, T), performing basic session setup procedure

Another example with the same call scenario is shown in Figure 3.6. This example depicts the user as (V_1, O) starts the session procedure, which is re-invite for new codec procedure, with (V_1, O) user. However, in the path of the originating routing scenario, the I-CSCF #1 and HSS #1 are not involved as in the basic session setup procedure for both paths of originating and terminating routing scenarios. I-CSCF servers are responsible for assigning the address of S-CSCF. HSS servers are responsible for offering information about S-CSCF. The address of S-CSCF #1 is decided in the first session setup, with the help of both I-CSCF and HSS servers. Once the address of S-CSCF is decided, I-CSCF and HSS servers are no longer needed in the current session procedure. Therefore, it can be observed that only S-CSCF #1 is involved twice in the end-to-end call scenario: once in the path of originating routing scenario and once in the terminating routing scenario.

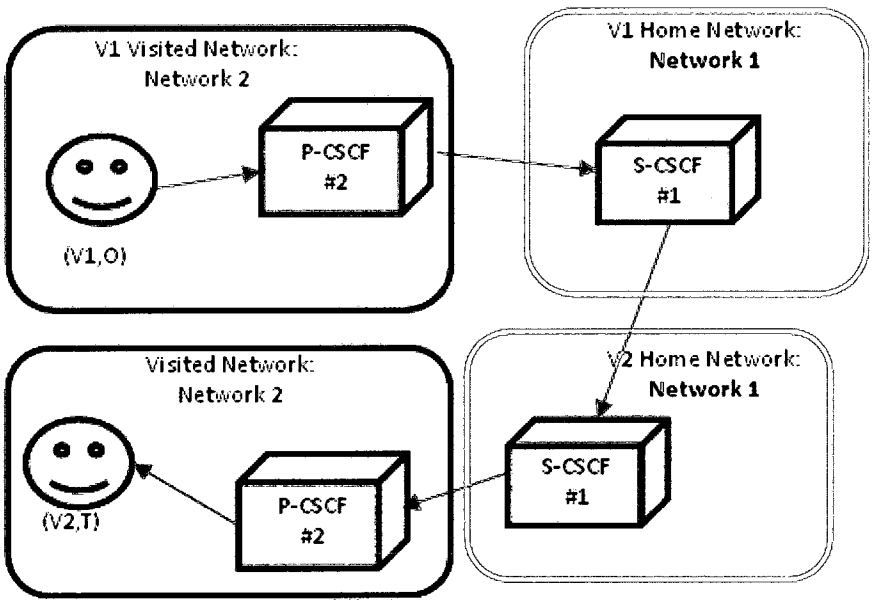


Figure 3.6: IMS end-to-end call scenario where (V_1, O) calls (V2, T), performing re-invite for new codec procedure

It should be noted that although session procedures are grouped into routing scenarios as discussed above, not all servers in a routing scenario will appear in all the signaling session procedures. For example, I-CSCF #1 server in Figure 3.5 is not involved in the path of the originating routing scenario. In addition, only S-CSCF #1 server in Figure 3.6 is involved in both the path of the originating routing scenario and the terminating routing scenario. Although we defined the routing scenarios that somewhat help us reduce the complexity of analysis, the estimation of traffic load on a particular server is still complicated due to the different servers involved in different session procedures for a given routing scenario. Hence, seeking a technique that can capture the load correlation between different servers is our main task. This leads to the introduction of signaling flow concept.

In summary, out of 15 call scenarios, we identified 6 routing scenarios. With each routing scenario, 20 session procedures are defined. Next, a signaling flow based approach for modeling the IMS signaling traffic is presented in Chapter 4; this is the main part of our thesis.

Chapter 4

IMS Traffic Model

This chapter explains the technique that we use to capture the characteristics of signaling traffic in IMS network. We also interpret how to apply the technique in order to characterize the traffic in each routing scenario for the selected session procedures. A mathematical model is successfully formulated in order to evaluate the performance of a given IMS system. By applying the model, the server load and the mean server queuing time can be predicted by means of a simple mathematical calculation. In order to better understand how this model works to predict the server load, we offer an example by introducing a new application to the market. The details of the procedures of predicting the server load in this example are provided. Moreover, we develop a measurements architecture based on ESP and ENF frameworks in order to collect the network statistics and user profile. Then, the input data of the proposed traffic model can be obtained from the collected data.

4.1 SIP Signaling Flows

We propose the signaling flow in this section. The procedures of characterizing the SIP signaling traffic by using the signaling flows are discussed.

4.1.1 Motivation

We understand that IMS network is an extremely complex system because of the combination of various routing scenarios and the large number of session procedures defined in [10] for various signaling functions. Each combination is required to analyze individually, so that the server load can be obtained accurately. As we discussed before, it is a challenging task to engineer all kinds of IMS servers in order to satisfy the real-time call performance requirements; therefore, our task is to propose an effective solution by utilizing the characteristics of SIP messages, so that the server load can be easily predicted by following a simple procedure.

After carefully considering the diagrams of session procedures, there are two findings that give us the inspiration. We take the registration procedure (user registers for the first time), which is shown in Figure 4.1, as an example. The first finding is that one message typically triggers a sequence of messages to complete a transaction. For instance, as show in Figure 4.1, REGISTER message (1) triggers REGISTER message (2) to be sent to I-CSCF server. The following REGISTER message (5) and its response 401 messages (8, 9, 10) are all triggered by the previous REGISTER message. The same concept can be applied to the messages that travel between I-CSCF and HSS and HSS and S-CSCF. The request (3) sent by I-CSCF triggers the response (4) to be sent back to

I-CSCF. These perfectly present a causal relationship among a sequence of messages, and the load at their corresponding servers is therefore correlated. This finding leads us to utilize an approach in order to capture such relationship among a sequence of messages.

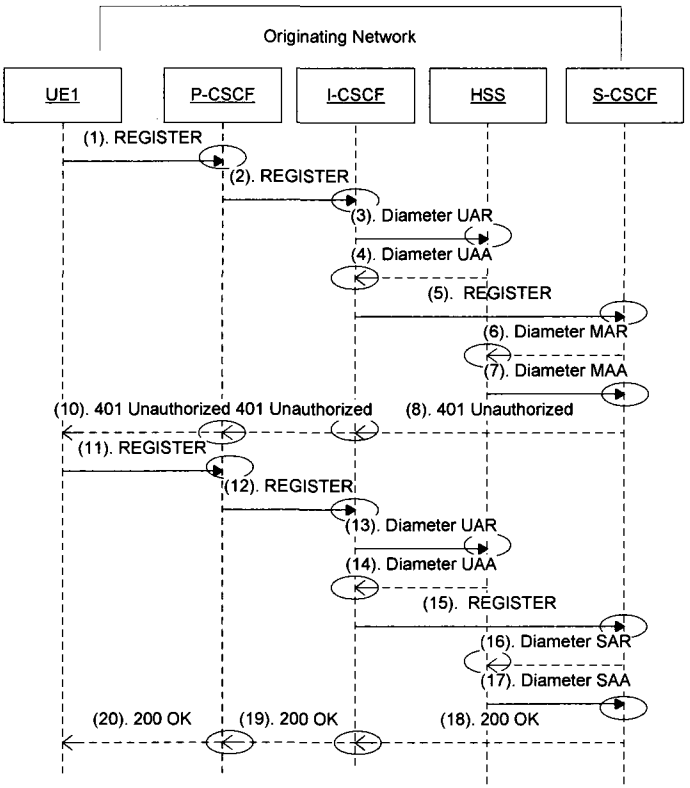


Figure 4.1: The diagram of registration procedure (user registers for the first time)

Secondly, we found similar sequences of messages existing in different session procedures. These similar sequences of messages follow the same path and even contain the same messages in some cases. In Figure 4.1, a sequence of messages, (REGISTER, 401) follows the same path with a sequence of messages (REGISTER, 200). More details can be found in Appendix B.2and B.3.

Based on these two useful findings, server load can be calculated by utilizing the causal relationship among a sequence of messages. Table 4.1 lists the procedures for calculating the server load in a registration procedure (user registers for the first time), as depicted in Figure 4.1.

Table 4.1: Server load for registration procedure (user registers for the first time), where number1 represents the number of message sequences passed by this server, number2 represents the message number, in bold type

SIP Message Sequence	Message Number	Load on server for each SIP message sequence unit: number of messages number1 x number2			
		P-CSCF	I-CSCF	HSS	S-CSCF
(REGISTER, 401)	**2** (for P/I-CSCF); **1** (for S-CSCF)	1x**2**	1x**2**	0	1x**1**
(REGISTER, 200)	**2** (for P/I-CSCF); **1** (for S-CSCF)	1x**2**	1x**2**	0	1x**1**
(Diameter UAR, Diameter UAA)	**1** (for HSS & I-CSCF)	0	2x**1**	2x**1**	0
(Diameter SAR, Diameter SAA)	**1** (for HSS & I-CSCF)	0	0	2x**1**	2x**1**
Total load at server:		**4**	**6**	**4**	**4**

In a given session procedure, a SIP message sequence describes a message sent from a node, traversing a list of nodes, and arrived at the destination node. If there is a response to the previous request, the request along with the response is considered a message sequence. However, if there is no response, the request becomes a single message sequence. There are four SIP message sequences extracted from the registration procedure (user registers for the first time), and they are listed in Table 4.1. The message number is defined as the number of messages of the specific message sequence passing a

server for a given unit of time. As shown in Table 4.1 , since any of servers, except S-CSCF, on the routing chain needs to process the message sequences, (REGISTER, 401) and (REGISTER, 200). Each of these two message sequences consists of a request and a response, the message number of these two message sequences is 2. S-CSCF only processes the REGISTER message once as it is received. By contrast, the message sequences, (Diameter UAR, Diameter UAA) and (Diameter SAR, Diameter SAA), are one way trips, while HSS processes Diameter UAR and Diameter SAR, and I-CSCF processes Diameter UAA and Diameter SAA, respectively. The message number of these two message sequences is 1.

Next, the load of a message sequence at one server is equivalent to the number of message sequences passed by this server (number1) times the number of the message sequences passing a node (number2), and it is represented as a format of number1 x number2, as depicted in Table 4.1, and number2 is in bold type. For example, message sequence (REGISTER, 401) is passed by P-CSCF once (number1 = 1), and the message number of this message sequence is 2 (number2 = 2), then the load at P-CSCF is 1x2 (number1 x number2). If a message sequence does not traverse a certain server, the load of this message at this server would be zero. For instance, P-CSCF is not involved in the message sequence, (Diameter UAR, Diameter UAA), then the corresponding load for this node is marked as zero.

Lastly, the load at different servers involved in registration procedure (user registers for the first time) can be obtained, as shown in Table 4.1, where server load is calculated as the number of messages to be processed in this server for this session procedure. The

calculated results in Table 4.1 can be verified with the diagram of the registration procedure (user registers for the first time), as shown in Figure 4.1. It demonstrates this approach to characterize server load by using the causal relationship among a sequence of messages.

In addition, this causal relationship not only exists in a registration procedure (user registers for the first time), but it also exists in the rest of the identified 20 session procedures. Therefore, to better present the causal relationship among the sequence of messages, a signaling flow based approach is introduced and applied. Moreover, this approach aims to map various combinations of routing scenarios and session procedures to the limited number of signaling flows; this allows the load of each IMS server to be easily estimated. When a new session procedure is introduced into the IMS system, we hope the load can still be characterized through our signaling flow based approach with little modification. The detailed procedures of our signaling flow based approach are provided in the following sections.

4.1.2 Introduction to SIP Signaling Flows

It is well known that the data flow level analysis on user data traffic is appealing since it provides a useful assessment of performance requirements for a router [13]. A popular definition of data flow adopted for data plane traffic is defined as a unidirectional packet stream with a unique <*source-IP-address, source-port, destination-IP-address, destination-port, IP-Protocol*> tuple [13]. We expand the flow concept into signaling traffic analysis in IMS network. A signaling flow is defined to be the aggregation of a

sequence of messages that follows the same path in a network of IMS servers. The characteristics of a signaling flow include both signaling flow path and signaling flow volume.

The path of a signaling flow identifies the servers that a specific sequence of messages will traverse as well as the order that these servers will be traversed, while the specific sequence of messages is established by the causal structure among these sequences: one message triggers another one. The signaling flow path is determined from the session procedures.

It should be noted that multiple sequences of messages can be mapped to the same signaling flow as long as they traverse the same path. This means that a server may need to process different types of messages for the same signaling flow, and therefore, the message processing times for a signaling flow at a server are variable. Furthermore, although a sequence of messages in a signaling flow may have a causal relationship, the messages may consist of different types at different servers. Therefore, their processing times at different servers are also different.

The signaling flow volume represents the mean number of messages of the specific signaling flow passing a server for a given unit of time. The key characteristic of a signaling flow is that its volume stays constant across all the servers that the signaling flow traverses. Therefore, the signaling flow concept captures the correlation structure among different servers due to various sequences of messages. Below, we will explore how to quantify the signaling traffic at signaling flow level.

4.1.3 Quantification of SIP Signaling Traffic at Signaling Flow Level

Next, we will show the method used to extract signaling flows from a typical IMS session procedure, namely, registration procedure (user registers for the first time), in the home network, with the originating routing scenario (H1, O). As shown in Figure 4.2, since all the servers, P-CSCF, I-CSCF, HSS, and S-CSCF are located in the designated home network, Network #1, the signaling traffic traversing through these servers is taken into account.

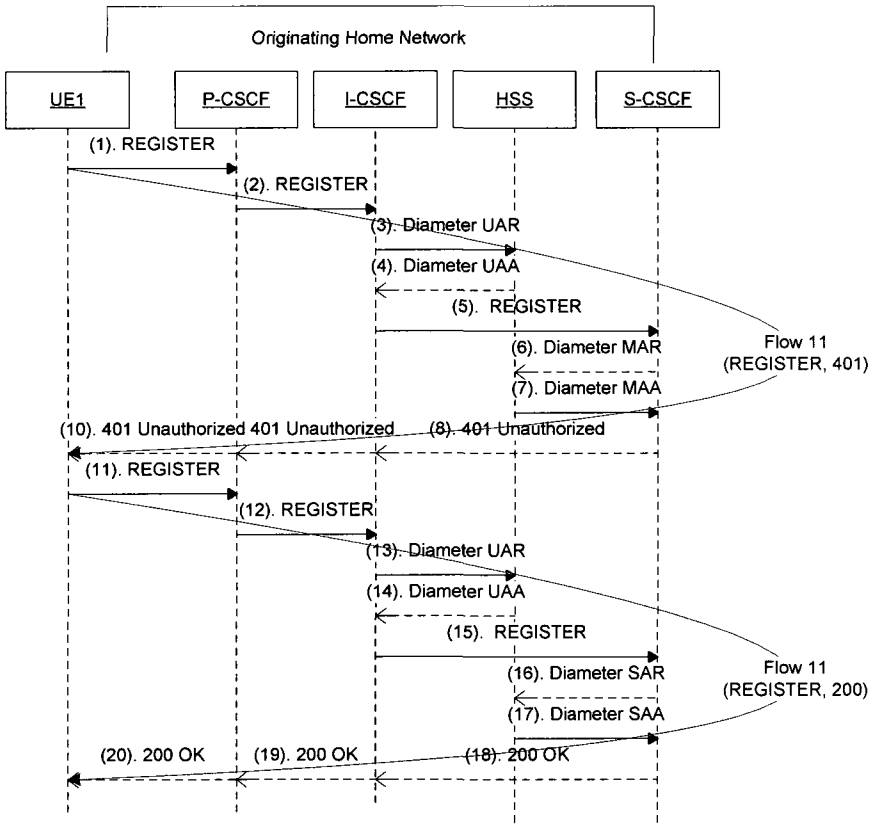


Figure 4.2: IMS registration procedure (user registers for the first time), with signaling flow analysis, in (H_1, O)

As we discussed before, there is a causal relationship existing in the SIP messages generated in the session procedure. This leads to the correlation structure among different servers. The message sequences, (REGISTER, 401) and (REGISTER, 200), traverse all the servers except HSS and follow the same path; therefore, the load at the involved servers is correlated and can be easily captured. Once the message sequences follow this path, it is considered one signaling flow. This signaling flow is numbered as 11 due to its order in our signaling flow analysis, and we have identified 17 signaling flows. Later, we will introduce the rest of the signaling flows defined in our analysis. In regards to signaling flow 11, as illustrated in Figure 4.2, in total, there are two message sequences that follow the same path, and they are (REGISTER, 401) and (REGISTER, 200). Each message sequence contributes two messages to the signaling flow for each server. Therefore, there are four messages passing any one of the servers along this signaling flow, except S-CSCF server. Thus, the volume of this signaling flow is four. As it can be seen in the figure, different parts of a session procedure may belong to different signaling flows. In the above registration procedure, two more signaling flows are identified. The details of these different signaling flows are presented in Table 4.2. The detailed explanation of the message sequences in registration procedure (user registers for the first time) is provided in Appendix B.1.

Table 4.2: Signaling flow summary for registration procedure (user registers for the first time), in (H_1, O)

Signaling Flow	SIP Message Sequence	Signaling Flow Path	Signaling Flow Volume
11	(REGISTER, 401), (REGISTER, 200)	→P→I→ S→I→P→	4
15	(Diameter UAR, Diameter UAA)	→HSS → I	2
16	(Diameter SAR, Diameter SAA)	→HSS → S	2

Now, we understand how to extract the signaling flows from a given IMS session procedure. Next, we will discuss the differences between signaling flow based modeling for signaling traffic and the data flow based modeling for data traffic.

4.1.4 Differences between Signaling Flow based Modeling for Signaling Traffic and for Data Traffic

Data flow based modeling has been widely used in data plane traffic characterization. A popular definition of data flow adopted for data plane traffic is defined in [13]. The data flow is defined as a unidirectional packet stream with a unique <*source-IP-address, source-port, destination-IP-address, destination-port, IP-Protocol*> tuple [13]. The data plane traffic transported by the protocol among the end users can be decomposed based on source, destination, application, and/or transport protocol, depending on the definition of the data flow adopted. Figure 4.3 illustrates an example of transporting two types of data traffic between two users by UDP. By adopting the above data flow definition provided in [13], there are two data flows can be identified, as follows:

1. Data flow 1 for Video Streaming application: <UserA_IP_address, UserA

_source_port_1, UserB_IP_address, UserB _source_port_1, UDP>

2. Data flow 2 for Internet Gaming application: <UserA_IP_address, UserA _source_port_2, UserB_IP_address, UserB source_port_2, UDP>

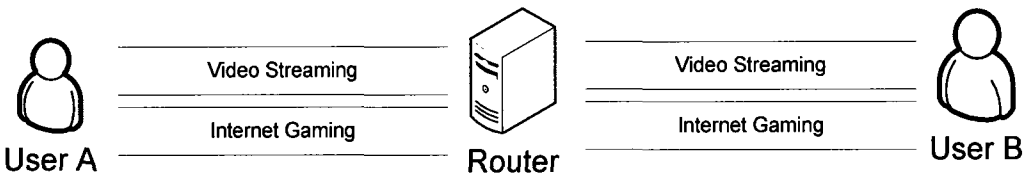


Figure 4.3: An example of data traffic transported by UDP, between two users

If the ideas of data flow are adopted by the signaling traffic, each SIP message or any sequence of messages can be treated as a single flow based on the sender and the receiver of message or a sequence of messages. Therefore, there are many different ways of formulating the flows in a session procedure. Also, it may procedure many flows afterwards. Following the idea of data flow, for a system that has a large number of session procedures, there will be huge number of flows that can be formed. Thus, it does not reduce the complexity of the signaling traffic analysis, but it also makes it worse.

On the other hand, our proposed signaling flow is different from the data flow concept, given that we utilize the causal relationship among a sequence of messages in the session procedures. As long as a sequence of messages follows the same path in a chosen set of session procedures, it is considered as a signaling flow. In this case, a number of session procedures can be mapped into a limited number of signaling flows, so that the signaling traffic analysis is simplified to a certain extent. Thus, the signaling

flow based modeling for signaling traffic is quite different from the data flow based modeling of data plane traffic, since it is heavily dependent on the session procedures. In the following section, we will show the SIP signaling flow analysis.

4.2 SIP Signaling Flow Analysis

In this session, we present the signaling flow analysis on another typical session procedure, a basic session setup procedure with 6 different routing scenarios.

4.2.1 Basic Session Setup, (H_1, O)

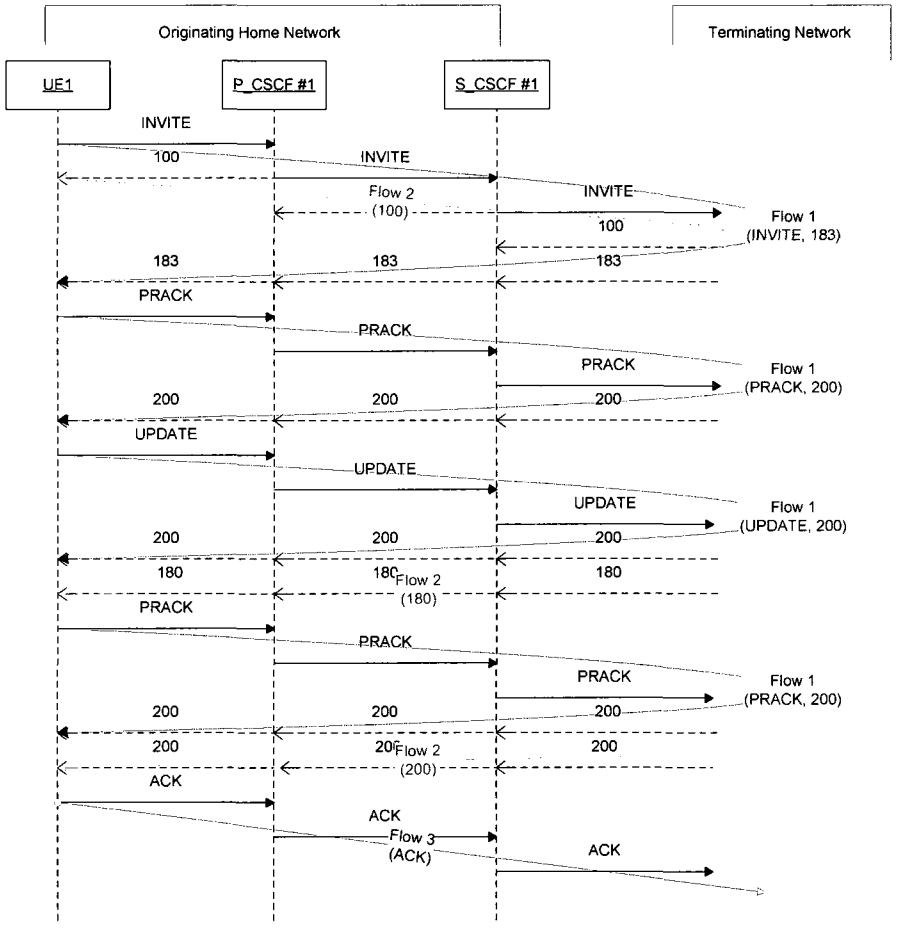


Figure 4.4: IMS basic session setup procedure with signaling flow analysis, in (H_1, O)

As shown in Figure 4.4, since P-CSCF #1 and S-CSCF #1 are located in the designated home network, which is Network #1, the signaling traffic traversing through these two servers is taken into account. It is observed that an INVITE message is initiated by UE1. Once P-CSCF #1 receives it, P-CSCF #1 sends a 100 message back to UE1 in order to inform UE1. Meanwhile, P-CSCF#1 forwards the INIVTE request to the next node in the network, and then, a 183 message as the response of INVITE request is sent from the terminating network, and it follows the same path as the INVITE does. Thus, 183 message is a formal response to the INVITE request, compared to the 100 message. Furthermore, they are noticed as one response followed by one request, so that the correlation between P-CSCF #1 and S-CSCF #1 server load can be captured, namely 1:1. They are treated as one signaling flow, which is numbered as 1. Signaling flow 1 is a round trip signaling flow; the details are listed in Table 4.3. The detailed explanation of the sequences of messages in basic session setup procedure is provided in Appendix C.1.

Table 4.3: Signaling flow summery for basic session setup procedure, in (H_1, O)

Signaling Flow	Message Sequence	Signaling Flow Path	Signaling Flow Volume
1	(INVITE, 183), (PRACK, 200), (UPDATE, 200), (PRACK, 200)	→P1 → S1 → --- → S1 → P1 →	8
2	100,180,200	→ S1 → P1 →	3
3	ACK	→ P1 →S1 →	1

Moreover, the message sequences (PRACK, 200), (UPDATE, 200), and (PRACK, 200) follow the same path as (INVITE, 183), and they are also considered as signaling flow 1. Then, the volume of signaling flow 1 is 8 messages per dialog. The rest of message sequences in the diagram contain 100, 180, 200, and ACK message. Since the message sequences 100, 180, and 200 follow the same path, they are treated as signaling flow 2; its volume is 3 messages per dialog. ACK message follows in the opposite direction of signaling flow 2, and it is considered to be the last signaling flow, called signaling flow 3, whose volume is 1 message per dialog. Both signaling flow 2 and signaling flow 3 are one way trips in nature. In a conclusion, we divide all the signaling messages in this session procedure into three signaling flows; the details are listed in Table 4.3. Section 4.2.2 uses the same logic and concludes the signaling flows analysis of this session procedure in the other 5 different routing scenarios.

4.2.2 Signaling Flow Analysis on Basic Session Setup Procedure

The signaling flow analysis on a basic session setup procedure in the remaining 5 different routing scenarios is discussed separately in Appendix H. Now, we understand how to extract the signaling flows from different routing scenarios. After a performed analysis on basic session setup procedure in different routing scenarios, 11 signaling flows are found, and the summary is provided in Table 4.4. The numbers in the check box represent the signaling flow volume (number of messages) of a particular signaling flow for the corresponding routing scenario. Thereby, the signaling flow concept maps the combinations of 6 routing scenarios and this session procedure into a limited number

of signaling flows. This helps us estimate the server load for a given session procedure performed in a particular routing scenario. The detailed procedures of estimating the server load are provided in next section.

Table 4.4: Signaling flow analysis on basic session setup procedure, in 6 different routing scenarios

Routing Scenario	Signaling Flow										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**15**
(H_1, O)	8	3	1								
(H_1, T)		1		2	6	3					1
(V_1, O)										12	
(V_1, T)										12	
(V_2, O)							12				
(V_2, T)							7	2	3		1

4.3 Characterization of Server Load

Following the same approach, we have analyzed the 20 session procedures with the 6 routing scenarios as discussed above. As shown in Table 4.5, in the end, 17 signaling flows were identified.

Table 4.5: Summary of 17 signaling flows

Signaling Flow	Signaling Flow Path
1	→P1 → S1 → --- → S1 → P1 →
2	→ S1 → P1 →
3	→ P1 →S1 →
4	→I1 → S1 → P1 →---→ P1 → S1 → I1 →
5	→ S1 → P1 →---→ P1 → S1 →
6	→ P1 → S1 → I1 →
7	--- S1 ---
8	→I1 → S1 →---→ S1 → I1 →
9	→ S1 → I1 →
10	--- P1 ---
11	→ P1 → I1 → S1 → I1 → P1 →
12	→ I1 → S1 → I1 →
13	→ I1 → S1 → P1 →
14	→ I1 →S1 →
15	→ HSS → I1
16	→ HSS → S1
17	→ S1 → HSS1

Figure 4.5 represents the volumes (number of messages) of signaling flows per session procedure in one routing scenario, (H_1, O). We will call the table matrix $\mathbf{X_1}$ (also shown in Appendix I.1). The matrices for the remaining 5 routing scenarios, $\mathbf{X_2}$,..., $\mathbf{X_6}$, are all listed in Appendix I.2, I.3, I.4, I.5, and I.6.

$X_1 =$

Procedure \ Flow	1	2	3	4	5	…	…	…	…	11	…	…	15	16	17
1	8	3	1	0	0	0	0	0	0	0	0	0	0	0	0
2	8	3	1	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	4	0	0	2	2	0
⋮	0	0	0	0	0	0	0	0	0	2	0	0	1	2	0
⋮	0	0	0	0	0	0	0	0	0	2	0	0	1	1	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
⋮	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
⋮	0	0	1	0	0	0	0	0	0	1	0	0	0	0	2
⋮	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	10	3	1	0	0	0	0	0	0	0	0	0	0	0	0
⋮	8	4	2	0	0	0	0	0	0	0	0	0	0	0	0
⋮	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
⋮	8	3	1	0	0	0	0	0	0	0	0	0	0	0	0
⋮	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.5: Matrix of session procedures and signaling flows in (H_1, O)

As discussed in the previous section, every signaling flow traverses different servers according to the signaling flow path. We use a matrix in order to identify the relationship between the signaling flows and the servers, including P/I/S-CSCF, and HSS. We denote this matrix as **I**, shown in Figure 4.6. If the number of times that a signaling flow traverses all of the servers is the same, the corresponding value will be 1. For example, signaling flow 1 is a round trip signaling flow, and it traverses both P-CSCF and S-CSCF twice. Then, the corresponding value is indicated as 1. Moreover, the signaling flow 2 is a single trip signaling flow, and it only traverses the servers once. The corresponding value is also marked as 1. If the number of times that a signaling flow traverses one of servers is half of the other servers, the values will be 1/2, such as signaling flow 11 at S node and signaling flow 12 at S node. If a signaling flow does not traverse a server at all, the corresponding value will be 0.

$$
\mathbf{I}=\begin{array}{c} \\ Flow1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \end{array}
\begin{array}{c}
\begin{array}{cccc} P & I & S & HSS \end{array} \\
\left[\begin{array}{cccc}
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 \\
1 & 1 & 1/2 & 0 \\
0 & 1 & 1/2 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1
\end{array}\right]
\end{array}
$$

Figure 4.6: Matrix represents the relationship between the signaling flows and the servers

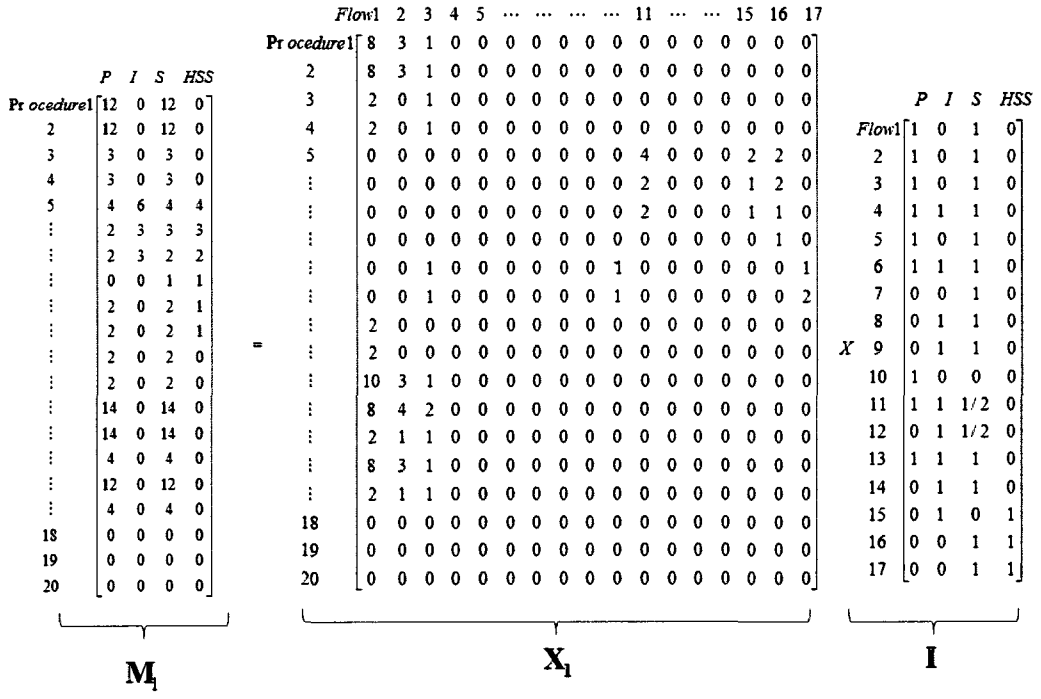


Figure 4.7: The computation of M_1, by the multiplication of X_1 and I

Through the quantitative analysis from the previous section, we know $\mathbf{X_1}$ represents the volumes (number of messages) of the signaling flows per session procedure in one routing scenario, (H_1, O), and **I** identifies the relationship between the signaling flows and the servers. The multiplication of $\mathbf{X_1}$ and **I** results in a matrix $\mathbf{M_1}$, which represents the load carried by each server as generated per session procedure for the (H_1, O) routing scenario in terms of message per unit time. The result is shown in Figure 4.7 (also presented in Appendix I.1). Following the same ideas, matrix $\mathbf{M_2}$, ..., $\mathbf{M_6}$ can be obtained for the 5 remaining routing scenarios, and they are listed in Appendix I.2, I.3, I.4, I.5, and I.6.

Furthermore, let $\mathbf{T_1}$ be the row vector representing the average arrival rate of the session procedures for the routing scenario (H_1, O) in a unit of session procedure per unit time. An example is shown in Figure 4.8. The first value in the vector $\mathbf{T_1}$, namely 5, stands for an average of 5 session procedures (session procedure here is referred to as session procedure 1) arriving every unit time in the routing scenario (H_1, O).

procedure	1	2	3	4	5	6	7	8	9	10	...	...	...	...	...	...	19	20
$\mathbf{T_1}$ =	5	5	5	5	4	4	4	4	4	4	3	3	2	2	2	2	1	1	1	1

Figure 4.8: Example of row vector $\mathbf{T_1}$

Then, $\mathbf{L_1}$ is calculated by,

$$\mathbf{L_1=T_1M_1=T_1(X_1I)} \qquad \textbf{4.1}$$

$\mathbf{L_1}$ is a row vector, which represents the load carried by each server for the (H_1, O) routing scenario in terms of message per unit time.

Considering the fact that there are 6 routing scenarios, the total average load of all the servers can be calculated as:

$$\mathbf{Л} = \sum_{r=1}^{6} \mathbf{L}_r \qquad \textbf{4.2}$$

where **Л** is a row vector denoting the total average load (the number of messages) of all servers in terms of message per unit time. Let λ_v denote the v^{th} element of the row vector **Л** and μ_v denote the mean service rate of server v, then the utilization of server v is

$$p_v = \lambda_v / \mu_v \qquad \textbf{4.3}$$

In a conclusion, the signaling flow traffic model is built based on the signaling flow analysis of the session procedures. In this traffic model, the server load can be predicted by applying the input data $\mathbf{T}_1$ into the Equation 4.1 and 4.2. In order to better understand how this model works for predicting the server load in a specific example, we offer an example by introducing a new application to the market in next section.

4.4 Example

IMS supports a wide range of IP-based applications or services over wireline and wireless networks. It allows third-party vendors to develop new applications or services for both network operators and end users. In order to guarantee the stability of a network of servers, it is important for network providers to predict the impact on servers when introducing new applications. Now, let us offer an example that shows how to utilize the signaling flow analysis in order to predict the impact on different servers when a new application is introduced into the market. Now, we assume that there is a new application

called ABC, and the involved session procedures are only performed in an originating routing scenario, (H_1, O), and a terminating routing scenario, (H_1, T).

The detailed procedures for predicting server load are as follows:

1. Find the session procedures involved when end users request this application, as shown in Table 4.6. Those session procedures may contain some of the 20 session procedures discussed above, and they are numbered in Table 2.1. Decide the distribution of 6 session procedures, as provided in Table 4.6.

Table 4.6: Example of predicting the server load by introducing a new application

New Application	No.	Involved Session Procedures	Frequency
Application ABC	1	ABC session setup	30%
	2	Re-invite for new codec, without I-CSCF	2.5%
	5	Registration, user not registered	30%
	6	Re-registration, user registered	2.5%
	11	Mobile terminal initiated session release	30%
	21	A new session procedure: involved signaling flow 1, 2, 3, 21	5%

2. Determine the signaling flows extracted from the involved session procedures and their volume (number of messages) of corresponding signaling flow, as shown in Table 4.7 and Table 4.8 for the two routing scenarios. Those signaling flows may contain some of the 17 signaling flows already extracted. If new session procedure(s) is/are introduced, an approach similar to the one above can be used in order to identify the signaling flows. Insert the new signaling flow(s) into the signaling flows table.

Table 4.7: Signaling flow analysis on the session procedures involved in the example, in (H_1, O)

Involved Session Procedures	Signaling Flow					
	1	**2**	**3**	**11**	**15**	**16**
1	8	3	1			
2	8	3	1			
5				4	2	2
6				2	1	2
11	2					
21	2	3	4			5

Table 4.8: Signaling flow analysis on the session procedures involved in the example, in (H_1, T)

Involved Session Procedures	Signaling Flow					
	2	**3**	**4**	**5**	**6**	**15**
1	1		2	6	3	
2	1	3			8	
5						
6						
11	2					
21	2	3	4			5

3. Estimate $\mathbf{T_1}$ and $\mathbf{T_2}$, the average arrival rate of the session procedures for both of routing scenarios, (H_1, O) and (H_1, T).

We assume the following situation:

Step a. The average number of users arriving at the network to request the new application in an hour is $3.6x10^3$.

Step b. The average number of session procedures triggered by one user is 2 per hour.

Step c. The distribution of these two routing scenarios is 50% and 50%.

Step d. The distribution of these 6 session procedures is provided in Table 4.6.

The procedures of calculating $\mathbf{T_1}$ and $\mathbf{T_2}$:

1. The average rate of session procedures triggered by the new application that requested by users arriving at the network every second is,

$3.6x10^3$ (users/hour, from Step a) x 2 (session procedures/user, from Step b) / 3600 (seconds/hour) = 2 (session procedures/second) **4.4**

2. The average rate of session procedures in each routing scenario per second is,

$$2x50\% = 1 \quad \textbf{4.5}$$

3. $\mathbf{T_1}$ and $\mathbf{T_2}$ are row vectors, and they are equal to the above value (=1) times the distribution of 6 session procedures that is presented in Table 4.6. Then,

$$\mathbf{T_1} = \mathbf{T_2} = [1x30\%\ 1x2.5\%\ 1x30\%\ 1x2.5\%\ 1x30\%\ 1x5\%] = [0.3\ 0.025\ 0.3\ 0.025\ 0.3\ 0.05] \quad \textbf{4.6}$$

4. On the basis of all the signaling flows we obtained from the involved session procedures, the matrices $\mathbf{X_1}$ and $\mathbf{X_2}$ are obtained from Table 4.7 and Table 4.8, for routing scenario (H_1, O) and (H_1, T), respectively. Matrix **I** shrinks from Figure 4.6 for the attendance to the involved signaling flows. The matrices are all shown in Figure 4.9.

$$\mathbf{X_1} = \begin{bmatrix} 8 & 3 & 1 & 0 & 0 & 0 \\ 8 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 2 & 2 \\ 0 & 0 & 0 & 2 & 1 & 2 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 3 & 4 & 0 & 0 & 5 \end{bmatrix}, \mathbf{X_2} = \begin{bmatrix} 1 & 0 & 2 & 6 & 3 & 1 \\ 1 & 3 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 3 & 4 & 0 & 0 & 5 \end{bmatrix}, \mathbf{I} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1/2 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Figure 4.9: The matrices X_1, X_2, and I in the example

5. Then, apply $\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{I}$, $\mathbf{T_1}$ and $\mathbf{T_2}$ into Equations 4.1 and 4.2. The calculation steps are provided below.

Step a. Apply Equations 4.1, we obtain the load carried by each server for the (H_1, O) and (H_2, O) routing scenarios, $\mathbf{L_1}$ and $\mathbf{L_2}$.

$$\mathbf{M_1} = \mathbf{X_1 I} = \begin{bmatrix} 12 & 0 & 12 & 0 \\ 12 & 0 & 12 & 0 \\ 4 & 6 & 4 & 4 \\ 2 & 3 & 3 & 3 \\ 2 & 0 & 2 & 0 \\ 9 & 0 & 14 & 5 \end{bmatrix} \quad \textbf{4.7}$$

$$\mathbf{M_2} = \mathbf{X_2 I} = \begin{bmatrix} 9 & 9 & 7 & 4 \\ 4 & 8 & 4 & 8 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 9 & 5 & 14 & 5 \end{bmatrix} \quad \textbf{4.8}$$

$$\mathbf{L_1} = \mathbf{T_1 M_1} = [6.2 \quad 8.075 \quad 14.55 \quad 16.075] \quad \textbf{4.9}$$

$$\mathbf{L_2} = \mathbf{T_2 M_2} = [3.85 \quad 7.0 \quad 10.5 \quad 12.15] \quad \textbf{4.10}$$

Step b. Apply Equation4.2, we obtain the total average load of all the servers, **JI**.

$$\mathbf{JI} = \mathbf{L_1} + \mathbf{L_2} = [10.05 \quad 15.075 \quad 20.05 \quad 28.225] \quad \textbf{4.11}$$

Now we understand how this traffic models works for predicting the server load in a specific example. Next, we will discuss the proposed measurements architecture that

collects the useful event information from the network. The collected event information can be calculated in order to obtain the input data of the proposed traffic model, $\mathbf{T_r}$.

4.5 Measurements Architecture Based on Event State Publication (ESP) and Event Notification Framework (ENF)

In this section, we propose a measurements architecture that is based on Event State Publication (ESP) and Event Notification Framework (ENF). The details regarding how to calculate the input data of the proposed traffic model, $\mathbf{T_r}$, based on the collected data are provided.

4.5.1 Motivation

The server load can be predicted by the proposed signaling flow based traffic model with a given input data, $\mathbf{T_r}$, which represents the average arrival rate of 20 session procedures in routing scenario *r*. The formulation of computing the server load by applying $\mathbf{T_r}$ is presented in Equation 4.1 and 4.2. In general, $\mathbf{T_r}$ is given as a statistic for user profile from the network provider. However, we know that the users' behavior is dynamic, and real-time network conditions are determined by various network factors at the time. For example, users request different applications according to their needs in different time slots. The value of $\mathbf{T_r}$ is then unknown; therefore, we need to obtain $\mathbf{T_r}$ from IMS network during the run time.. The obtained $\mathbf{T_r}$ should guarantee to effectively reflect network information in time. Thus, we need to collect system statistics and user profile during the run time, which can directly reflect the user behavior and network conditions at the moment. The system statistics can include the number of physical nodes failed, the

number of physical nodes overload, and the number of certain session procedures triggered, which are all measured within a given time period. In addition, the user profile is a collection of personal data, including the number of users entering to the network, the number of certain applications requested, and the average duration of users staying in the network, all of which are measured within a given time period, and so forth.

In order to obtain the input data of the proposed traffic model, $\mathbf{T_r}$, we have to develop a method to collect user profile. Meanwhile, such a method can be applied in order to collect all the network statistics, so that it is more convenient for network providers. The collected data can be evaluated in order to obtain information regarding to server overload, network entity failure, and application-dependent user behavior. Then, network providers can rely on this conclusion in order to perform the next tasks, for example, inserting backup node, increasing server capacity, creating different marketing strategies, among other things. The network providers are allowed to subscribe to the network statistics and the user profile at anytime online or offline according to their needs.

Therefore, a method for collecting network statistics and user profile during the run time in IMS network is developed. In our method, we proposed a new measurements architecture that can be integrated into existing IMS network. The proposed measurements architecture is based on ESP and ENF frameworks for collecting, evaluating, and subscribing the network and user data for network providers. According to a list of network statistics and user profile collected from the network, $\mathbf{T_r}$ can be

obtained by applying a set of mathematical calculations. In the following section, we introduce the proposed measurements architecture and the computation of $\mathbf{T_r}$.

4.5.2 Measurements Architecture in IMS

The measurements architecture is designed for collecting network statistics and user profile from UA and SIP servers; it is also designed to allow SIP nodes to subscribe the data. The proposed measurements architecture is established based on ESP and ENF frameworks. The definitions for the involved network entities and SIP messages are shown below:

Event Publication Agent (EPA): A network entity that can provide certain event information to Event Server (ES) through PUBLISH SIP messages periodically or when the event information is updated.

Event Server (ES): A network entity that collects the event information from EPA(s) through PUBLISH SIP messages. ES is also responsible for managing event information and notifying the watcher about the requested event information through a NOTIFY SIP messages.

Watcher: A network entity that subscribes to or requests event information from an ES through SUBSCRIBE SIP messages.

"PUBLISH": A SIP method for EPA to publish event information to ES.

"SUBSCRIBE": A SIP method that is used to request an asynchronous notification of an event or set of events at a later time. SUBSCRIBE requests are generated by a watcher, and sent to ES.

"NOTIFY": This SIP method is used by the ES in order to notify watcher of event(s) information, which has been requested by an earlier SUBSCRIBE method. The NOTIFY method contains information about the event which the watcher is interested in.

ESP is defined in [43] as a framework for the publication of certain event information from an EPA to a network entity, namely, ES that composites this event information through a PUBLISH SIP message and distributes it to interested network nodes, namely watcher. The distribution of event information is standardized in ENF, which is discussed in more detail later on. ESP consists of two network entities, ES and EPA. We use the ESP framework in order to collect the network statistics and the user profile from a network entity, which is EPA. An UA can become an EPA, because user information, like applications requested, duration of certain application, and routing scenarios can be collected. A SIP server implemented with a function that collects certain network events can also be an EPA. Such an SIP server can provide information on server utilization, the average server queuing delay, and the node failure rate. All information is sent to ES periodically or once the information is updated by using the PUBLISH SIP message. Then, ES can apply data filtering and the calculation of useful information, such as the average number of certain applications requested by users. In this thesis, we apply it on collecting network statistics and user profile, which are used to calculate the input data of the proposed traffic model, $\mathbf{T}_r$. Further details are provided in Section 4.5.3.

ENF is defined in [44] as an extensible framework. It is a general-purpose infrastructure for all classes of SIP asynchronous event subscriptions and notifications

[44]. ENF consists of two network entities, a watcher and an ES. A watcher can be a server that subscribes information from the ES in order to optimize service. An ES can also send notifications to the watecher when those event information is updated. Network providers can request the information of a certain event that is of interest through the watcher. ENF allows the watcher to request information from ES, which indicates the occurrence of certain events. In this thesis, we apply ENF to subscribe the value of $\mathbf{T_r}$. Further details are provided in Section 4.5.4.

The proposed measurements architecture based on ESP and ENF frameworks has three advantages. First, the measurements architecture is SIP based, and it can be implemented as a new application, which can be integrated into current the IMS architecture smoothly. Once an application of this measurements architecture is developed, it can be installed on different network entities just like other applications. Moreover, it allows massive deployment of this measurements architecture to different networks. Second, the network data can be effectively collected during the run time, since the proposed measurements architecture collects information from both network entities and end users using ESP. Measurements of asynchronous events can be processed and stored at ES as a later supply for network providers. Finally, since ENF uses ES in order to process raw data and to store a large number of network information, network provides can subscribe the information from ES according to their needs in a timely and efficient manner. This allows the providers to monitor the IMS network system during the run time and take effective actions before problems occur. Thus, the working efficiency of network providers can be improved.

Figure 4.10 illustrates the measurements architecture based on ESP and ENF frameworks. The watcher can request the information of certain event that they interested in by sending the SUBSCRIBE message to ES, as illustrated in Figure 4.10. When ES receives the SUBSCRIBE message from a watcher, it verifies whether this watcher is authorized to subscribe to this ES. If so, ES acknowledges the watcher with a 200 message. At the same time, the information regarding the requested event is sent to the watcher via a NOTIFY message. Meanwhile, the watcher acknowledges ES with a 200 message. The updated event information is sent to ES by EPA using the PUBLISH message periodically or once the information is updated. Similarly, ES uses the NOTIFY message to notify the subscribed watcher for updating the event information.

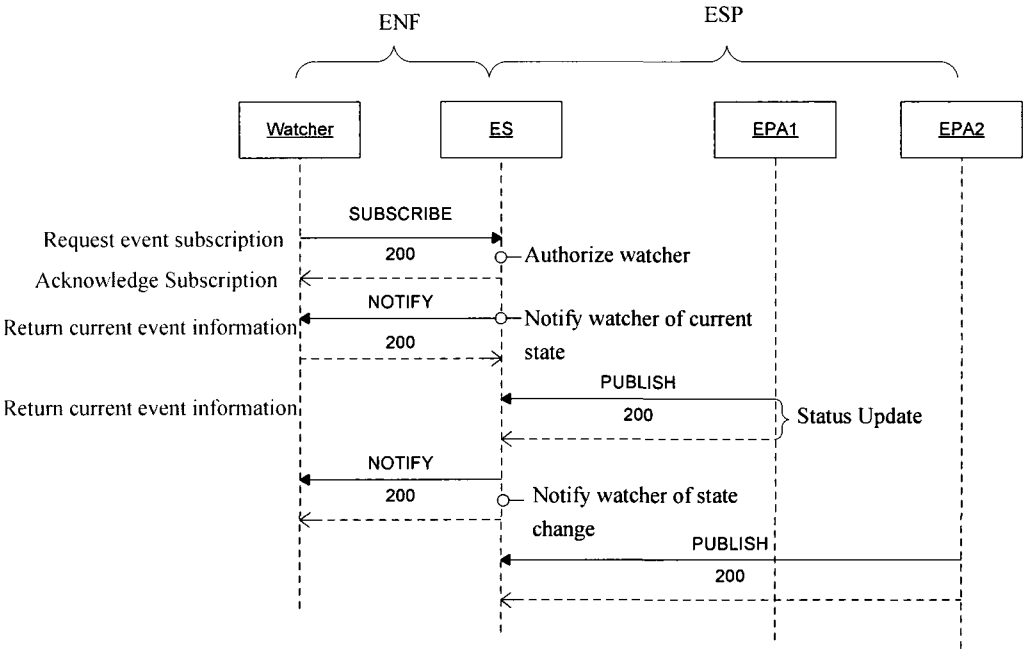


Figure 4.10: The measurements architecture based on ESP and ENF frameworks

In the next section, we provide the procedures of computing $\mathbf{T_r}$, by applying a series of the network statistics and user profile, which are obtained in the proposed measurements architecture. After that, we offer an example in an attempt to describe how ESP and ENF work for collecting, evaluating, and subscribing the network statistics and user profile from SIP servers and UA, respectively. In this case, both the SIP server and the UA act as an EPA. Eventually, the value of $\mathbf{T_r}$ is calculated, and the server load can be obtained from the proposed traffic model for the network design.

4.5.3 The Computation of the Input Data, $\mathbf{T_r}$

$\mathbf{T_r}$, which is a row vector, represents the average arrival rate of 20 session procedures in routing scenario r, as the input data of the proposed traffic model for the calculation of the server utilization and the mean server queuing delay. However, $\mathbf{T_r}$ is calculated from three useful data, and they are as follows:

1. N, average arrival rate of session procedures requested by users, in a unit of session procedure per second.
2. **A**, a 1x20 row vector represents the distribution of 20 session procedures. For example,

   A=[0.1 0.1 0.1 0.1 0.05 0.05 0.05 0.05 0.05 0.05 0.02 0.02 0.03 0.03 0.03 0.03 0.035 0.035 0.035 0.035]

3. B_r, the frequency of routing scenario r, where $r = 1,2,3,4,5,6$.

Then, following the equation shows the calculation of $\mathbf{T_r}$.

$$\mathbf{T_r} = NB_r\mathbf{A}, \quad r = 1,2,3,4,5,6 \qquad \textbf{4.12}$$

In order to obtain N, **A**, and B_r, a list of network statistics and user profile is needed to be gathered from the network by applying the proposed measurements architecture, as shown in Table 4.9.

Table 4.9: A list of network and user data

Parameter	Network Statistics and User Profile
N_1	Average arrival rate of users (user per second).
U_{rj}	The number of users entered to the system in routing scenario r at time point j, where $r = 1,2,3,4,5,6$, j=1, 2, 3, 4..., J, J is the number of time points.
U_j	The number of users entered to the system at time point j, where j=1, 2, 3, 4..., J.
Q_j	The number of applications requested by users in the system at time point j ,where j=1, 2, 3, 4..., J.
Z_j	The number of session procedures triggered at time point j, where j=1, 2, 3, 4..., J.
Z_{sj}	The number of session procedure s triggered at time point j, where $s = 1,2,3,...,20$, and j=1, 2, 3, 4..., J.

Besides, there are parameters need to be introduced. They are N_2 and N_3. N_2 represents the average number of applications requested by one user in a unit of application per user. N_3 denotes the average number of 20 session procedures triggered by one application in a unit of session procedure per application.

The procedures of calculating N, **A**, and B_r are listed below.

1. Computing N, in a unit of session procedure per second

$$N = N_1(\frac{user}{second}) \cdot N_2(\frac{application}{user}) \cdot N_3(\frac{session\ \ procedure}{application}) \qquad \textbf{4.13}$$

   a. Computing N_2, by taking the average number of applications requested by one user.

$$N_2 = \frac{1}{J}\sum_{j=1}^{J}\frac{Q_j}{U_j}, \quad j=1, 2, 3, 4..., J \qquad \textbf{4.14}$$

   b. Computing N_3, by taking the average number of session procedures triggered by one application.

$$N_3 = \frac{1}{J}\sum_{j=1}^{J}\frac{Z_j}{Q_j}, \quad j=1, 2, 3, 4..., J \qquad \textbf{4.15}$$

2. Computing **A**. At the time point j, the frequency of one session procedure is obtained by taking the average of the number of the session procedures and dividing it by the total number of 20 session procedures triggered.

$$\mathbf{A} = [\frac{1}{J}\sum_{j=1}^{J}\frac{Z_{1j}}{Z_j} \quad \frac{1}{J}\sum_{j=1}^{J}\frac{Z_{2j}}{Z_j} \quad \cdots \quad \frac{1}{J}\sum_{j=1}^{J}\frac{Z_{20j}}{Z_j}], \quad j=1, 2, 3, 4..., J \qquad \textbf{4.16}$$

3. Computing B_r. This follows the ideas of computing **A**.

$$B_r = \frac{1}{J}\sum_{j=1}^{J}\frac{U_{rj}}{U_j}, \quad j=1, 2, 3, 4..., J \qquad \textbf{4.17}$$

Now we conclude a list of network statistics and a user profile, which are required in order to calculate $\mathbf{T_r}$. In the following section, we offer an example in an attempt to explain the procedures of obtaining $\mathbf{T_r}$, and then, the server load can be calculated.

4.5.4 Example

In this thesis, we collect the network statistics and user profile using the proposed measurements architecture to obtain $\mathbf{T_r}$ ($r = 1,2,3,4,5,6$) by a set of mathematics calculations. A list of network statistics and user profile is summarized in Table 4.9. Four of them, N_1, U_{rj}, U_j, Q_j, are the user information, which are provided from UA. The remaining two network statistics are Z_j and Z_{sj}, which are collected from SIP servers, P/I/S-CSCF and HSS. So, both UA and SIP servers become EPA. Figure 4.11 illustrates the proposed measurements architecture based on ESP and ENF frameworks implemented in the IMS network system. The procedures of collecting, evaluating, and subscribing the list of network and user data are provided in the following.

When a UA joins a network, it sends a PUBLISH message to notify ES the time it joins the network. ES can calculate N_1 and U_j during a period of time. In addition, when a user requests an application, it publishes the type of application and the type of its routing scenario to ES, so that Q_j and U_{rj} can be obtained at ES.

Session procedures are processed in different types of SIP servers in IMS network, and each of them has a corresponding starting SIP server, as shown in Table 4.10. In order to obtain the number of session procedures triggered in the network, we need to collect information from its starting SIP server. In Figure 4.11 SIP servers are shown as EPA2 for P-CSCF, EPA3 for I-CSCF, EPA4 for S-CSCF, and EPA5 for HSS. Each SIP server maintains a session procedure table that records the number of different types of existing session procedures that started from it. Each of them publishes this information

to the ES periodically. During the run time, Z_{sj}, which is the number of session procedure s triggered in the system at time point j, can be obtained by evaluating the session procedure tables from EPA2, EPA3, EPA4, and EPA5. The total numbers of session procedures in the system trigged at time point j, Z_j can be obtain by adding the number of existing session procedures from EPA2, EPA2, EPA3, EPA4, and EPA5.

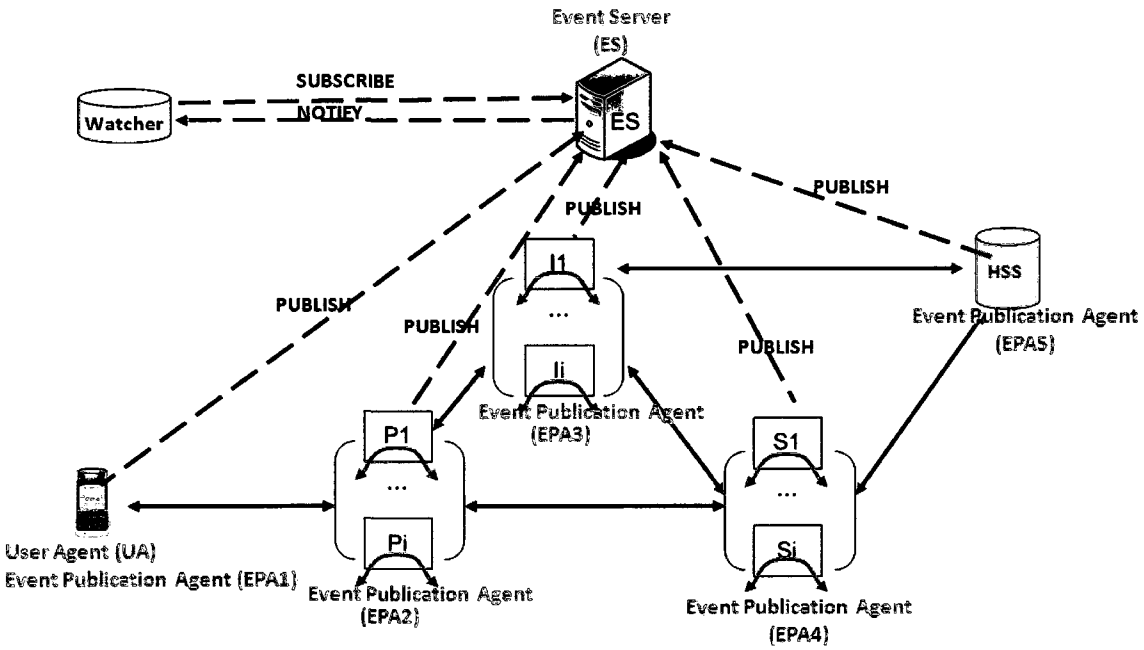


Figure 4.11: The measurements architecture based on ESP and ENF frameworks implemented in IMS network system; UA and SIP servers become EPA

Table 4.10: The starting SIP servers for different session procedures

Procedure Category	No.	Session Procedures	Starting SIP Server
Session Initiation	1	Basic Session Setup	P-CSCF
	2	Re-invite for new codec, without I-CSCF	P-CSCF
	3	Re-invite for reserved codec	P-CSCF
	4	Re-invite, failure happen	P-CSCF
Registration	5	Registration, user not registered	P-CSCF
	6	Re-registration, user currently registered	P-CSCF
	7	Mobile initiated	P-CSCF
De-registration	8	Network initiated, registration timeout	S-CSCF
	9	Network initiated by HSS, administration	HSS
	10	Network initiated, service platform	P-CSCF
Session Termination	11	Mobile terminal initiated session release	P-CSCF
	12	Network initiated session release P-CSCF initiated	P-CSCF
Session Failure	13	Failure in session abandon, origination procedure	P-CSCF
	14	Failure in obtaining resource, origination procedure	P-CSCF
	15	Failure in termination procedure	P-CSCF
	16	Rejection by termination procedure	P-CSCF
Session Redirection	17	Initiated by S-CSCF to CS-domain	S-CSCF
	18	Initiated by S-CSCF to IM CN subsystem	S-CSCF
	19	Initiated by P-CSCF	P-CSCF
	20	Initiated by UE	P-CSCF

Since N_1, U_j, U_{rj}, Q_j, Z_j, and Z_{sj} can all be obtained periodically from EPAs, $\mathbf{T_r}$ can be calculated at the requested time frame. Thus, the server load can be obtained. The following chapter presents the simulation for verification of the proposed traffic model. In the simulation, the proposed measurements architecture is implemented in order to collect the network statistics and user profile, and then, the input data is obtained by the list of calculations discussed above.

Chapter 5

Simulation Results and Discussions

The purpose of this chapter is to evaluate the behavior of the proposed traffic model in IMS network with known traffic parameters such as $\mathbf{T_r}$, which is the average arrival rate of the session procedures for routing scenario r, where $r = 1,2,3,4,5,6$. We devise a source model that simulates the users' behaviors in the IMS network, and we implement the measurements architecture based on ESP and ENF frameworks for collecting the network statistics and user profile. Accordingly, the average arrival rate of session procedures requested by users, N, the distribution of session procedures, $\mathbf{A}$, and the frequency of routing scenario r ($r = 1,2,3,4,5,6$), B_r, are obtained by collecting the user data during the simulation. These information are then use to calculate the input data of the proposed traffic model, $\mathbf{T_r}$. The results calculated from the proposed traffic model and the server utilization and the mean server queuing delay obtained from the simulation are used to verify the proposed traffic model. The proposed traffic model is proven to be acceptable.

5.1 Network Model

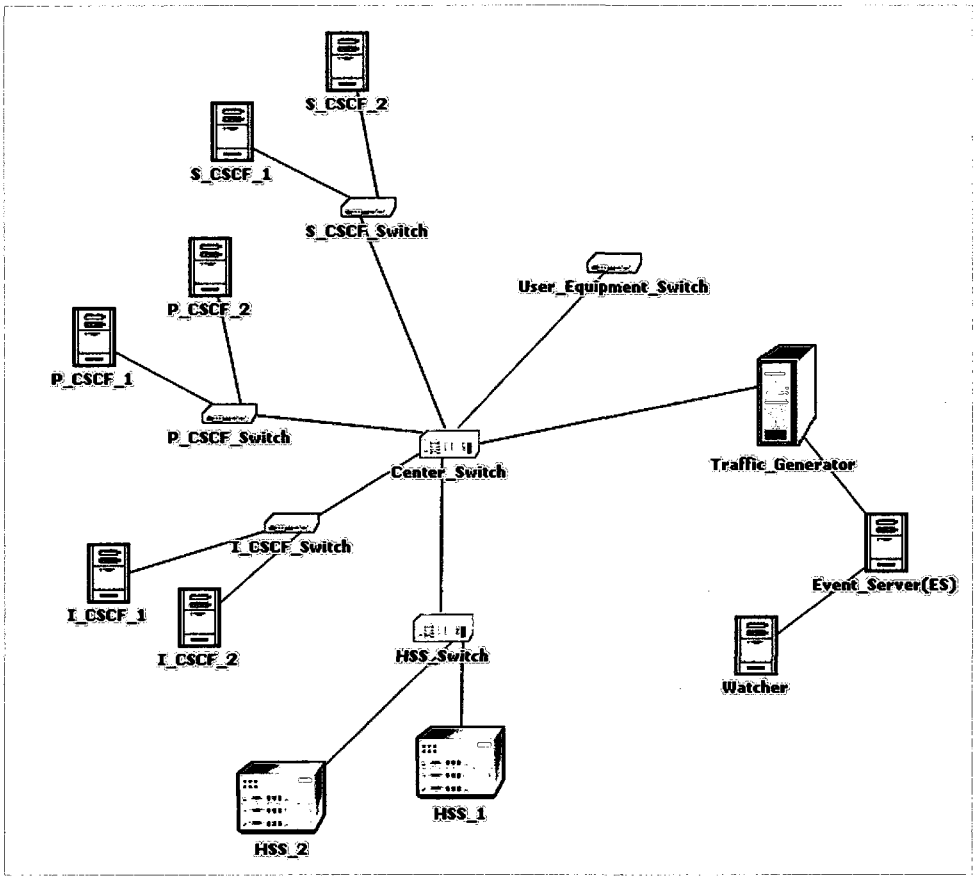


Figure 5.1: Test bench of IMS core network in OPNET, with one traffic generator

Figure 5.1 illustrates the test bench we developed in the OPNET Modeler 14.5 for verifying the proposed traffic model. This model allows us to collect and monitor the network statistics and user profile during the run time. It is a simplified IMS core network, and contains main IMS entities. It effectively realizes the session interactions that allow SIP messages to be exchanged within the network. As shown in Figure 5.1, the network topology consists of 4 types of IMS servers: P-CSCF, I-CSCF, S-CSCF, and HSS, a single unit for UEs, a traffic generator, an ES, a watcher, and a center switch. Traffic

generator acts like EPA, and it publishes network statistics and user profile to ES during the run time. Watcher can subscribe information from ES. The measurement architecture, based on ESP and ENF frameworks, is implemented with the purpose of obtaining $\mathbf{T_r}$.

For simplicity, we deployed two parallel load balancing servers for every type of IMS server. The two paralleled servers are connected to a switch presented in Figure 5.1 with suffix *ServerName_Switch.* This switch plays two roles: first of all, it exchanges messages between the servers and the center switch. Second, it assigns the server to new coming users. Once a server is assigned to a user, this server is attached to this particular user during his period of registration.

One unit of the UE named *User_Equipment_Switch* in Figure 5.1 acts like the end users. The traffic generator in Figure 5.1 is implemented with a source model with the purpose of providing the input data to the simulation. The source model simulates the behavior of users requesting applications in a real network. Each requested application triggers different session procedures. These session procedures generate different SIP messages between IMS servers, which form SIP signaling traffic we discussed in previous chapter. Details on this source model are provided in the next section. The generated traffic sent by the traffic generator is initially received by the UEs, and then forwarded to the destination through the center switch.

Since the traffic generator generates both network and user information, as an EPA, it sends the information to ES every 30 seconds. ES is devised to analyze the network conditions and the characteristics of the IMS users. During the run time, the collected network statistics and user information are used to calculate $\mathbf{T_r}$.

ES has functions to collect the network statistics and user profile, which include $N_1, U_j, U_{rj}, Q_j, Z_j, Z_{sj}$, where j=1, 2, 3, 4..., J, r=*1,2,3,...,6*, and s=*1,2,3,...,20*, as shown in Table 4.9. The details of how to calculate $\mathbf{T_r}$ are discussed in Section 4.5.3. The results, $\mathbf{T_r}$, are stored in ES. Once the received information is updated, it requires recalculation. The calculated results, $\mathbf{T_r}$, are sent to watcher every 30 seconds when the watcher subscribes to ES.

The watcher connects directly to the ES. It subscribes to the value of $\mathbf{T_r}$ from ES. $\mathbf{T_r}$ represents the average arrival rate of 20 session procedures in routing scenario r in the simulation.

In the simulated IMS network, each server records its utilization and the mean server queuing delay during the simulation. The server utilization, ρ_v, is obtained by dividing the server busy time by the sum of the server busy time and server idle time. The method of obtaining server utilization is identical to the formulation provided in Equation 4.3, and the mean server queuing delay is the average waiting time in which SIP messages waiting for the process. The calculations are performed every 60 seconds. These data is used to verify the proposed traffic model. If two calculated results obtained from the proposed traffic model fall in the 95% confidence interval of the simulation results, we say the proposed traffic model is acceptable.

In order to facilitate message exchanges among the individual entities, a center switch is used to connect the servers and the traffic generator. The center switch emulates the operation of an IP network. Messages are routed as defined in IMS. In the following section, we provide the assumptions made to the simulation.

5.2 Network Assumptions

In the simulation, we need to develop a source model for the traffic generator, which can simulate the user behavior in IMS network. The IMS servers are implemented as a queuing model for processing the received SIP messages.

Source Model

The purpose of developing a source model for the simulation is to provide input data to later verify the proposed traffic model. This input data should be a representative of the users' behavior in the IMS network in real life. Therefore, we develop a source model to replicate the behavior of IMS users as they act in reality.

Before devising the source model, we should understand the general users' behavior in IMS network. Every user enters into the IMS network with the purpose of requesting various available applications. In general, users request one application at a time and stay for this application within a time period, say, 10 minutes on average. They may request another application after some time, say, 30 minutes on average. The users may request a number of applications in total during their stay in the IMS network. Then, users leave the IMS network when no more applications needed to be requested. The user's life in IMS generally lasts several hours, say 6 hours on average.

From the understanding of users' behavior in the IMS, a source model takes into account: i) the user inter-arrival time, ii) the number of applications requested by one user, iii) the application inter-arrival time, iv) the application duration, v) the duration of the user's staying in the system. Since we need to simulate the behavior of IMS users in a

source model, we make the assumptions for each of above listed factors, so that they can approach the real case.

Figure 5.2 shows the procedures of every user's behavior in IMS network. The user entered into the IMS network is assumed to follow with the exponential distribution with the mean of 0.5 second. The user is assumed to request one available application at a time as well as to stay for this application, with a mean duration of 10 minutes. We assume that the inter-arrival time of the requested applications follow an exponential distribution with the mean of 30 minutes, respectively. The duration of users staying in the system is assumed to be exponential distributed and is set to be 6 hours (9:00am to 3pm) per user.

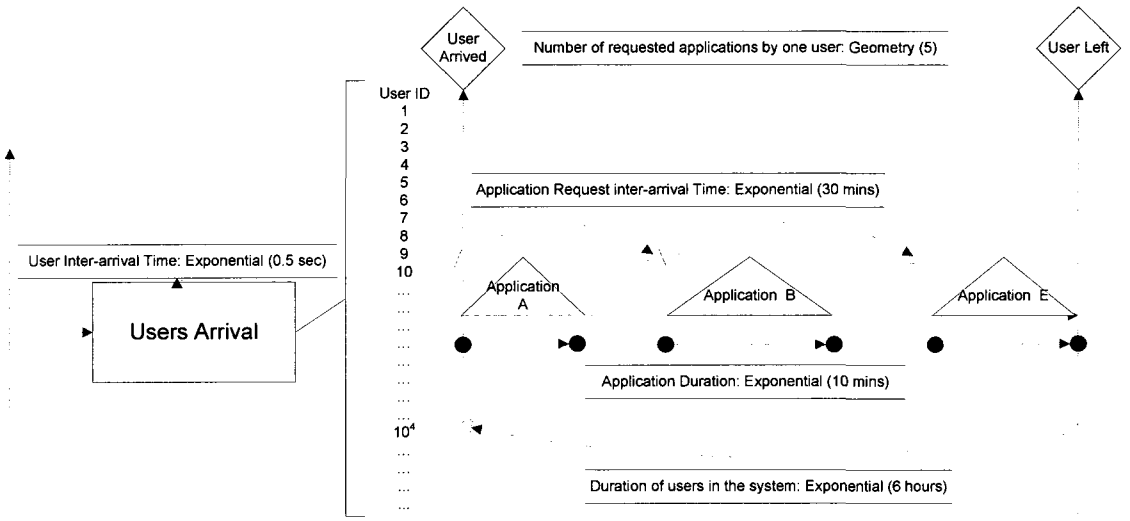


Figure 5.2: A source model developed for the simulation

The routing scenario that a user belongs to, as a parameter of source model, is determined and applied into the simulation. The frequencies of 6 routing scenarios should be given by the network operators from the statistics of the past user data. However, since

there are no such publications that are related to this issue, we have to make assumptions regarding the frequencies of 6 routing scenarios to the system for the time being. Since we understand that the users request the applications from their own network operator with a much higher possibility rather than from other network operators, we assume there are 60% chances that a user as originator in (H_1, O) routing scenario. Moreover, we assume the user is located in a different network operator from his own as an originator in (V_1, O) or (V_2, O) routing scenario with the same possibility. Then, each of (V_1, O) and (V_2, O) routing scenarios is assumed to be 20% chances. As we know, one session procedure is composed of both originating and terminating routing scenarios. We also understand that a user, as terminator in his own network operator, (H_1, T), is called and involved in the session with the callee/originator, with a much higher chances, compared with the user as terminator in (V_1, T) or (V_2, T). Then, we assume 60% chances that a user as terminator in (H_1, T) routing scenario as well as each of (V_1, O) and (V_2, O) routing scenario accounts for 20% chances.

When a user arrives, he requests one or more applications during his stay in the IMS. Table 5.1 lists five general application classes as well as the available applications for each class. The distribution of the five application classes is obtained from the measurements of the real experiments provided in [45]. We assume there are 5 application classes available for requesting by the users, and the user can only request one application class at one time. The frequency of requesting the application classes are assumed to follow the data presented in Table 5.1. For simplicity, we use the word "application" to replace the application class, in later text.

Table 5.1: Distribution of application classes requested by users through laptop PC

Class #	Application Class	Packet Data Applications	Frequency (%)
1	Internet Gaming	Quake II, World of Warcraft	2.62
2	Voice over IP (VoIP)/Video Conference	VoIP, Video Conference	26.19
3	Streaming Media	Video Clip, Movie Streaming, Music/Speech	10.37
4	Information Technology	IM, Email(POP3, IMAP), Telemetry	59.08
5	Media Content Download/Backup	FTP, P2P	1.74

In a real IMS network, a number of session procedures are sequentially triggered by the requested applications, and the procedures for triggering the session procedures are logical. Figure 5.3 shows the logical diagram of 20 session procedures that are triggered one by one. It is well known that every user is required to register before requesting any available applications for the first time, and re-registration may be required after some time, due to the new requested applications or other reasonable cases. In our case, for simplicity, we require the user to register at the time of requesting any new application. The registration procedure, session procedure 5 or 6, is triggered and determined by a system randomly. In either of these two session procedures, the user registers successfully, thus the authorized user to start requesting the applications. Any requested applications require a session establishment, where a session redirection may occur by the determination of various network nodes; the involved session procedure includes

session procedure 17, 18, 19, and 20. If the session redirection does not take place, the session initiation, session procedure 1, 2, 3 or 4, is triggered. Session procedures 1, 2, and 3 allow the users to establish the session successfully. Users fail to establish the session in session procedure 4. Moreover, the session may also fail due to the potential factors in a real network. In this case, a session failure, session procedure 13, 14, 15, or 16, is performed. For simplicity, the session is only allowed to be established once, and if it fails, it is final. After the session is established, the media data passes between two or more end users. At the same time, network nodes or end users may attempt to deregister in some particular cases. Accordingly, in the case of deregistration (session procedure 7, 8, 9, 10), the session will be automatically terminated. For simplicity, a deregistration attempt is only allowed to happen after the session is successfully established in our case. If the ongoing session is terminated at the users' request or network's request, a session termination, session procedure 11 or 12, is performed. At this point, the session is finally released. The choices of session procedures are made by a system of random selection at the particular points, as shown in Figure 5.3. Therefore, every requested application triggers different session procedures.

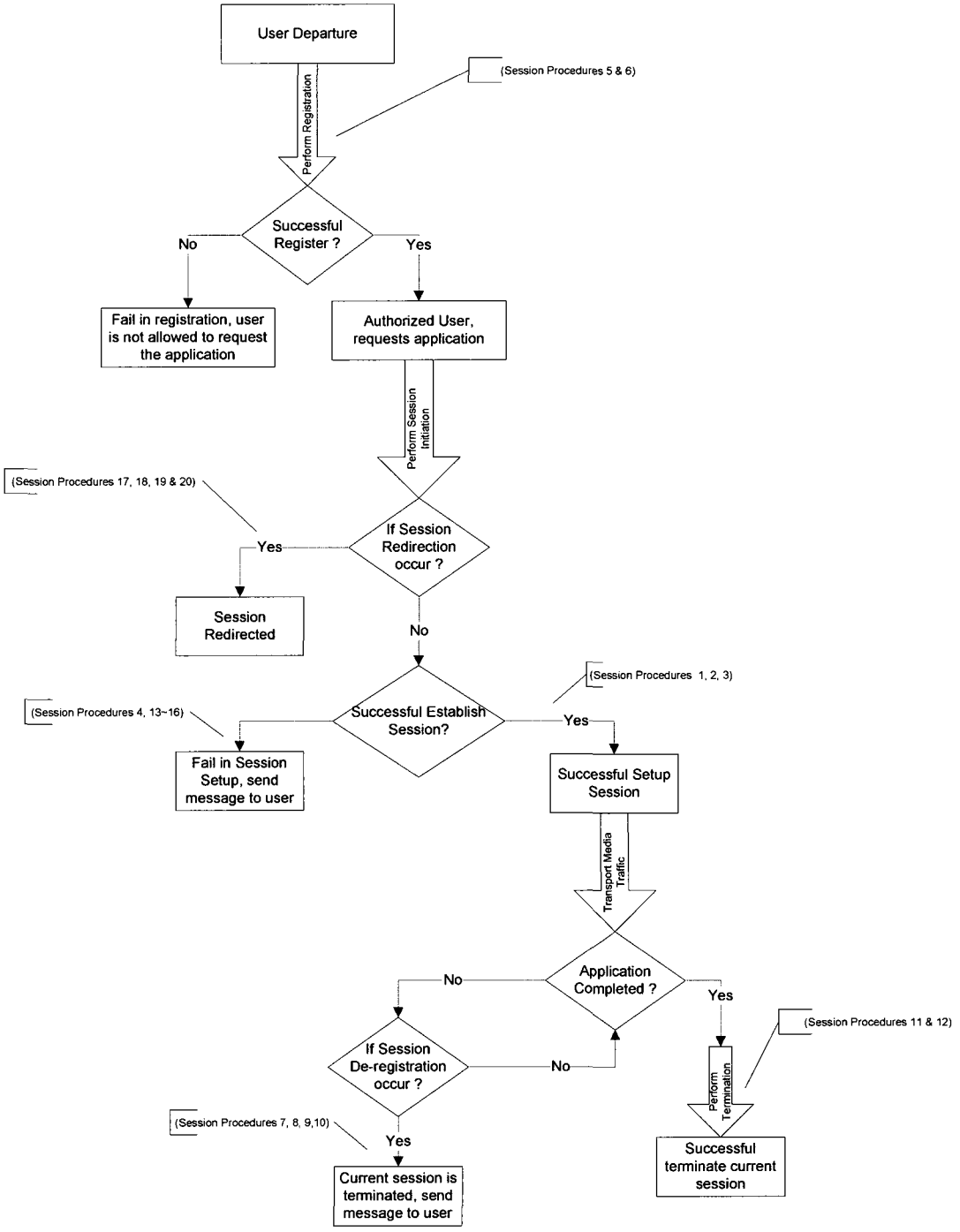


Figure 5.3: The logical diagram for 20 session procedures SIP Servers

We used the M/M/1 queuing model for the network IMS servers, as illustrated in Figure 5.4, and assumed service time for each SIP message arrived at IMS servers follows the exponential distribution. The simulation results, server utilization, and the mean server queuing delay are collected and discussed in the following section.

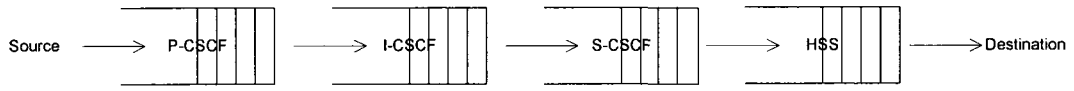


Figure 5.4: IMS queuing network

5.3 Simulation Results

We collect two sets of simulation results based on the different service time settings. In the first simulation, the average service times for four IMS servers are set to be the same, which are 2.5 milliseconds per message. As mentioned earlier, the service of every two parallel servers is equally distributed. The utilization values of these two servers remain at the same level; therefore, we only collect one of them. Then, the simulation is set up and has duration of 48 hours. Figure 5.5 demonstrates the instantaneous utilizations of the four IMS servers recorded by every 60 seconds within the 48 hours of simulation period. Moreover, the mean server queuing delays are shown in Figure 5.6. In this simulation, the service times for all servers are set to be the same. Subsequently, the utilization of S-CSCF server is shown to be the highest value than the rest of servers.

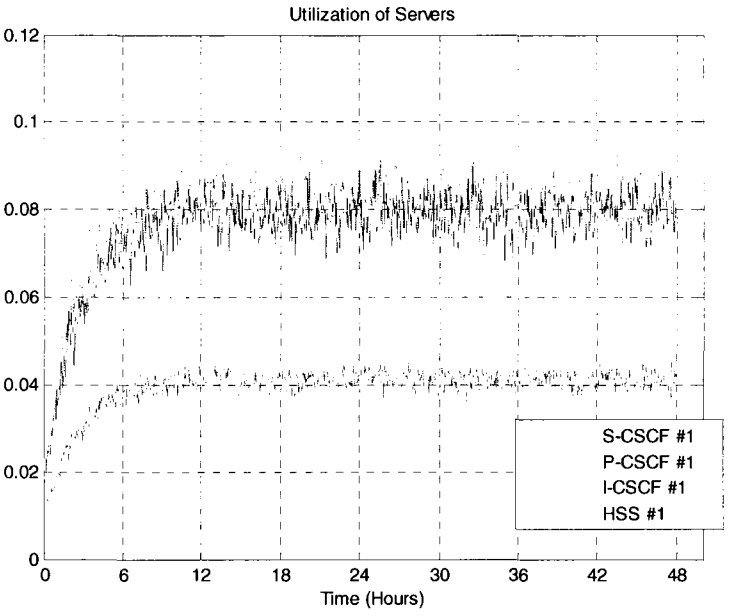


Figure 5.5: Server utilization recorded every 60 seconds for 48 hours, with the service time of 2.5 milliseconds per message for 4 IMS servers

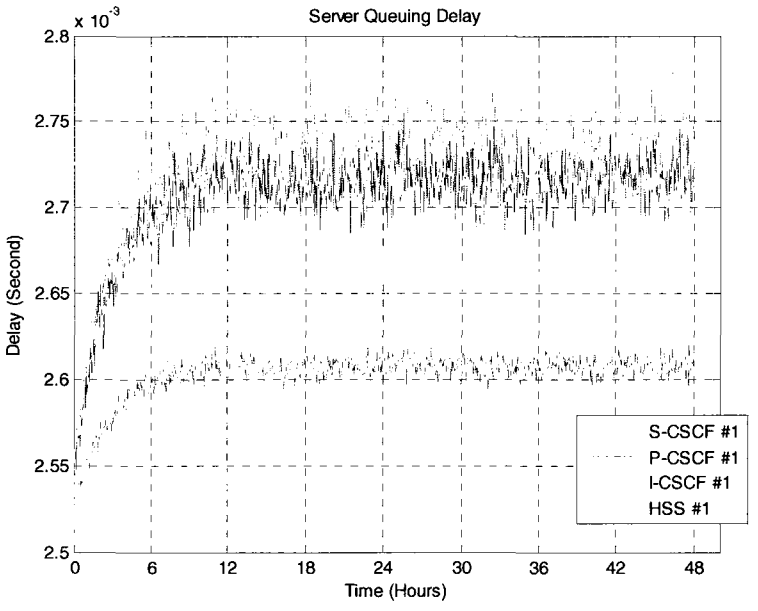


Figure 5.6: Mean server queuing delay recorded every 60 seconds for 48 hours, with the service time of 2.5 milliseconds per message for 4 IMS servers

However, in IMS systems, every IMS server is defined to have their task of processing messages that contain various information come from users. Some servers are designed to perform more complex and time-consuming message processing tasks. For instance, S-CSCF, as a central node of the signaling plane, not only takes on the responsibilities of the registration for the users, but it also performs the service control for them via the interaction with the HSS. Therefore, the service times for four servers should be set differently. According to message processing capability of each server, the service times in the second set simulation are set to be 1/100, 1/70, 1/50, and 1/50 seconds per message for S-CSCF, P-CSCF, I-CSCF, and HSS, respectively. The simulation also has the duration of 48 hours. The simulation results, instantaneous utilizations and the mean queuing delay for the 4 servers are discussed in Section 5.5.

In these two sets of simulations, a source model that simulates the user behavior is implemented. We require the data from the second simulation for calculating the input data of the proposed traffic model, $\mathbf{T_r}$. During the second simulation, a list of user profile and network statistics are stored in ES, as presented in Table 4.9. The following section discusses the computations performed on ES in order to calculate the three data N, $\mathbf{A}$, B_r, where r=1,2,3,...6; this is done so that $\mathbf{T_r}$ can eventually be obtained.

5.4 The Computations of the Input Data $\mathbf{T_r}$ Based on Simulation Results

This section provides details regarding the calculation of N, $\mathbf{A}$, B_r, based on the collected information on ES. Also, the procedures of the computations are provided in Section 4.5.3. Finally, the input data of the proposed traffic model, $\mathbf{T_r}$, can be obtained.

All the computations are performed and stored at ES. The network provider can subscribe $\mathbf{T_r}$ through the watcher when the simulation is finished, so that we can verify the proposed traffic model, since we obtained the simulation results from previous section.

5.4.1 Average Arrival Rates of Session Procedures, N

As mentioned, the users enter into the IMS network following the exponential distribution with the mean of 0.5 second. The average user arrival rate, N_1 (user/second) is 2, and the duration in which users remain in the system is assumed to be exponential distribution and is set to be 6 hours. Figure 5.7 demonstrates the current number of users in the system during the 48 hours simulation. We can see that during the first 6 hours of the simulation, the number of users in the system increases linearly. This indicates that there are few or no users leaving the system. However, in the next 6 hours, the increase in the number of users in the system decreases in comparison to the previous 6 hours. This implies that some users who entered into the system during the first 6 hours are now leaving, yet the number of users entering is still larger than the number leaving. Up to this point, the number of users in the system is still increasing and does not reach the stable point. Moreover, in the fourth 6 hours, namely, during the18^{th} and 24^{th} hours, the number of users in the system is nearly constant. After this time point, the number of users which entered and left the system almost stabilizes. Finally, the system reaches stability after the 18^{th} hour of simulation. In addition, Figure 5.8 illustrates the cumulative number of users which entered the system up to the time point during the simulation, which reaches $3.5x10^5$.

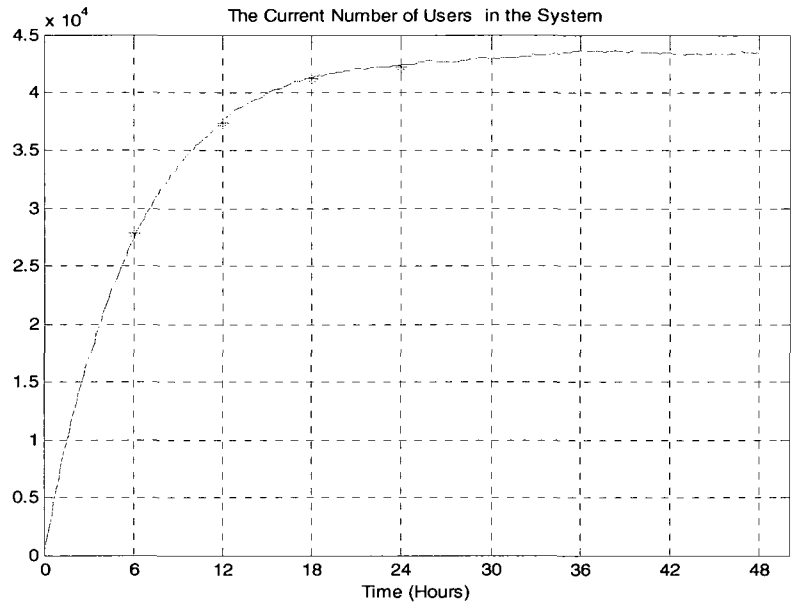


Figure 5.7: The current number of users in the system during the 48 hours simulation

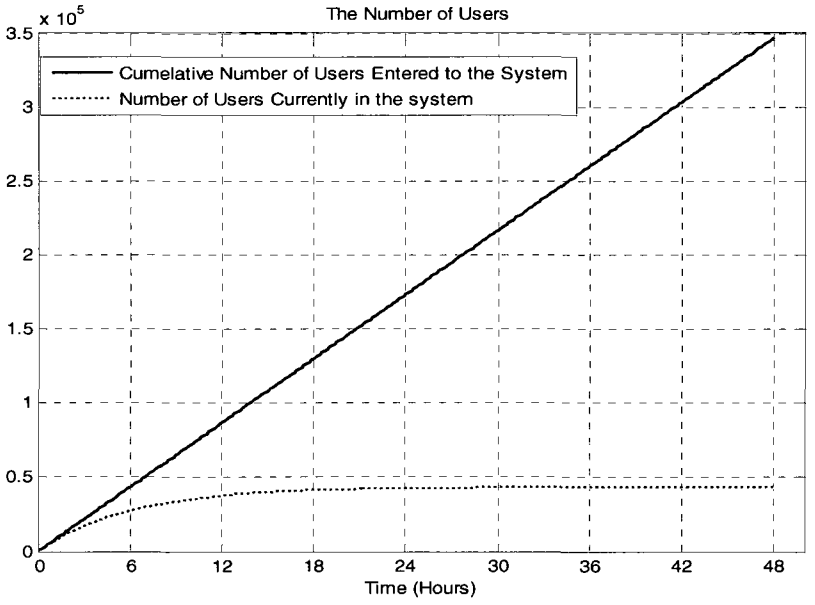


Figure 5.8: The cumulative number of users entered the system and the current number of users in the system

Every user can request one or more applications during his staying in the system. There are five applications available for users to choose from. The application's duration and inter-arrival time of requested applications both followed exponential distribution with the means of 10 minutes and 30 minutes, respectively. Figure 5.9 demonstrates the number of applications currently running in the system. During the 12^{th} hour, the number of applications reaches 6000 and remains constant for the duration of the simulation.

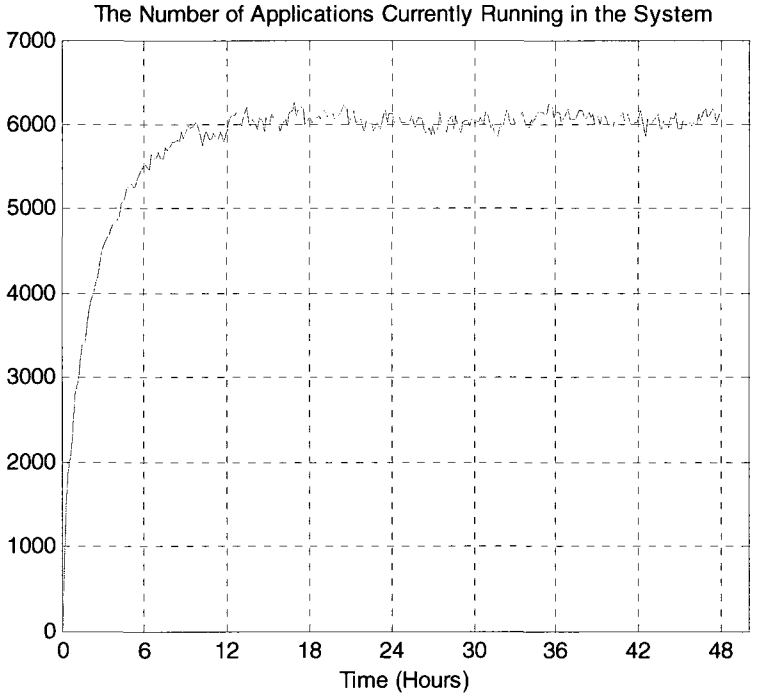


Figure 5.9: The current number of applications running in the system.

In addition, Figure 5.10 shows the total number of applications requested by the users at the time point j, Q_j, and the total number of users which enter the system at the time point j, U_j, where j=1, 2, 3, 4..., J, and J is the number of time points. These two

sets of data increase linearly as the simulation time passed. The average number of applications requested by one user, N_2, can be calculated via Equation 4.14 using the values of 3 points picked from Figure 5.10.

$$N_2 = \frac{1}{J}\sum_{j=1}^{J}\frac{Q_j}{U_j} = 4.67 \qquad \textbf{5.1}$$

The value of N_2 is close to the network setting for the number of requested applications by one user, which abides by the geometry distribution with the mean of 5, as illustrated in Figure 5.2.

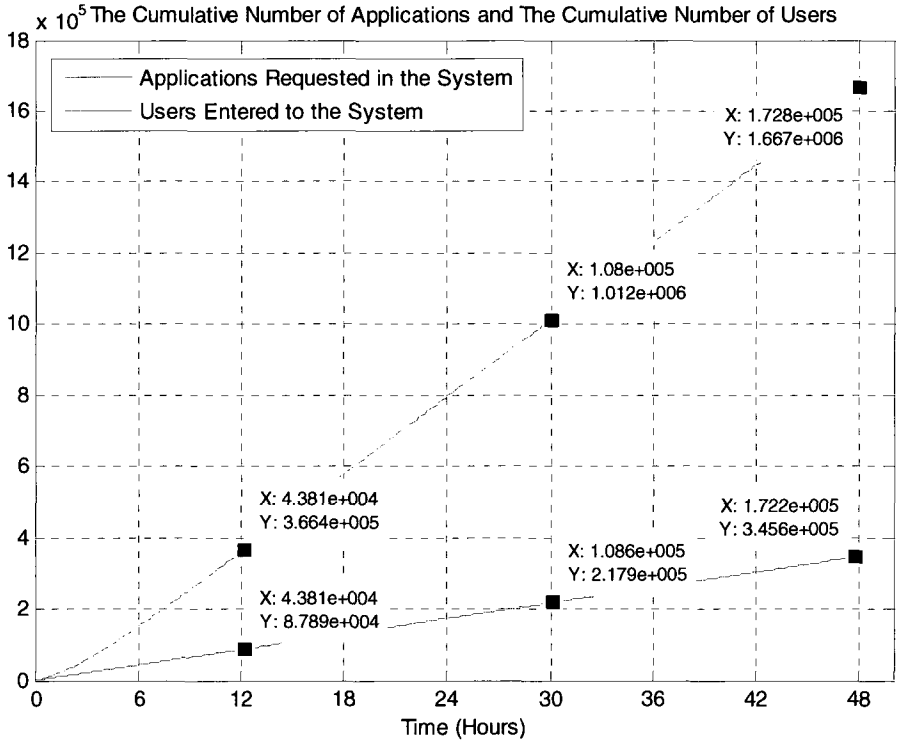


Figure 5.10: The cumulative number of applications requested and the cumulative of users entered the system up to the time point

Furthermore, as we have discussed, a number of session procedures are sequentially triggered by the requested applications. The relationship between the total number of requested applications at the time point j , Q_j , and the total number of session procedures triggered by applications at the time point j , Z_j , where j=1, 2, 3, 4..., J, are shown in Figure 5.11. Then, the average number of session procedures triggered by one application, N_3, can be calculated via Equation 4.15 using the values of 3 points picked from Figure 5.11 as follows:

$$N_3 = \frac{1}{J}\sum_{j=1}^{J}\frac{Z_j}{Q_j} = 2.151 \qquad \textbf{5.2}$$

Thereby, the first useful user data is the average arrival rates of session procedures triggered, N , and it can be calculated by Equation 4.13, as follows:

$$N = N_1 \cdot N_2 \cdot N_3 = 2 \cdot 4.67 \cdot 2.151 = 20.09 \text{ (session procedure/second)} \qquad \textbf{5.3}$$

It indicates that there are 20 session procedures triggered by every second. The following section focuses on calculating **A**.

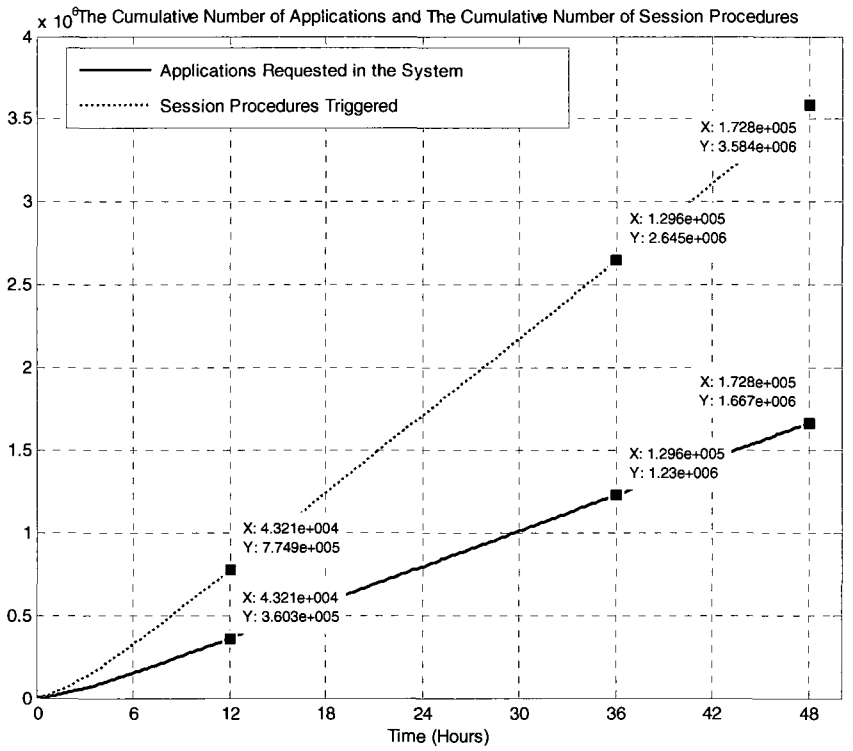


Figure 5.11: The cumulative number of applications requested and the cumulative number of session procedures triggered up to the time point

5.4.2 Distribution of Session Procedures, A

In order to obtain the distribution of 20 session procedures, **A**, the total number of every session procedure triggered in the system is recorded by the ES, and it is shown in Figure 5.12. There are five different categories of procedures. Each category is further divided into several cases. However, deregistration is considered a separate category and is now split from the registration category. It is noticed that there is an overlap between the values of the failure category and the termination category, as shown in Figure 5.12.

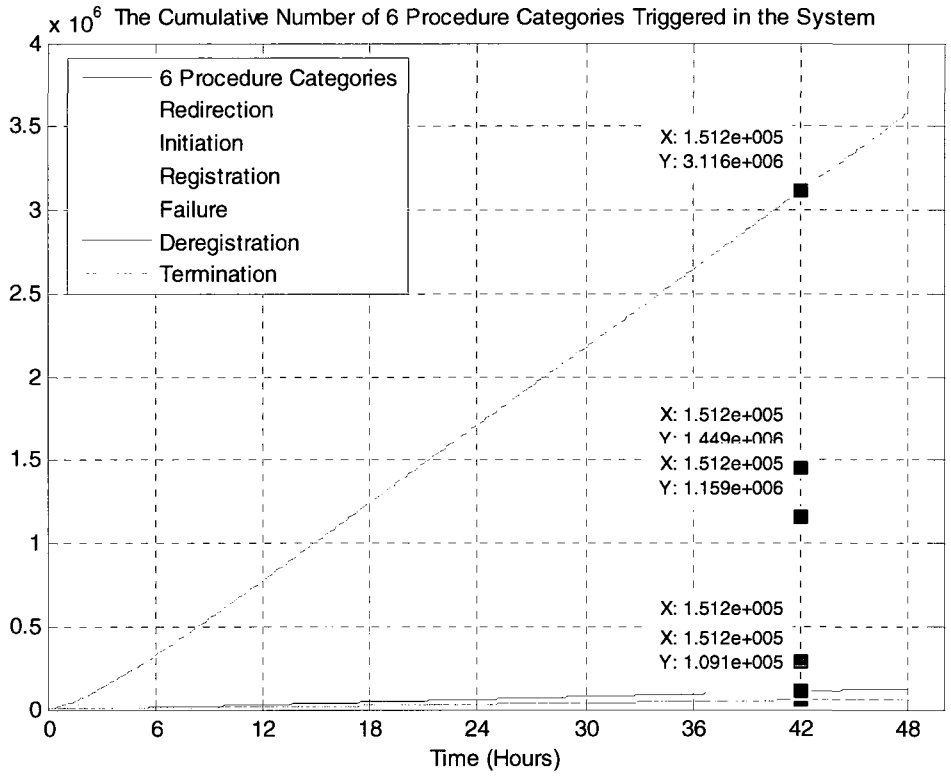


Figure 5.12: The cumulative number of 6 procedure categories triggered up to the time point

Figure 5.13 is a zoom in version of Figure 5.12 in the first 6 hours of simulation. The distribution of 6 procedure categories is calculated by the data collected from Figure 5.12, given that their values appear linearly. Table 5.2 provides the final distribution of the 6 procedure categories. We know that the session procedures for each category are selected randomly by the program, and that in a long run, their amounts are showed as equally distributed. Finally, Table 5.3 provides the distribution of 20 session procedures for all the requested applications that is **A**, which is calculated by Equation 4.16 according to the

points picked from Figure 5.12. Since we obtain N and $\mathbf{A}$, B_r is calculated in the following section.

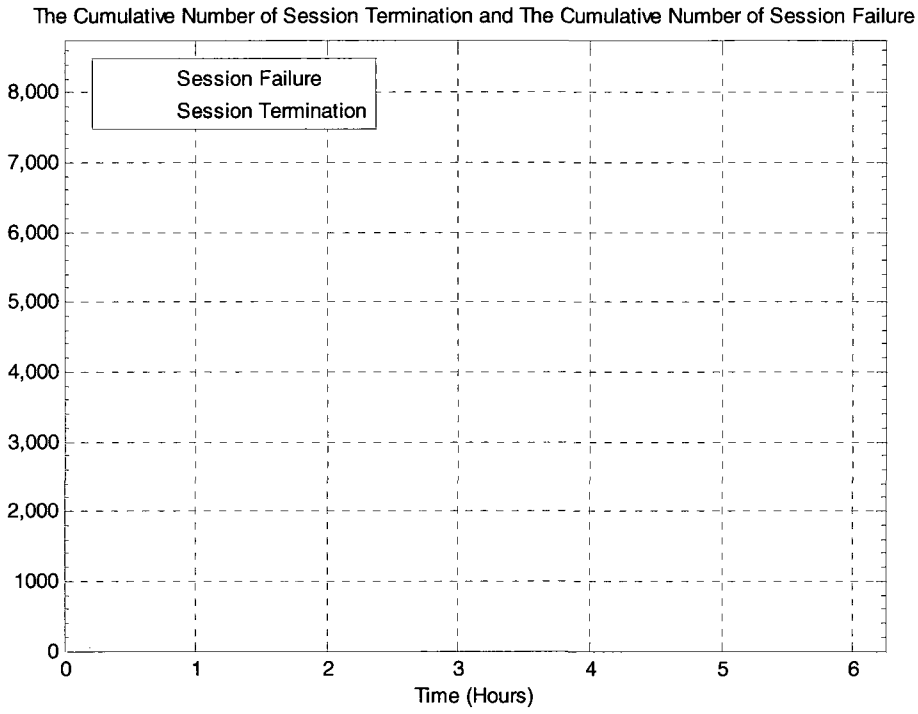


Figure 5.13: Zoom in Figure 5.12, the cumulative number of session termination and the cumulative number of session failure up to the time point

Table 5.2: The distribution of 6 categories procedures obtained from Figure 5.12

Procedure Category	The number of 6 categories of procedures	Proportion
6 categories	3.129×10^6	100%
Registration	1.449×10^6	46.3%
Session Initiation	2.902×10^5	9.27%
Session Redirection	1.159×10^6	37.33%
De-registration	1.091×10^5	3.5%
Session Failure	5.435×10^4	1.8%
Session Termination	5.435×10^4	1.8%

Table 5.3: The distribution of 20 session procedures obtained from Figure 5.12

Procedure Category	*No.*	Session Procedures	Frequency
Session Initiation	1	Basic Session Setup	2.3175%
	2	Re-invite for new codec, without I-CSCF	2.3175%
	3	Re-invite for server codec	2.3175%
	4	Re-invite, failure happen	2.3175%
Registration / De-registration	5	Registration, user not registered	23.15%
	6	Re-registration, user registered	23.15%
	7	Mobile initiated	1.75%
	8	Network initiated, registration timeout	1.75%
	9	Network initiated by HSS, Administration	1.75%
	10	Network initiated, service platform	1.75%
Session Termination	11	Mobile terminal initiated Session release	0.9%
	12	Network initiated session release P-CSCF initiated	0.9%
Session Failure	13	Failure in Session abandoned, or resource	0.45%
	14	Failure in origination procedure	0.45%
	15	Failure in termination procedure	0.45%
	16	Rejection by termination procedure	0.45%
Session Redirection	17	Initiated by S-CSCF to CS-domain	9.3325%
	18	Initiated by S-CSCF to IM CN subsystem	9.3325%
	19	Initiated by P-CSCF	9.3325%
	20	Initiated by UE	9.3325%

5.4.3 The Frequency of Routing Scenario *r*, B_r

As reviewed, the implementation of this network determines the status of the users before entering the system, namely, which routing scenario the user belongs to. There are 60% chances that a user as originator in (H_1, O) routing scenario, and for the remaining 40% chances, each of (V_1, O) and (V_2, O) routing scenarios accounts for half. Moreover, 60% chances are set up for (H_1, T) routing scenario, and each of (V_1, O) and (V_2, O) routing scenarios accounts for a half of the remaining 40% chances. Let's see if the simulation results match the assumption made to the system. Figure 5.14 presents the number of

users in each routing scenario throughout the 48 hours simulation. In this figure, we can see that the values of (V_1, O), (V_1, T), (V_2, O), and (V_2, T) are well matched together in a long run, since each of them accounts for 20% chances. Moreover, the value of (H_1, O) matches together with (H_1, T) due to 60% chances. In order to examine the small differences among them in the short run, both Figure 5.15 and Figure 5.16 present a zoom in version of Figure 5.14 in the first 20 minutes, for originating and terminating routing scenarios, respectively. Then, we need to calculate the frequency of routing scenario *r*, B_r, where *r=1,2,3,...6*, by means of Equation 4.17 and using the data collected from Figure 5.14.

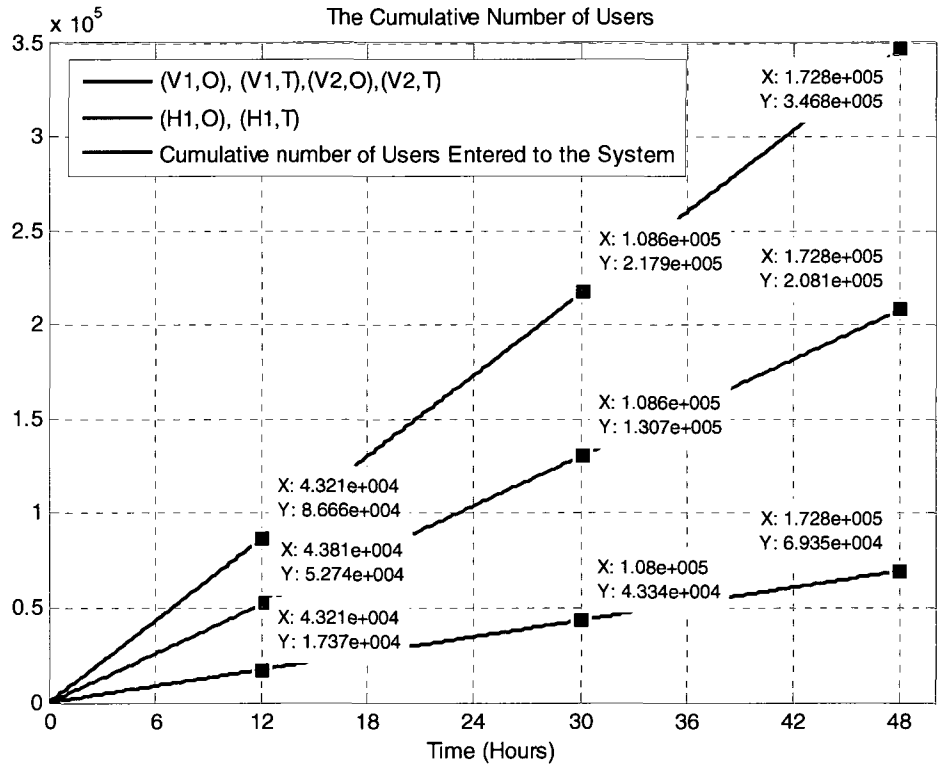


Figure 5.14 The cumulative number of user belongs to each routing scenario and the cumulative number of users entered to the system

Table 5.4 provides the final frequency of routing scenario r, B_r. Then, we utilize the values of N, **A**, and B_r in order to calculate $\mathbf{T_r}$ for further verification of the proposed traffic model in the next section. Now, N, **A**, and B_r,are obtained from the previous sections, the input data of the proposed traffic model, $\mathbf{T_r}$, can be calculated in next section.

Table 5.4: The frequency of routing scenario obtained from Figure 5.14

Routing Scenario	Average Frequency (%)
6 Routing Scenarios	100
(H_1, O), (H_1, T)	59.97%≈60%
(V_1, O) ,(V_2, O) (V_1, T) (V_2, T)	20.01% ≈ 20%

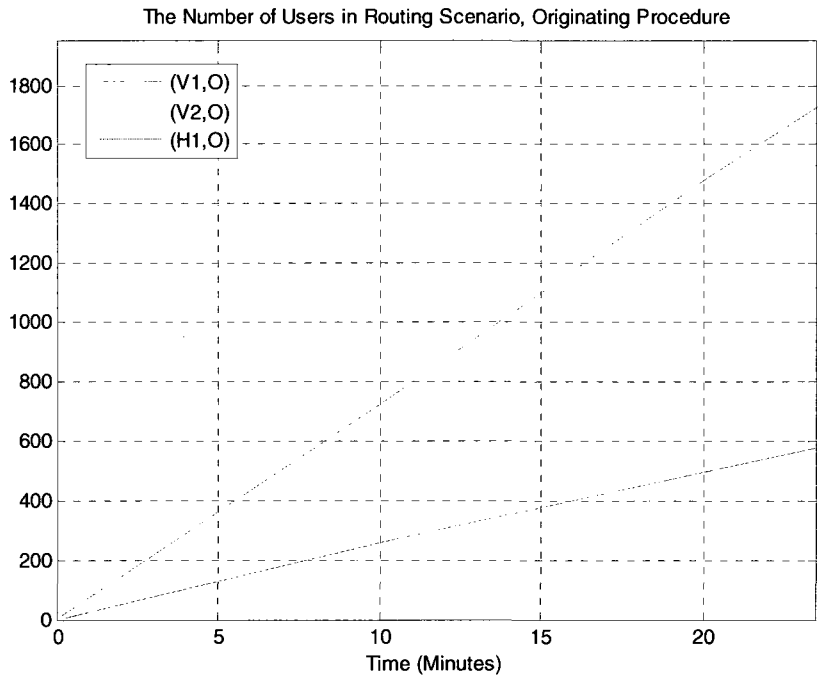


Figure 5.15 Zoom in Figure 5.14, for originating routing scenarios

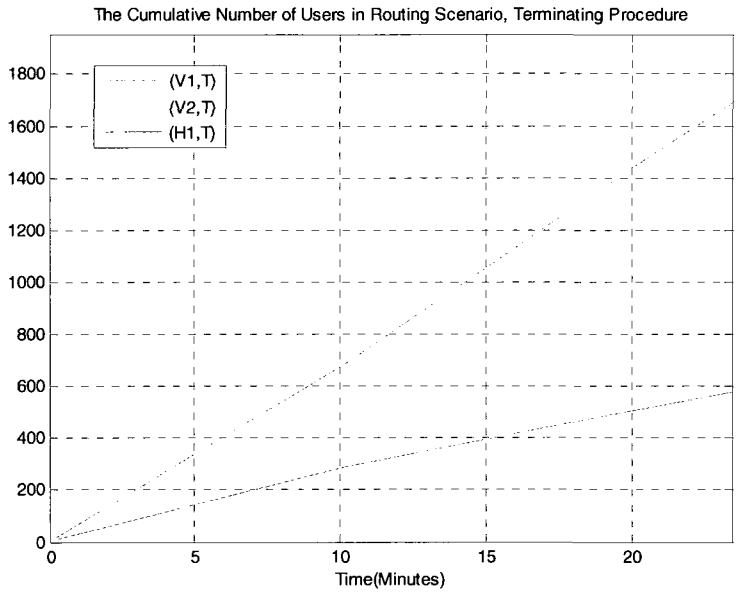


Figure 5.16: Zoom in Figure 5.14, for terminating routing scenarios

5.4.4 The Computations of Input Data, $\mathbf{T_r}$

Based on the distribution of session procedures, **A**, the frequency of routing scenario *r*, B_r, and the average arrival rate of session procedures, N, the average arrival rate for the session procedures in routing scenario *r*, namely, a row vector $\mathbf{T_r}$, can be estimated according to Equation 4.12. The following section presents the verification of the proposed traffic model by applying $\mathbf{T_r}$ as the input data.

5.5 Verification of the Proposed Traffic Model

We apply $\mathbf{T_r}$ into Equation 4.1 and 4.2 in order to obtain the load of each IMS server. Since the mean service time is set up for each server differently and the mean service rates μ_v can be obtained by taking the reciprocal of the mean service time. The utilization of each server, ρ_v, is calculated based on Equation 4.3.

The utilization values, ρ_v, which are calculated via Equation 4.3, are illustrated in Figure 5.17 as straight lines; their values are 0.375 for S-CSCF (95% confidence interval, 0.294 to 0.3797), 0.447 for P-CSCF (95% confidence interval, 0.376 to 0.498), 0.482 for I-CSCF (95% confidence interval, 0.415 to 0.4747), and 0.342 for HSS (95% confidence interval, 0.278 to 0.358). The utilization values in Figure 5.17 remain constant throughout the 48 hours simulation for all four servers. Since the calculated results fall in the 95% confidence interval of the simulation result, the proposed traffic model is verified.

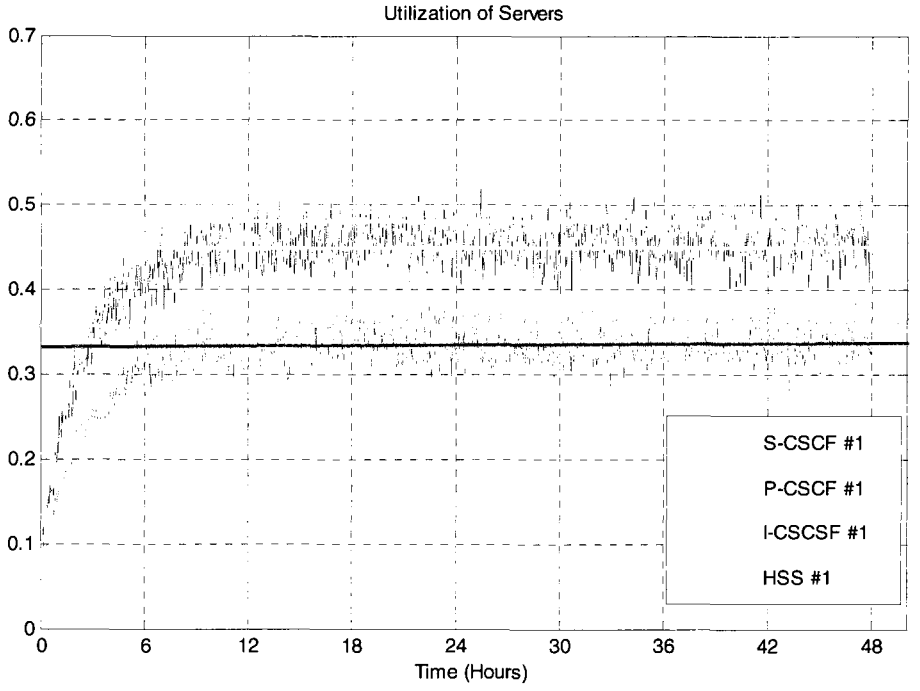


Figure 5.17: Server utilization recorded every 60 seconds for 48 hours, with the different service times for 4 servers, and calculated server utilization as straight lines

Figure 5.18 shows the average server queuing delay recorded by 60 second observation instances over a period of 48 hours. Since the M/M/1 queuing model is

assumed for the network servers P/I/S-CSCF and HSS by using results from the queuing theory [46], the average queuing delay at the server v is as follows,

$$D_v = \frac{1}{\mu_v - \lambda_v} = \frac{1}{\mu_v - \mu_v \rho_v} \quad \textbf{5.4}$$

where μ_v is the mean service rate (message/second) for server v, λ_v is the average arrival rate (message/second) for server v, and ρ_v is the utilization of server v. The server utilization for four servers can be calculated as followed:

S-CSCF #1:

$$D_1 = \frac{1}{\mu_1 - \mu_1 \rho_1} = \frac{1}{100 * (1 - 0.357)} = 0.015552 \text{ sec} \quad \textbf{5.5}$$

P-CSCF #1:

$$D_2 = \frac{1}{\mu_2 - \mu_2 \rho_2} = \frac{1}{70 * (1 - 0.447)} = 0.02583 \text{ sec} \quad \textbf{5.6}$$

I-CSCF #1:

$$D_3 = \frac{1}{\mu_3 - \mu_3 \rho_3} = \frac{1}{50 * (1 - 0.482)} = 0.03861 \text{ sec} \quad \textbf{5.7}$$

HSS #1:

$$D_4 = \frac{1}{\mu_4 - \mu_4 \rho_4} = \frac{1}{50 * (1 - 0.342)} = 0.03039 \text{ sec} \quad \textbf{5.8}$$

The mean server queuing delays for the four servers are illustrated in Figure 5.18 as straight lines; they are 15.52 ms for S-CSCF (95% confidence interval, 13.78 to 15.88), 25.83 ms for P-CSCF (95% confidence interval, 24.78 to 27.16), 38.61 ms for I-CSCF (95% confidence interval, 35.22 to 39.38), 30.39 ms for HSS (95% confidence interval,

29.06 to 31.56). From the simulation results, the proposed model is proven to be acceptable.

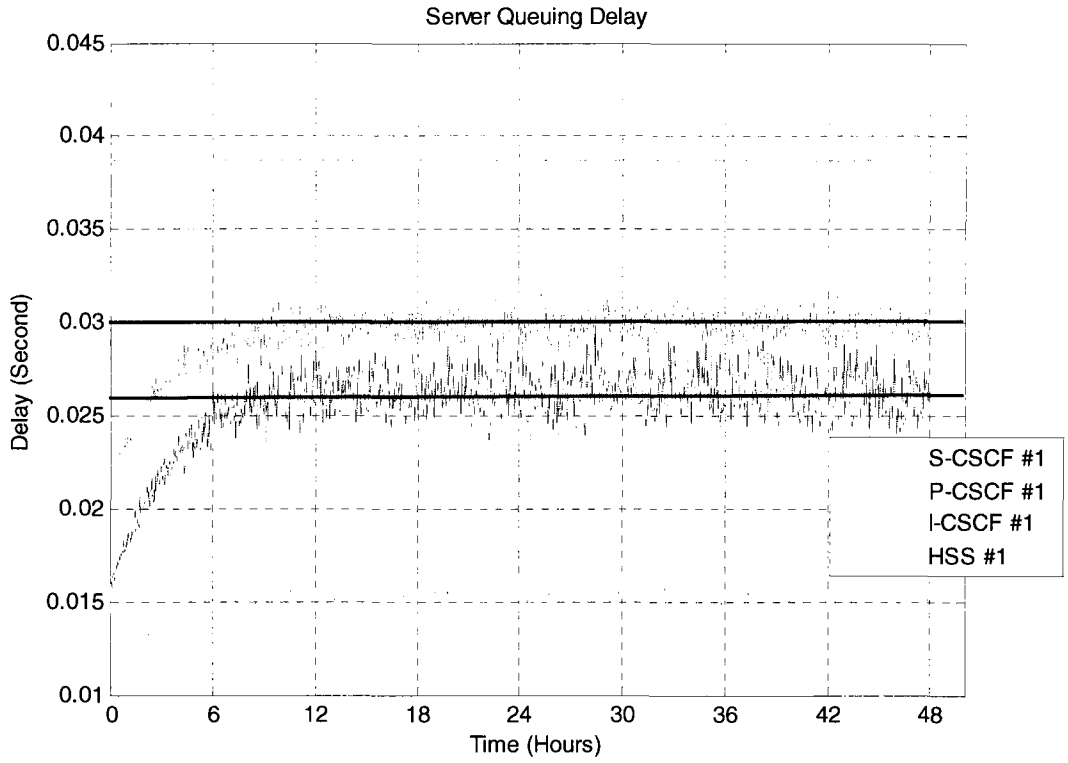


Figure 5.18: Mean server queuing delay recorded for 48 hours, with the different service times for 4 servers, and calculated mean server queuing delays as straight lines

Chapter 6

IMS Network Design Problem

This chapter concentrates on formulating IMS network deployment cost optimization issue as a linear programming problem. The proper mathematical notations for network modeling are introduced. The logical IMS network can be mapped to different IMS physical networks with different mapping strategies, where the logical SIP servers are mapped into the physical node(s). In this chapter, we discuss on three potential mapping strategies. In addition, the proposed signaling flow based traffic model is used to characterize the SIP traffic and reflect the traffic load on SIP servers. Then, the load of the physical nodes can be obtained, depending on the applied mapping strategy. Each mapping strategy is formulated as an IMS network deployment cost optimization problem. In order to better understand how to formulate the cost optimization in a specific strategy, we present an example for each mapping strategy and provide the detailed procedures.

6.1 Motivation

Modeling and design of IMS network have always been an important area to both researchers and network providers. Our interest is in the area of developing efficient

design models and optimization methods for IMS network. In our research, we focus on IMS network deployment cost optimization to produce a good network design potentially capable of securing considerable saving. In this Chapter, the mathematical modeling and the application of efficient optimization methods are applied. We, as designers, have to make selective use of various available theoretical models and different approximations, such as physical node capacity and physical link bandwidth. Also we consider various practical constraints in specific models.

Previously, all the works we have done are in IMS logical network domain. The SIP servers (P/S/I-CSCF and HSS) discussed are all logical entities. In a real network, all logical servers need to be implemented on the physical node(s). How to map logical servers located in a logical IMS core network topology to physical node(s) located in physical IMS network topology is not standardized [5], but is of great interest to the network providers. Network providers may choose different mapping strategies to achieve their own objective. For example, a starting network provider may want to choose a mapping strategy that can minimize the cost, while maintaining acceptable applications to users. On the other hand, an industry-leading network provider may want a mapping strategy that provides high reliability and high expandability. Moreover, each mapping strategy has its own advantages and disadvantages.

There are various mapping strategies available on the public domain for network providers to choose. The providers select the one with the best performance results according to their needs and actual network conditions, including the number of users, the capacity of physical nodes, the budget plan, and so forth. This requires the providers

to consider both advantages and disadvantages of each mapping strategy, in order to determine the one that is satisfied by themselves and their users. In the following sections, we start with a generic mapping strategy, and then focus on two special mapping strategies, which can be widely used in the public domain.

More importantly, we use our proposed signaling flow based traffic model to provide constraints to the cost optimization problem, so that the optimization method can be developed. In the mean time, the disadvantages and advantages of different mapping strategies are discussed in detail.

6.2 A Generic Mapping Strategy

A generic mapping strategy is a method that allows for the mapping of a logical IMS core network topology into a physical IMS network topology in a general case. The description of this mapping strategy is provided in this section. The strategy is represented with the proper mathematical formulation in order to take advantage of the signaling flow based traffic model. An example of showing the formulation is also provided.

6.2.1 Introduction

The upper part of Figure 6.1 illustrates a logical IMS core network topology, which is the way that the SIP messages pass through the network from one logical SIP server to the next without regard to the physical interconnection of the physical nodes. The load of each logical server can be predicted by applying the proposed traffic model. However, in the physical structure of the IMS network, also called physical IMS network topology,

which is depicted in the lower part of Figure 6.1, the load of each physical node can be estimated according to the different mapping strategies, each determining how the logical servers are mapped into the physical node(s). Therefore, we need to study various strategies for this mapping while meeting all the available network requirements.

Figure 6.1 shows a generic strategy of mapping logical servers in the IMS core network into physical nodes interconnected through a network. The physical network topology consists of 5 physical nodes. In this case, any one physical node can host one or more logical server(s). On the other hand, two or more physical nodes can host one or more identical logical servers. Any two or more physical nodes can be identical, which means that they can host the same logical servers. A generic mapping strategy includes all possible mapping ways.

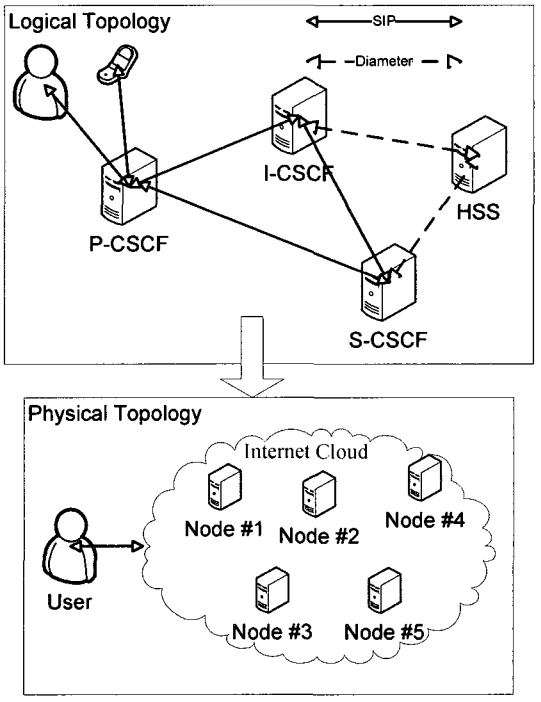


Figure 6.1: A generic mapping strategy

The generic mapping strategy is suitable for all network providers with different network conditions. The assignment of logical servers into physical nodes depends on the traffic load on logical servers and the capacity of the particular physical node. When the capacity of a physical node is large enough for implementing two logical servers, this physical node can host two different logical servers. The network conditions can include the number of physical nodes available, and the cost of implementation and maintenance on physical nodes, and so on. Therefore, compared to the other two special mapping strategies that we discuss later, the generic mapping strategy provides the flexibility to determine the number of, and the type of logical server(s) mapped to a physical node. In conclusion, the generic mapping strategy satisfies the needs of different providers. Next, we will discuss the assumptions made to the generic mapping strategy.

6.2.2 Assumptions and Conditions

Although the mapping strategy is called generic, certain constraint is necessary to minimize the search for the optimal solution. The constraint is related to the signaling flow concept. All signaling flows listed in Table 4.5 can be classified into two types: round trip signaling flow and single trip signaling flow. The round trip signaling flow is defined as a signaling flow that traverses the involved logical servers twice: one in the forward direction and one the reverse direction. The single trip signaling flow passes the involved logical servers only once. The selection of physical nodes for performing both directions in a round trip signaling flow should be identical. This is because the physical nodes chosen in the forwarding direction may hold some information regarding the end users. It will minimize the information to be duplicated on different physical nodes by

choosing the reverse path to be the same physical nodes except the order is reversed. Next, we introduce the proper mathematical notations for modeling IMS network.

6.2.3 Notation for Network Modeling

In this section, we introduce the mathematical notations for network entities. As you will see, a good mathematical notation can represent a specific design problem in a compact and unambiguous way. Furthermore, it helps us to understand the formulation better.

Physical Node, Logical Server, and Signaling Flow

We first introduce the generic labels for physical node, logical server, and signaling flow. The four different logical SIP servers in logical IMS network are labeled with the generic label v, where $v = 1,2,3,4$, and the physical nodes are denoted as y, where $y = 1,2,\cdots,Y$, and Y is the number of physical nodes in the physical IMS network topology. A signaling flow is measured in IMS logical network topology, with label f, where $f = 1,2,\cdots,17$. A direct physical link connects its physical end nodes directly.

Signaling Flow demand, Physical Path

Secondly, we introduce signaling flow demand volume. It is denoted as h_f, where $f = 1,2,\cdots,17$, and it represents the traffic volume (number of messages) in a given unit of time. Since we have a list of matrices $\mathbf{X}_r$, which represents the volume (number of messages) of signaling flows per session procedure in one routing scenario r, where $r = 1,2,...,6$, and a list of row vectors $\mathbf{T}_r$, the average arrival rate of the session procedures for routing scenario r, where $r = 1,2,...,6$, a row vector $\mathbf{H}$ representing the total traffic

volume (number of messages) of the signaling flows for 6 routing scenarios in a given unit of time, can be calculated as follows:

$$\mathbf{H} = \sum_{r=1}^{6} \mathbf{T}_r \mathbf{X}_r , \quad r = 1,2,...,20 \qquad \textbf{6.1}$$

Let h_f denotes the f^{th} element of the row vector **H**, where $f = 1,2,\cdots,17$.

For signaling flow *f*, the total number of available physical paths is denoted by P_f, and they are labeled with p from the first physical path to the total number of physical paths, i.e. $p = 1,2,\cdots,P_f$. This sequence is called the list of candidate physical paths. Each physical path p connects the physical end nodes of signaling flow *f*, and it is described as the set of physical links of which the physical path is composed of. In this paper, we assume the candidate paths for a signaling flow are known to the carrier. A carrier may decide the candidate paths based on its own policy.

Figure 6.2 depicts an example of mapping the logical servers into four physical nodes, which are hosting the corresponding logical servers, as shown in the bracket. A list of physical paths that can carry signaling flow 3 (signaling flow path: →P→S→) is drawn in the lower part of Figure 6.2. Table 6.1 lists the candidate physical paths for signaling flow 3 under the network topology in Figure 6.2. Note that, since there is no physical link connecting physical node #2 to #3, the physical path 3 traverses the physical node #4 in order to reach the destination, which is physical node #3. Physical node #4 plays the role of a passer-by and does not process the messages of this signaling flow. Moreover, for physical path 5, physical node #3 can perform signaling flow 3 alone.

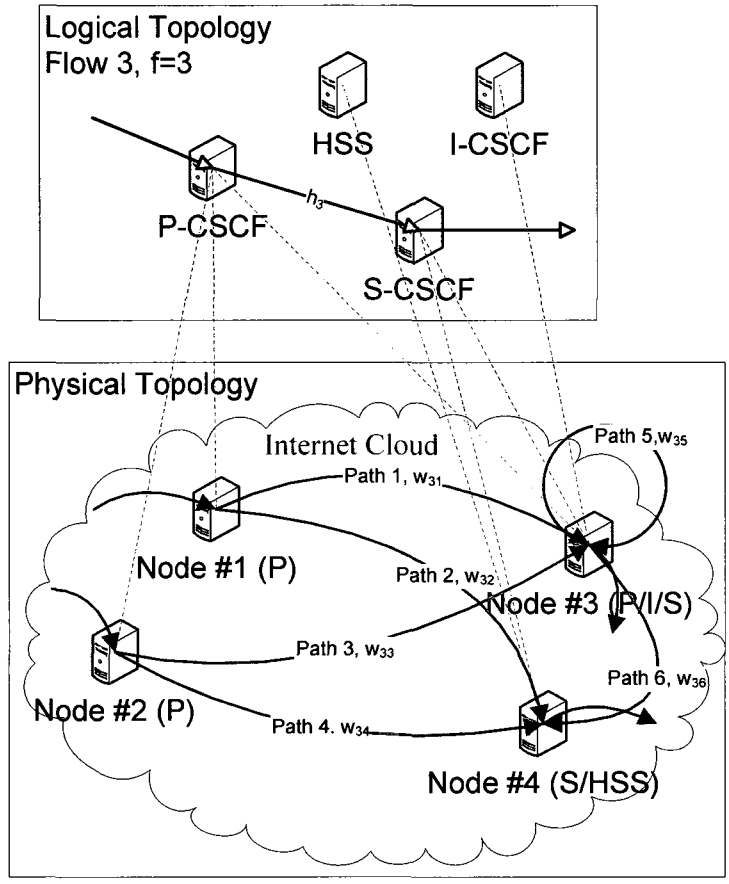


Figure 6.2: An example of mapping, 6 available physical paths for signaling flow 3 (signaling flow path: →P→S→)

Table 6.1: A list of candidate physical paths for signaling flow 3, under the network topology in Figure 6.2

p	**Candidate physical paths for signaling flow 3 (signaling flow path: →P→S→)**	w_{3p}
1	Physical node #1 → #3	w_{31}
2	Physical node #1 → #4	w_{32}
3	Physical node #2 → #4→ #3	w_{33}
4	Physical node #2 → #4	w_{34}
5	Physical node #3	w_{35}
6	Physical node #3 → #4	w_{36}

Now, signaling flow demand volume is assigned to the available physical paths. The load assigned to physical path, a candidate physical path of signaling flow f is denoted by w_{fp}, where $f = 1,2,\cdots,17$ and $p = 1,2,\cdots,P_f$, as shown in Table 6.1. Since the demand volume of signaling flow f needs to be realized by the traffic on all the candidate physical paths, we can write the following equation:

$$w_{31} + w_{32} + w_{33} + w_{34} + w_{35} + w_{36} = h_3 \qquad \textbf{6.2}$$

It leads to the demand constraint, which can be written in a general form as follows:

$$w_{f1} + w_{f2} + w_{f3} + \cdots + w_{fP_f} = h_f, \quad f = 1,2,\cdots,17 \qquad \textbf{6.3}$$

In summation notation, we can write this as:

$$\sum_{p=1}^{P_f} w_{fp} = h_f, \quad p = 1,2,\cdots,P_f \quad f = 1,2,\cdots,17 \qquad \textbf{6.4}$$

And, we know the physical path p is numbered from 1 to P_f for each signaling flow f, the previous expression becomes as follows:

$$\sum_{p} w_{fp} = h_f, \quad p = 1,2,\cdots,P_f \quad f = 1,2,\cdots,17 \qquad \textbf{6.5}$$

where P_f is the number of candidate physical paths for signaling flow f.

Indicator

Thirdly, we define two indicators here for formulating the design problem. The first indicator, denoted by α_{fpyv}, indicates the relationship among physical node y ($y = 1,2,\cdots,Y$), logical server v ($v = 1,2,3,4$), physical path p ($p = 1,2,\cdots,P_f$), and signaling flow f ($f = 1,2,\cdots,17$). When it is one, the physical node y hosts logical server

v along physical path p that is one available physical path of signaling flow f; it is zero for the rest of the cases. The advantage of defining the indicator here is to provide a nice compact manner by using the notation, α_{fpyv}, to present the required information. Formally, the indicator α_{fpyv} is defined for each quadruple (f, p, v, y), and it is written as:

$$\alpha_{fpvy} = \begin{cases} 1, \text{if physical node y hosts logical server v along} \\ \text{physical path p that is one physical path of signalling flow f} \\ 0, \text{otherwise.} \end{cases} \quad \textbf{6.6}$$

where $f = 1,2,\cdots,17$, $p = 1,2,\cdots,P_f$, $v = 1,2,3,4$, and $y = 1,2,\cdots,Y$. α_{fpvy} is constant and obtained from the analysis performed on network topology assuming that the carrier knows all candidate paths for each flow based on its policy..

The second indicator determines the number of times that logical server v is involved in signaling flow f, and it is denoted by β_{fv}. As we know, there are two types of signaling flow, which are the round trip signaling flow and the single trip signaling flow as summarized in Table 4.9. The logical servers along a round trip signaling flow are involved twice. And, the logical servers along a single trip signaling flow are involved once. However, signaling flow 11 and signaling flow 12 are special cases due to the signaling flow traverses logical S-CSCF server only once although they look like a round trip flow. Although both coefficient β_{fv} and the previously defined matrix **I** identify the relationship between the signaling flows and the logical servers, β_{fv} records the values that are different from matrix **I**.

β_{fv} is written as follows:

$$\beta_{fv} = \begin{cases} 2, \text{if logical server v is involved in signalling flow f,} \\ \text{and signallingflow f is a round trip} \\ 1, \text{if logical server v is involved in signallingflow f,} \\ \text{and signallingflow f is a single trip} \\ 2, \text{if logical server v is P - CSCF, for signalling flow 11} \\ 2, \text{if logical server v is I - CSCF, for signalling flow 11} \\ 1, \text{if logical server v is S - CSCF, for signalling flow 11} \\ 2, \text{if logical server v is I - CSCF, for signalling flow 12} \\ 1, \text{if logical server v is S - CSCF, for signalling flow 12} \\ 0, \text{otherwise.} \end{cases} \quad \textbf{6.7}$$

where $f = 1,2,\cdots,17$ and $v = 1,2,3,4$.

Table 6.2: The types of signaling flows

Single Trip Signaling Flow		**Round Trip Signaling Flow**		**Special Case**	
Signaling Flow	Signaling Flow Path	*Signaling Flow*	Signaling Flow Path	*Signaling Flow*	Signaling Flow Path
2	→ S1 → P1 →	*1*	→P1 → S1 → --- → S1 → P1 →	*11*	→ P1 → I1 → S1 → I1 → P1 →
3	→ P1 →S1 →	*4*	→I1 → S1 → P1 →--->P1 → S1 → I1 →	*12*	→ I1 → S1 → I1 →
6	→ P1 → S1 → I1 →	*5*	→ S1 → P1 →---→ P1 → S1 →		
7	--- S1 ---	*8*	→I1 → S1 →---→ S1 → I1 →		
9	→ S1 → I1 →				
10	--- P1 ---				
13	→ I1 → S1 → P1 →				
14	→ I1 →S1 →				
15	→ HSS → I1				
16	→ HSS → S1				
17	→ S1 → HSS1				

Load, Rate, Capacity

Lastly, we introduce additional issues that help to formulate the network deployment cost optimization problem. We want to lower the cost of deploying the physical nodes in the network, and the cost of a physical node mainly depends on its capacity. Thus, we should predict the traffic load on physical node, and determine the best choice for the capacity of a physical node.

The load on each physical node indicates the number of actual SIP messages processed in a given unit of time, and we denote this load as k_y for physical node y. As we can see in Figure 6.2, the load of physical node is associated with the traffic of the individual physical link connected to the physical node. Thereby, the load of the physical node is calculated from the load on each connected physical link and can be represented in a linear mathematic expression. We take the physical nodes in Figure 6.2 as examples here.

$$\begin{aligned} k_1 &= w_{31} + w_{32} \\ k_2 &= w_{33} + w_{34} \\ k_3 &= w_{31} + w_{33} + w_{35} + w_{36} \\ k_4 &= w_{33} + w_{34} + w_{36} \end{aligned} \qquad \textbf{6.8}$$

Since we define the indicators, α_{fpvy} and β_{fv}, the use of notation is extremely helpful to correctly present the relationship between physical link traffic and the load of the physical node in a nice compact way, in terms of signaling flows and the available

physical paths for each signaling flow. Hence, the load of physical node y is defined as follows:

$$k_y = \sum_f \left[\sum_v \beta_{fv} \left(\sum_p \alpha_{fpvy} w_{fp} \right) \right] \qquad \textbf{6.9}$$

$$y = 1,2,\cdots,Y,\ f = 1,2,\cdots,17,\ p = 1,2,\cdots,P_f,\ v = 1,2,3,4$$

Next, we utilize the example illustrated in Figure 6.2 to better explain the Equation 6.9, and the details are drawn in Figure 6.3. In general, this equation expression can be divided into three steps.

Step a. α_{fpvy} =1 if physical node y hosts logical server v along physical path p that is one available physical path of signaling flow f, and α_{fpvy}=0, otherwise. Then, $\left(\sum_p \alpha_{fpvy} w_{fp} \right)$, accumulates the load allocated to each available physical path p of the signaling flow f, with a given logical server v and physical node y.

a) In Figure 6.3, we assume y = 3 (physical node #3), v = 1 (logical server P-CSCF), f = 3 (signaling flow 3).

b) Then, $\left(\sum_p \alpha_{3p13} w_{3p} \right) = w_{31} + w_{33} + w_{35} + w_{36}$

Step b. β_{fv}, $\neq 0$ in the case of logical server v is traversed by signaling flow f; =0, otherwise. Then, $\left[\sum_v \beta_{fv} \left(\sum_p \alpha_{fpvy} w_{fp} \right) \right]$, represents the total load on physical node y, for one signaling flow, namely, signaling flow f.

a) Since physical node #3 hosts P-CSCF, I-CSCF and S-CSCF, then

$$\left\lfloor \sum_v \beta_{3v} \left\lfloor \sum_p \alpha_{3pv3} w_{3p} \right\rfloor \right\rfloor = 1 \cdot \underbrace{(w_{31} + w_{33} + w_{35} + w_{36})}_{\text{For logical server P-CSCF}} + \underbrace{0}_{\text{For logical server I-CSCF}} + 1 \cdot \underbrace{(w_{31} + w_{33} + w_{35} + w_{36})}_{\text{For logical server S-CSCF}}$$

Step c. $\sum_f \left\lfloor \sum_v \beta_{fv} \left(\sum_p \alpha_{fpvy} w_{fp} \right) \right\rfloor$, sums up the load for the involved signaling flows.

a) Since in Figure 6.3, only one signaling flow is involved.

b) $k_3 = \sum_f \left\lfloor \sum_v \beta_{fv} \left(\sum_p \alpha_{fpv3} w_{fp} \right) \right\rfloor = 2 \cdot (w_{31} + w_{33} + w_{35} + w_{36})$

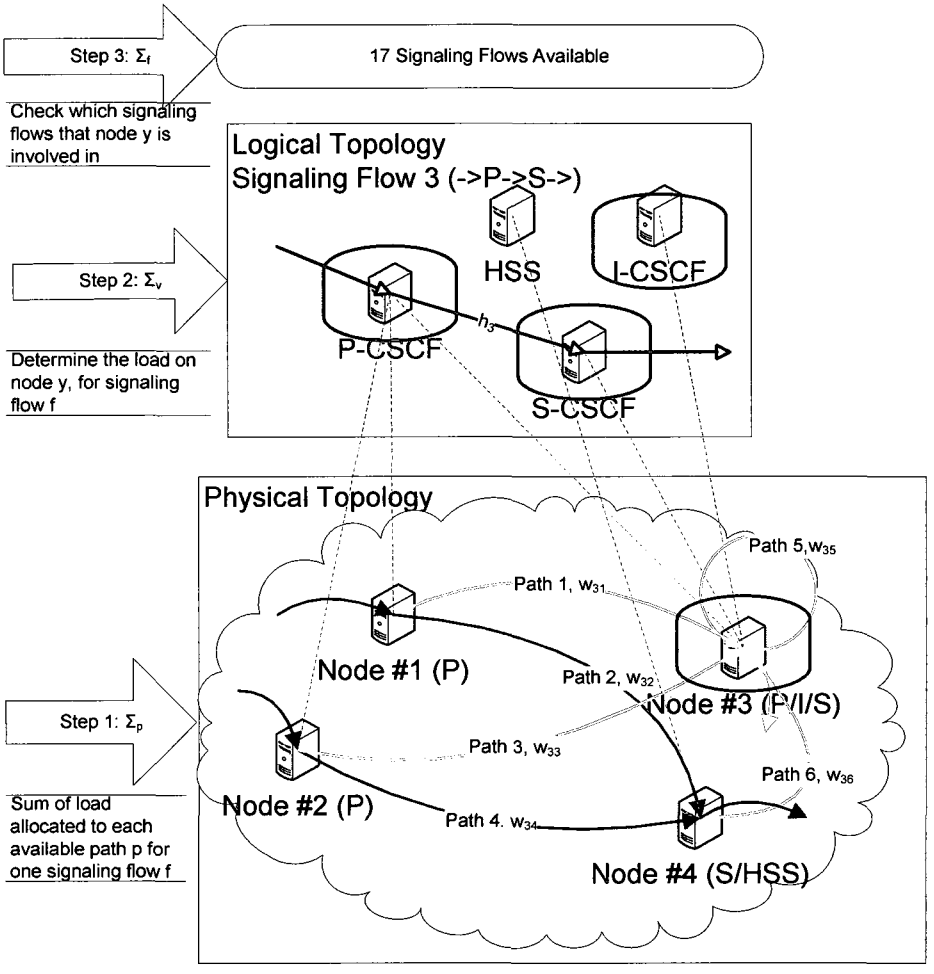


Figure 6.3: 3 steps for explaining Equation 6.9, with an example of Figure 6.2

However, we notice that indicator α_{fpvy} holds the important information extracted from the physical network topology, and the value of it is obtained through processing the given network construction. When the number of physical nodes located in the physical topology is low, it is easily to obtain the information for the indicator α_{fpvy}, which can even be found by manual computation. However, in the case of a large and complex network topology that holds tons of physical nodes and a large set of physical links, more time is required as well as more power for processors to obtain the value of α_{fpvy}.

Our goal is to find the capacity of a physical node that can satisfy the load requirement. Let c_y represents the processing capability of a physical node. c_y can be in various units that are related to the cost of the physical node. For example, c_y can be the number of certain CPUs the physical node carries.

Since in IMS network, each logical server has different message functions to be processed, we need to decide how much capacity is required to process messages for each type of logical server. This capacity coefficient is denoted as κ_v for logical server v. κ_v is in the unit of time-capacity product. For example, κ_v=2 can mean that a message requires two time units for a logical server with a single CPU or one time unit for a logical server with two CPUs. Here we assume the overhead associated with multiple CPUs is negligible to simplify the analysis. However our analysis can be generalized to include those overheads.

It should be noted that messages processed by the same logical server do not necessarily take the same amount processing time due to their different types. To simplify our analysis, we take κ_v as the statistical mean of the capacity required by all types of messages processed by the logical server.

Equation 6.9 calculates the load of physical node y, which is represented in the number of messages processed within a unit of time, and it is a sum of the load on logical server v that is hosted in physical node y. Hereby, the capacity of physical node y should be greater than the accumulation of the load of logical server v times κ_v, for all possible logical server v that the physical node y hosts. This is a second set of constraints that can be generally written as:

$$\sum_f \left[\sum_v \kappa_v \beta_{fv} \left(\sum_p \alpha_{fpvy} w_{fp}\right)\right] \le c_y \qquad \textbf{6.10}$$

$$f = 1,2,\cdots,17,\ v = 1,2,3,4,\ p = 1,2,\cdots,P_f,\ y = 1,2,\cdots,Y$$

Furthermore, in the design problem, we aim to minimize the physical node capacity cost. Therefore, we introduce a rate ε_y that represents the cost per unit processing capability for node y. A list of notations we defined is listed in Table 6.3. The details of formulating the cost optimization problem are provided in the following section.

Table 6.3: Mathematical notations defined for the generic mapping strategy

Parameter	Notation	Description
Physical Node, Logical Server Signaling Flow	y	Physical node, $y = 1,2,\cdots,Y$, where Y is the number of physical nodes in the network.
	v	Logical server, $v = 1,2,3,4$.
	f	Signaling flow, $f = 1,2,\cdots,17$.
Physical Path, Demand Volume	p	Candidate physical path for signaling flow f, $p = 1,2,\cdots,P_f$, where P_f is the number of candidate physical paths for signaling flow f.
	h_f	Signaling flow demand volume for signaling flow f (message per unit time).
	w_{fp}	Load allocated to physical path p that is one available physical path of signaling flow f (message per unit time).
Indicator	α_{fpvy}	=1, if physical node y hosts logical server v along physical path p that is one available physical path of signaling flow f; =0, otherwise.
	β_{fv}	The number of times that logical server v is involved in signaling flow f.
Rate	κ_v	The capacity coefficient in time-capacity product unit for each logical server v.
	ε_y	The cost of processing one message on physical node y.
Load, Capacity	k_y	The load of physical node y (message per unit time).
	c_y	The cost coefficient per unit processing capacity.

6.2.4 Formulation of Cost Optimization Problem

The performance of physical nodes is our main concern in physical IMS network topology, and we assume the physical link bandwidth to be large enough so that they are not the bottleneck of performance. Also, the physical node performance is usually evaluated by service response time. However, the service response time is dependent on

the capacity of physical node, which decides the physical node cost. Thereby, we concern the physical node cost.

In this section, we consider IMS network deployment cost optimization issue with a set of given signaling flow demand volume in the IMS physical network and provide a complete version of formulating the cost optimization issue as a linear programming problem for a generic mapping strategy. When the physical network topology is given, the formulation can be formed.

Cost Optimization Problem Formulation:

- Indices
  - f=1, 2,..., 17, signaling flow
  - v= 1, 2, 3, 4, logical server
  - p =1, 2, ... P_f , candidate physical path for signaling flow f
  - y=1, 2, ...Y, physical node
- Constants
  - α_{fpyv} :=1, if physical node y hosts logical server v along physical path p that is one available physical path of signaling flow f; 0, otherwise.
  - β_{fv} :The number of times that logical server v is involved in signaling flow f.
  - h_f :Signaling flow demand volume for signaling flow f.
  - κ_v: The capacity coefficient in time-capacity product unit for each logical server v.

- ε_y : The cost coefficient per unit processing capacity for physical node y.

- Variables

  - w_{fp} :The load allocated to physical path p that is one available physical path of signaling flow f

- Objective

  - Minimize total network physical nodes cost:

$$F = \sum_y \varepsilon_y \cdot c_y, \quad y=1, 2, \ldots Y \tag{6.11}$$

- Constraints

  - Demand Constraints:

$$\sum_p w_{fp} = h_f, \quad p = 1,2,\cdots,P_f \quad f = 1,2,\cdots,17 \tag{6.12}$$

  - Capacity Constraints:

$$k_y = \sum_f \left[\sum_v \beta_{fv} \left(\sum_p \alpha_{fpvy} w_{fp}\right)\right] \tag{6.13}$$

$$y=1, 2, \ldots Y,\ f = 1,2,\cdots,17,\ v = 1,2,3,4,\ p = 1,2,\cdots,P_f$$

$$\sum_f \left[\sum_v \kappa_v \beta_{fv} \left(\sum_p \alpha_{fpvy} w_{fp}\right)\right] \le c_y \tag{6.14}$$

$$f = 1,2,\cdots,17,\ v = 1,2,3,4,\ p = 1,2,\cdots,P_f,\ y=1, 2, \ldots Y$$

  - Constraints on variables,

$$w_{fp} \ge 0 \text{ (continuous, non - negative)}, \quad f = 1,2,\cdots,17 \quad p = 1,2,\cdots,P_f \tag{6.15}$$

According to Equation 6.12, 6.13, and 6.14, the cost optimization issue can be formulated as a linear programming problem. In the optimal solution of this problem, all

constraints presented in Equation 6.14 are binding, i.e., the physical node load is equal to the physical nodes capacities; however, the capacity of the physical node may not come in continuous value. To reduce the unused capacity, we can set c_y to integer. Then the problem becomes an integer programming problem which is more difficult to solve.

6.2.5 Example

Here, we give an example to show the formulation of the cost optimization problem. The simple network is shown in Figure 6.4, with 5 physical nodes. Each physical node is hosting one or more logical server(s), as shown in the bracket after the name of physical node. The assumptions are made as follows:

1. There are 3 signaling flows involved; they are signaling flow 11, signaling flow 15, and signaling flow 16, i.e. $f = 11,15,16$. It is easy to see that the number of potential paths that can be used as candidate paths is large. To simplify the example, we assume only the paths listed in Table 6.4 are the candidate paths.
2. Signaling flow 11 is a round trip signaling flow. P-CSCF and I-CSCF logical servers are involved twice. The selected physical nodes, hosting either P-CSCF or I-CSCF logical servers, should be identical for both forward direction and back direction in signaling flow 11.

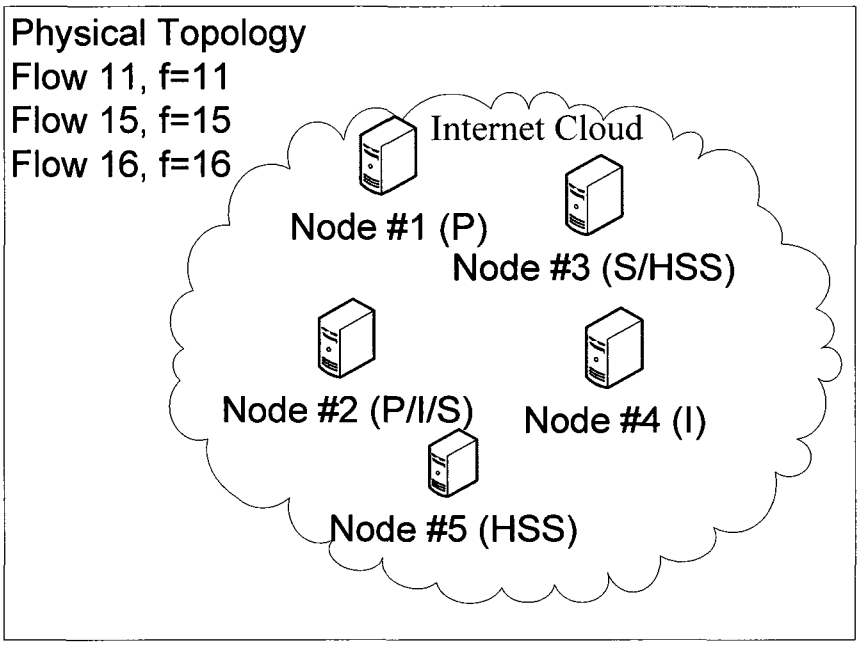


Figure 6.4: An example of formulating cost optimization problem, with 3 signaling flows involved, for a generic mapping strategy

Table 6.4: The candidate physical paths for signaling flow 11, 15 and 16, reference to Figure 6.4

p	The choice of physical nodes				Candidate physical paths for 3 signaling flows
	P	I	S	HSS	**Signaling Flow 11 (→P→I→S→I→P→)**
1	#1	#4	#2		→#1(P) → #4 (I) → #2 (S) → #4 (I) →#1(P) →
2	#1	#4	#3		→#1(P) →#4(I) → #3 (S) → #4(I) →#1(P) →
					Signaling Flow 15 (→HSS→I→)
1		#4		#5	→#5 (HSS) → #4 (I) →
					Signaling Flow 16 (→HSS→S→)
1			#2	#5	→#5 (HSS) → #2 (S)→
2			#3	#5	→#5 (HSS) →#3 (S) →

The Given information is provided as follows:

a) Signaling flow demand volume,

$$h_f = \begin{cases} 400, \text{f} = 11 \\ 250, \text{f} = 15 \\ 200, \text{f} = 16 \end{cases} \quad \textbf{6.16}$$

b) Coefficient in a ratio for logical server v,

$$\kappa_v = \begin{cases} 2, \text{v} = 1 \\ 1, \text{v} = 2 \\ 5, \text{v} = 3 \\ 2, \text{v} = 4 \end{cases} \quad \textbf{6.17}$$

S-CSCF (v=3) requires more information processing on the received messages. The processing time for processing S-CSCF messages are larger than other logical servers.

c) Cost of one message processed on physical node y,

$$\varepsilon_y = \begin{cases} 5, \text{y} = 1 \\ 10, \text{y} = 2 \\ 10, \text{y} = 3 \\ 5, \text{y} = 4 \\ 5, \text{y} = 5 \end{cases} \quad \textbf{6.18}$$

Physical node #2 and #3 are hosting S-CSCF that is required more power in order to process the messages. Then, the unit cost of processing a message on these two physical nodes is higher than the others.

d) β_{fv}, an indicator determines the number of times that logical server v is involved in signaling flow f.

$$\beta_{fv} = \begin{cases} 2, \text{if logical server v is involved in signalling flow f,} \\ \text{and signallingflow f is round trip} \\ 1, \text{if logical server v is involved in signallingflow f,} \\ \text{and signallingflow f is single trip} \\ 2, \text{if logical server v is P - CSCF, for signalling flow 11} \\ 2, \text{if logical server v is I - CSCF, for signalling flow 11} \\ 1, \text{if logical server v is S - CSCF, for signalling flow 11} \\ 2, \text{if logical server v is I - CSCF, for signalling flow 12} \\ 1, \text{if logical server v is S - CSCF, for signalling flow 12} \\ 0, \text{otherwise.} \end{cases} \quad \textbf{6.19}$$

e) α_{fpyv}, a coefficient can be extracted from the network topology, as shown in Figure 6.4. The candidate physical paths for each signaling flow and the involved physical nodes are listed in Table 6.4.

The variables we need to obtain:

$$w_{fp}, \text{ for} \begin{cases} p = 1, 2, \text{for f} = 11 \\ p = 1, \text{for f} = 15 \\ p = 1, 2, \text{for f} = 16 \end{cases} \quad \textbf{6.20}$$

The objective function can be written as:

$$\textit{Minimize} \qquad F = 5c_1 + 10c_2 + 10c_3 + 5c_4 + 5c_5 \quad \textbf{6.21}$$

To obtain constraints:

a) Demand Constraints:

$$w_{11,1} + w_{11,2} = h_{11} = 400 \quad \textbf{6.22}$$

$$w_{15,1} = h_{15} = 250 \quad \textbf{6.23}$$

$$w_{16,1} + w_{16,2} = h_{16} = 200 \quad \textbf{6.24}$$

b) Capacity Constraints:

$$2 \cdot 2 \cdot (w_{11,1} + w_{11,2}) \leq c_1 \quad \textbf{6.25}$$

$$5 \cdot (w_{11,1}) + 5 \cdot (w_{16,1}) \leq c_2 \quad \textbf{6.26}$$

$$5 \cdot (w_{11,2}) + 5 \cdot (w_{16,2}) \leq c_3 \quad \textbf{6.27}$$

$$2 \cdot (w_{11,1} + w_{11,2}) + (w_{15,1}) \leq c_4 \quad \textbf{6.28}$$

$$2 \cdot (w_{15,1}) + 2 \cdot (w_{16,1} + w_{16,2}) \leq c_5 \quad \textbf{6.29}$$

c) Constraints on variables:

$$w_{11,1} \geq 0, w_{11,2} \geq 0 \quad \textbf{6.30}$$

$$w_{15,1} \geq 0 \quad \textbf{6.31}$$

$$w_{16,1} \geq 0, w_{16,2} \geq 0 \quad \textbf{6.32}$$

In this example, there are only 5 physical nodes, and each logical server is hosted in each 2 physical nodes. We will introduce two mapping strategies in the following sections.

6.3 Customized Mapping Strategy 1: One Physical Node Hosts One Logical Server

The customized mapping strategy 1 is a method that only allows one physical node to host one logical server. In this section, we discuss this mapping strategy, including its

advantages and disadvantages. Follow this, the cost optimization problem for this mapping strategy is formulated, and an example of presenting the formulation is provided at the end of this section.

6.3.1 Introduction

In IMS core network architecture, the logical IMS servers are classified into four different categories, which are P-CSCF, S-CSCF, I-CSCF, and HSS, according to their own assigned tasks in IMS. Four logical servers that we concentrated on play different roles in IMS system. When using customized mapping strategy 1 to map these logical servers to the physical nodes, each physical node only hosts one logical server, and it focuses on performing one type of task. The task on each physical node is clearly demarcated. This mapping strategy is desired when the load of two or more logical servers exceeds the capacity of a physical node. This is often the case for the network providers with a large number of users.

Overall, there are three advantages by using this mapping strategy. First, it is easier to create a backup physical node and upgrade capacity for the future. Second, this strategy brings a small impact to the system when the failure occurs on the physical nodes, because each physical node only takes care of one type of tasks. If a physical node that hosts two or more logical servers fails, other physical nodes take over the work of the failed physical node. The work contains the processing of all SIP messages, which are needed to be processed in two or more logical servers. In this case, a network system is required in order to deal with the distribution of different types of SIP messages. Therefore, it brings a large impact to the network system, compared to the case of a

physical node only hosting a logical server. Third, it is easy to implement and maintain the physical nodes. The main disadvantage of this mapping strategy is cost. The remaining capacity of the physical nodes which host one type of logical server can not be allocated to other logical servers and therefore will be wasted.

Figure 6.5 depicts this mapping strategy, where each physical node hosts only one logical server. Multiple physical nodes, which host the same type of logical server, can then perform a load balance. Next, the formulation of the cost optimization issue is presented.

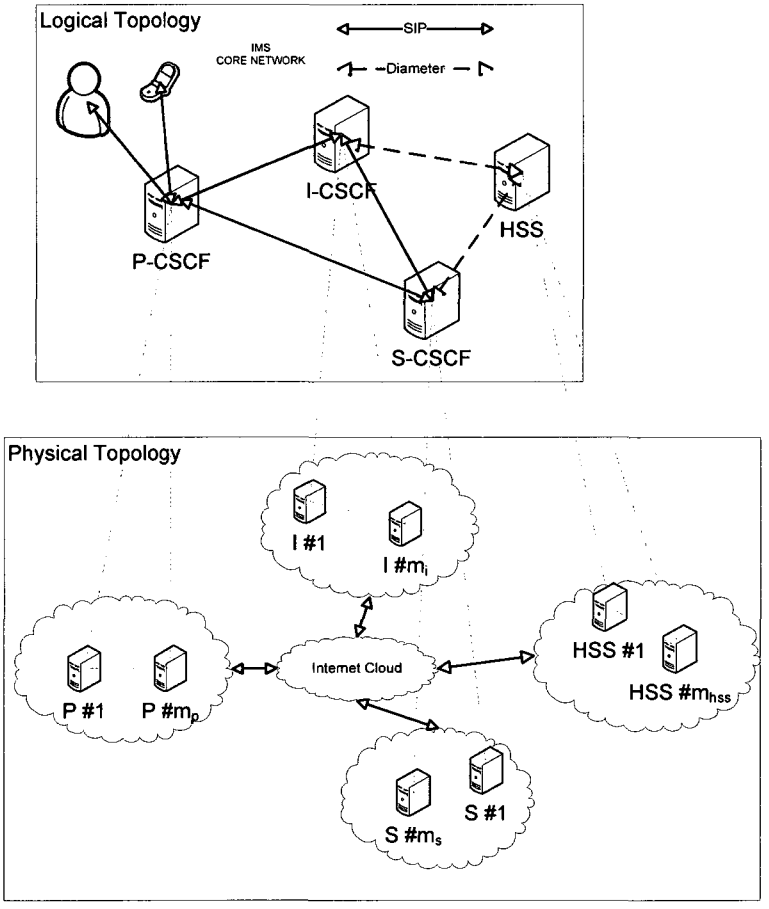


Figure 6.5: Customized mapping strategy 1

6.3.2 Formulation of Cost Optimization Problem

In this mapping strategy, it is easy to see that the physical nodes that support one logical server play load balance among themselves. Because there is no sharing of residual capacities among different logical servers, the physical nodes that support different types of logical servers can be optimized separately. This can significantly simplify the optimization process because the number of variables to be optimized only depends on the number of physical nodes hosting the same logical server rather than the number of candidate paths. The optimization problem described in the last section can be reformulated as follows.

Cost Optimization Problem Formulation:

- Indices
  - f=1, 2,..., 17, signaling flow
  - v= 1, 2, 3, 4, logical server
  - y_v=1, 2, ...Y_v, physical node that hosts logical server v
- Constants
  - α_{fv}: =1, if signaling flow f traverses logical server v; =0, otherwise.
  - β_{fv} :The number of times that logical server v is involved in signaling flow f.
  - h_f :Signaling flow demand volume of signaling flow f.
  - κ_v:The capacity coefficient in time-capacity product unit for each logical server v.

- ε_{vy} :The cost coefficient per unit processing capacity for physical node y that hosts logical server v.

- Variables

  - w_{fvy_v} :The loads allocated to physical node y of signaling flow f for logical server v.

- Objective

  - Minimize total physical nodes cost for logical server v :

$$F_v = \sum_{y_v} \varepsilon_{vy_v} c_{vy_v}, \quad y_v = 1,2,\cdots,Y_v \tag{6.33}$$

- Constraints

  - Demand Constraints:

$$\sum_{y_v} w_{fvy_v} = h_f \alpha_{fv}$$

$$y_v = 1,2,\cdots,Y_v, \ f = 1,2,\cdots,17, \ v = 1,2,3,4 \tag{6.34}$$

  - **Capacity Constraints:**

$$k_y = \sum_f \beta_{fv} \alpha_{fv} w_{fvy_v}$$

$$y_v = 1,2,\cdots,Y_v, \ f = 1,2,\cdots,17, \ v = 1,2,3,4 \tag{6.35}$$

$$\sum_f \kappa_v \beta_{fv} \alpha_{fv} w_{fvy_v} \le c_{vy_v}$$

$$f = 1,2,\cdots,17, \ v = 1,2,3,4, \ y_v = 1,2,\cdots,Y_v \tag{6.36}$$

  - Constraints on variables:

$$w_{fvy_v} \geq 0,\ f = 1,2,\cdots,17,\ v = 1,2,3,4,\ y_v = 1,2,\cdots,Y_v \quad \textbf{6.37}$$

$$c_{vy_v} \geq 0,\ v = 1,2,3,4,\ y_v = 1,2,\cdots,Y_v \quad \textbf{6.38}$$

Next, we provide an example to show the formulation of customized mapping strategy 1.

6.3.3 Example

An example is provided to formulate the cost optimization problem when using customized mapping strategy 1. The network topology illustrated in Figure 6.6 is the topology shown in Figure 6.4 with the relocation of logical servers in 5 physical nodes. The assumptions and given information remain the same as provided in. In this case, only Nodes #2 and #3 need to be optimized for logical server S-CSCF.

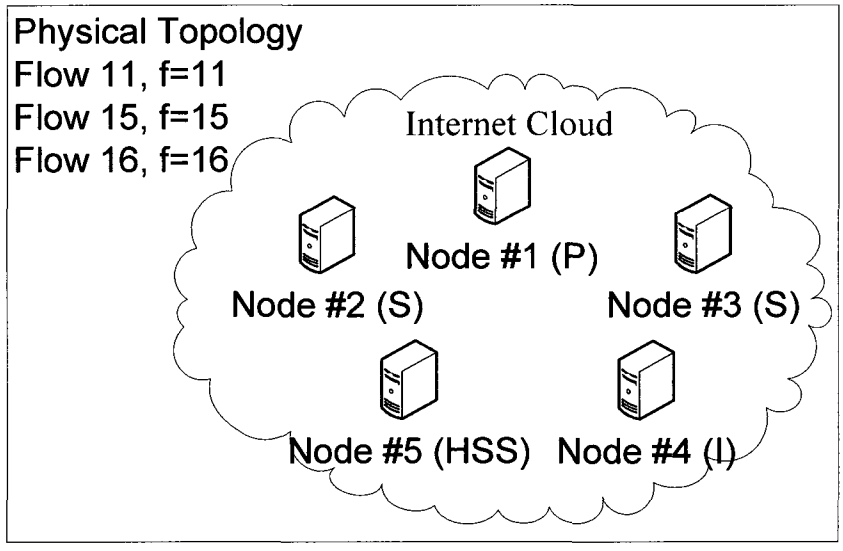


Figure 6.6: An example of formulating the network cost optimization problem, with 3 signaling flows involved, for customized mapping strategy 1

The objective function can be written as:

$$Minimize \quad F_3 = 10c_{3,2} + 10c_{3,3} \tag{6.39}$$

To obtain constraints:

- Demand Constraints:

$$w_{11,3,2} + w_{11,3,3} = h_{11} = 400 \tag{6.40}$$

$$w_{16,3,2} + w_{16,3,3} = h_{16} = 200 \tag{6.41}$$

- Capacity Constraints:

$$5 \cdot (w_{11,3,2} + w_{16,3,3}) \leq c_{3,2} \tag{6.42}$$

$$5 \cdot (w_{11,3,2} + w_{16,3,2}) \leq c_{3,3} \tag{6.43}$$

- Constraints on variables:

$$w_{11,3,2} \geq 0, w_{11,3,3} \geq 0 \tag{6.44}$$

$$w_{16,3,2} \geq 0, w_{16,3,3} \geq 0 \tag{6.45}$$

In this example, each physical node hosts only one logical server. It reduces the complexity of the problem formulation. Moreover, it carries a small number of variables so that it requires the less computation time to solve the problem, compared with the generic mapping strategy. Next, the discussion of customized mapping strategy 2 is provided.

6.4 Customized Mapping Strategy 2: One or More Logical Server(s) Map(s) to One Type of Physical Node

The customized mapping strategy 2 is a method in which one or more logical server(s) can map to one type of physical node. In this section, we discuss this mapping strategy, including its advantages and disadvantages. Following this, the cost optimization issue for this mapping strategy is formulated, and an example of presenting the formulation is provided at the end of this section.

6.4.1 Introduction

While the customized mapping strategy 1 discussed in the last section fits large carriers with numerous users, the mapping strategy discussed in this section fits small carriers who try to pact different logical servers into the same physical node to save footprint and cost. The customized mapping strategy 2 is illustrated in Figure 6.7. In this strategy, physical nodes are divided into different groups. One logical server can be hosted by the physical nodes located in one group only. One group of physical nodes can host one or more than one logical servers. Furthermore, we assume that a message that traverses the logical servers that belong to the same group will be processed by one physical nodeonly in the group. This constraint can reduce the traveling time within a group. In the extreme case, if each group hosts only one logical server, this becomes the customized mapping strategy 1.

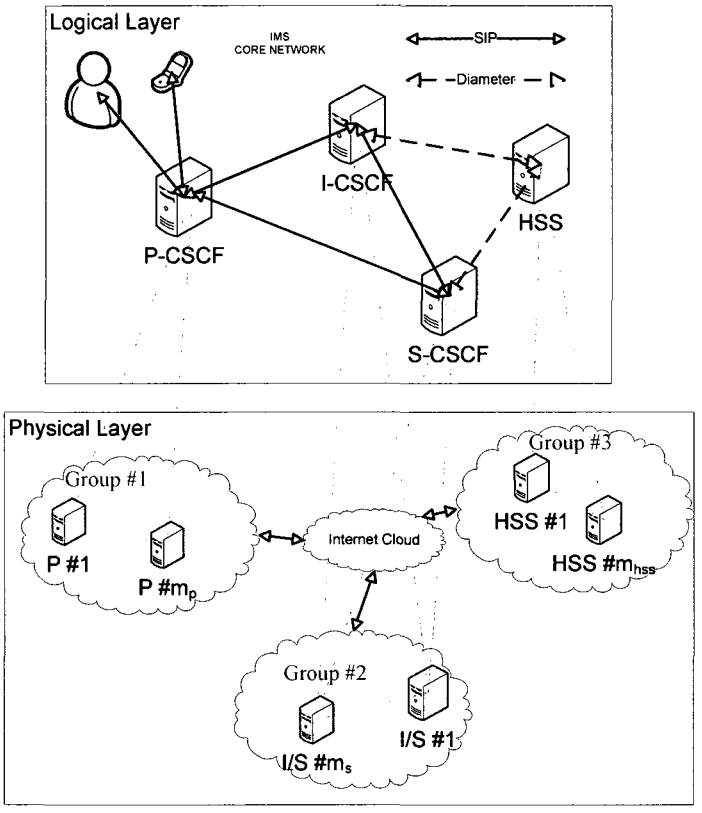


Figure 6.7: Customized mapping strategy 2

The customized mapping strategy 2 allows a physical node to host two or more logical servers. For many network providers, the capacity of one physical node may be more than the load of one logical server. This mapping strategy is more practical than the previous strategy for this type of carriers.

In general, there are two main advantages of using this mapping strategy over the first one. First, mapping more than one logical server to a physical node can utilize the existing physical node capacity, thus reduce costs. Second, this method can reduce the messages' traveling time.

The main disadvantage of this mapping strategy over the previous one is the complexity involved. When a physical node hosts more than one logical server, the

physical node has to handle more types of tasks which may interfere with each other. The maintenance cost will clearly be higher. The details of formulating cost optimization problem are presented in next section.

6.4.2 Formulation of Cost Optimization Problem

In this mapping strategy, the group concept is introduced. A group of physical nodes is denoted as g, where $g = 1,2,3,...,G$ and G is the total number of groups we have in the network topology. In a group, the physical nodes host the same logical servers. Every physical node y belongs to one group. Because the residual capacities of the physical nodes can only be shared among the physical nodes in the same group, the optimization can be decomposed into the optimization of each group of physical nodes. It can be formulated as the follows.

Cost Optimization Problem Formulation:

- Indices
  - f=1, 2,..., 17, signaling flow
  - v = 1, 2, 3, 4, logical server
  - g=1,2,3,...,G, group number
  - y_v=1, 2, ...Y_v, physical node that hosts logical server v
- Constants
  - α_{fg} : =1, if signaling flow f traverses group g; =0, otherwise.
  - δ_{fvg} : =1, if signaling flow f traverses logical server v mapped to group g.
  - h_f :Signaling flow demand volume of signaling flow f.

- κ_v:The capacity coefficient in time-capacity product unit for each logical server v.
- ε_{gy} :The cost coefficient per unit processing capacity for physical node y in group g.

- Variables
  - w_{fgy_g} : The loads of signaling flow f allocated to physical node y in group g.
- Objective
  - Minimize total physical nodes cost for logical server v :

$$F_g = \sum_{y_v} \varepsilon_{gy_g} c_{gy_g}, \quad g = 1,2,\cdots,G, \; y_v = 1,2,\cdots,Y_v \tag{6.46}$$

- Constraints
  - Demand Constraints:

$$\sum_{y_v} w_{fgy_g} = h_f \alpha_{fg}$$
$$y_v = 1,2,\cdots,Y_v, \; f = 1,2,\cdots,17, g = 1,2,\cdots,G \tag{6.47}$$

  - **Capacity Constraints:**

$$k_y = \sum_f \sum_v \beta_{fv} \delta_{fvg} w_{fgy_g}$$
$$y = 1,2,\cdots,Y, \; f = 1,2,\cdots,17, \; v = 1,2,3,4, \; g = 1,2,\cdots,G \tag{6.48}$$

$$\sum_f \sum_v \kappa_v \beta_{fv} \delta_{fvg} w_{fgy_g} \le c_{gy_g}$$
$$f = 1,2,\cdots,17, \; v = 1,2,3,4, \; g = 1,2,\cdots,G, \; y = 1,2,\cdots,Y \tag{6.49}$$

  - Constraints on variables:

$$w_{fgy_g} \geq 0,\ f = 1,2,\cdots,17,\ g = 1,2,\cdots,G,\ y = 1,2,\cdots,Y \quad \textbf{6.50}$$

$$c_{gy_g} \geq 0,\ g = 1,2,\cdots,G,\ y = 1,2,\cdots,Y \quad \textbf{6.51}$$

An example is given in the next section in an attempt to illustrate the procedures of formulating the cost optimization problem for customized mapping strategy 2.

6.4.3 Example

The network topology illustrated in Figure 6.8 is provided to show the formulation of cost optimization problem using the customized mapping strategy 2. This network topology is the topology discussed in Section 6.2.5 with a relocation of the logical servers in the 5 physical nodes. The assumptions and given information remain the same as provided in Section 6.2.5. Clearly only Groups #1 and #2 can be optimized.

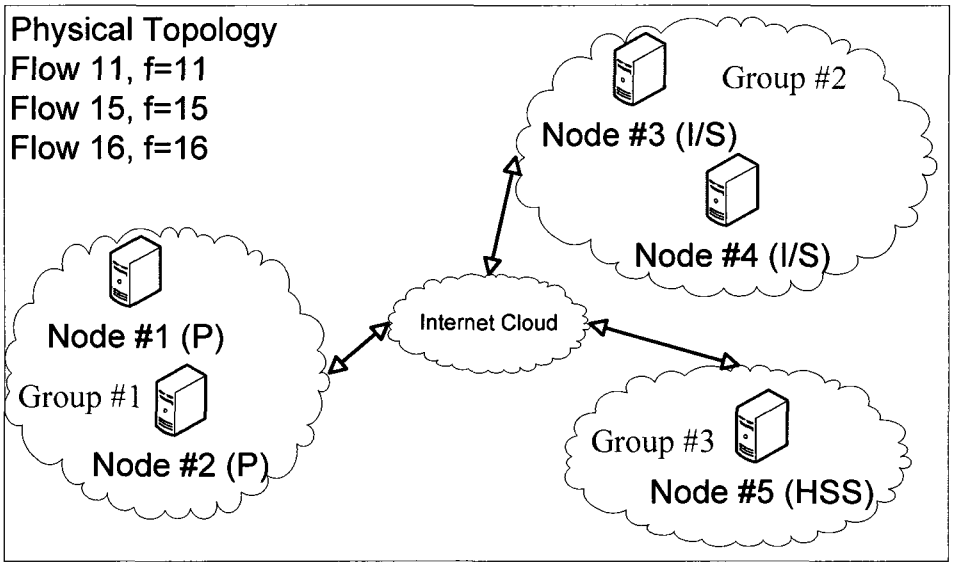


Figure 6.8: An example of formulating network cost optimization problem, with 3 signaling flows involved, for customized mapping strategy 2

The objective function can be written as:

$$Minimize \quad F_1 = 5c_{1,1} + 10c_{1,2} \tag{6.52}$$

$$Minimize \quad F_2 = 10c_{2,1} + 5c_{2,2} \tag{6.53}$$

To obtain constraints:

- Demand Constraints:

$$w_{11,1,1} + w_{11,1,2} = h_{11} = 400 \tag{6.54}$$

$$w_{11,2,1} + w_{11,2,2} = h_{11} = 400 \tag{6.55}$$

$$w_{15,2,1} + w_{15,2,2} = h_{15} = 250 \tag{6.56}$$

$$w_{16,2,1} + w_{16,2,2} = h_{16} = 200 \tag{6.57}$$

- Capacity Constraints:

$$2 \cdot 2w_{11,1,1} \leq c_{1,1} \tag{6.58}$$

$$2 \cdot 2w_{11,1,2} \leq c_{1,2} \tag{6.59}$$

$$2w_{11,2,1} + 5w_{11,2,1} + w_{15,2,1} + 5w_{16,2,1} \leq c_{2,1} \tag{6.60}$$

$$2w_{11,2,2} + 5w_{11,2,2} + w_{15,2,2} + 5w_{16,2,2} \leq c_{2,2} \tag{6.61}$$

- Constraints on variables:

$$w_{11,1,1} \geq 0, w_{11,1,2} \geq 0, w_{11,2,3} \geq 0, w_{11,2,4} \geq 0, \\ w_{15,2,3} \geq 0, w_{15,2,4} \geq 0, w_{16,2,3} \geq 0, w_{16,2,4} \geq 0 \tag{6.62}$$

$$w_{15,1} \geq 0, w_{15,2} \geq 0 \tag{6.63}$$

$$w_{16,1} \geq 0, w_{16,2} \geq 0 \qquad \textbf{6.64}$$

6.5 Discussion

There are three potential mapping strategies introduced in this paper. One of them is the Generic Mapping Strategy that includes all the possible mapping ways. Network providers can decide the type and the number of logical servers hosted in the physical nodes according to their needs. Therefore, once the physical network topology is given by the network provider, cost optimization problem can be formulated. Since the physical node can host one or more logical servers, the complexity of the optimization formulation increases quickly due to the dramatic increase of potential candidate paths.

However, when new constraints are introduced into the formulation, the computation time can be reduced significantly. This has been shown in the two special mapping strategies. The customized mapping strategy 1 only allows a physical node host one logical server. The overall optimization problem can then be decomposed into the optimization problem for each logical server. The complexity only depends on the number of physical nodes that support the specific logical server. This mapping strategy can be applied to networks owned by large carriers. The customized mapping strategy 2 is, on the other hand, designed for small carriers. In this strategy, physical nodes are divided into groups. Its complexity then depends on the number of physical nodes in a group. While the optimization complexity is reduced, it also allows multiple logical

serves mapped to the same physical node. This will also make server utilization more efficient.

Last but not the least, the formulations of the mapping strategies proposed in this paper are all based on the novel signaling flow concept we proposed. Without the signaling flow concept, it's not possible to formulate the problems in a scalable way. The next chapter is a summary of this thesis and concludes the main contributions we made.

Chapter 7

Conclusion and Suggestions for Future Work

7.1 Conclusion

As we know, the number of IMS users and their demands on new applications or services are rapidly increasing; therefore, network providers need a model in order to predict the impact of new applications on servers when the new applications are introduced into the market. By having such a model, network providers can predict the network availability as well as maintain the stability of the system, and thereby, increasing their revenue potential. Moreover, due to the combination of various routing scenarios and a large number of session procedures, IMS is an extremely complex system. This leads to difficulty in developing a traffic model that allows for predicting the impact on IMS servers when various available applications are requested by users. Besides, we found in the public domain that there is no such reference related to propose an effective model for IMS system, and the model can ensure the prediction of the load of important IMS servers.

Therefore, in this thesis, we mainly address the problem of developing an efficient traffic model that allows us to predict the server load for IMS system by utilizing the

characteristics of SIP messages. We found there is a potential causal relationship among a sequence of messages in the session procedures, so that the load at their corresponding servers is therefore correlated. Then, we propose a signaling flow based approach in order to present such a relationship among SIP messages, so that the complex correlation structure of the load across different signaling servers can be captured. This approach defines the signaling flow as the aggregation of a sequence of messages that follows the same path in a network of IMS servers. Various combinations of routing scenarios and session procedures can be mapped to the limited number of signaling flows. The routing scenario is defined so as to simplify the signaling flow analysis based on the analysis of call scenario in IMS network. By utilizing the signaling flow analysis, the model is built so that the load of servers can be predicted with a simple mathematical calculation. Thus, information becomes available for us to access the impact on the different servers, while new applications are introduced into the telecommunication market. This model also allows for flexibility when expanding the SIP session procedures in IMS network.

In addition, we develop a measurements architecture based on ESP and ENF in order to collect the network statistics and user behavior. This architecture is established upon IMS network and relies on SIP messages in order to collect and subscribe the information that is of interest to network providers. According to a list of user data collected from the network, the input data of the proposed traffic model can be obtained by a set of mathematical calculations.

We perform the simulation in OPNET. We devise and implement a source model to simulate the users' behavior in IMS network. During the simulation, a set of the network

statistics and user data is collected from the network in order to calculate the input data of the proposed traffic model. The results calculated from the proposed traffic model are verified with the server utilization and mean server queuing delay from the simulation. The proposed traffic model is proven to be acceptable.

Lastly, we formulate IMS network deployment cost optimization issue as a linear programming problem by utilizing the signaling flow based approach. The IMS network mapping strategies are discussed, and we represent them into the proper mathematical notations. Also, the example for each mapping strategies is provided, and we show the formulation for each of them.

In conclusion, the SIP traffic model with a proposed signaling flow based approach is verified through OPNET simulation. It achieves our goal and becomes a new traffic modeling field for IMS system.

7.2 Suggestions for Future Work

In order to perform a real IMS network, retransmission scenario needs to be considered. In the future, the work will extend the proposed traffic model to cover this issue. The retransmission occurs among the physical nodes, in the case of time out for sequential messages, when SIP chooses UDP as a transport protocol. We can formulate each of possible retransmission paths as one individual signaling flow. We set up the packet loss possibility for each signaling flow and apply them into the proposed traffic model. Thereby, we can easily access the impact of retransmission on different IMS servers.

Moreover, since we formulate IMS network deployment cost optimization issue, the capacity of each physical node can be obtained. The assignment of users to each physical

node is considered another potential network mapping issue. Within this topic, IMS network constraints in terms of physical node, logical server, and user are required to consider. One constraint is mentioned in Section 2.1.4, and it is the selection of P-CSCF and S-CSCF always remains fixed for one particular user during his life of registration.

Appendix A Methods of SIP

INVITE:

- When a user agent desires to initiate a session, it formulates an INVITE request. The request asks a server to establish a session and will be forwarded by proxies, and finally arriving at one or more servers that can potentially accept the invitation.

REGISTER:

- When a user agent desires to register to start the services, it formulates a REGISTER request. The request can bind a permanent address to the current location, namely, to notify the SIP network servers about a user agent's current location information.

ACK:

- When the response of the INVITE request is being acknowledged, it formulates a ACK message that contains the header fields of the INVITE request. This message is used in order to confirm the session establishment.

BYE:

- When a user agent or server desires to terminate the session, it formulas the BYE message. The BYE requires a response and acknowledgment prior to the session being released.

CANCEL:

- This request is used to cancel a previous request sent by a client. In the case where a final response to a previous request is given, the CANCEL request has no effect.

UPDATE:

- This message is used to send new information about the session without altering the current established session.

Appendix B IMS-Level Session Registration & De-registration

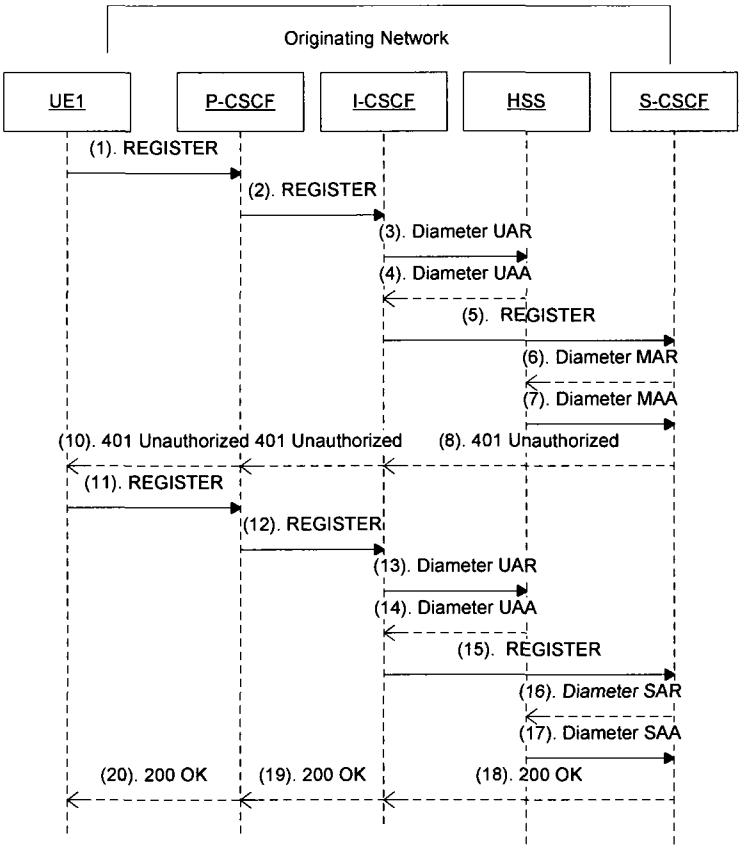
IMS registration is a procedure where the IMS user requests authentication in order to access IMS services in the IMS network [1][5][14][47]. The registration to the IMS network happens prior to the performance of other session procedures; it is mandatory before the IMS user can establish a session. This registration process requires the binding of the IP address of the user device and the user's public identities (e.g. his phone number or SIP address). By doing that, both the device and the core network are protected from unsafe access or attack to the IMS network. The basic registration procedure, when user registers for the first time, is illustrated in Appendix B.1. It is noticed that the originating network can be the home network or the visited network.

Besides the basic registration procedure (when user registers the first time), another scenario, re-registration procedure (when user currently registered), is illustrated in Appendix B.2. The re-registration is initiated by the user who needs to refresh the existing registration or to update a change in the registration status. The details of message sequences are provided in [9].

The de-registration procedure is performed either by the user or the IMS network. When the user wants to de-register from the IMS network, the mobile initiated de-

registration is triggered. Also, the de-registration procedure is illustrated in Appendix B.3; it follows the same signaling path as a basic registration procedure. On the other hand, the IMS core network initiates a network initiated de-registration procedure due to some potential reasons discussed in [9]. In addition, there are two types of network-initiated de-registration procedures specified. One type is designed to deal with the case of registration expiration, and the other type is designed to deal with the cases of network forcing de-registration under any of the approved possible cases. Thus, the network initiated de-registration we covered in the thesis includes de-registration initiated by registration timeout, initiated by HSS, and initiated by service platform [9]. The diagrams of session procedures are illustrated in Appendix B.4, B.5 and B.6, respectively.

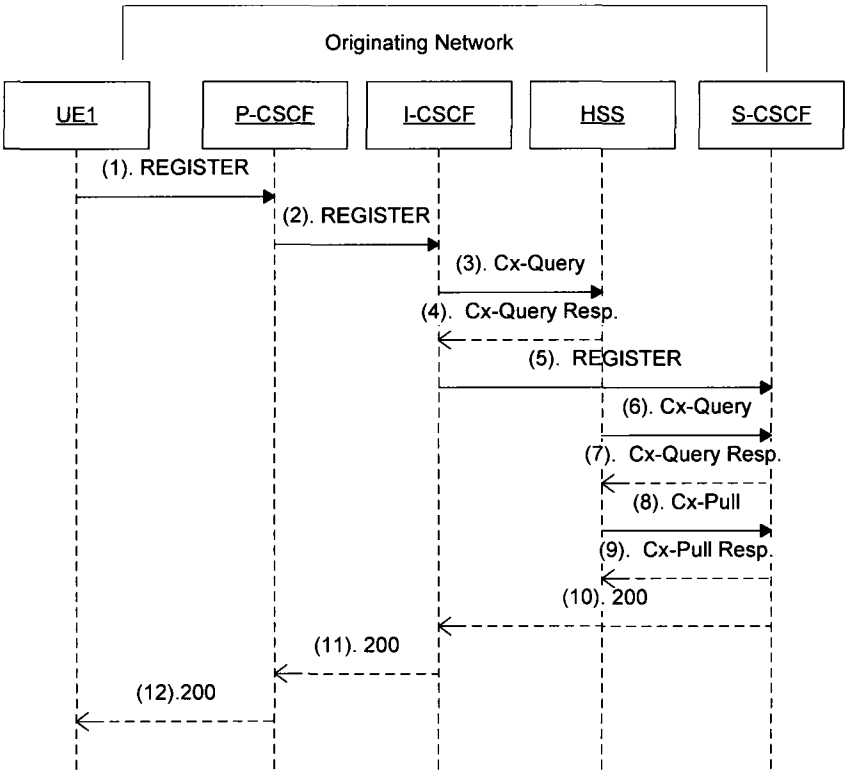
B.1 IMS Registration, when user registers for the first time



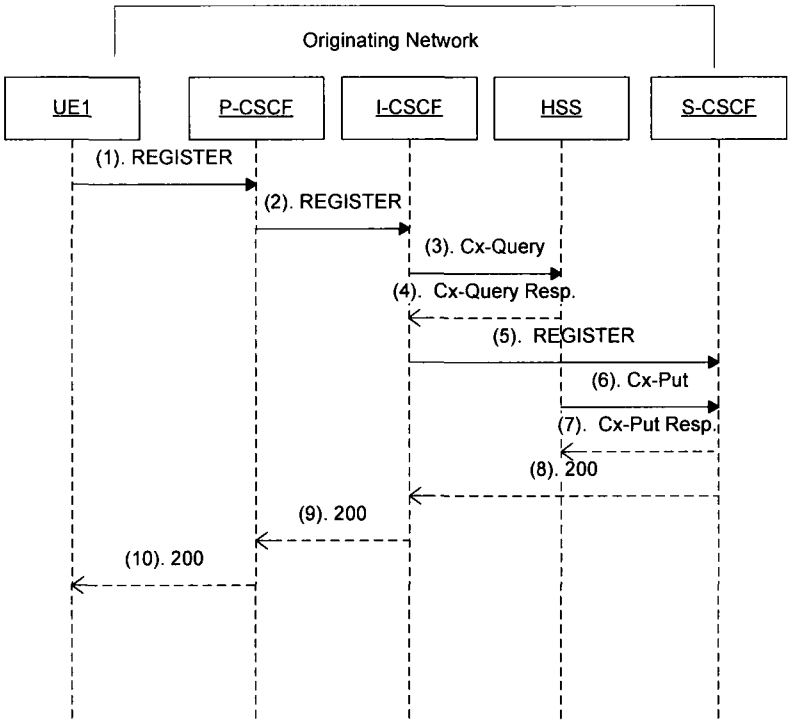
As shown above figure, the registration process is completed with two round trips of message exchange. A SIP REGISTER message (1) is sent by a user (UE1), who requests the registration to a P-CSCF. As we discussed previously, the P-CSCF is located in the same network where the user is located. It continues with the REGISTER message (2) forwarded to the I-CSCF. Once I-CSCF receives it, it does a Diameter UAR (User-Authorization-Request) / UAA (User-Authorization-Answer) dialogue (3, 4) [21] with HSS in order to download a set of S-CSCF capability for I-CSCF to perform an S-CSCF selection procedure. This process mainly requires choosing an appropriate S-CSCF for

this particular UE based on the capabilities received from the Diameter UAA message (4). Then, the REGISTER message (5) is forwarded to the chosen S-CSCF. Once S-CSCF receives the message (5), it contacts the HSS in Diameter MAR (Multimedia-Auth-Request) / MAA ((Multimedia-Auth-Answer) (6, 7) with two purposes, 1). Download authentication vectors for authenticating this particular user; 2). Store this S-CSCF URI in the HSS for future routing this user to the assigned S-CSCF. After that, S-CSCF creates a SIP 401 Unauthorized response (8) to the user, via the I-CSCF and P-CSCF. This message contains a challenge that the user needs to answer. When the user receives the 401 message (10), it produces an appropriate response (11) to this message sent by the user in another REGISTER message. Then, the recipients, P-CSCF and I-CSCF do the same operation as the first REGISTER message. In particular, S-CSCF performs the authentication to the user by received message (15) that contains the user credentials. If the authentication is successful, the S-CSCF does Diameter SAR (Server-Assignment-Request) / SAA (Server-Assignment-Answer) (16, 17) [21] in order to inform the HSS that the user is now registered and to download the user profile. The 200OK response (18) is created and forwarded to the user, and it indicates the success of the registration.

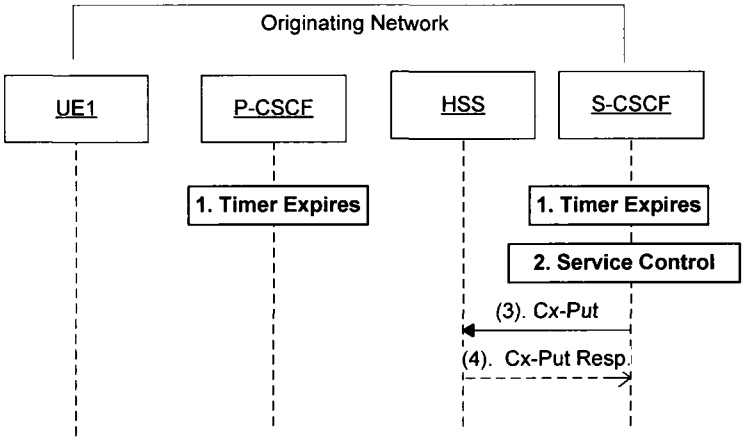
B.2 IMS Re-registration, when user registered already



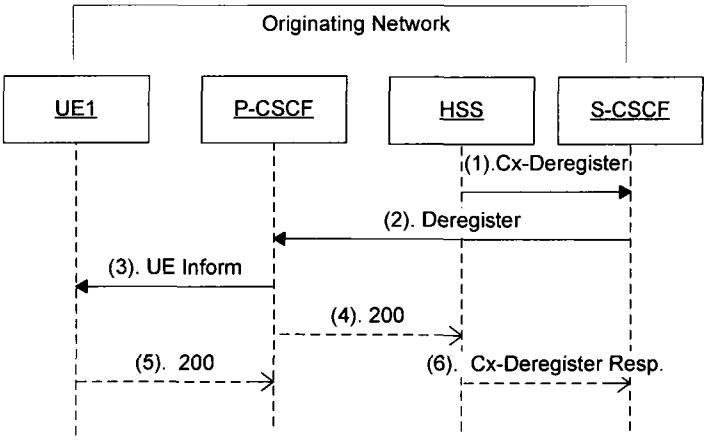
B.3 IMS De-registration, mobile initiated



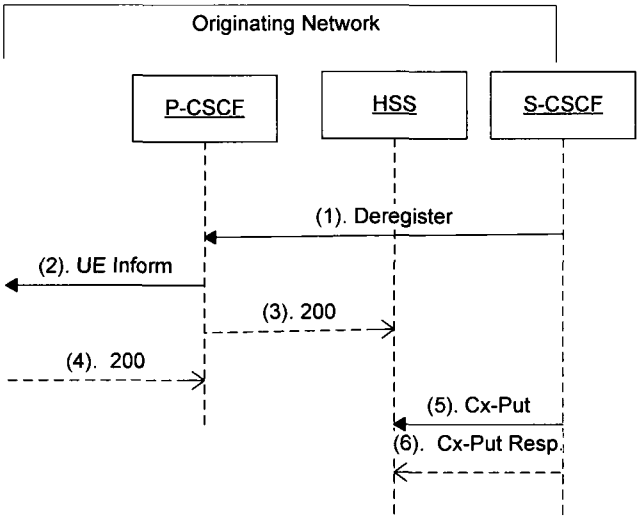
B.4 IMS De-registration, network initiated, registration timeout



B.5 IMS De-registration, network initiated by HSS, administration



B.6 IMS De-registration, network initiated, service platform

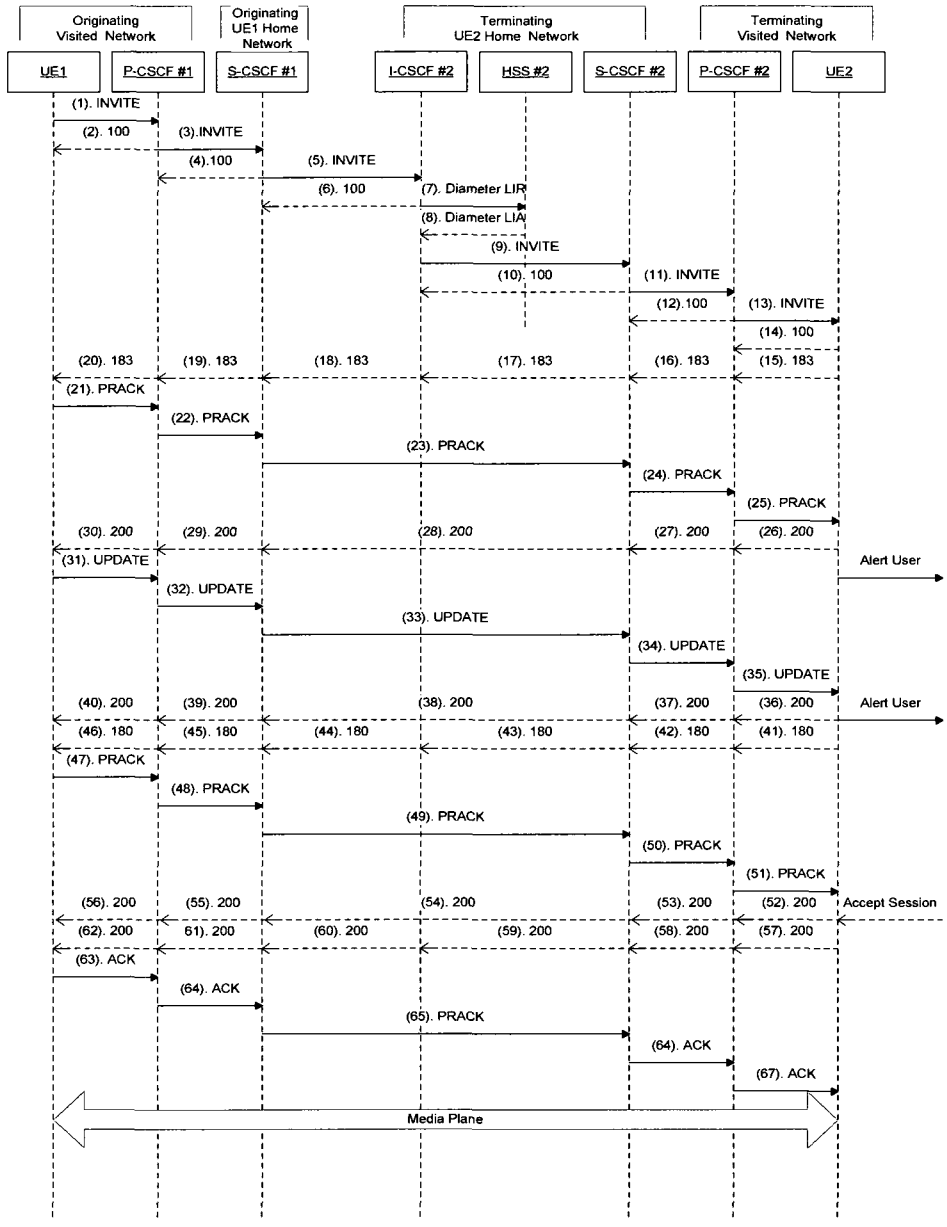


Appendix C IMS-Level Session Initiation

Basic session setup [9][34], as a typical IMS session procedure, is responsible for establishing the multimedia session, where the two end-users are involved in order to start the communication. This session procedure allows the participant parties to negotiate the characteristics of the media data exchanged during the session and also to reserve adequate resources to support the multimedia session. A diagram that describes the signaling involved in basic session setup procedure is presented in Appendix C.1. For the sake of simplicity, we assume as following: 1). the two IMS users establish a session. 2). the two IMS users support the same sort of capabilities. 3). the services are provided by application server, the message exchange with application server is ignored here. 4). both users are roaming at two different visited networks. P-CSCF #1 and S-CSCF #1 located in the originating network are serving UE1, while P-CSCF #1 resides in the visited network of UE1, and S-CSCF #1 is in home network of UE1. P-CSCF #2, S-CSCF #2, I-CSCF #2, and HSS #2 located in the terminating networks, are serving UE 2, while P-CSCF #2 is in the visited network, and S-CSCF #2, I-CSCF #2, and HSS #2 are in home network of UE 2. This assumption constructs the most complicated and complex situation when establishing the session.

In addition to the basic session setup procedure, three re-invite session procedures are listed in Table 2.1. One of them, re-invite for new codec (without I- CSCF) procedure [5][9], shown in Appendix C.2, deals with the case of user generating an invite that is adding another video media to existing established session. Moreover, once the multimedia session is established, a set of media data or codec for a media data may be requested to change, and such a change within the resource is reserved, the re-invite for reserved codec [5][9] is triggered as shown in Appendix C.3. At the end, this procedure at the end successfully executes the new resource request. Nevertheless, if the error occurs in changing codec or media data, the requesting of new resource fails. This refers to the re-invite session procedure (failure happen) [5][9], and it is illustrated in Appendix C.4.

C.1 Basic session setup



As shown in the above figure, we might observe some features from the signaling messages exchange. Firstly, the signaling and media plane are completely separate, since the signaling messages traverse a set of CSCFs and HSS servers, while the media data is

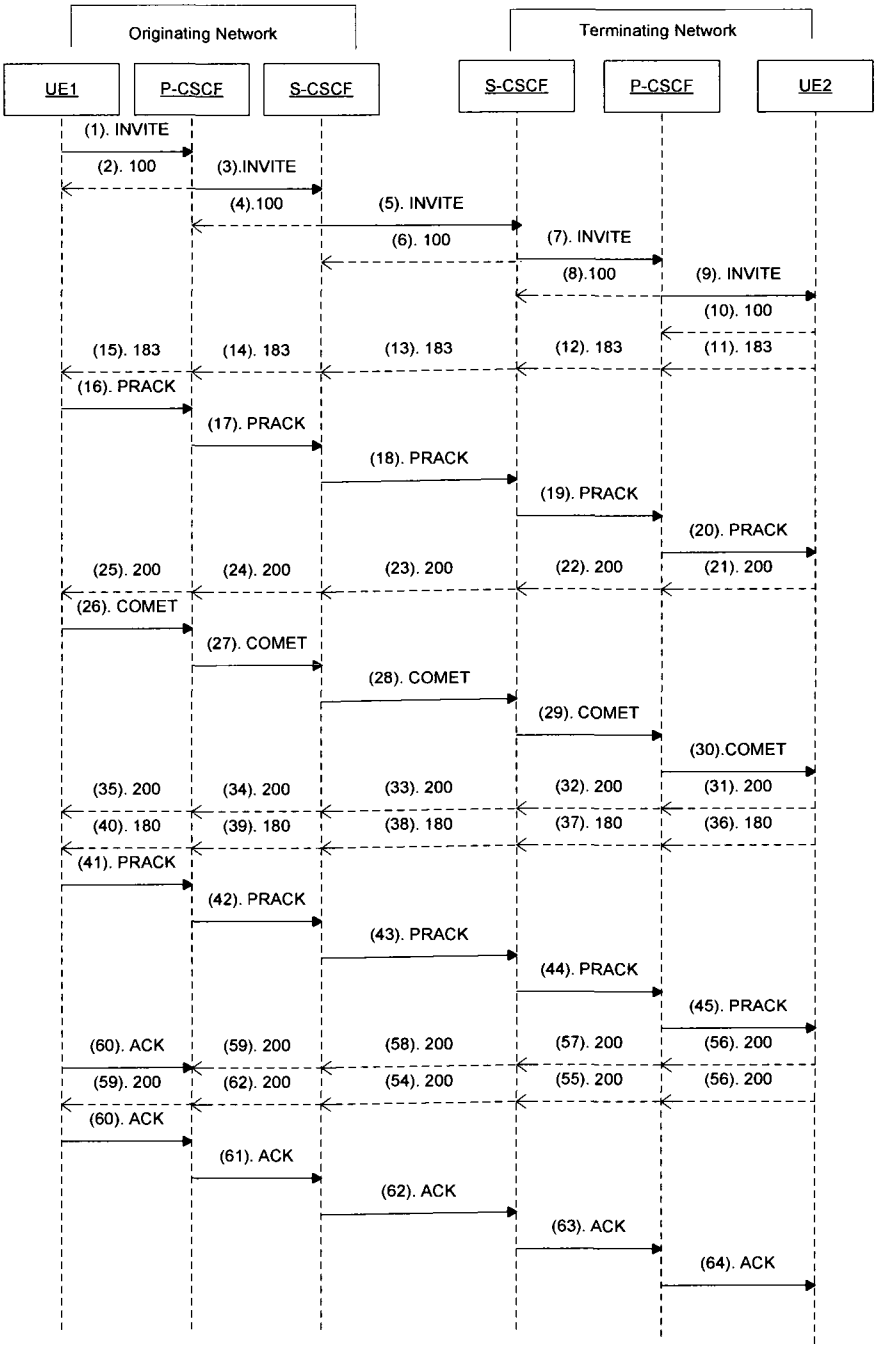
passing between the two end-users. Secondly, all the signaling messages traverse both P-CSCF and S-CSCF in the originating network. This is because P-CSCF is in charge of compressing/decompressing the SIP messages for the end user. At the same time, S-CSCF provides the multimedia services, which the users potentially request, no matter whether the user is roaming or not. The third observation is the interaction between the I-CSCF and the HSS in the terminating network, whereas there is no such interaction in the originating network. This is because the I-CSCF #2 requires discovering the address of S-CSCF #2 serving the destination user. Thus, the signaling message sequences are asymmetric and this requires we do the signaling traffic analysis on the originating and terminating networks separately.

The above figure illustrates that the basic session setup procedure starts with the sending of SIP INVITE request (1) from one user (UE1) as the caller to S-CSCF #1, via P-CSCF #1. As the INVITE request (3) arrives at S-CSCF #1, who is allocated to the caller at the registration procedure, S-CSCF #1 identifies the user who originated the INIVITE request. Since the user file was downloaded by the S-CSCF #1 at the registration, the so-called initial Filter Criteria (iFC) stored in the user file can be obtained. Based on the iFC, S-CSCF #1 can control the services being requested by the user from application server. Besides that, S-CSCF #1 is the first node that tries to route the INVITE request according to the destination stored in the Request-URI (Uniform Resource Identifier) field of the request. There are two types of contents of Request-URI, a SIP URI [48], or a TEL URL (Uniform Resource Locator) [49]. In the case of a SIP URI found in a Request-URI filed, the regular SIP routing procedures are applied;

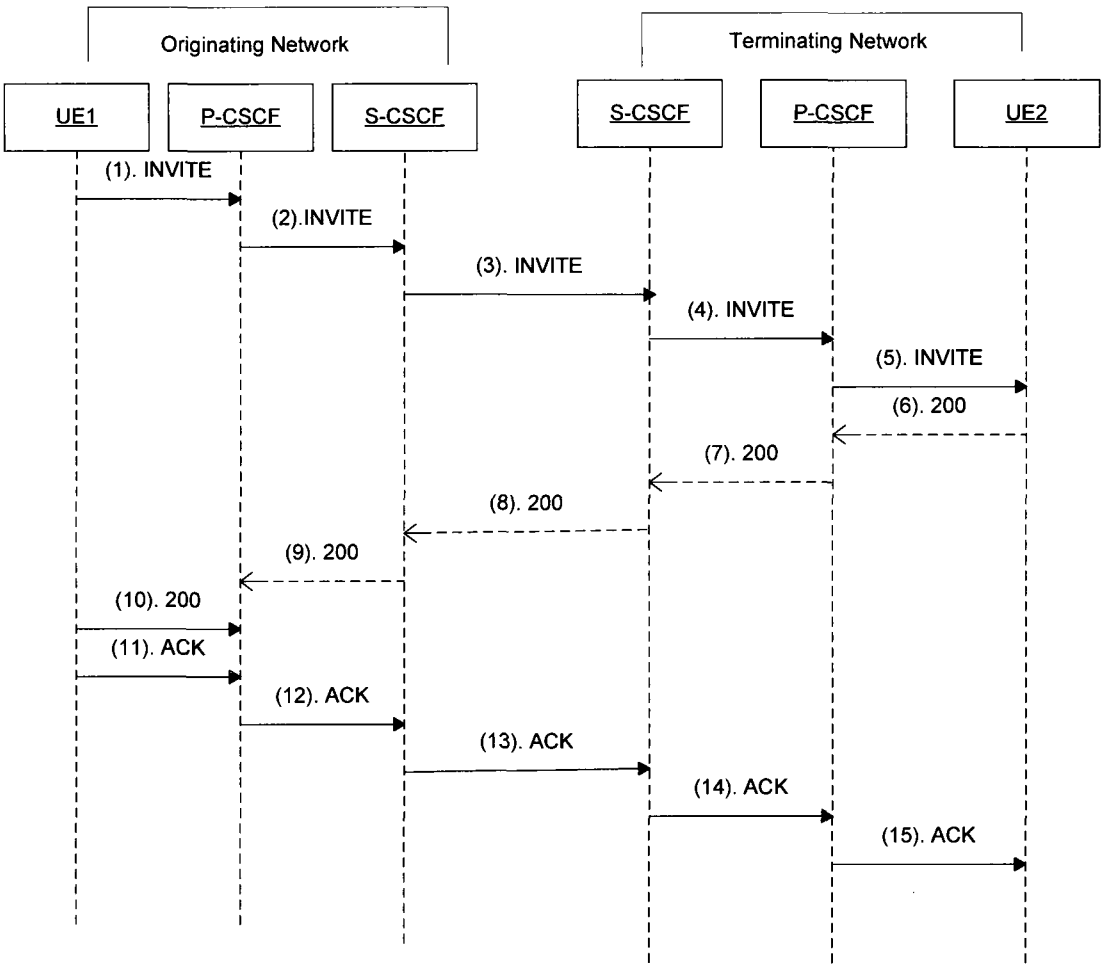
however, in the case of finding a TEL URI, the mapping of the TEL URL to a SIP URI is applied. Sequentially, a SIP server in the destination home network is found through querying DNS. Then, S-CSCF #1 forwards the request (5) to it, I-CSCF #2. After I-CSCF #2 identifies the callee through the Request-URI of the INVITE request and it obtains the address of S-CSCF #2 from the HSS by message (7, 8), the INVITE request (9) is forwarded to S-CSCF #2 allocated to the callee. As the S-CSCF #2 receives the request (9), the evaluation of iFC of the called user is performed, which includes the search of services that apply to sessions established towards the user. Finally, after S-CSCF #2 completes the evaluation of iFC, the request (11, 13) is replayed to UE2 via P-CSCF #2. The following SIP messages, 183 SESSION IN PROGRESS (15-20) , PRACK (21-25) and UPDATE (31-35), and the first INVITE request (1,3,5,9,11,13) carry an SDP part in the body of message that describes the media available and the resource requested to reserve at either end users. Then, the media negotiation and resource reservation are completed under the process of SIP messages (3-40) [9][34].

After the resource is successfully reserved at UE1, the UPDATE message (31-35) as a notification is sent to UE2. The SIP 200OK message (36-40) sent to UE1 is used to confirm the UE1's resource reservation. The rest of SIP messages (41-67) passing between two UEs are used in order to complete the session establishment. This is the simplified description of a basic session setup procedure; please refer to [5][9] for more details.

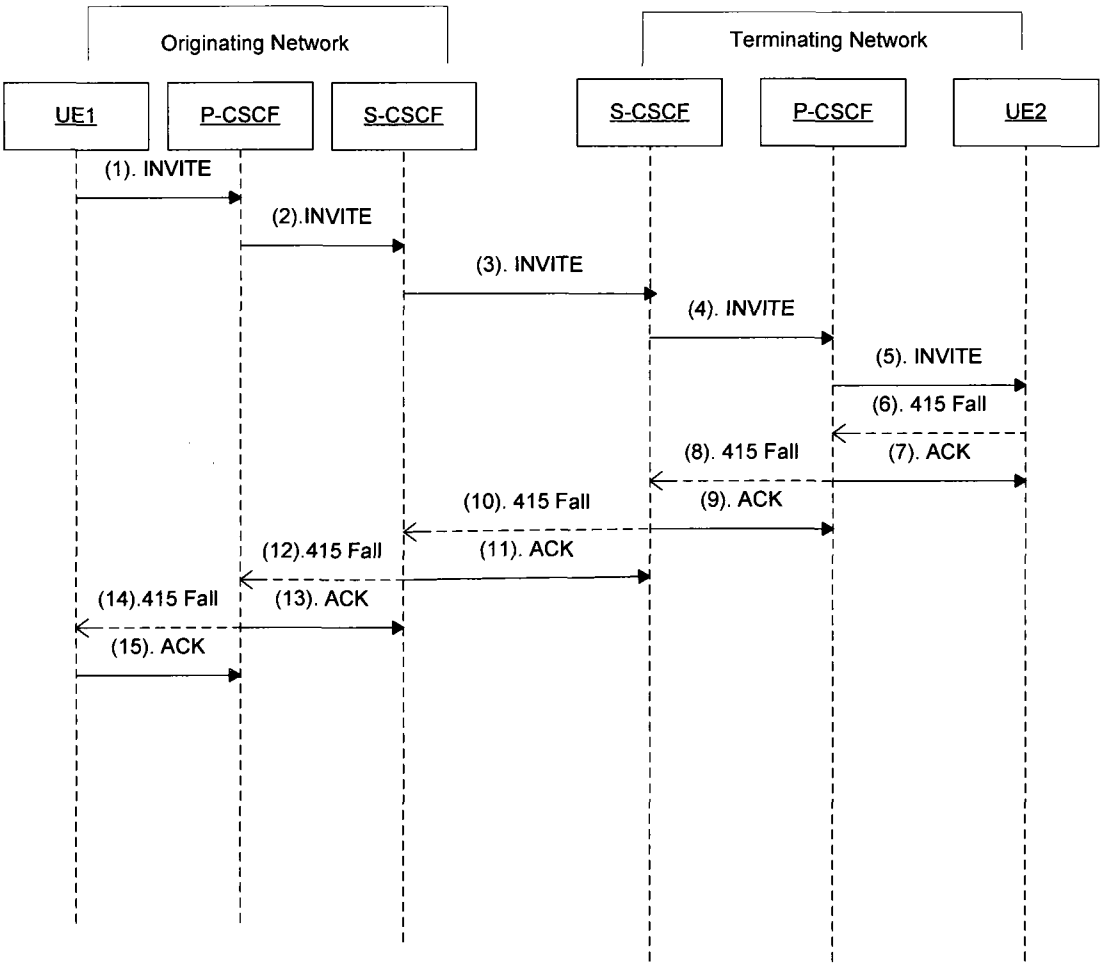
C.2 Re-Invite for new codex, without I-CSCF



C.3 Re-Invite for reserved codec



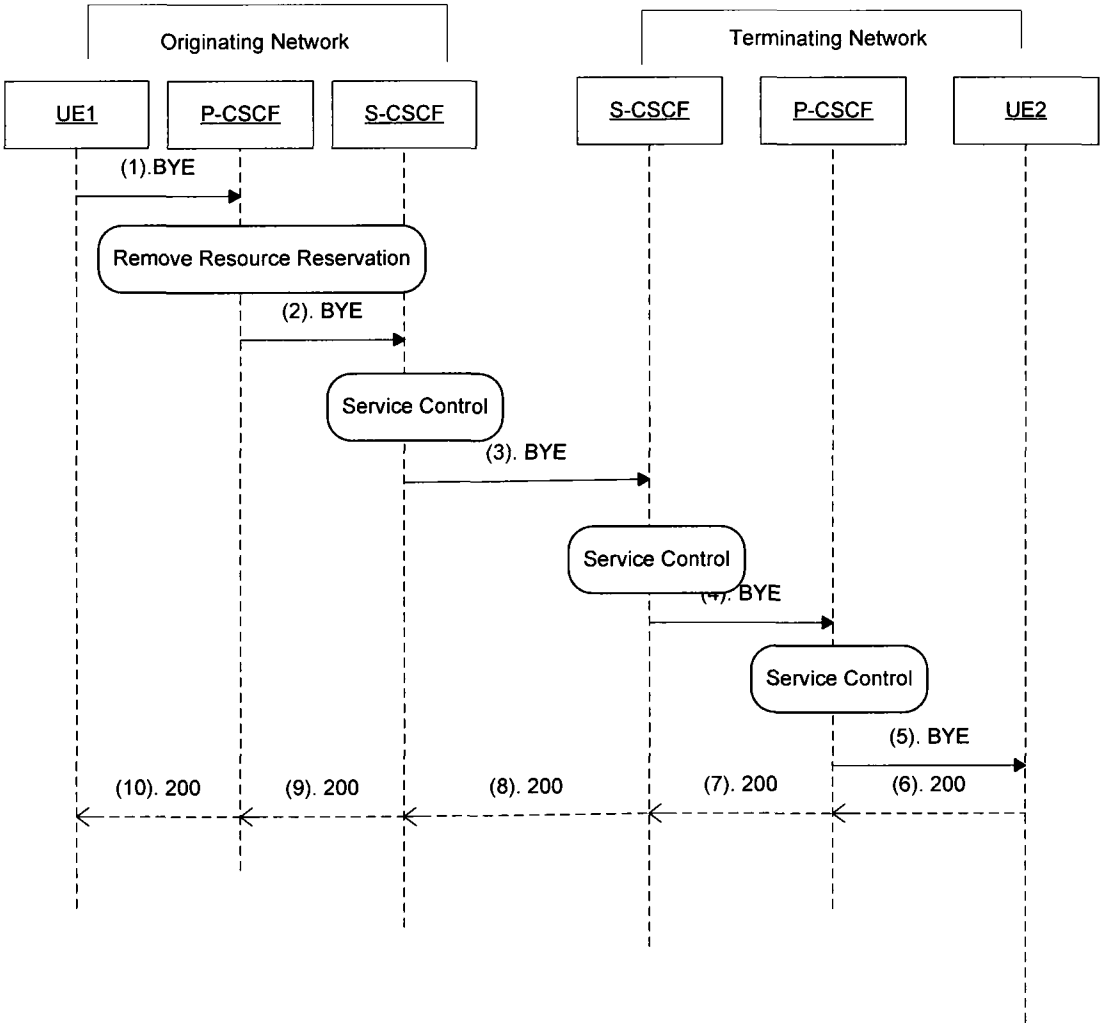
C.4 Re-Invite, failure happen



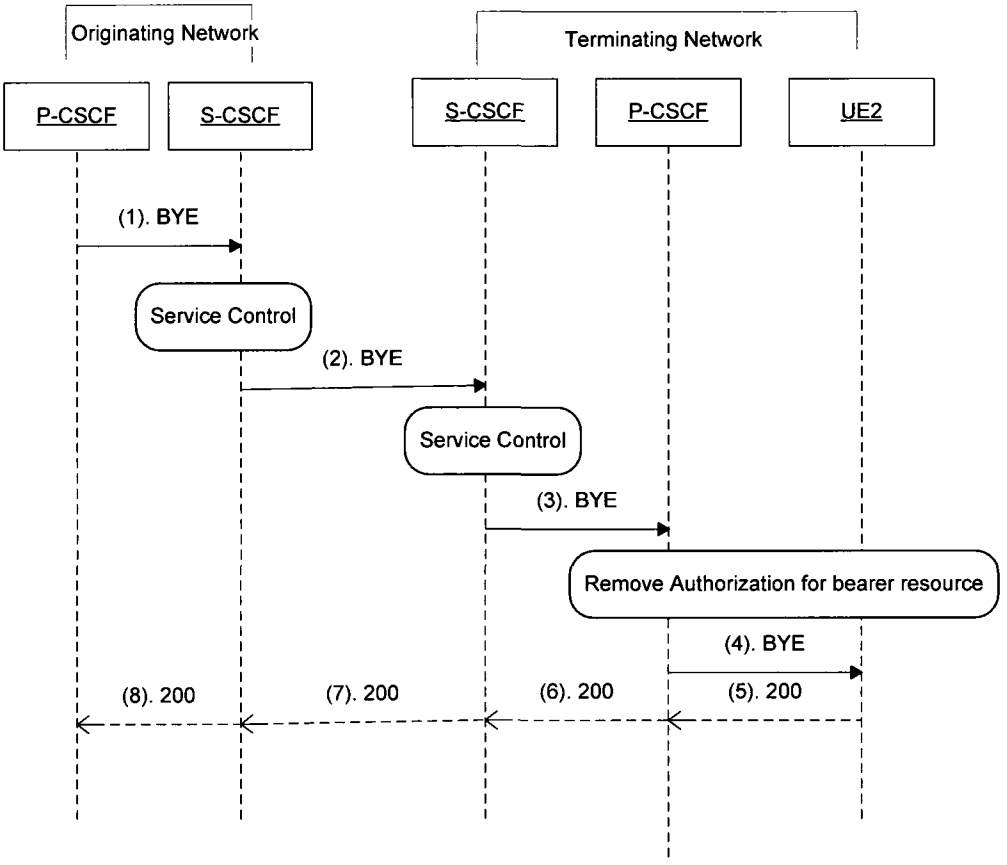
Appendix D IMS-Level Session Termination

Session termination allows the session to be released, as listed in Table 2.1. Session termination is either initiated by mobile terminal or network. Once data in a particular session has been exchanged, the end users from either side can request to release the session. This refers to the session procedure of mobile terminal initiated session release. The network initiated session release (P-CSCF initiated) procedure is triggered by some possible reasons, such as UE out of signaling coverage, or UE accidental removal [5][9]. The illustrations for these two session procedures are given in Appendix D.1and D.2, and the details of SIP message sequences are provided in [5][9].

D.1 Mobile terminal initiated session release



D.2 Network initiated session release P-CSCF initiated



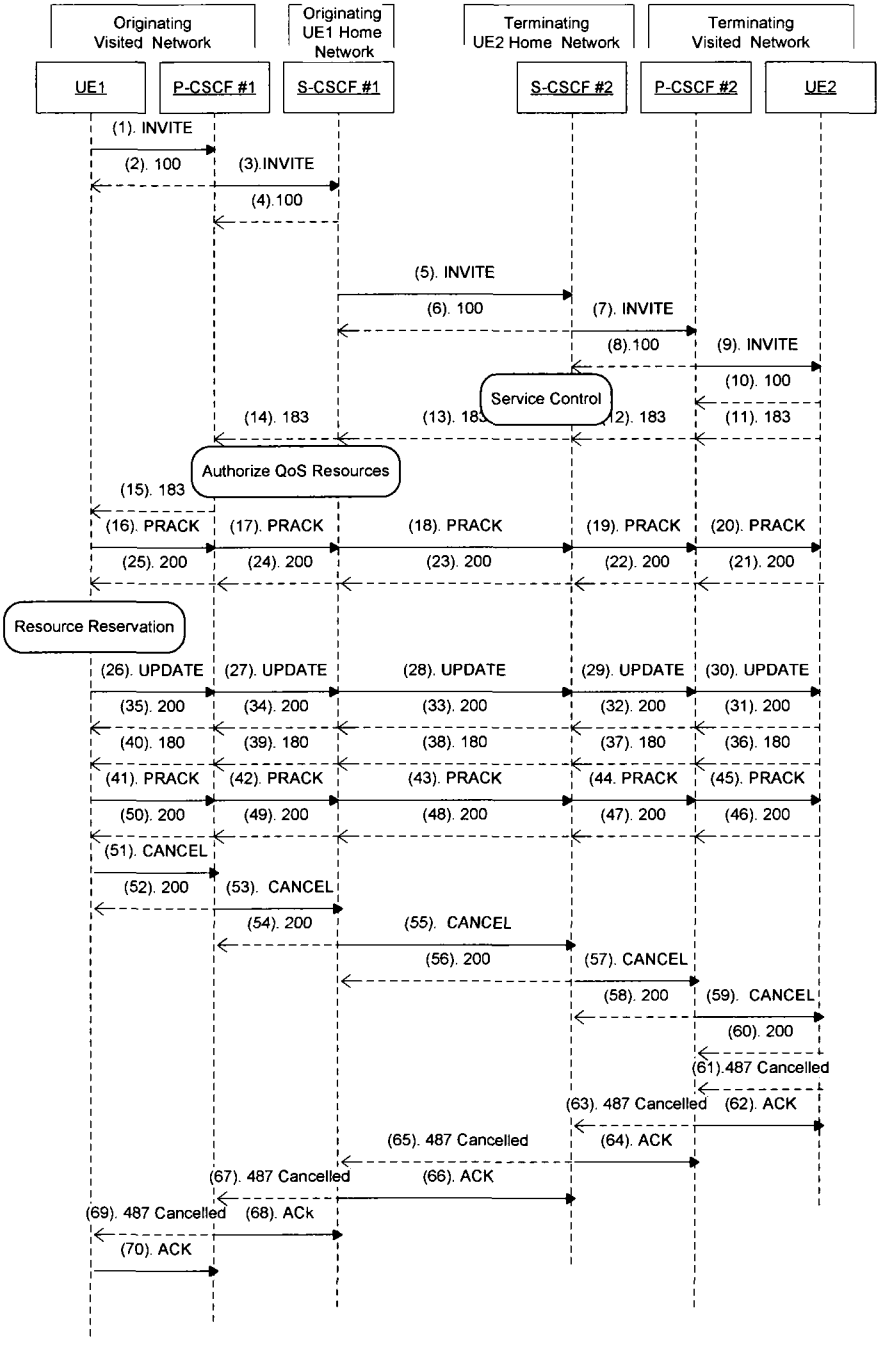
Appendix E IMS-Level Session Failure

Session failure specifies a set of session procedures in IMS for handling failures in the session establishment. In general, they are classified into two categories, failure in origination procedure and failure in termination procedure. Furthermore, the failure occurs in origination procedure due to two possible cases, depicted in Appendix E.1 and Appendix E.2, respectively. One case specifies that session is abandoned due to user command as shown in Appendix E.1; it could happen at any point between messages #11 to #50. However, in our signaling analysis, we assume that the failure occurs at message #50 for this case. Also, another case, in Appendix E.2, describes the session as aborted due to a failure to obtain resources by the originator; it could occur prior to message #26. However, in our signaling analysis, we assume that the failure occurs at message #26 for this case.

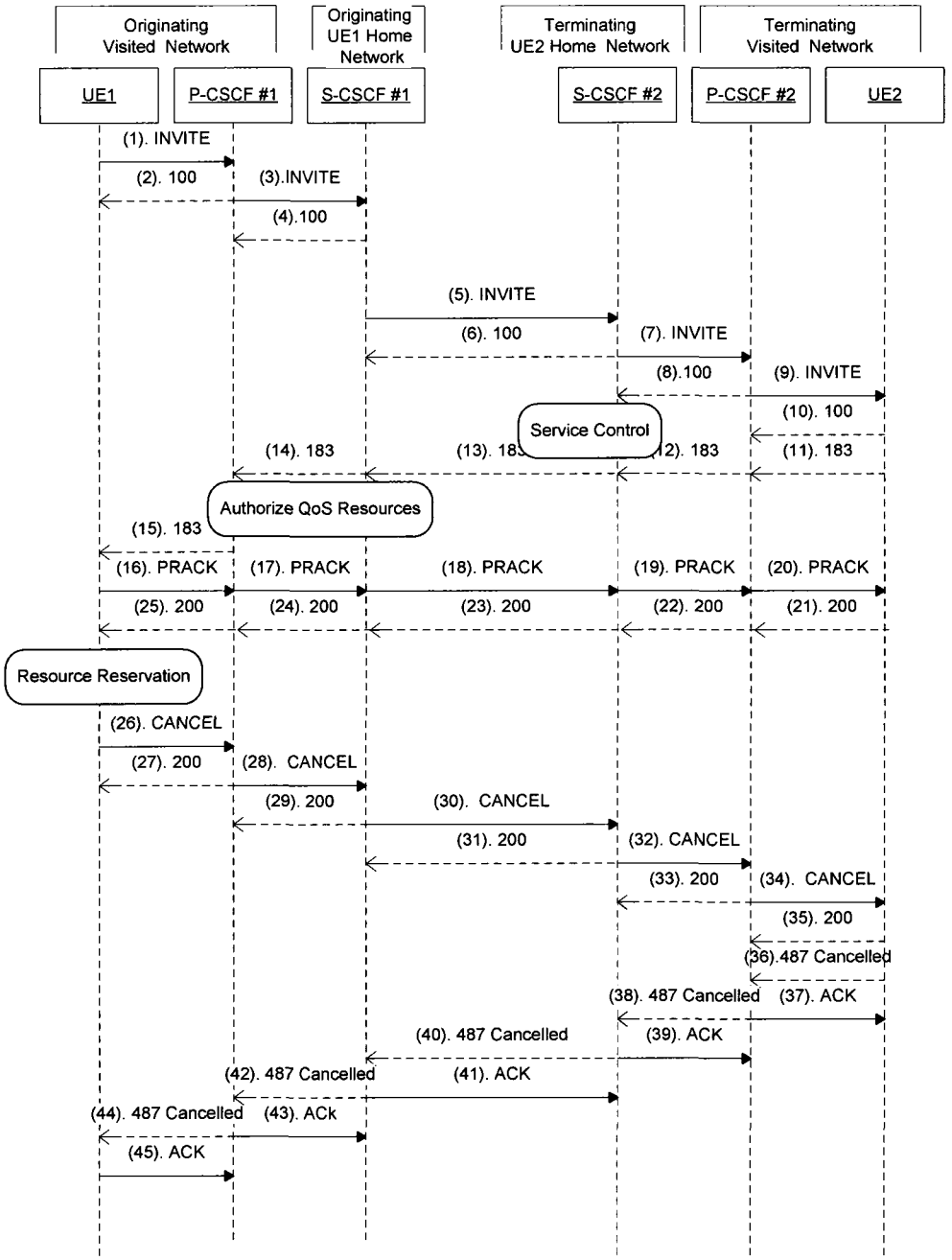
Besides that, in the case where failure happens in the termination procedure, the end user that initiates a session encounters a failure due to an error detected in the termination procedure, depicted in Appendix E.3. This error could be from, for example, destination busy (error code 486), destination service denied (error code 403), destination currently out of coverage (error code 480), or some other error [5]. When the error situation is detected during a set of procedures in Session Initiation, UE could be at different stages

between messages #6 to #50, as shown in Appendix E.3. In our signaling analysis, we assume the error happens at message #50. The last failure case is that terminator rejects to establish the session with originator, and it is illustrated in Appendix E.4.

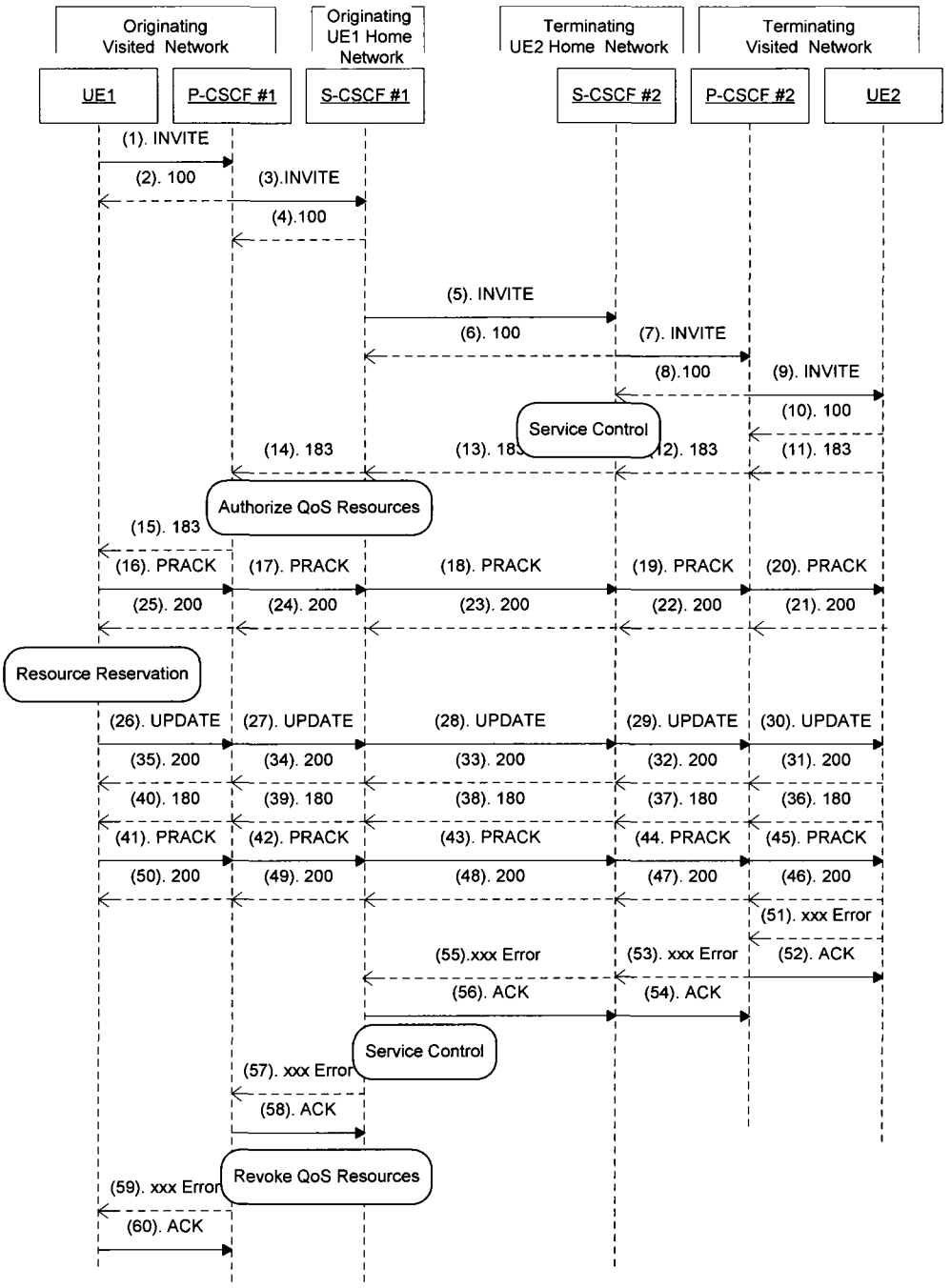
E.1 Failure in session abandon, in origination procedure



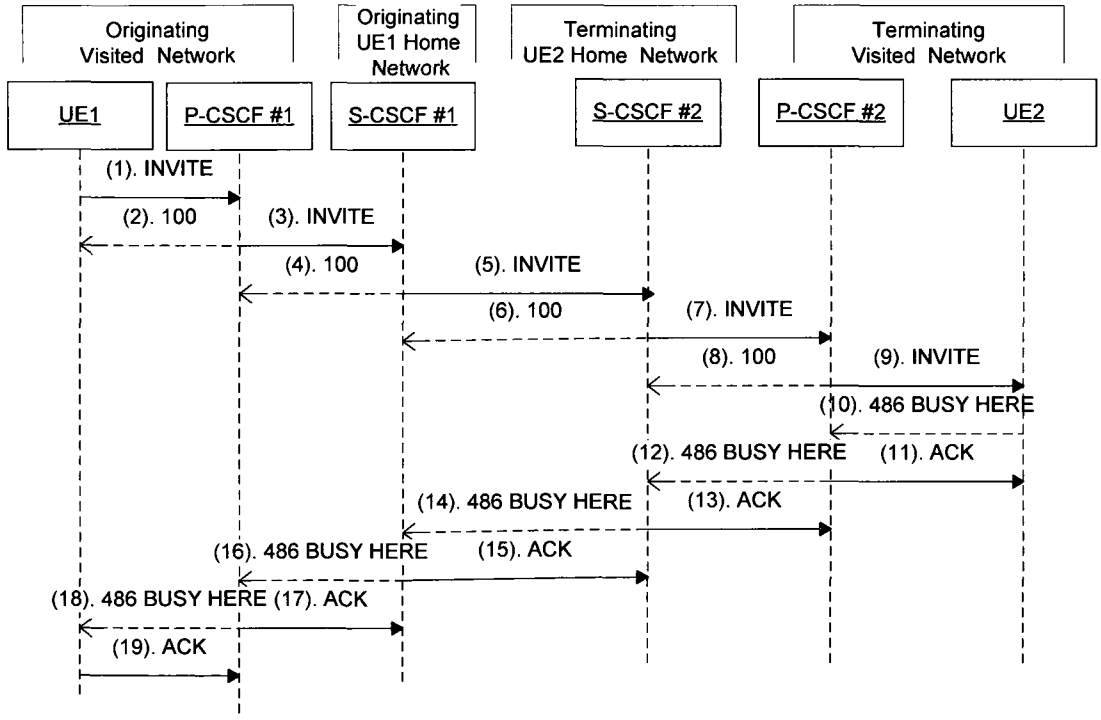
E.2 Failure in obtaining resource, in origination procedure



E.3 Failure in termination procedure



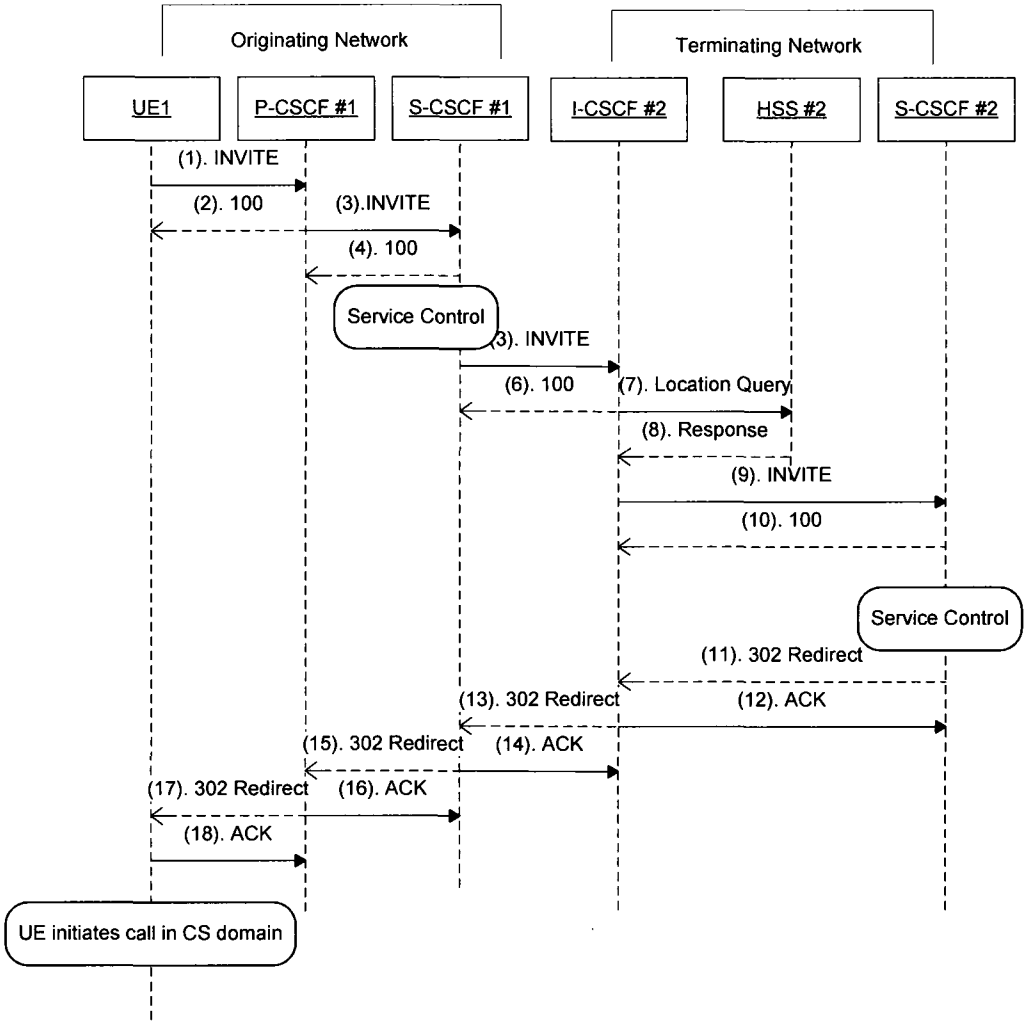
E.4 Rejection by termination procedure



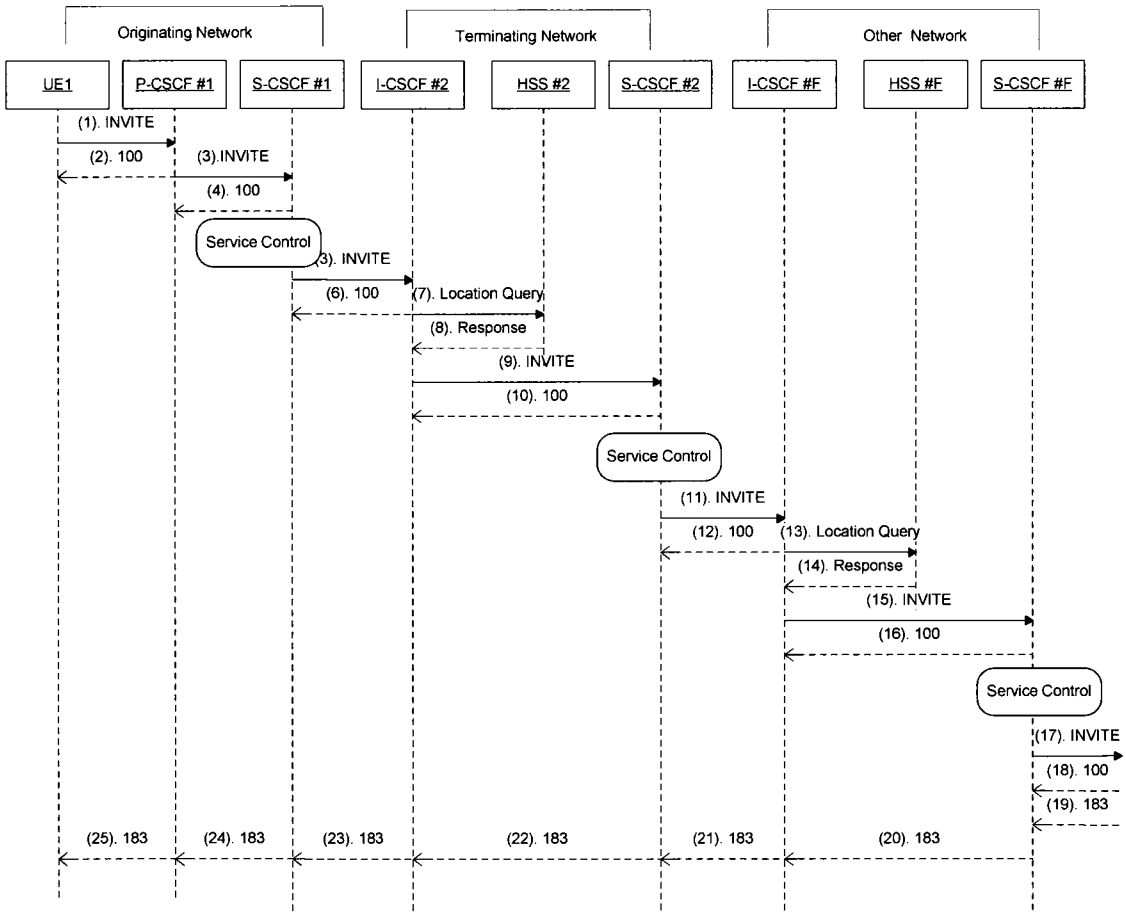
Appendix F IMS-Level Session Redirection

The session redirection performs the purpose of redirecting a session to a different destination. However, the decision to redirect a session can be made by many different reasons and at different points during the session establishment. It also may be involved in a large number of different functional elements. The detailed interpretation of causing a redirection is provided in [5]. Generally speaking, session redirection is initiated by S-CSCF to CS-domain, by S-CSCF to IM CN Subsystem, by P-CSCF, or by UE, illustrated in Appendix F.1, F.2, F.3, and F.4, respectively. The explanation of message flows in each session procedure can be found in [5].

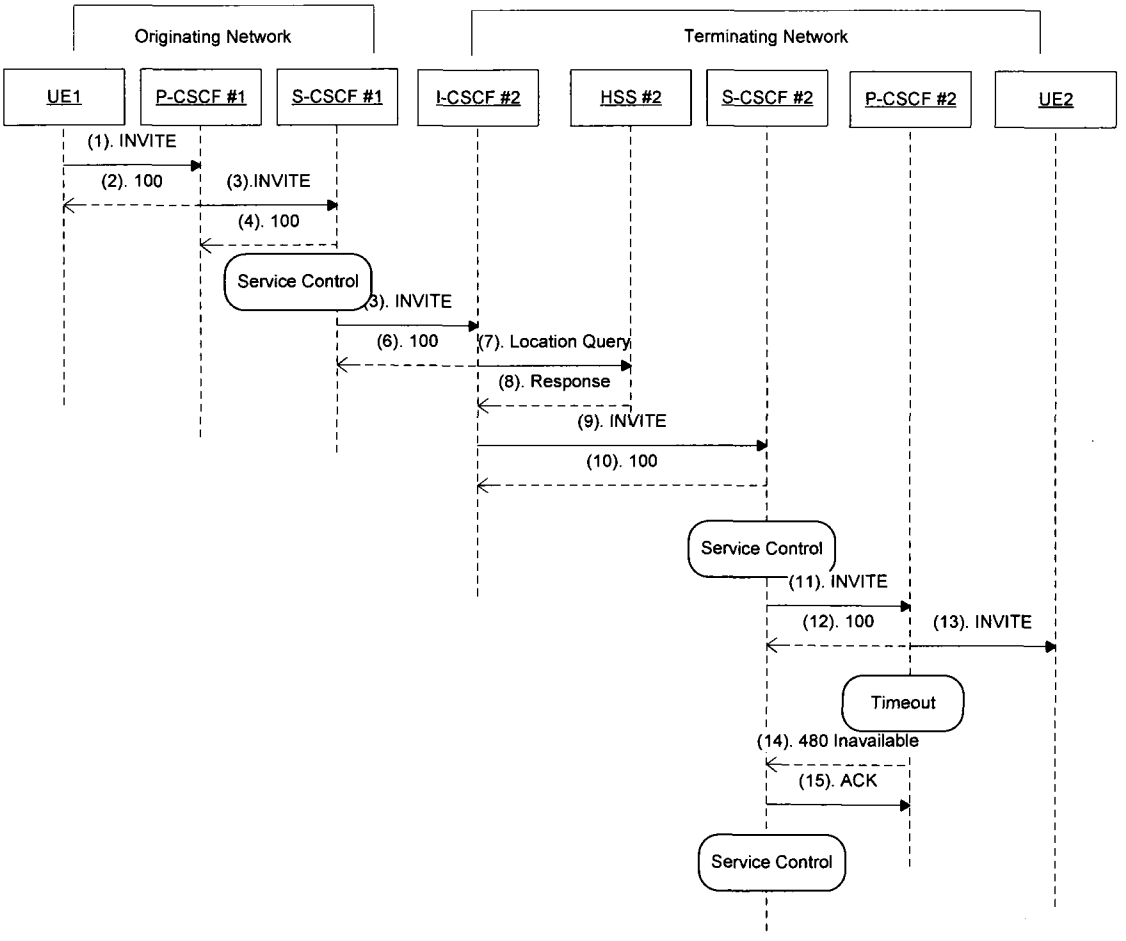
F.1 Redirection initiated by S-CSCF to CS-domain



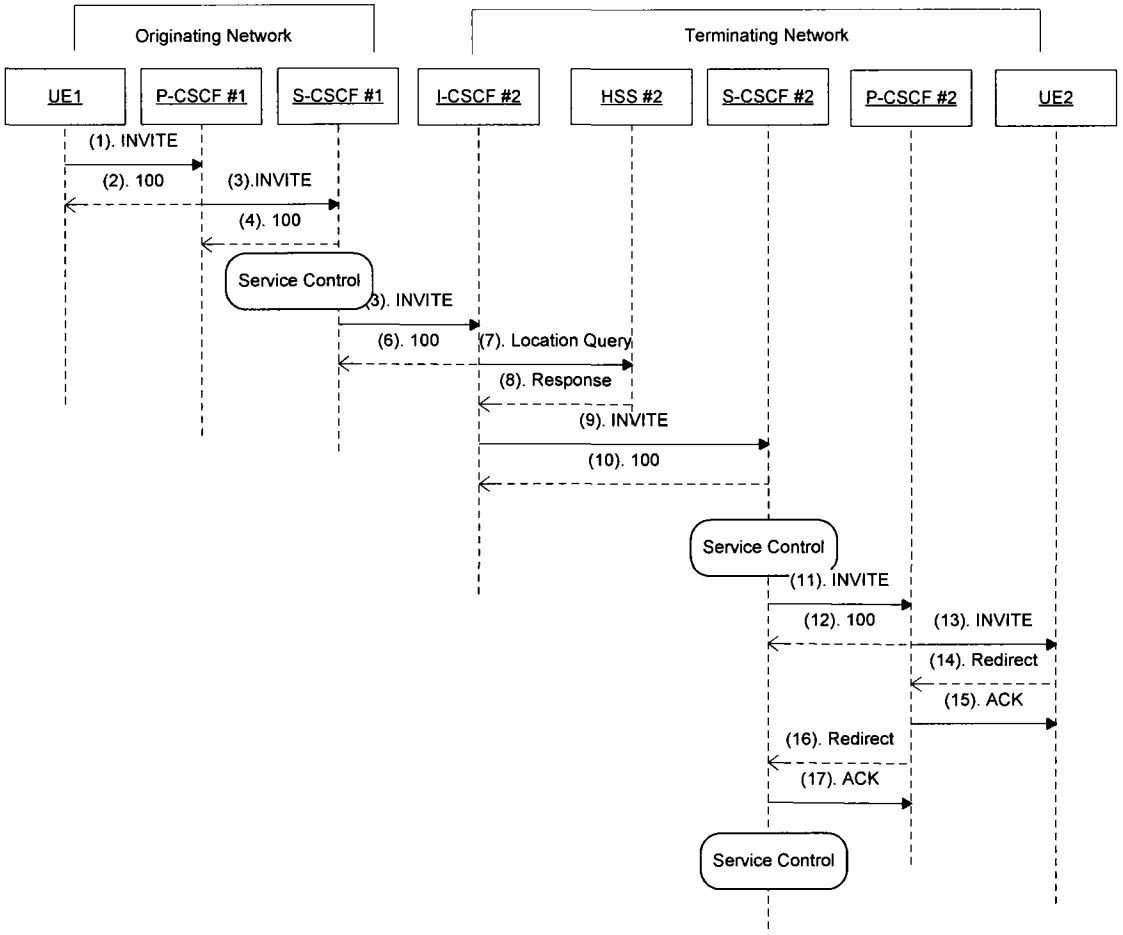
F.2 Redirection initiated by S-CSCF to IM CN subsystem



F.3 Redirection initiated by P-CSCF



F.4 Redirection initiated by UE



Appendix G All Possible Call Scenarios

No.	Call Scenarios	Descriptions
1	H1 → H1	(H1,O) calls (H1,T): A non-roaming Network #1 user calls a non-roaming Network #1 user.
2	H1 → V1	(H1,O) calls (V1,T): A non-roaming Network #1 user calls a Network #2 user roaming at Network #1.
3	V1 → H1	(V1,O) calls (H1,T): A Network #2 user roaming at Network #1 calls a non-roaming Network #1 user.
4	H1 → H2	(H1,O) calls (H2,T): A non-roaming Network #1 user calls a non-roaming Network #2 user.
5	H2 → H1	(H2,O) calls (H1,T): A non-roaming Network #2 user calls a non-roaming Network #1 user.
6	H1 → V2	(H1,O) calls (V2,T): A non-roaming Network #1 user calls a Network #1 user roaming at Network #2.
7	V2 → H1	(V2,O) calls (H1,T): A Network #1 user roaming at Network #2 calls a non-roaming Network #1 user.
8	V1 → V1	(V1,O) calls (V1,T): A Network #2 user roaming at Network #1 calls a Network #2 user roaming at Network #1.
9	V1 → H2	(V1,O) calls (H2,T): A Network #2 user roaming at Network #1 calls a non-roaming Network #2 user.
10	H2 → V1	(H2,O) calls (V1,T): A non-roaming Network #2 user calls a Network #2 user roaming at Network #1.
11	V1 → V2	(V1,O) calls (V2,T): A Network #2 user roaming at Network #1 calls a Network #1 user roaming at Network #2.
12	V2 → V1	(V2,O) calls (V1,T): A Network #1 user roaming at Network #2 calls a Network #2 user roaming at Network #1.
13	V2 → V2	(V2,O) calls (V2,T): A Network #1 user roaming at Network #2 calls a Network #1 user roaming at Network #2.
14	V2 → H2	(V2,O) calls (H2,T): A Network #1 user roaming at Network #2 calls a non-roaming Network #2 user.
15	H2 → V2	(H2,O) calls (V2,T): A non-roaming Network #2 user calls a Network #1 user roaming at Network #2.
16	H2 → H2	(H2,O) calls (H2,T): A non-roaming Network #2 user calls a non-roaming Network #2 user.

Appendix H Signaling Flow Analysis on Basic Session Setup, in 5 Routing Scenarios

H.1 Basic session setup, (H_1, T)

In a routing scenario, (H_1, T), all the servers, P-CSCF #1, HSS #1, S-CSCF #1, and I-CSCF #1, at the terminating network are located in the home network, Network #1, as shown in Figure 1. The signaling traffic traversing through these four servers is taken into account. The detailed explanation of the message sequences is provided in Appendix C.1. It is found that the message sequences, (INVITE, 183) are forwarded from the originating network via I-CSCF #1, S-CSCF #1, and P-CSCF #1. This message path is considered to be a signaling flow, numbered as 4. Since signaling flow 4 is a round trip and has no more coming message sequences following the same path, the volume of this signaling flow is 2. For the message sequences, (PRACK, 200), the PRACK message received by S-CSCF #1 is forwarded to P-CSCF #1. Then, the 200 message, as a response to PRACK, follows the reversed path as the PRACK message has. This message path, → S1 → P1 →---→ P1 → S1→, is treated as signaling flow 5. Other two message sequences, (UPDATE, 200) and (PRACK, 200) are found to follow this path, and the volume of this signaling flow is 6. Finally, signaling flow 2 and signaling flow 6 are extracted from the procedure. The details are shown in Table 1.

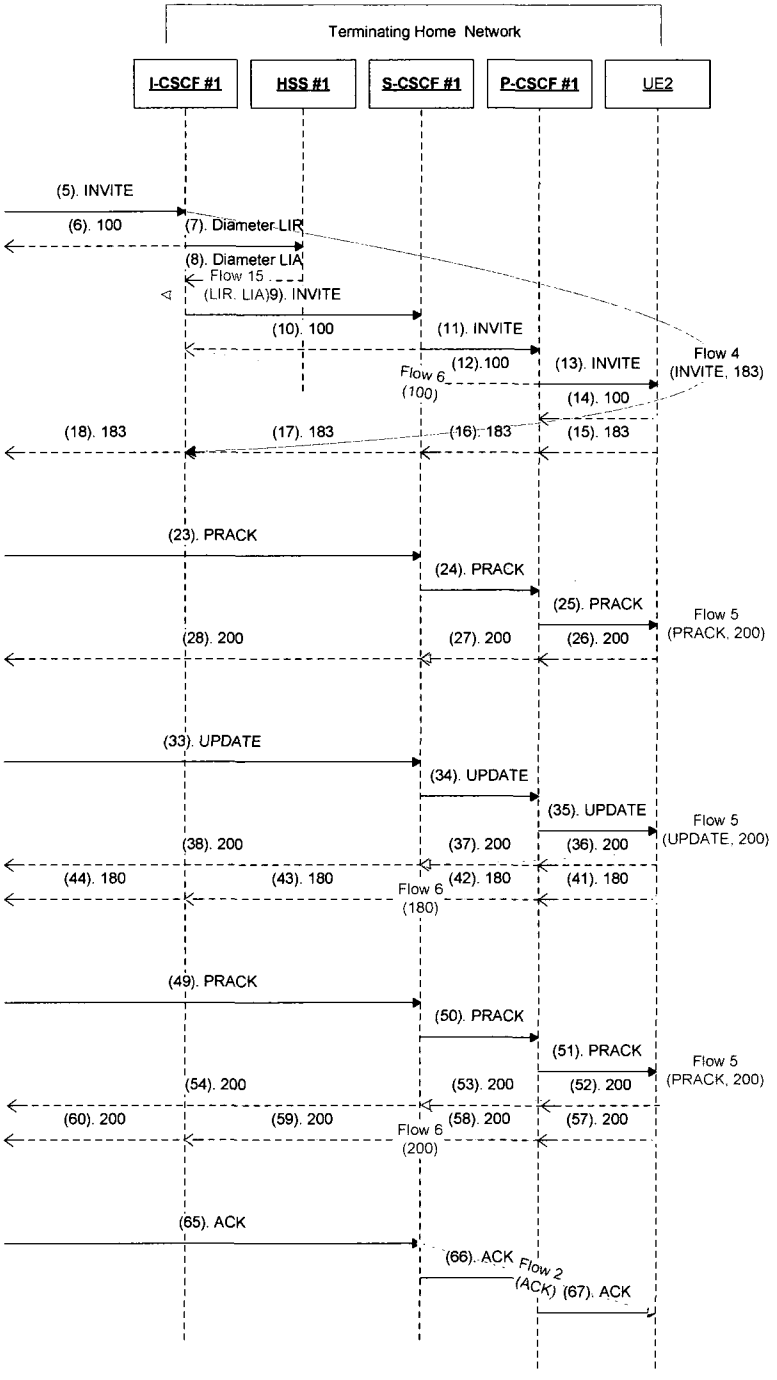


Figure 1: IMS basic session setup procedure with signaling flow analysis, in (H_1, T)

Table 1: Signaling flow summery for basic session setup, in (H_1, T)

Signaling Flows	Message Sequences	Signaling Flow Path	Signaling Flow Volume
2	ACK	→ S1 → P1 →	1
4	(INVITE, 183)	→I1 → S1 → P1 →--- → P1 → S1 → I1 →	2
5	(PRACK, 200), (UPDATE, 200), (PRACK, 200)	→ S1 → P1 →---→ P1 → S1 →	6
6	100,180,200	→ P1 → S1 → I1 →	3
15	(Diameter LIR, Diameter LIA)	→ HSS → I1	1

H.2 Basic session setup, (V_1, O)

When a basic session setup procedure is performed in an originating routing scenario, (V_1, O), Network #2 user is roaming at the visited network, Network #1, to initiate the invitation, as show in Figure 2. Also, the P-CSCF is our concern since it is located in Network #1, as stated in Table 3.2. The signaling traffic traversing through the P-CSCF #1 server is taken into account, no matter which direction that the traffic travels from. Since only P-CSCF #1 is involved in this case, the signaling flow can be defined as the SIP message traversing the server, P-CSCF #1. The path of this signaling flow can be in two directions, which are → P1 and ← P1. This signaling flow is numbered as 10. There are 12 message sequences found in this procedure and that follow this path; the volume of signaling flow 10 is 12. The details are presented in Table 2.

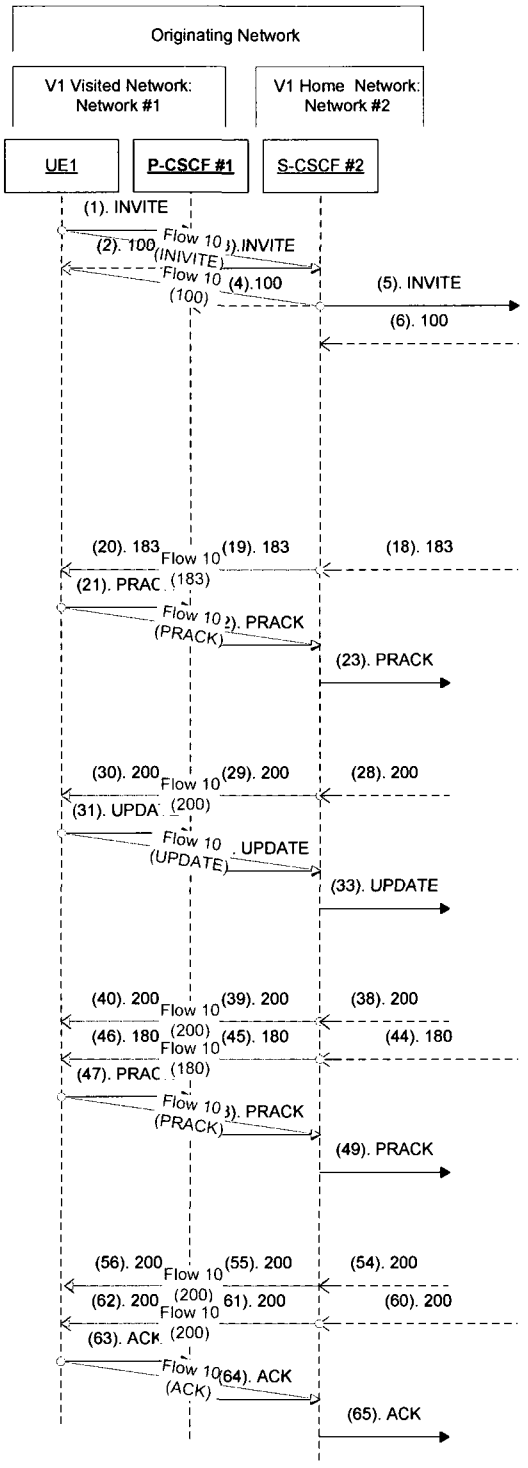


Figure 2: Signaling flow analysis on basic session setup, in (V_1, O)

Table 2: Signaling flow summery for basic session setup, in (V_1, O)

Signaling Flows	Message Sequences	Signaling Flow Path	Signaling Flow Volume
10	INIVITE,100,183, PRACK, 200, UPDATE, 200, 180, PRACK, 200, 200, ACK	--- P1 ---	12

H.3 Basic session setup, (V_1, T)

In the case of a terminating routing scenario, (V_1, T), as illustrated in Figure 3, P-CSCF is still our concern since it is located in the Network #1, as stated in Table 3.2. The signaling traffic traversing through the P-CSCF #1 server is taken into account, no matter which direction the traffic travels from. Following the same idea of an originating routing scenario, (V_1, O), the signaling flow 10 is used in order to qualify all the message sequences traversing the P-CSCF #1. The details are provided in Table 3.

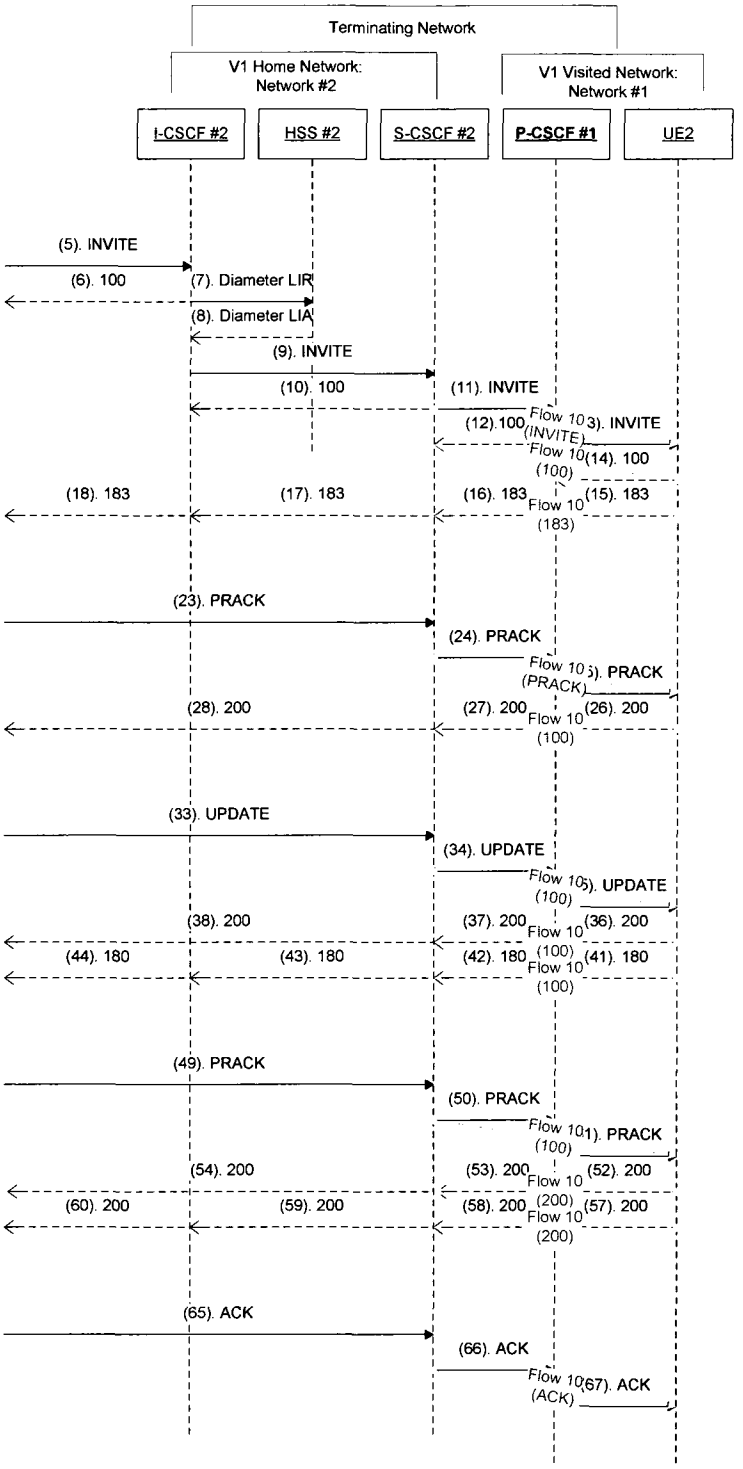


Figure 3: IMS basic session setup procedure with signaling flow analysis, in (V_1, T)

Table 3: Signaling flow summery for basic session setup, in (V_1, T)

Signaling Flows	Message Sequences	Signaling Flow Path	Signaling Flow Volume
10	INIVITE,100,183, PRACK, 200, UPDATE, 200, 180, PRACK, 200, 200, ACK	--- P1 ---	12

H.4 Basic session setup, (V_2, O)

(V_2, O) user belongs to Network #1, and he is roaming at a visited network, namely, Network #2. Then, P-CSCF is located in the same network, Network #2, as the user resides, as opposed to S-CSCF, who resides in the user's home network, Network #1. Figure 4 illustrates the case of a basic session setup procedure in (V_2, O). Then, S-CSCF #1 becomes our concern. The signaling traffic traversing through S-CSCF #1 server in the originating network is taken into account, no matter which direction that the traffic travels from. As long as the traffic traverses the S-CSCF #1, it is considered as a new signaling flow, numbered as 7. In total, there are 12 message sequences found. The summary of signaling flow 7 is given in Table 4.

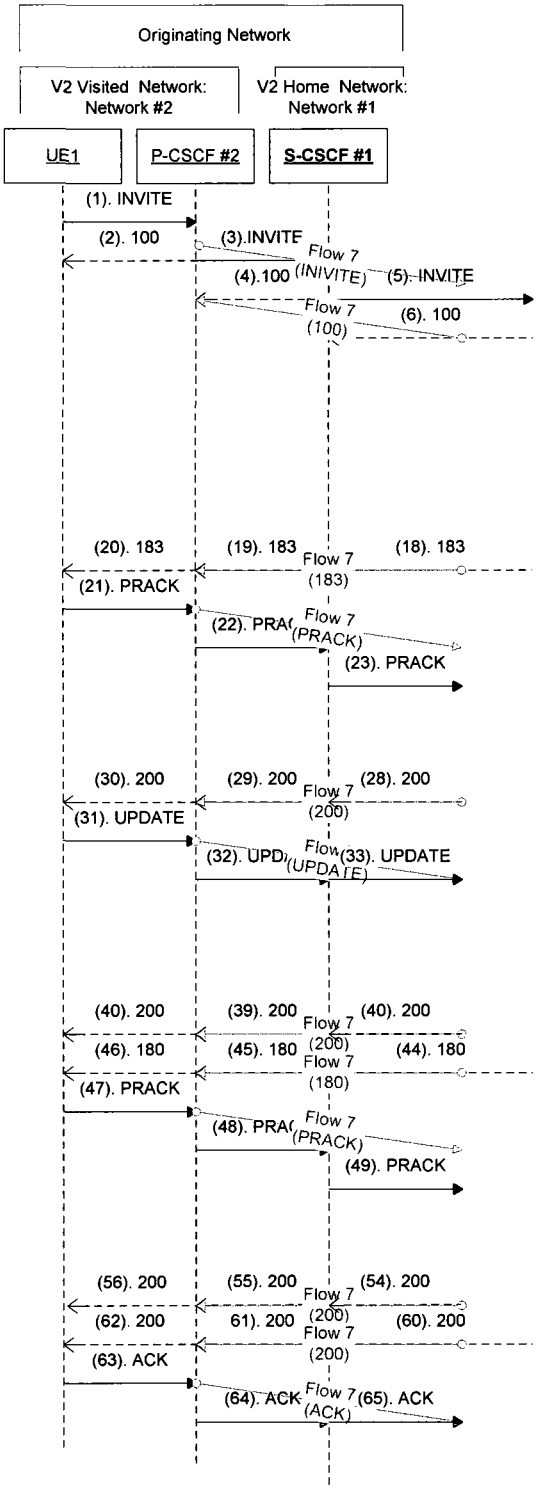


Figure 4: IMS basic session setup procedure with signaling flow analysis, in (V_2, O)

Table 4: Signaling flow summery for basic session setup, in (V_2, O)

Signaling Flows	Message Sequences	Signaling Flow Path	Signaling Flow Volume
7	INIVITE,100,183, PRACK, 200, UPDATE, 200, 180, PRACK, 200, 200, ACK	--- S1 ---	12

H.5 Basic session setup, (V_2, T)

The last case refers to the Network #1 user as callee/terminator roaming at Network #2. Figure 5 shows the basic session setup procedure in a routing scenario, (V_2, T). In this case, I -CSCF #1, HSS #1, and S-CSCF #1 are involved and found to be located in Network #1. The signaling traffic traversing through them is taken into account. The signaling flow analysis in this particular case is similar to the case of (H_1, T), except that P-CSCF #1 is not of our concern. The signaling flow analysis is provided in Figure 5. Four signaling flows are found, and the SIP messages are well qualified by them. Table 5 provides the information.

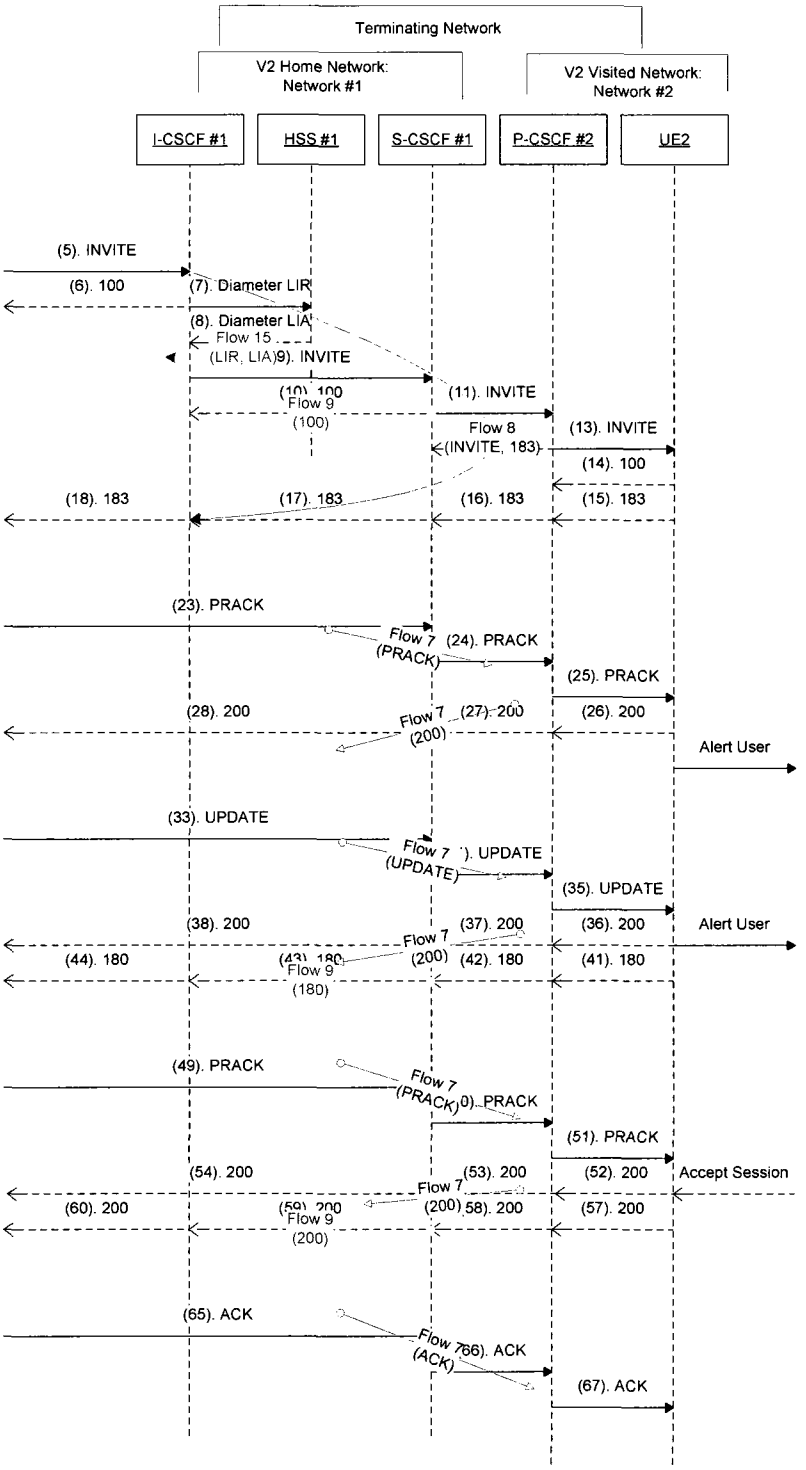


Figure 5: IMS basic session setup procedure with signaling flow analysis, in (V_2, T)

Table 5: Signaling flow summery for basic session setup, in (V_2, T)

Signaling Flows	Message Sequences	Signaling Flow Path	Signaling Flow Volume
7	PRACK,200, UPDATE,200, PRACK, 200, ACK	--- S1 ---	7
8	(INVITE, 183)	→I1 → S1 →---→ S1 → I1 →	2
9	100,180, 200	→ S1 → I1 →	3
15	(Diameter LIR, Diameter LIA)	→ HSS → I1	1

Appendix I Matrices, Volume of Signaling Flows per Session Procedure, X_r & Load Carried by Each Server, M_r

I.1 In (H1, O)

$$\mathbf{X}_1 = \begin{array}{c} \\ \text{Procedure}1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 18 \\ 19 \\ 20 \end{array} \begin{array}{c} \begin{array}{cccccccccccccccc} \textit{Flow}1 & 2 & 3 & 4 & 5 & \cdots & \cdots & \cdots & \cdots & 11 & \cdots & \cdots & 15 & 16 & 17 \end{array} \\ \begin{bmatrix} 8 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\mathbf{M}_1 = \begin{array}{c} \\ \text{Procedure}1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 18 \\ 19 \\ 20 \end{array} \begin{array}{c} \begin{array}{cccc} P & I & S & HSS \end{array} \\ \begin{bmatrix} 12 & 0 & 12 & 0 \\ 12 & 0 & 12 & 0 \\ 3 & 0 & 3 & 0 \\ 3 & 0 & 3 & 0 \\ 4 & 6 & 4 & 4 \\ 2 & 3 & 3 & 3 \\ 2 & 3 & 2 & 2 \\ 0 & 0 & 1 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 0 & 2 & 0 \\ 2 & 0 & 2 & 0 \\ 14 & 0 & 14 & 0 \\ 14 & 0 & 14 & 0 \\ 4 & 0 & 4 & 0 \\ 12 & 0 & 12 & 0 \\ 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

I.2 In (H1, T)

$X_2 =$

	*Flow*1	2	3	4	5	⋯	⋯	⋯	⋯	11	⋯	⋯	15	16	17		
Pr*ocedure*1	0	1	0	2	6	3	0	0	0	0	0	0	0	0	1	0	0
2	0	1	3	0	0	8	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	4	6	3	0	0	0	0	0	0	1	0	1	0	0
⋮	0	2	0	2	4	4	0	0	0	0	0	0	2	0	1	0	0
⋮	0	0	0	2	0	1	0	0	0	0	0	0	1	0	0	0	0
⋮	0	0	0	2	6	3	0	0	0	0	0	0	1	0	1	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	4	0	0	1	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$M_2 =$

	P	*I*	*S*	*HSS*
Pr*ocedure*1	12	6	12	1
2	12	0	12	0
3	3	0	3	0
4	3	0	3	0
5	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	2	0	2	0
⋮	2	0	2	0
⋮	14	9	14	1
⋮	14	9	14	1
⋮	4	4	4	0
⋮	12	7	12	1
⋮	0	5	2	1
18	0	0	0	0
19	2	0	2	0
20	2	0	2	0

I.3 In (V1, O)

$\mathbf{X}_3 =$

Procedure \ Flow	1	2	3	4	5	⋯	⋯	⋯	⋯	11	⋯	⋯	15	16	17
1	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	2	0	0	0	0	2	0	0	0	0	0
⋮	0	0	0	0	2	0	0	0	0	2	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$\mathbf{M}_3 =$

Procedure	*P*	*I*	*S*	*HSS*
1	12	0	0	0
2	12	0	0	0
3	3	0	0	0
4	3	0	0	0
5	4	0	0	0
⋮	2	0	0	0
⋮	2	0	0	0
⋮	0	0	0	0
⋮	2	0	0	0
⋮	2	0	0	0
⋮	2	0	0	0
⋮	2	0	0	0
⋮	14	0	0	0
⋮	14	0	0	0
⋮	4	0	0	0
⋮	12	0	0	0
⋮	4	0	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0

I.4 In (V1, T)

$\mathbf{X}_4 =$

Flow	1	2	3	4	5	…	…	…	…	11	…	…		15		16	17
Procedure 1	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

$\mathbf{M}_4 =$

	P	I	S	HSS
Procedure 1	12	0	0	0
2	12	0	0	0
3	3	0	0	0
4	3	0	0	0
5	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	2	0	0	0
⋮	2	0	0	0
⋮	14	0	0	0
⋮	14	0	0	0
⋮	4	0	0	0
⋮	12	0	0	0
⋮	0	0	0	0
18	0	0	0	0
19	1	0	0	0
20	1	0	0	0

I.5 In (V2, O)

$\mathbf{X}_5 =$

	*Flow*1	2	3	4	5	…	…	…	…	11	…	…	15	16	17
Pr*ocedure*1	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	4	0	0	2	2	0
⋮	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	2	0
⋮	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	1	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
⋮	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
⋮	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
⋮	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$\mathbf{M}_5 =$

	P	*I*	*S*	*HSS*
Pr*ocedure*1	0	0	12	0
2	0	0	12	0
3	0	0	3	0
4	0	0	3	0
5	0	6	4	4
⋮	0	3	3	3
⋮	0	3	2	2
⋮	0	0	1	1
⋮	0	0	2	1
⋮	0	0	2	1
⋮	0	0	2	0
⋮	0	0	2	0
⋮	0	0	14	0
⋮	0	0	14	0
⋮	0	0	4	0
⋮	0	0	12	0
⋮	0	0	4	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0

I.6 In (V2, T)

$\mathbf{X}_6 =$

	*Flow*1	2	3	4	5	⋯	⋯	⋯	⋯	11	⋯	⋯	15	16	17		
Pr*ocedure*1	0	0	0	0	0	0	7	2	3	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
⋮	0	0	0	0	0	0	6	4	3	0	0	0	0	1	1	0	0
⋮	0	0	0	0	0	0	6	2	4	0	0	0	0	2	1	0	0
⋮	0	0	0	0	0	0	0	2	1	0	0	0	0	1	0	0	0
⋮	0	0	0	0	0	0	6	2	3	0	0	0	0	1	1	0	0
⋮	0	0	0	0	0	0	0	0	0	0	0	4	0	0	1	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

$\mathbf{M}_6 =$

	P	*I*	*S*	*HSS*
Pr*ocedure*1	0	6	12	1
2	0	0	12	0
3	0	0	3	0
4	0	0	3	0
5	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	0	0
⋮	0	0	2	0
⋮	0	0	2	0
⋮	0	9	14	1
⋮	0	9	14	1
⋮	0	4	4	0
⋮	0	7	12	1
⋮	0	5	2	1
18	0	0	0	0
19	0	0	1	0
20	0	0	1	0

References

[1] G. Camarillo and M. A. Garcia-Martin, "*The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular worlds*," Second Edition, Wiley, 2005.

[2] J. C. Chen and T. Zhang, "*IP-based Next-Generation Wireless Network*," Wiley, 2004.

[3] G. T. "*Digital Cellular Telecommunications System (Phase 2+); Universal Mobile Telecommunications System (UMTS); IP Multimedia Subsystem (IMS); Stage 2*," version 7.5.0, ETSI TS123228, 2006.

[4] S. Pandey, V. Jain, D. Das, V. Planat and R. Periannan, "Performance Study of IMS Signaling Plane," *Proceeding of International Conference on IP Multimedia Subsystem Architecture and Applications*, pp.1-5, December 2007.

[5] V. Koukoulidis and M. Shah, "The IP Multimedia Domain in Wireless Networks: Concepts, Architecture, Protocols and Applications," *Proceeding of IEEE 6th International Symposium on Multimedia Software Engineering*, pp. 484-490, Miami, December 2004.

[6] M. Handley, H. Schulzrinne, E. Schooler and J. D. Rosenberg, "*SIP: Session Initiation Protocol*," IETF RFC 2543, March 1999.

[7] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnson, J. Peterson, R. Spar M. Handley and E. Schooler, "*The session initiation protocol (SIP)*," IETF RFC 3261, 2001.

[8] T. Borosa, B. Marsic and S. Pocuca, "QoS Support in IP Multimedia Subsystem Using DiffServ," *Proceeding of the 7th International Conference on Telecommunications*, Vol. 2, pp. 669-672, Zagreb, June 2003.

[9] "*IP Multimedia Subsystem (IMS), Stage 2*," 3GPP TS23.228 v7.6.0, 2006.

[10] J. Rosenberg, "*Requirements for management of Overload in the Session Initiation Protocol*," IETF RFC 5390, December 2008.

[11] *"IP Multimedia Subsystem Cx and Dx Interfaces; Signaling Flows and Message Contents,"* 3GPP TS29.228 v7.6.0, December 2006.

[12] N. Rajagopal and M. Devetsikiotis, "Modeling and Optimization for the Design of IMS Networks," *Proceeding of the 39th Annual Simulation Symposium*, pp. 7, Huntsville, April 2006.

[13] B. Ryu, "Modeling and Simulation of Broadband Satellite Networks— Part II: Traffic Modeling," *IEEE Communications Magazine,* Vol. 37, Issue 7, pp. 48-56, July 1999.

[14] *"Signaling flows for the IP Multimedia call control based on SIP and SDP; Stage 3 (Release 5),"* 3GPP TS 24.228 v5.13.0, June 2005.

[15] I. Vidal, I. Soto, F. Valera, J. Garcia and A. Azcorra, "IMS Signaling for Multiparty Services Based on Network Level Multicast," *Proceeding of the 3rd EuroNGI Conference on Next Generation Internet Networks*, pp. 103-110, May 2007.

[16] V. Koukoulidis and M. Shah, "The IP Multimedia Domain: Service Architecture for the Delivery of Voice, Data, and Next Generation Multimedia Applications," *Journal of Multimedia Tools and Applications*, Vol. 28, No. 2, pp. 203-220, Springer Netherlands, February 2006.

[17] M. Handley, H. Schulzrinne, E. Schooler and J. Rosenberg, *"SIP: Session Initiation Protocol,"* IETF RFC 2543, March 1999.

[18] B. Rong, Y. Qian and H. Chen, "An Enhanced SIP Proxy Server for Wireless VoIP in Wireless Mesh Networks," *IEEE Communications Magazine*, Vol. 46, Issue 1, pp. 108-113, January 2008.

[19] A. Cuevas, J. I. Moreno, P. Vidales and H. Einsiedler, "The IMS Service Platform: A Solution for Next-generation Network Operators to be More than Bit Pipes," *IEEE Communications Magazine*, Vol. 44, Issue 8, pp. 75-81, August 2006.

[20] H. Fathi, S. Chakraborty and R. Prasad, "Optimization of VoIP Session Setup Delay over Wireless Links Ssing SIP," *Technical Session of IEEE Globe Telecommunications Conference 2004*, WC42-6, Dallas, December 2004.

[21] P. Calhoun et al., *"Diameter base protocol,"* IETF RFC 3588, September 2003.

[22] *"IP Multimedia Subsystem Cx and Dx interface; Signaling Flows and Message Contents (Release 6),"* 3GPP TS 29.228 v6.9.0, December 2005.

[23] M. Koukal and R. Bestak, "Architecture of IP Multimedia Subsystem," *Proceeding of the 48th International Symposium ELMAR-2006 Focused on Multimedia Signal Processing and Communications*, pp. 323-326, Zadar, June 2006.

[24] P. Agrawal, Y. Jui-Hung, C. Jyh-Cheng and Z. Tao, "IP Multimedia Subsystems in 3GPP and 3GPP2: Overview and Scalability Issues," *IEEE Communications Magazine*, Vol. 46, Issue 1, pp. 138-145, January 2008.

[25] U. Olsson, "Toward the all-IP Vision," *Ericsson Review*, Vol. 82, Issue 1, pp. 44-53, 2005.

[26] T. Raty, J. Sankala and M. Sihvonen, "Network Traffic Analyzing and Monitoring Locations in the IP Multimedia Subsystem," *Proceeding of the 31st EUROMICRO Conference on Software Engineering and Advanced Applications*, pp. 362-369, Porto, September 2005.

[27] T. Russel, "*The IP Multimedia Subsystem (IMS): Session Control and Other Network Operations*," First Edition, McGraw-Hill, 2008.

[28] A. Munir, "Analysis of SIP-Based IMS Session Establishment Signaling for WiMax-3G Networks," *Proceeding of the 4th International Conference on Networking and Services*, pp. 282-287, Gosier, March 2008.

[29] A. A Kist, E. A. Kist and R. J. Harris, "SIP Signaling Delay in 3GPP," *Proceeding of the 6th International Symposium on Communications Interworking of IFIP interworking 2002*, pp. 13-16, Fremantle, October 2002.

[30] I. D. D. Curcio and M. Lundan, "SIP Call Setup Delay in 3G Networks," *Proceeding of the 7th International Symposium on Computers and Communications*, pp. 835, Taormina, July 2002.

[31] Eyers and H. Schulzrinne, "Predicting Internet Telephony Call Setup Delay," *Proceeding of the 1st IP Telephony Workshop*, pp.107-126, Berlin, April 2000.

[32] V. K. Gurbani, L. Jagadeesan and V. B. Mendiratta, "Characterizing Session Initiation Protocol (SIP) Network Performance and Reliability," *Proceeding of the 2nd International Service Availability Symposium*, pp. 196-211, Berlin, April 2005.

[33] M. Cortes, J. R. Ensor and J. O. Esteban, "On SIP Performance," *Bell Labs Technique*, pp.155-172, 2004.

[34] Y. Fei, J. Liao, Q. Qi and X. Zhu, "A Cache Based Session Setup Mechanism for IMS," *Workshops of IEEE International Conference on Communications 2008*, pp.261-265, Beijing, May 2008.

[35] J. Hwang, N. Kim, S. Kang and J. Koh, "A Framework for IMS Interworking Networks with Quality of Service Guarantee," *Proceeding of the 7th International Conference on Networking*, pp. 454-459, Cancun, April 2008.

[36] M. Hammer and W. Franx, "Redundancy and Scalability in IMS," *Proceeding of the 12th International Telecommunication Network Strategy and Planning Symposium*, pp. 1-6, New Delhi, November 2006.

[37] V.S. Abhayawardhana and R. Babbage, "A Traffic Model for the IP Multimedia Subsystem (IMS)," *Proceeding of the 65th IEEE Vehicular Technology Conference*, pp.783-787, Dublin, April 2007.

[38] U. V. Carlos, M. Amit and E. S. Mohanmed, "Presence and Availability with IMS: Application Architecture, Traffic Analysis, and Capacity Impacts," *Bell Labs Technical Journal*, Vol. 10, No. 4, pp. 101-107, 2006.

[39] M. Day, J. Rosenberg and H. Sugano, "*A Model for Presence and Instant Messaging*," IETF RFC 2778, 2000.

[40] M. Pous, D. Pesch, G. Foster and A. Sesmun, "Performance Evaluation of a SIP Based Presence and Instant Messaging Service for UMTS," *Proceeding of the 4th International Conference on 3G Mobile Communication Technologies*, pp. 254-258, London, June 2003.

[41] B. Panwar and K. Singh, "IMS SIP Core Server Test Bed," *Proceeding of International Conference on IP Multimedia Subsystem Architecture and Applications*, pp.1-5, Bangalore, December 2007.

[42] M. Cortes, J. O. Esteban and H. Jun, "Diabelli: An IMS Simulation Tool," *Bell Labs Technical Journal*, Vol. 10, No. 4, pp. 255-259, 2006.

[43] A. Niemi, "*Session Initiation Protocol (SIP) Extension for Event State Publication*," IETF RFC 3903, October 2004.

[44] A. B. Roach, "*Session Initiation Protocol (SIP) - Specific Event Notification*," IETF RFC 3265, June 2002.

[45] B. Kim et al., "Capacity Estimation and TCP Performance Enhancement over Mobile WiMAX Networks," *IEEE Communications Magazine,* Vol. 47, Issue 6, pp. 133-141, June 2009.

[46] L. Kleinrock, "*Queuing Systems, Vol. 1, Theory*," Wiley, 1975.

[47] D. Vingarzan and P. Weik, "End-to-end Performance of the IP Multimedia Subsystem over Various Wireless Networks," *Proceeding of IEEE Wireless Communications and Networking Conference 2006*, Vol. 1, pp.183-188, Las Vegas, April 2006.

[48] J. Rosenberg and H.Schulzrinne, "*Session Initiation Protocol (SIP): Locating SIP Servers*," IETF RFC 3263, June 2002.

[49] A. Vaha-Sipila, "*URLs for Telephone Calls*," IETF RFC 2806, April 2000.