國立交通大學

資訊科學與工程研究所

博士論文

第三代行動通訊系統之通話控制與身份認證

Design of Call Control and Authentication for UMTS

研究生： 吳怜儀

指導教授： 林一平　博士

中華民國九十五年七月

第三代行動通訊系統之通話控制與身份認證

Design of Call Control and Authentication for UMTS

研 究 生：吳怜儀　　Student：Lin-Yi Wu

指導教授：林一平 博士　　Advisor：Dr. Yi-Bing Lin

國立交通大學

資訊科學與工程研究所

博士論文

A Dissertation
Submitted to
Department of Computer Science
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Computer Science

July 2006
Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

第三代行動通訊系統之通話控制與身份認證

學生：吳怜儀　　　　　　　　指導教授：林一平博士

國立交通大學　　資訊工程研究所

摘要

Universal Mobile Telecommunication System (UMTS) 為第三代行動通訊標準的主流之一，該系統整合了數據服務與多媒體服務，並且具有高服務容量、高速傳輸、服務品質保證、客製化服務等特性。UMTS 網路可以分為四層，包括無線傳輸層、UMTS 核心網路、多媒體子系統、與應用服務層。在本論文中，我們分別就各層討論設計的議題。在無線傳輸層中，無線網路與 UMTS 的整合將 UMTS 的服務擴及到無線網路的涵蓋範圍中，因此使用者可以透過無線網路存取服務，獲得較高的服務品質。在本論文，我們提出了一個整合的架構稱之為 *WLAN-based GPRS Support Node* (WGSN)。WGSN 符合 3GPP 標準文件 TS 22.934 中第三階段的整合特徵。WGSN 為了省電，手機上的網路電話程式可能會被關閉，因而打到該手機的電話便無法接通。為了解決這樣的問題，我們提出了 push mechanism 以簡訊的方法開啟受話端手機上的程式，以接通電話。

在 UMTS 網路中，*Authentication Vector* (AV) usage mechanism 可以減低 SGSN 與 HSS/AuC 之間的訊號交換，然而該機制也會造成 SGSN 所需的記憶體增多。本論文利用數學分析以及電腦模擬的方法來研究 AV usage mechanism 的效能。我們的研究結果可以提供行動通訊業者用以設定 AV usage mechanism 的相關參數。

針對使用者的身分認證問題，3GPP 制定了 two-pass authentication 程序，分別在 General Packet Radio Service (GPRS)網路以及多媒體子系統認證使用者。我們發現在 two-pass authentication 中，有許多步驟是重複的，因此在論文中我們提出了 one-pass authentication 的方法。在此方法中，GPRS 網路執行相同的認證程序，但是多媒體子系統以簡化的方法在使用者註冊的過程完成認證。我們證明，one-pass authentication 可以正確的認證使用者，同時省下了 50%的訊息交換。

UMTS 的應用服務層可以 *Open Service Access* (OSA)做為服務開發的平台。在這個平台之上，我們實作了 *Push to Talk over Cellular* (PoC)服務。我們描述了使用者端程式的設計架構，也詳盡說明了服務運作的流程。

以上的研究成果提供讀者在研究 UMTS 通話控制以及身分認證的議題上，可供參考之基礎。

關鍵字：第三代行動通訊、Universal Mobile Telecommunications System (UMTS)、General Packet Radio Service (GPRS)、多媒體子系統、無線網路、Push to Talk over Cellular (PoC), 身分認證、*Session Initiation Protocol* (SIP)、通話控制。

Design of Call Control and Authentication for UMTS

Student: Lin-Yi Wu Advisor: Dr. Yi-Bing Lin

Department of Computer Science and Information Engineering,
Nation Chiao Tung University

ABSTRACT

Universal Mobile Telecommunication System (UMTS) is an integrated solution for multimedia and data services with wide area coverage. UMTS is developed towards large system capacity, high data transmission, and customized services with quality of services. The UMTS all-IP architecture can be horizontally partitioned into four layers: radio network layer, UMTS core network, *IP Multimedia Subsystem* (IMS), application layer. In this dissertation, we discuss design issues of each layer. In the radio network layer, UMTS and WLAN interworking extends the UMTS services to the WLAN coverage, and the UMTS subscribers can acquire services with better quality through WLAN. We propose *WLAN-based GPRS Support Node* (WGSN), which is a loosely-coupled architecture satisfying Scenario 3 features in 3GPP TS 22.934. A push mechanism is implemented in WGSN to connect the MS-terminated call where the *Voice over IP (VoIP)* client in the callee is not activated.

In the UMTS authentication, the *Authentication Vector* (AV) usage mechanism is used to reduce the signaling traffic between the SGSN and the HSS/AuC. However the AV usage mechanism also consumes extra storage at the SGSN. Therefore, we propose analytic and simulation models to investigate the performance of the AV usage mechanism.

In UMTS two-pass authentication, many steps in the GPRS authentication and IMS authentication are duplicated. Therefore, we propose an one-pass authentication procedure, in which only the GPRS authentication procedure is performed. In the IMS network, the authentication is implicitly executed in the IMS registration. We formally prove that the IMS user is correctly authenticated, and the one-pass authentication saves up to 50% of the IMS registration/authentication traffic.

In the application layer, we implement the *Push to Talk over Cellular* (PoC) on the *Open Service Access* (OSA) platform. We focus on the design and implementation of the PoC client. The detailed architecture and message flows are described.

These research results presented in this dissertation can be viewed as a useful foundation for further study in UMTS call control and authentication.

Key Words: Third Generation (3G), Universal Mobile Telecommunications System (UMTS), General Packet Radio Service (GPRS), IP Multimedia Subsystem (IMS), Call Session Control Function (CSCF), *Wireless LAN* (WLAN), Push to Talk over Cellular (PoC), floor control, cellular network, authentication, security function, *Session Initiation Protocol* (SIP).

Acknowledgement

I would like to express my deep and sincere gratitude to my advisor Prof. Yi-Bing Lin for his continuous support, encouragement, and guidance throughout my graduate study. His extensive knowledge and creative thinking have been an invaluable help for me. Without his perspicacious advice, I can not complete this dissertation. Also, I would like to deliver a special gratitude to Prof. Ming-Feng Chang for his stimulating suggestions and guidance helped me to accomplish several researches. Special thank to my committee members, Prof. Ming-Feng Chang, Prof. Han-Chieh Chao, Dr. Sheng-Lin Chou, Dr. Herman Chunng-Hwa Rao, Prof. Wen-Nung Tsai, Prof. Chu-Sing Yang for their valuable comments and suggestations. Moerover, I am grateful to the colleagues in the Laboratory 117 for their friendship and many helpful discussions.

Especially, I am indebted to my dear father, mother, sister, amd brother. This dissertation is dedicated to them. It is their love, dedication, and encouragement that made everything I have possible. Last but not least, I would like to express my deepest love and gratitude to my boyfriend and soul mate, Fu-Yuan, for his company over the past ten years. His patient love, warm support, and sharing of successful academic experience help me to overcome all kinds of difficulties and challenges. Also thank my little pet Dolly for his absolute trust and selfless company that make my life full of joy and happiness.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Universal Mobile Telecommunication System (UMTS) [37] is a *Third-Generation* (3G) mobile telecommunication system evolved from *Global System for Mobile Communication* (GSM) and *General Packet Radio Service* (GPRS) [5]. UMTS provides an integrated solution for multimedia and data services with wide area coverage. Comparing to GSM and GPRS, UMTS is developed towards larger system capacity, higher data transmission (384 kbps for high mobility situations, and 2Mbps for stationary user environments), and customized services with quality of services. Currently, the UMTS service is commercially available in 47 countries supported by more than 100 mobile operators.

The UMTS all-IP network can be horizontally partitioned into four layers as illustrated in Figure 1-1.

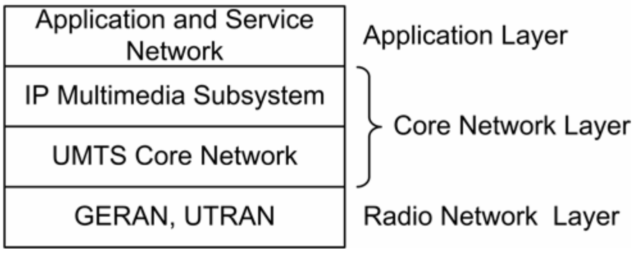


Figure 1-1. Horizontal Structure of UMTS All-IP Network

Radio Network Layer provides wireless access for the subscribers to obtain the UMTS services. Two radio networks are defined in the *Third Generation Partnership Project* (3GPP) specifications: *GSM Enhanced Data Rates for Global Evolution (EDGE) Radio Access Network* (GERAN) and *UMTS Terrestrial Radio Access Network* (UTRAN).

UMTS Core Network provides switching, routing, transit for data and multimedia services. The UMTS core network is also responsible for mobility management and session management. The core network is divided in *circuit switch* (CS) and *packet switch* (PS) domains. In the PS domain, IP technology is utilized as the transport protocol.

***IP Multimedia Subsystem* (IMS)** is responsible to deliver signaling and voice data for IP multimedia services. The IMS is connected to the UMTS core network, IP multimedia network, and *Public Switch Telephone Network* (PSTN). The IMS routes the signaling between different networks and maps the *Session Initiation Protocol* (SIP) signaling used in *Voice over IP* (VoIP) service form/to SS7 signaling used in PSTN.

Application Layer supports flexible services through a common service platform. The third parties can implement the application servers following the *Application Programming Interfaces* (APIs) provided by the mobile operators, and run the application server on the common service platform.

Many research studies have been conducted to investigate various aspects of the UMTS design in each of the layers. In the radio network layer, integration of UMTS with *Wireless LAN* (WLAN) has been intensively studies. UMTS and WLAN interworking extends the 3G services and functionality to the WLAN environment. Since WLAN provides higher data transmission rate and bandwidth, the subscribers can acquire 3G service with better service quality through WLAN environment.

Authentication is an important issue in the core network layer. Before a subscriber accesses the UMTS services or IMS services, authentication function is performed to identify and authenticate the subscriber, and then validates the service request type to ensure that the subscriber is authorized to access the service. The 3GPP specifications define mutual authentication mechanisms in both the UMTS core network and the IMS. In this dissertation, we will investigate the performance of authentication function, and provide the suggestions to setup the related configuration.

In the application layer, the design on a flexible common service platform speeds up the development of various 3G service. *Open Service Access* (OSA) is a common service platform specified in the 3GPP specification. OSA provides a service creation and execution environment that is independent of the underlying communication technologies.

In the following sections, we first briefly present the UMTS all-IP architecture, and then elaborate more on the UMTS design issues.

1.1 UMTS All-IP Architecture

Figure 1-2 illustrates the UMTS all-IP network architecture. The UMTS network includes five parts: *Mobile Station* (MS; Figure 1-2 (1)) is the equipment through which a user accesses UMTS services. Core Network (Figure 1-2 (3)) provides mobility management, session management and transport for IP-based services. *UMTS Terrestrial Radio Access Network* (UTRAN; Figure 1-2 (2)) provides wireless connectivity between the MS and the core network. *IP Multimedia Subsystem* (IMS; Figure 1-2 (4)) supports multimedia services such as voice telephony, video, real-time interactive games, messaging, and multimedia conferencing [7]. Application and Service Network (Figure 1-2 (5)) supports flexible services through a common service platform.

UTRAN consists of *Radio Network Controllers* (RNCs; Figure 1-2 (10)) and Node Bs (Figure 1-2 (9)). In the core network, *Serving GPRS Support Node* (SGSN; Figure 1-2 (11)) and *Gateway GPRS Support Node* (GGSN; Figure 1-2 (12)) provide mobility and session services to mobile users. One SGSN connects to several RNCs, and one RNC connects to one or more Node Bs. The coverage of the Node Bs connected to the same SGSN is called an SGSN area. In Figure 1-2, SGSN area 1 (Figure 1-2 (13)) corresponds to SGSN1, and SGSN area 2 (Figure 1-2 (14)) corresponds to SGSN2. The GGSN connects to the external *Packet Data Network* (PDN; Figure 1-2 (6)) by an IP-based interface. The MS obtains the

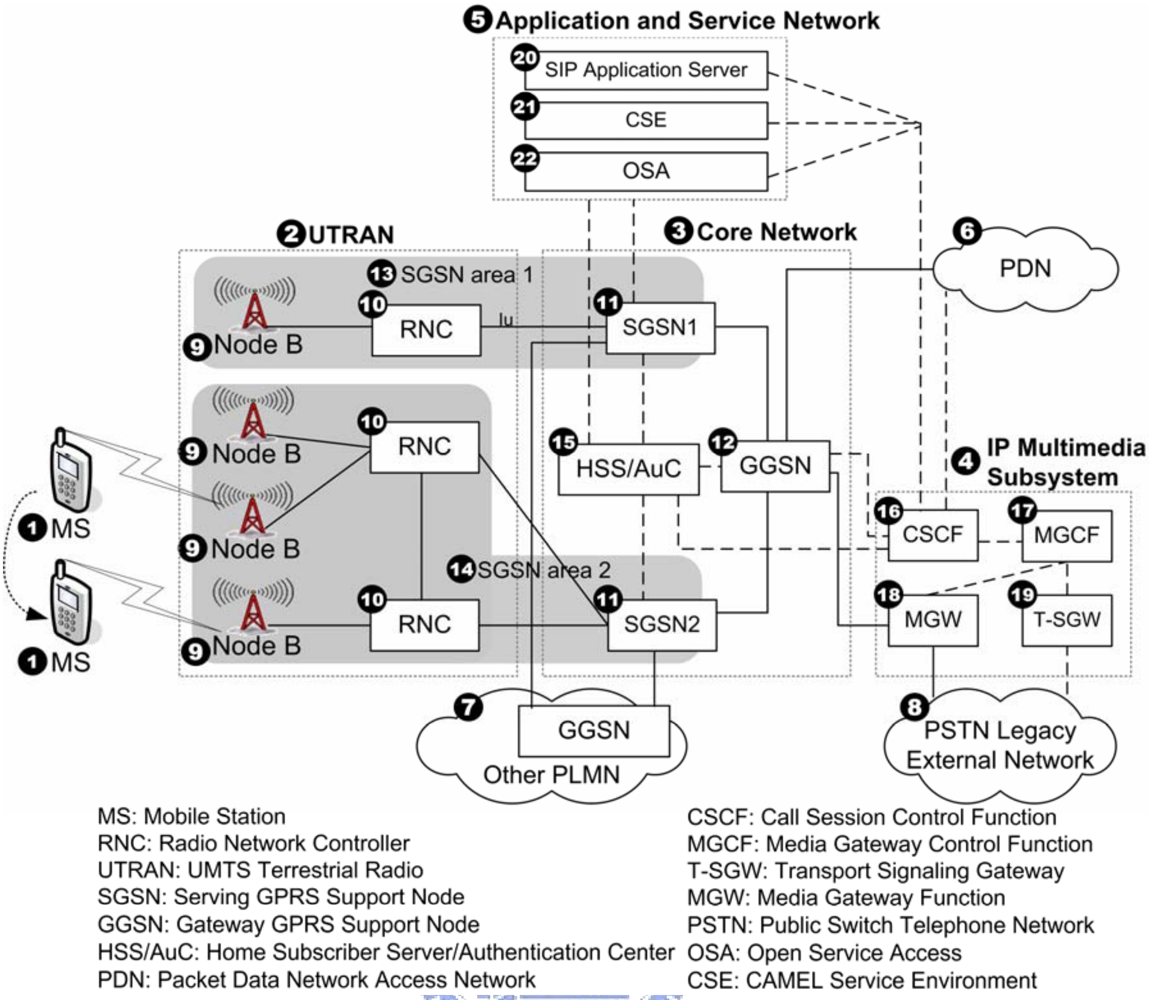

5 Application and Service Network
20 SIP Application Server
21 CSE
22 OSA
2 UTRAN
3 Core Network
6 PDN
13 SGSN area 1
10 RNC
Iu
11 SGSN1
9 Node B
15 HSS/AuC
12 GGSN
4 IP Multimedia Subsystem
1 MS
16 CSCF
17 MGCF
14 SGSN area 2
11 SGSN2
18 MGW
19 T-SGW
7 GGSN
Other PLMN
8 PSTN Legacy External Network
MS: Mobile Station
RNC: Radio Network Controller
UTRAN: UMTS Terrestrial Radio
SGSN: Serving GPRS Support Node
GGSN: Gateway GPRS Support Node
HSS/AuC: Home Subscriber Server/Authentication Center
PDN: Packet Data Network Access Network
CSCF: Call Session Control Function
MGCF: Media Gateway Control Function
T-SGW: Transport Signaling Gateway
MGW: Media Gateway Function
PSTN: Public Switch Telephone Network
OSA: Open Service Access
CSE: CAMEL Service Environment

Figure 1-2. UMTS All-IP Architecture

PS services by connecting to the SGSN via UTRAN, and then accesses the external data network through the GGSN. Both SGSN and GGSN communicate with the *Home Subscriber Server/Authentication Center* (HSS/AuC; Figure 1-2 (15)) through *Mobile Application Part* (MAP) [21] to retrieve subscriber data and authentication information of an MS. The HSS/AuC is the master database containing all user-related subscription and location information.

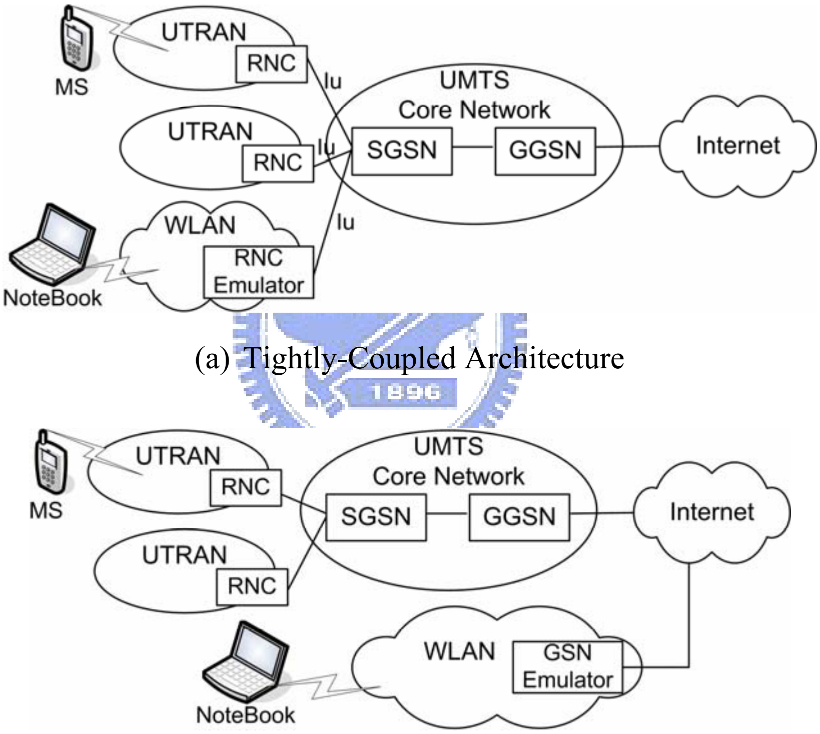
The IMS, which provides the SIP-based multimedia service, is located between the GGSN and the PDN. In IMS, the *Call Session Control Function* (CSCF; Figure 1-2 (16)) is the SIP server handling the call setup procedures. If the calling party is in the PDN, the CSCF transport the signaling to VoIP call control server or a terminal in the PDN. After the call is connected, the voice data is transferred from the GGSN to the PDN directly. If the calling party resides in the *Public Switch Telephone Network* (PSTN; Figure 1-2 (8)), the signaling is transferred from the GGNS to the *Media Gateway Control Function* (MGCF; Figure 1-2 (17)), and then is forwarded to the *Transport Signaling Gateway* (T-SGW; Figure 1-2 (19)). The T-SGW maps the SIP signaling to the SS7 signaling, and transfers the SS7 signaling to the PSTN. If the signaling requests to setup, modify, or disconnect the media channels, the MGCF controls the *Media Gateway Function* (MGW Figure 1-2 (18)) to provide the voice data transportation between the UMTS and the PSTN. While a call is successfully setup, the voice data is transferred from GGSN to MGW, and then is delivered to the PSTN.

In order to provide flexible and global services, the 3GPP defines three possible alternatives to construct the Application and Service Network: SIP application server (Figure 1-2 (20)) is either developed by the mobile operator or purchased from the trusted third parties. *Customized Application Mobile Enhanced Logic* (CAMEL) *Service Environment* (CSE; Figure 1-2 (21)), which has already been built in the UMTS CS domain, is reused by the mobile operator to provide CAMEL services to IMS user. *Open Service Access* (OSA; Figure 1-2 (22)) is constructed by the mobile operator, and provides the third parties a platform to run their own applications without concerning the underlying network environment

1.2 UMTS and WLAN Interworking

Two approaches for interconnection of UMTS and WLAN are proposed in HiperLan/2, which is developed by the *European Telecommunication Standard Institute* (ETSI) *Broadband Radio Access Network* (BRAN) project [20]. The first approach is a tightly-coupled architecture shown in Figure 1-3 (a). In this architecture, the WLAN is treated as another radio access network of the UMTS. The RNC emulator plays as the gateway in the WLAN, which hides the details of the WLAN network to the UMTS network. The RNC emulator is responsible for all the functionalities required in a UMTS radio access network, and provides the Iu interface connected to the SGSN. The WLAN uses the same authentication, signaling, transport, and billing infrastructures as those in UMTS network.

Figure 1-3 (b) illustrates the loosely-coupled architecture, in which the WLAN is an independent and parallel network to the UMTS network. In the WLAN, the GSN emulator integrates both SGSN and GGSN functionalities, and connects to the UMTS core network through Internet. The interconnection between the WLAN and the UMTS network relies on IP-based protocols. For example, the *Authentication, Authorization, and Accounting* (AAA) mechanism [19] is used to perform the authentication and accounting procedure, and Mobile IP [44] is used for managing the mobility and roaming between the UMTS access network and WLAN.



(a) Tightly-Coupled Architecture

(b) Loosely-Coupled Architecture

Figure 1-3. Two Approaches for Interconnection of UMTS and WLAN

3GPP Technical Report 22.934 [3] conducts a feasibility study on *Third Generation* (3G) system and *Wireless LAN* (WLAN) interworking that extends 3G services to the WLAN environment. In this interworking, WLAN serves as an access technology to the 3G system, which scales up the coverage of 3G services. Six scenarios were proposed for incremental development of 3G and WLAN interworking. Each scenario enhances interworking functionalities over the previous scenarios as illustrated in Table 1-1. The service and operational capabilities of each scenario are described as follows.

Table 1-1. Interworking Scenarios and Service Capabilities

Service Capabilities	Scenario1	Scenario2	Scenario3	Scenario4	Scenario5	Scenario6
Common Billing	V	V	V	V	V	V
Common Customer Care	V	V	V	V	V	V
3G-based Access Control		V	V	V	V	V
3G-based Access Charging		V	V	V	V	V
Access to 3G PS Services			V	V	V	V
Service Continuity				V	V	V
Seamless Service Continuity					V	V
Access to 3G CS Services with Seamless Mobility						V

Scenario 1 provides common billing and customer care for both WLAN and 3G mobile operators. That is, a customer receives single monthly billing statements combining both 3G and WLAN services. The customer also consults the same customer care center about the problems for both services.

Scenario 2 reuses 3G-access control and charging mechanisms for WLAN services. The WLAN customers are authenticated by the 3G core network without introducing a separate procedure. In addition, the roaming mechanism between 3G system and WLAN is supported. In this scenario, users can access traditional Internet services but cannot access 3G services (such as *Circuit-Switched* (CS) voice and GPRS data services) through WLAN.

Scenario 3 allows a customer to access 3G *Packet-Switched* (PS) services over WLAN. The PS services include *Short Message Service* (SMS) [4], *Multimedia Message Service* (MMS) [1], and *IP Multimedia Subsystem Service* (IMS) [2]. Customers equipped with both WLAN card and 3G module can simultaneously but independently access WLAN and 3G networks.

Scenario 4 allows a customer to change access between 3G and WLAN networks during a service session. The system is responsible for re-establishing the session without user involvement. Service interruption during system switching is allowed in this scenario.

Scenario 5 provides seamless service switching (i.e., handover) between 3G system and WLAN. Techniques must be developed to minimize data lost rate and delay time during switching so that the customer would not experience significant interruption during handover.

Scenario 6 supports 3G CS services in the WLAN environment. The seamless continuity feature described in Scenario 5 is also required to support CS services when customers roam between different networks.

1.3 Two-Pass Authentication

This section describes the 3GPP two-pass authentication procedure. We first describe the GPRS authentication, and then we elaborate more on the IMS authentication.

When an MS invokes the GPRS access (e.g., turns on its power), the MS sends an attach request to the SGSN. This message will trigger the GPRS authentication [5], which is implemented by *GPRS Mobility Management* (GMM) between the MS and the SGSN, and *Signaling System Number 7* (SS7) *Mobile Application Part* (MAP) between the SGSN and the HSS/AuC [35]. This procedure consists of the following steps (see Figure 1-4).

Step G.1. Consider an MS with the IMSI value *imsi* and the IMPI value *impi*. To access the GPRS services, the MS sends a GMM Attach Request (with the parameter IMSI = *imsi*) to the SGSN.

Step G.2. If the SGSN has the AVs of the MS, then Steps G.2 and G.3 are skipped. Otherwise, the SGSN must obtain the AV's from the HSS/AuC. That is, the SGSN invokes the authentication vector distribution procedure by sending a MAP_SEND_AUTHENTICATION_INFO Request message to the HSS/AuC (with

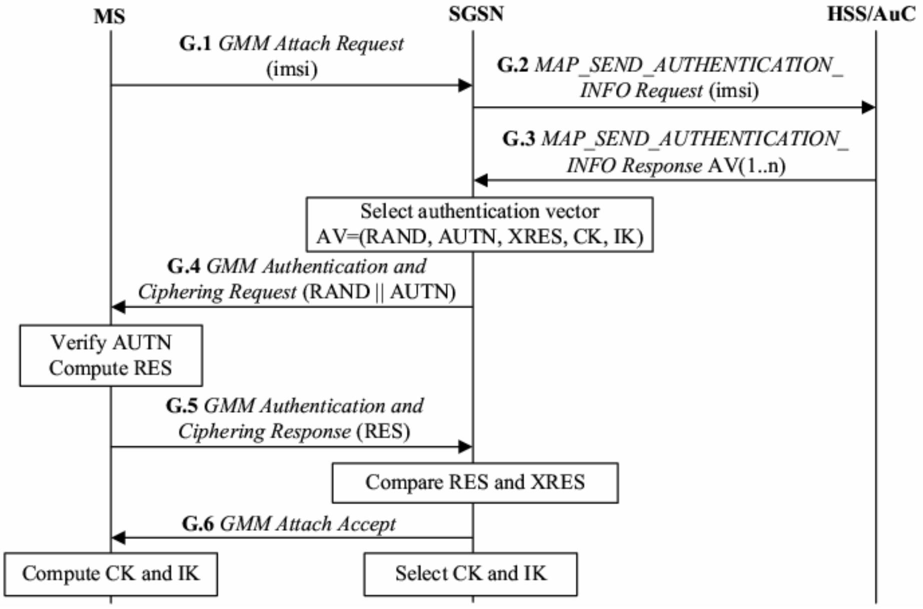

MS
SGSN
HSS/AuC
G.1 GMM Attach Request (imsi)
G.2 MAP_SEND_AUTHENTICATION_INFO Request (imsi)
G.3 MAP_SEND_AUTHENTICATION_INFO Response AV(1..n)
Select authentication vector AV=(RAND, AUTN, XRES, CK, IK)
G.4 GMM Authentication and Ciphering Request (RAND || AUTN)
Verify AUTN Compute RES
G.5 GMM Authentication and Ciphering Response (RES)
Compare RES and XRES
G.6 GMM Attach Accept
Compute CK and IK
Select CK and IK

Figure 1-4. Message Flow for 3GPP GPRS Authentication

the parameter IMSI = *imsi*).

Step G.3. The HSS/AuC uses *imsi* to retrieve the record of the MS, and generates an ordered array of AVs (based on the preshared secret key **K** in the MS record). The generated AV array is sent to the SGSN through a MAP_SEND_AUTHENTICATION_INFO Response message.

Step G.4. The SGSN selects the next unused authentication vector in the ordered AV array and sends the parameters **RAND** and **AUTN** (from the selected authentication vector) to the MS through a GMM Authentication and Ciphering Request message.

Step G.5. The MS checks whether the received **AUTN** can be accepted. If so, it produces a response **RES** that is sent back to the SGSN through a GMM Authentication and Ciphering Response message. The SGSN compares the received **RES** with the **XRES**. If they match, then the authentication and key agreement exchange is successfully completed.

Step G.6. The SGSN sends a GMM Attach Accept message to the MS, and the attach procedure is completed.

After GPRS authentication, GPRS registration follows (details of GPRS registration can be found in [36]). Then, the MS performs *Packet Data Protocol* (PDP) context activation to obtain access to the GPRS network. The PDP context specifies the application-layer packet data protocol and the routing information used for the GPRS communication session (see [37] for the details).

After PDP context activation, the MS can request the IMS services through the registration procedure illustrated in Figure 1-5. In this procedure, the MS interacts with the S-CSCF possibly through P-CSCF and I-CSCF. To simplify our discussion, Figure 1-5 uses the term "CSCF" to represent the proxy, interrogating, and service functions of CSCF. Details of message exchanges among these CSCFs are given in [37]. IMS authentication/registration is implemented by SIP and Cx protocols [8][9][11], which consists of the following steps.

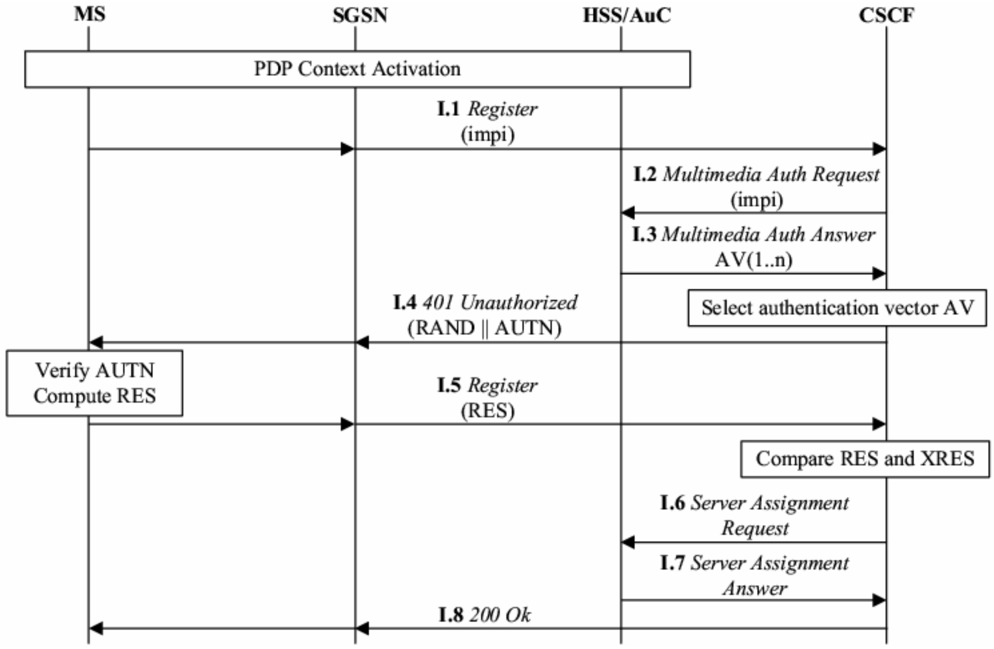

MS
SGSN
HSS/AuC
CSCF
PDP Context Activation
I.1 Register (impi)
I.2 Multimedia Auth Request (impi)
I.3 Multimedia Auth Answer AV(1..n)
Select authentication vector AV
I.4 401 Unauthorized (RAND || AUTN)
Verify AUTN Compute RES
I.5 Register (RES)
Compare RES and XRES
I.6 Server Assignment Request
I.7 Server Assignment Answer
I.8 200 Ok

Figure 1-5. Message Flow for 3GPP IMS Authentication

Step I.1. The MS sends a SIP Register message to the CSCF (with the parameter IMPI = *impi*) through the SGSN.

Step I.2. Assume that the CSCF does not have the AVs for the MS. The CSCF invokes the authentication vector distribution procedure by sending a Cx Multimedia Authentication Request message to the HSS/AuC (with the parameter IMPI = *impi*).

Step I.3. The HSS/AuC uses *impi* to retrieve the record of the MS, and generate an ordered array of AVs. The HSS/AuC sends the AV array to the CSCF through a Cx Multimedia Authentication Answer message.

Step I.4. The CSCF selects the next unused authentication vector from the ordered AV array and sends the parameters **RAND** and **AUTN** (from the selected authentication vector) to the MS through a SIP 401 Unauthorized message.

Step I.5. The MS checks whether the received **AUTN** can be accepted. If so, it produces a response **RES**. The MS sends this response back to the CSCF through a SIP Register message. The CSCF compares the received **RES** with the **XRES**. If they match, then the authentication and key agreement exchange is successfully completed.

Step I.6. The CSCF sends a Cx Server Assignment Request message to the HSS/AuC.

Step I.7. Upon receipt of the Server Assignment Request, the HSS/AuC stores the CSCF name and replies a Cx Server Assignment Answer message to the CSCF.

Step I.8. The CSCF sends a 200 ok message to the MS through the SGSN, and the IMS registration procedure is completed.

In the above procedure, Steps I.1–I.5 exercise authentication, and Steps I.6–I.8 perform registration.

1.4 Open Service Access

3GPP Technical Report 23.127 [6] specifies the OSA architecture deployed in the UMTS application and service network, which is illustrated in Figure 1-6. The OSA consists of three parts:

- Applications (Figure 1-6 (1)) implemented in one or more application servers.
- *Service Capability Servers* (SCSs; Figure 1-6 (2)) provide the applications the access

to the undering UMTS network functionalities. The network functionalities offered by the SCSs (e.g. call control, user location, etc.) are defined as a set of Service Capability Features (SCFs; Figure 1-6 (4)) in the OSA API (Figure 1-6 (5)).

- *Framework* (FW; Figure 1-6 (3)) authorizes the request of accessing the SCFs from the applications.

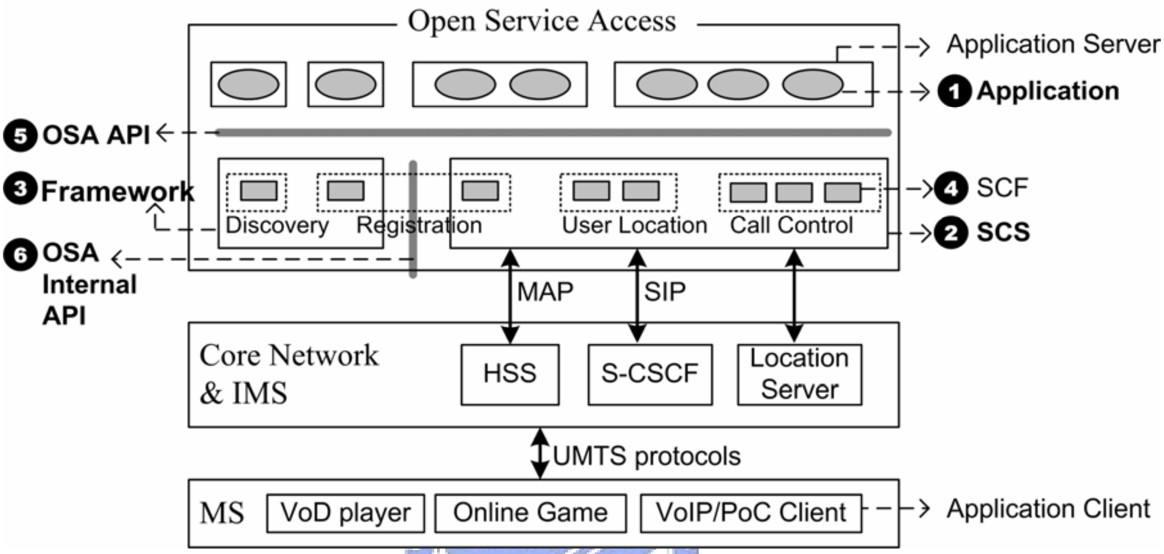


Figure 1-6. Open Service Access Platform

The SCSs are implemented all kinds of interfaces to communicate with the entities in the IMS and UMTS core networks. For example, the SCSs communicate with the HSS through MAP, and also play as a SIP server to connect to the S-CSCF through SIP protocol. Before a SCS provides services, the SCFs offered by the SCS have to be registered at the FW through OSA internal API (Figure 1-6 (6)). The FW is considered as one of the SCSs, and exactly one FW exists in an OSA environment. Before an application accesses the OSA API, the FW authenticates the application and determines if the application is authorized to access the certain SCFs. After successful authentication, the application obtains the information of the authorized SCFs through the discovery function provided by FW. Then the application accesses the network functionalities through SCFs.

Following the OSA APIs, the third party can implement their own applications on the

UMTS network, and the applications can also run on other networks which provide the same OSA APIs. If the mobile operator wants to release the restriction of operating system and programming language on implementing applications, the object-oriented techniques, such as CORBA, SOAP, etc., can be used to build the OSA environment.

1.5 Design and Performance Issues

In this dissertation, we will discuss the four design issues in each of the UMTS protocol layers. In the radio network layer, we propose an UMTS and WLAN interworking solution called *WLAN-based GPRS Support Node* (WGSN). WGSN is a loosely-coupled architecture that satisfies the Scenario 3 features in the 3GPP TS 22.934. Our survey with several mobile service providers indicates that the Scenario 3 features are essential for commercial operation of 3G/WLAN interworking in the first stage deployment. Depending on the business strategies, the Scenario 4 features may or may not be deployed in the long-term commercial operation. Scenarios 5 and 6 are typically ignored because the benefits of the extra features might not justify the deployment costs.

In the UMTS core network, the GPRS authentication function provides mutual authentication between the MS and UMTS core network. When a MS enters a SGSN area, the SGSN obtains an array of AVs from the HSS, and utilizes one AV for each authentication procedure. After the MS leaves the SGSN area, the SGSN may stores the unused AVs for a time period called *Reservation Timeout* (RT) period. If the MS returns to the SGSN area within the RT period, the SGSN uses the stored AVs for authentication instead of obtaining new AVs from the HSS. Because the AV usage mechanism consumes extra storage in the SGSN, it is desirable to selecting an appropriate RT period to reduce the access to the HSS and also consume acceptable storage in the SGSN. In this dissertation, we propose an analytic model to investigate the performance of AV usage mechanism. The results provide the mobile operators the suggestions to configure their system.

In addition to GPRS authentication, it is necessary to authenticate the MS before it can access IMS services. Without IMS authentication, a mobile user who passes the GPRS authentication can easily fake being another IMS user. Although both GPRS and IMS authentications are necessary, most steps in these two "authentication passes" are duplicated. Therefore, we propose an one-pass authentication procedure to reduce the IMS authentication traffic. We also formally prove that the one-pass procedure correctly authenticate the IMS users.

In the application layer, a CORBA-based OSA platform has been implemented in the *Industrial Technology Research Institute* (ITRI) and *National Chiao Tung University* (NCTU) Joint Research Center [17]. Based on the service platform, we implement the *Push to Talk over Cellular* (PoC) service, which provides a walkie-talkie like service in the cellular communication infrastructure. In this dissertation, we focus on the design and implementation of the PoC client. In the proposed PoC client architecture, most standard VoIP modules are reused for the PoC service, and the VoIP software can be easily extended to support PoC service.

The dissertation is organized as follows. Chapter 2 proposes the WGSN solution. A push mechanism is specified in the WGSN to forward the VoIP calls to the MS on which the VoIP client or WLAN module is not activated. Chapter 3 presents the analytic model for the AV usage mechanism. Chapter 4 proposes the one-pass authentication procedure for UMTS. Chapter 5 describes the design of the OSA-based PoC client. Finally, we conclude this dissertation with discussing our contribution and future work.

Chapter 2

WGSN: WLAN-based GPRS Support Node with Push Mechanism

This chapter proposes *WLAN-based GPRS Support Node* (WGSN), a solution for integrating 3G and WLAN services. We show that the 3G mechanisms can be re-used for WLAN user authentication and network access without introducing new procedures and without modifying the existing 3G network components. We describe the WGSN features and show how they are designed and implemented. To reduce the power consumption and computation complexity of a *Mobile Station* (MS), the WGSN applications may not be activated in the MS if they are not used. For an MS terminated application, a push mechanism is implemented in WGSN, which automatically activates the application at the MS side. An analytic model is proposed to investigate the requirements on the WGSN transmission delay of the push operation.

Our approach has similar development goals as the approaches described in [27][48]. We focus more on SIP applications (i.e., SIP *Application Level Gateway* (ALG) and the SIP push mechanism) in the WGSN system. A WGSN prototype has been implemented in *Industrial Technology Research Institute* (ITRI) and *National Chiao-Tung University* (NCTU) Joint Research Center.

2.1 The WGSN Approach

This section describes the architecture and the features of the *WLAN-based GPRS Support Node* (WGSN). WGSN interworks UMTS with WLAN to support Scenario 3 features described in Section 1.2.

2.1.1 WGSN Network Architecture

Figure 2-1 illustrates the inter-connection between a UMTS network and a WLAN network through WGSN. The UMTS network (Figure 2-1 (1)) provides 3G PS services. The WLAN network (Figure 2-1 (2)) provides access to Internet. The customers are allowed to roam between the two networks as long as the MS is equipped with both a 3G module and a WLAN card.

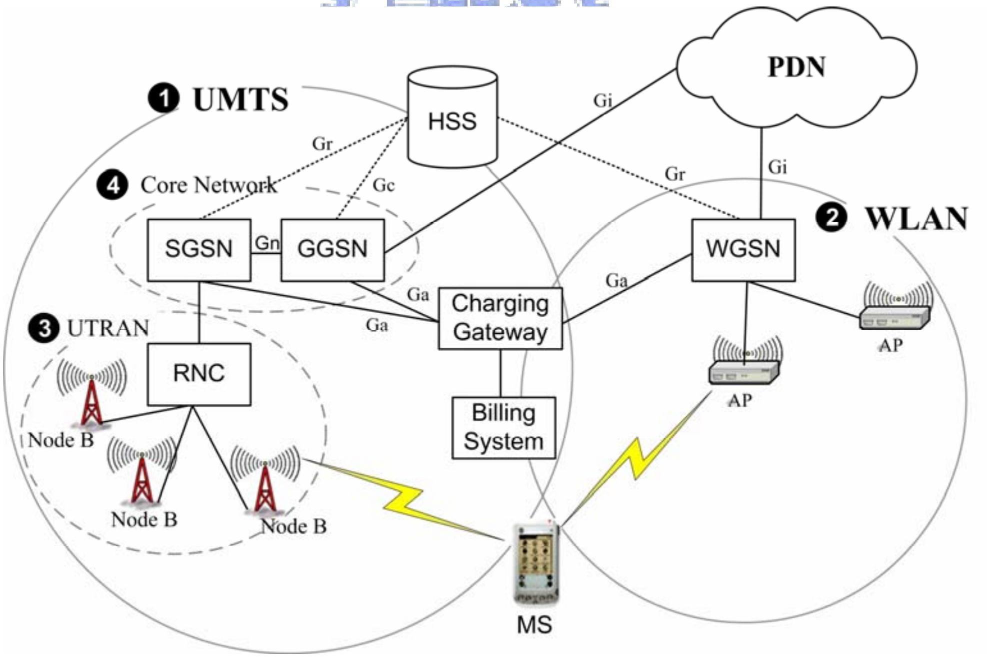


Figure 2-1. WGSN Architecture
(dashed lines: signaling; solid lines: data and signaling)

The UMTS network includes two sub-networks. The *UMTS Terrestrial Radio Access*

Network (UTRAN; Figure 2-1 (3)) consists of *Radio Network Controllers* (RNCs) and Node Bs (i.e., base stations). The radio interface between a Node B and an MS is based on WCDMA radio technology [28]. The UMTS core network (i.e., GPRS network; Figure 2-1 (4)) consists of *Serving GPRS Support Node* (SGSN) and *Gateway GPRS Support Node* (GGSN), which provide mobility management and session management services to mobile users. An SGSN connects to the UTRAN by *Asynchronous Transfer Mode* (ATM) links, and communicates with the GGSN through an IP-based backbone network. The GGSN connects to the external *Packet Data Network* (PDN) by an IP-based interface Gi. Both SGSN and GGSN communicate with the *Home Subscriber Server* (HSS) through the *Gr* and *Gc* interfaces, respectively. These two interfaces are based on the *Mobile Application Part* (MAP) [21]. The HSS is the master database containing all user-related subscription and location information.

The WLAN radio network includes 802.11-based *Access Points* (APs) that provide radio access for the MSs. The WGSN acts as a gateway between the PDN and the WLAN node, which obtains the IP address for an MS from a *Dynamic Host Configuration Protocol* (DHCP)[18] server and routes the packets between the MS and the external PDN. The WGSN node communicates with the HSS to support GPRS/UMTS mobility management following 3GPP Technical Specification 23.060 [5]. Therefore, the WLAN authentication and network access procedures are exact the same as that for GPRS/UMTS.

The WGSN node integrates both SGSN and GGSN functionalities. Like an SGSN, the WGSN communicates with the HSS through the Gr interface. On the other hand, like a GGSN, the WGSN communicates with the external PDN via the Gi interface. Therefore, for other GPRS/UMTS networks, the WGSN node and the corresponding WLAN network are considered as a separate GPRS network. The WGSN node can be plugged in any 3G core network without modifying the existing 3G nodes. To integrate the billing system for both UMTS and WLAN, WGSN communicates with the Charging Gateway using the same UMTS protocols (the *GPRS Tunneling Protocol* (GTP') protocol implemented in the Ga interface [5] or by FTP).

To access the WGSN services, the MS must be either a 3G-WLAN dual mode handset or a laptop/*Personal Data Assistant* (PDA) that equips with both WLAN *Network Interface Card* (NIC) and a 3G module.

2.1.2 WGSN Features

Based on the seven interworking aspects listed in 3GPP Technical Report 22.934 [3], we describe the features implemented in WGSN [23].

Service aspects: WGSN provides general Internet access and VoIP services based on SIP protocol [46]. Since a *Network Address Translator* (NAT) is built in the WGSN node, the VoIP voice packets delivered by the *Real Time Protocol* (RTP) [49] connection cannot pass through the WGSN node. This issue is resolved by implementing a SIP *Application Level Gateway* (ALG) [13] in the WGSN node, which interprets SIP messages and modifies the source IP address contained in these SIP messages.

In UMTS, an MS must activate the PDP [5] context for VoIP service before a caller from the external PDN can initiate a phone call to this MS. Also, for both UMTS and WLAN, a SIP *User Agent* (UA) must be activated in an MS before it can receive any incoming VoIP call. Therefore, a *SIP-based Push Center* (SPC) is implemented in the WGSN node to provide MS terminated SIP services. The SPC is implemented on a SMS-based IP service platform called iSMS [45], where none of the UMTS/GPRS components is modified. SPC also provides push mechanism through WLAN for a WGSN user who does not bring up the SIP UA. Therefore, the SIP terminated services (e.g., incoming VoIP calls) can be supported in WGSN.

Access control aspects: WGSN utilizes the standard UMTS access control for users to access WLAN services. Our mechanism reuses the existing UMTS *Subscriber Identity Module* (SIM) card and the subscriber data records in the HSS. Therefore, the WGSN customers do not need a separate WLAN access procedure, and the maintenance for customer information is simplified. User profiles for both UMTS and WLAN are combined in the same database (i.e., the HSS).

Security aspects: WGSN utilizes the existing UMTS authentication mechanism [34]. That is, the WLAN authentication is performed through the interaction between an MS (using 3G SIM card) and the 3G Authentication Center. Therefore, WGSN is as secured as existing 3G networks. We do not attempt to address the WLAN encryption issue [35]. It is well known that WLAN based on IEEE 802.11b is not secured. For a determined attack, *Wired Equivalent Privacy* (WEP) is not safe, which only makes a WLAN network more difficult for an attacker to intrude. The IEEE 802.11 Task Group I is investigating the current 802.11 *Media Access Control Address* (MAC) security. WGSN will follow the resulting solution.

Roaming aspects: WGSN provides roaming between UMTS and WLAN. We utilize the standard UMTS mobility management mechanism without introducing any new roaming procedures.

Terminal aspects: A terminal for accessing WGSN is installed with a *Universal IC Card* (UICC) reader (a smart card reader implemented as a standard device on the Microsoft Windows platform). The UICC reader interacts with the UMTS SIM card (i.e., the UICC containing the SIM application) to obtain authentication information for WGSN attach procedure.

Naming and addressing aspects: The WGSN user identification is based on *Network Access Identification* (NAI) format [12] following the 3GPP recommendation. Specifically, the *International Mobile Subscriber Identity* (IMSI) is used as WGSN user identification.

Charging and billing aspects: The WGSN acts as a router, which can monitor and control all traffic for the MSs. The WGSN node provides both offline charging and online charging (for pre-paid services) based on the *Call Detail Records* (CDRs) delivered to the charging gateway.

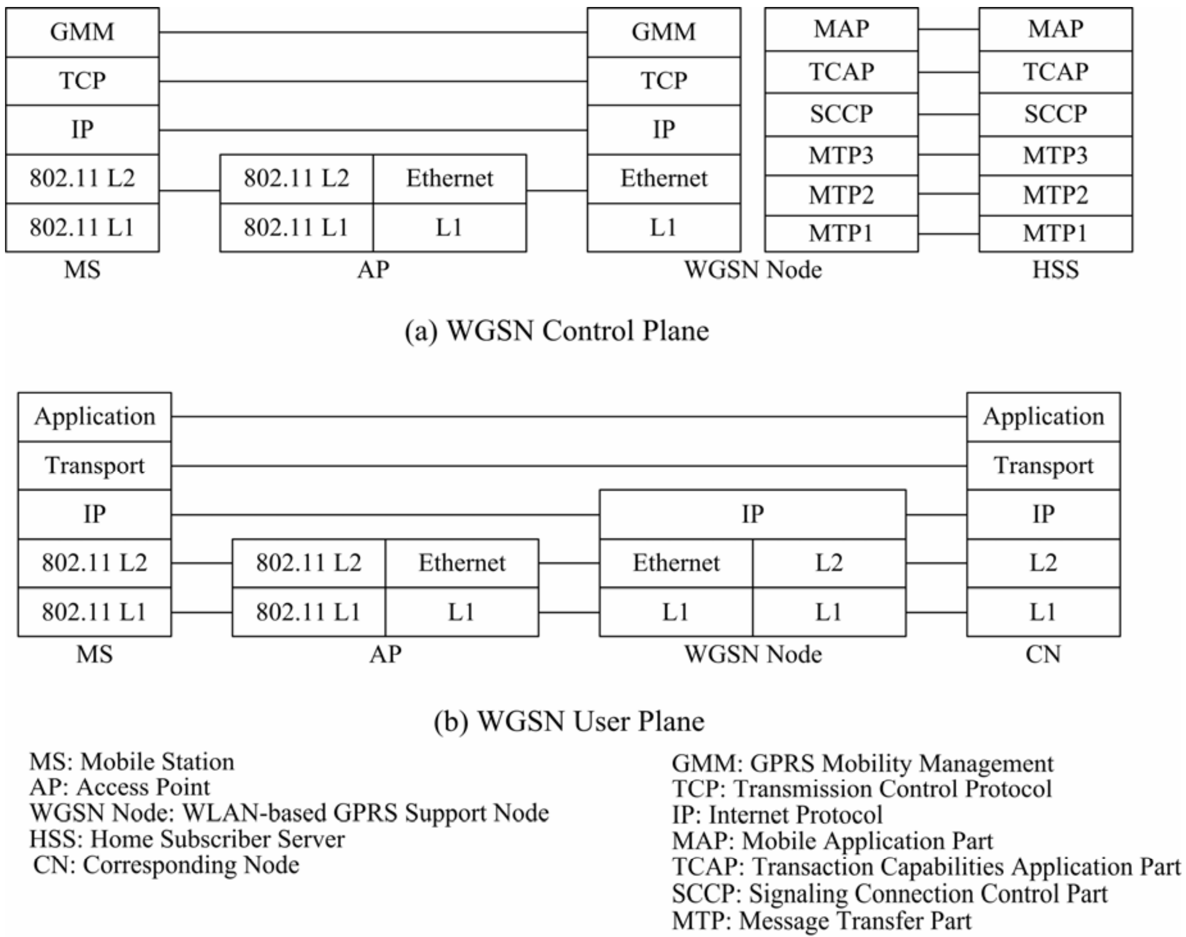
Besides the seven aspects listed above, WGSN also provides automatic WLAN network configuration recovery. A WGSN MS can be a notebook, which is used at home or office with different network configurations. The network configuration information includes IP address, subnet mask, default gateway, WLAN *Service Set Identifier* (SSID), etc. When the MS enters the WGSN service area, its network configuration is automatically reset to the

WGSN WLAN configuration if the MS is successfully authenticated. The original network configuration is automatically recovered when the MS detaches from the WGSN. This WGSN functionality is especially useful for those users who are unfamiliar with network configuration setup.

2.2 Implementation of WGSN

This section describes the implementation of WGSN. We first introduce the protocol stack among MS, AP, WGSN, and HSS. Then we elaborate on the WGSN components for the WGSN network node and the MS. Figure 2-2 illustrates the WGSN protocol stack. In the current implementation, the lower-layer protocol between the MS and the WGSN node is IP over 802.11 radio (through WLAN AP). In the control plane, standard *GPRS Mobility Management* (GMM) defined in 3GPP Technical Specification 23.060 [5] is implemented on top of TCP/IP between the MS and the WGSN node. The standard UMTS Gr interface is implemented between the WGSN node and the HSS through *Signaling System Number 7* (SS7)-based MAP protocol [35]. The layers of the SS7 protocol include *Message Transfer Part* (MTP), *Signaling Connection Control Part* (SCCP), and *Transaction Capabilities Application Part* (TCAP). Details of SS7 can be found in [35]. The WGSN node communicates with the charging gateway through the IP-based GTP' protocol, which is not shown in Figure 2-2. In the future, the TCP/IP layers in the control plan will be replaced by *Extensible Authentication Protocol / EAP Over LAN* (EAP/EAPOL) [14][31]. EAP/EAPOL operates over 802.11 MAC layer, which allows authentication of an MS before it is assigned an IP address. Therefore, the IP resource of WGSN system can be managed with better security. Also, between the WGSN node and the HSS, the lower-layer SS7 protocols (i.e., MTP and SCCP) will be replaced by IP-based *Stream Control Transmission Protocol* (SCTP) [50] to support all-IP architecture.

The WGSN user plane follows standard IP approach. That is, the MS and the WGSN node interact through the Internet protocol. The MS communicates with the CN in the external PDN using the transport layer over IP. In the user plane, the WGSN node serves as a



(a) WGSN Control Plane

(b) WGSN User Plane

Figure 2-2. WGSN Protocol Stack

gateway between the WLAN network and the external PDN.

The WGSN MS must be either a 3G-WLAN dual mode handset or a laptop/PDA that equips with both WLAN NIC and a 3G module. The UICC reader (which can be contained in the 3G module or a separate smart card reader) communicates with the standard SIM card to obtain the authentication information required in both 3G network and WLAN. In the current WGSN implementation, we use GPRS module instead of 3G module. The WGSN UICC reader is implemented as a standard device on the Microsoft Windows platform. The WGSN software modules are implemented on the Window 2000 and XP OS platforms for notebooks and WinCE for PDAs. A WGSN client is implemented to carry out tasks in the control plane. Several SIP user agents are implemented for SIP-based applications in the user plane. The modules for WGSN client are described as follows.

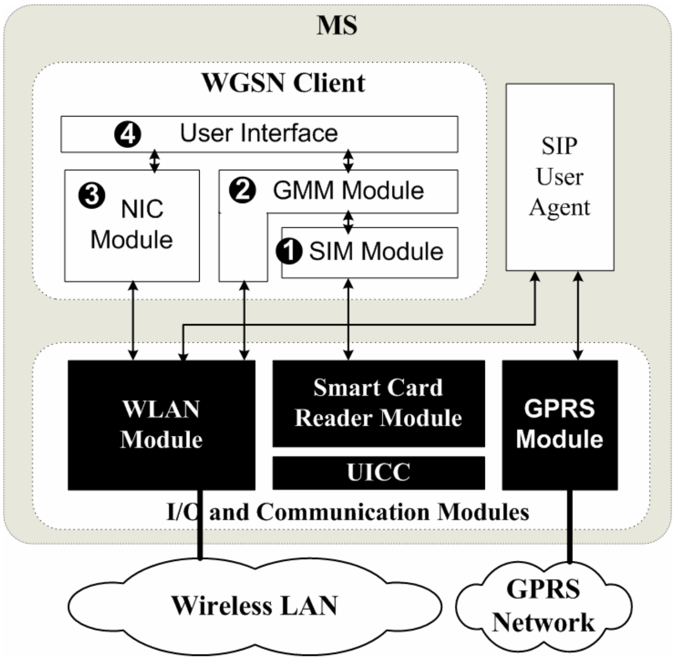


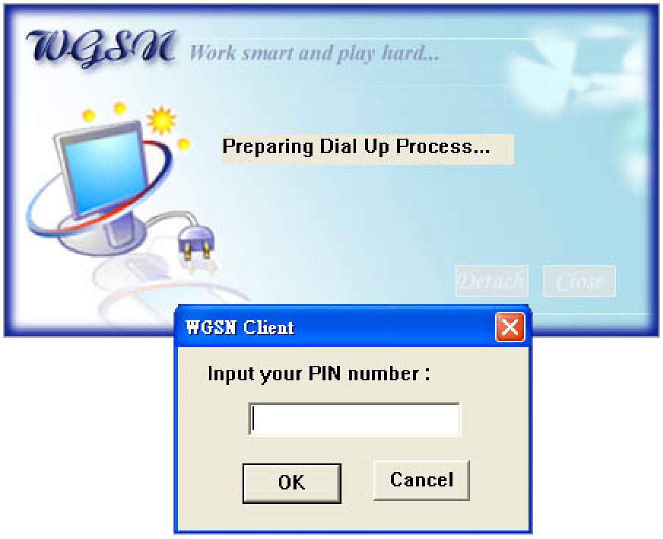
Figure 2-3. WGSN MS Architecture

SIM Module (Figure 2-3 (1)): As in UMTS, a WGSN user is authenticated using the UMTS SIM card (or GPRS SIM card in the current implementation) before the user can access the WLAN network. Through the UICC smart card reader, the SIM module retrieves the SIM information (including IMSI, SRES and Kc) [34] and forwards the information to the GMM module.

GMM Module (Figure 2-3 (2)): Based on the SIM information obtained from the SIM module, the GMM module communicates with the WGSN node to perform MS attach and detach. The authentication action is included in the attach procedure.

NIC Module (Figure 2-3 (3)): The network configurations of different WLANs may be different. With the *Operating System* (OS) support, the NIC module dynamically sets up appropriate network configurations when a WGSN user moves across different WLAN networks. WGSN utilizes DHCP for IP address management. The WGSN MS must obtain a legal IP address and the corresponding network configuration through the DHCP lease request. On the other hand, when the MS terminates a WGSN connection, it should send the IP release message to the WGSN node, and the IP address is reclaimed for the next WGSN user. The NIC module then recovers the original network configuration for the MS. If the MS is abnormally terminated, the NIC module cannot immediately recover the network configuration. Instead, the NIC module offers a Window OS program called WGSN Service. When the MS is re-started, this service will check if the network configuration has been recovered. If not, the configuration previously recorded by the NIC module is used.

User Interface (Figure 2-3 (4)): A user interacts with the WGSN system through the MS user interface. As illustrated in Figure 2-4, the user types the *Global System for Mobile communications* (GSM) / *General Packet Radio Service* (GPRS) pin number to initiate the WGSN connection. Like the usage of a GPRS handset, the pin number can be disabled. Based on the received command, the corresponding modules are instructed to carry out the desired tasks. During a WGSN session, the user interface indicates the status of the execution and displays the elapsed time of the WGSN connection.



WGSN Work smart and play hard...

Preparing Dial Up Process...

WGSN Client

Input your PIN number :

Figure 2-4. WGSN User Interface

On the network side, the WGSN node is implemented on the Advantech Industrial Computer platform S-ISXTV-141-W3. The black boxes in Figure 2-5 illustrate the WGSN communication modules, which include

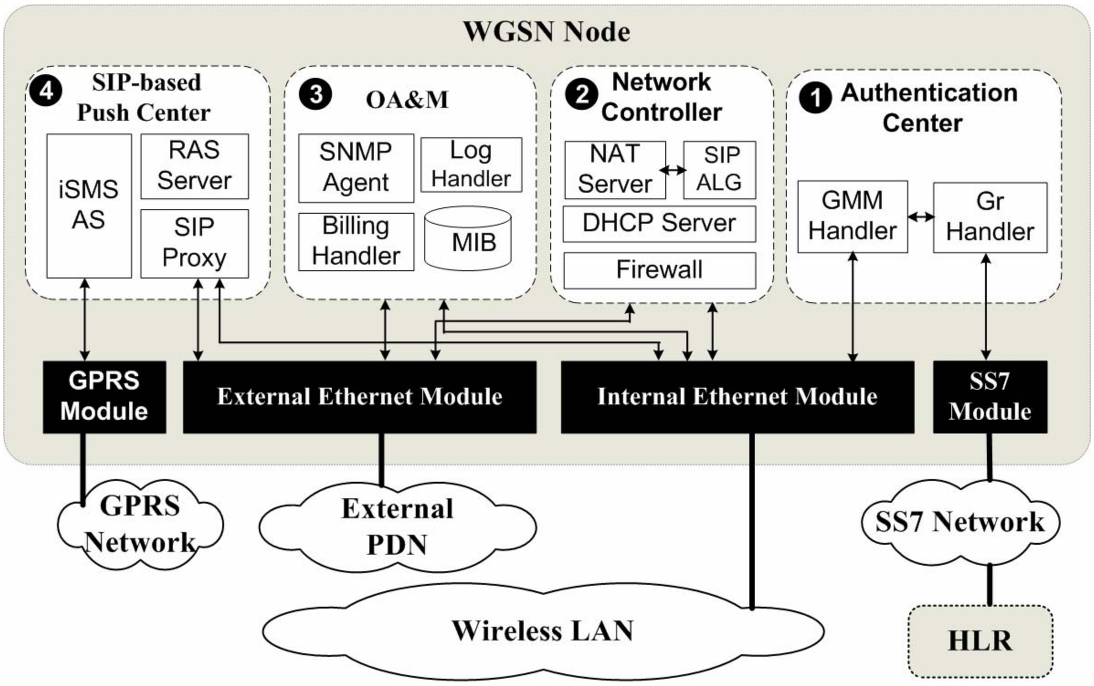


Figure 2-5. WGSN Node Architecture

- A SS7 module for communications with the HSS (through the SS7 network). In this module, the MTP, SCCP and TCAP layers (see Figure 2-2 (a)) are based on Connect7 2.4.0-Beta version software developed by SS8 Networks Cooperation.
- An internal Ethernet module for communications with the WLAN APs
- An external Ethernet module for communications with the external PDN
- A GPRS module for communications with the MS (through the GPRS network)

The software architecture of the WGSN node includes four major components:

Authentication Center (Figure 2-5 (1)) consists of the GMM and the Gr handlers. Through the internal Ethernet module, the GMM handler receives the GMM messages from the WGSN MS, and dispatches the corresponding tasks to the other WGSN modules (the details are described in Step 4 and Step 8 in the attach procedure in Section 2.3). The Gr handler implements the standard GMM primitives for the Gr interface [5]. Through the SS7 module, the Gr handler interacts with the HSS for MS network access and authentication. Specifically, it obtains an array of authentication vectors (including a random number Rand, a signed result SRES, and an encryption key Kc) from the GPRS authentication center (which may or may not be co-located with the HSS). The size of authentication array can be dynamically adjusted (see [34] for the details). Each time the WGSN MS requests for authentication, the Gr handler uses an authentication vector to carry out the task as specified in 3GPP Technical Specification 33.102 [10]. Furthermore, when an MS detaches, the Gr handler should inform the HSS to update the MS status. The current WGSN version has implemented two MAP service primitives: the MAP_SEND_AUTHENTICATION_INFO and MAP_PURGE_MS services. These primitives are implemented on MAP version 1.4 software developed by Trillium Digital Systems Inc.

Network Controller (Figure 2-5 (2)) provides the following functions for Internet access:

- IP address management: A DHCP server is implemented in the WGSN node to distribute private IP addresses to the MSs. An NAT server performs address translation when the IP packets are delivered between the private (WLAN) and the public (external PDN) IP address spaces.
- Internet access control: The WGSN node only allows the authenticated users to access Internet services. Unauthorized packets will be filtered out by the firewall.
- SIP application support: To support SIP-based applications under the NAT environment, the WGSN node implements a SIP ALG that modifies the formats of SIP packets so that these packets can be delivered to the WGSN MSs through the WGSN node.

Operation, Administration and Maintenance (OA&M; see Figure 2-5 (3)) controls and monitors individual WGSN user traffics. WGSN utilizes *Simple Network Monitoring Protocol* (SNMP) as the network management protocol. With *Management Information Base* (MIB), every managed network element is represented by an object with an identity and several attributes. An SNMP agent is implemented in the WGSN node, which interacts with the managed network element through SNMP. For example, the traffic statistics of an AP can be accessed by the OA&M (through the corresponding MIB object) and displayed in a web page using *Multi Router Traffic Grapher* (MRTG 2.9.22) [41]. The SNMP agent can also detach an MS through the MIB object of the MS. A log handler is implemented in the OA&M to record all events occurring in the WGSN node. A billing handler generates CDRs, which communicates with the billing gateway through the GTP' protocol or *File Transfer Protocol* (FTP).

SIP-based Push Center (SPC; see Figure 2-5 (4)) provides a push mechanism for GPRS networks [39] that support private IP addresses. Since GPRS significantly consumes the MS power, a mobile user typically turns on GSM but turns off GPRS unless he/she wants to originate a GPRS session. In this case, services cannot be pushed to the users from the network side. For an MS that is GSM attached but GPRS detached, the SPC can push a SIP request to the MS through a SMS application server called iSMS AS [45]. SPC also provides push mechanism through WLAN for a WGSN user who does not bring up the SIP UA. Therefore, the SIP terminated services (e.g., incoming VoIP calls) can be supported in WGSN. We will elaborate more on SPC in Section 2.4.

2.3 Attach and Detach Procedures

In WGSN attach and detach procedures, the message flows between the WGSN node and the HSS are the same as that for the SGSN and the HSS in UMTS. The message flows between the MS and the WGSN node are specific to the WGSN network, which are not found in UMTS.

The attach procedure is illustrated in Figure 2-6, which consists of the following steps:

Step 1: When the WGSN user brings up the MS user interface, the SIM module is invoked to configure the smart card reader and (optionally) requests the user to input the *Personal Identification Number* (PIN) number. The card reader

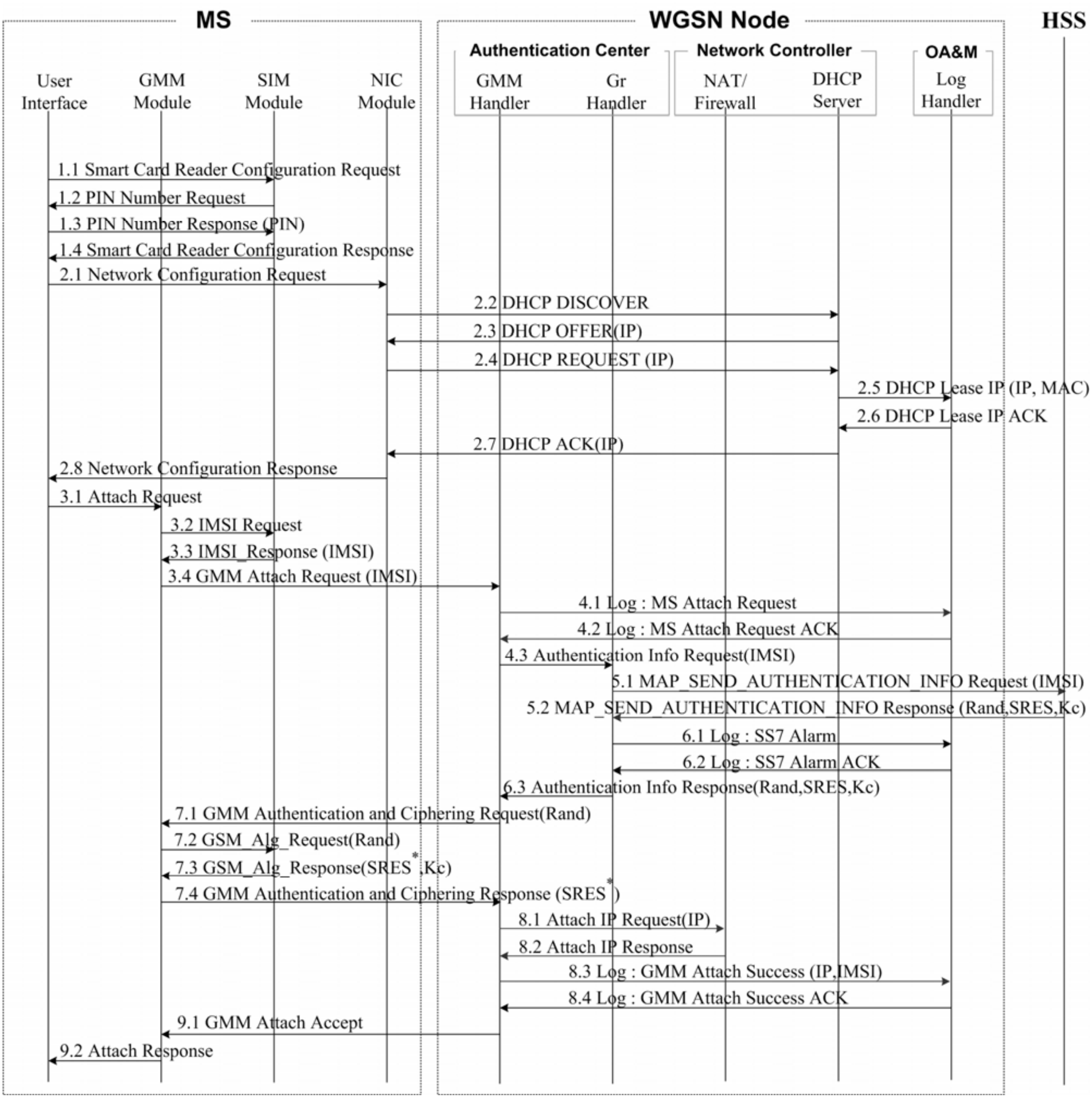

MS
WGSN Node
HSS
Authentication Center
Network Controller
OA&M
User Interface
GMM Module
SIM Module
NIC Module
GMM Handler
Gr Handler
NAT/ Firewall
DHCP Server
Log Handler
1.1 Smart Card Reader Configuration Request
1.2 PIN Number Request
1.3 PIN Number Response (PIN)
1.4 Smart Card Reader Configuration Response
2.1 Network Configuration Request
2.2 DHCP DISCOVER
2.3 DHCP OFFER(IP)
2.4 DHCP REQUEST (IP)
2.5 DHCP Lease IP (IP, MAC)
2.6 DHCP Lease IP ACK
2.7 DHCP ACK(IP)
2.8 Network Configuration Response
3.1 Attach Request
3.2 IMSI Request
3.3 IMSI_Response (IMSI)
3.4 GMM Attach Request (IMSI)
4.1 Log : MS Attach Request
4.2 Log : MS Attach Request ACK
4.3 Authentication Info Request(IMSI)
5.1 MAP_SEND_AUTHENTICATION_INFO Request (IMSI)
5.2 MAP_SEND_AUTHENTICATION_INFO Response (Rand,SRES,Kc)
6.1 Log : SS7 Alarm
6.2 Log : SS7 Alarm ACK
6.3 Authentication Info Response(Rand,SRES,Kc)
7.1 GMM Authentication and Ciphering Request(Rand)
7.2 GSM_Alg_Request(Rand)
7.3 GSM_Alg_Response(SRES*,Kc)
7.4 GMM Authentication and Ciphering Response (SRES*)
8.1 Attach IP Request(IP)
8.2 Attach IP Response
8.3 Log : GMM Attach Success (IP,IMSI)
8.4 Log : GMM Attach Success ACK
9.1 GMM Attach Accept
9.2 Attach Response

Figure 2-6. Message Flow for the Attach Procedure

authenticates the user through the pin number just like a GPRS mobile phone.

Step 2: The MS NIC module is invoked to store the current WLAN network configuration. To obtain the network configuration of WGSN, MS broadcasts a DHCP DISCOVER message on its subnet and looks for a DHCP server. The DHCP server in the WGSN node replies MS a DHCP OFFER message which includes an available IP address. Then, MS sends a DHCP REQUEST message to DHCP server and asks for the usage of the available IP address contained in DHCP OFFER message. If the DHCP server accepts the request, it reports the IP lease event to the Log handler and sends MS a DHCP ACK message with network configuration parameters. Finally, the MS NIC module sets up the new network configuration.

Step 3: The MS GMM module is invoked to perform the attach operation. The GMM module first obtains the IMSI from the SIM module. Then it sends the GMM Attach Request (with the parameter IMSI) to the WGSN node.

Step 4: When the GMM handler of the WGSN node receives the attach request, it reports this event to the Log handler, and sends the authentication information request to the Gr handler.

Step5: The GMM handler sends the MAP_SEND_AUTHENTICATION_INFO Request (with the argument IMSI) to the HSS. The HSS returns the authentication vector (Rand, SRES, Kc) through the MAP_SEND_AUTHENTICATION_INFO Response message.

Step 6: The WGSN Gr handler issues the SS7 Alarm message to the Log handler, and the event is logged. The Gr handler returns the authentication vector to the GMM handler.

Step 7: The GMM handler sends the GMM Authentication and Ciphering Request (with the parameters IMSI and Rand) to the GMM module of the MS. The GMM module passes the random number Rand to the SIM module, and the SIM module computes the signed result SRES* and the encryption key Kc based on the received Rand and the authentication key Ki stored in the SIM card. These results are returned to the GMM module. The GMM module returns the computed SRES* to the GMM handler of the WGSN node using the GMM Authentication and Ciphering Response (with the parameters IMSI and SRES*). The GMM handler compares SRES with SRES*. If they match, the authentication is successful.

Step 8: The GMM handler sends the Attach IP message to the firewall, which will allow the packets of this IP address to pass the WGSN node. Then the GMM handler reports to the Log handler that the attach is successful (with the corresponding IMSI and IP address).

Step 9: The GMM handler sends the GMM Attach Accept message to the GMM module of the MS, and the GMM module passes the Attach Response message to the user interface. At this point, the attach procedure is completed.

The WGSN connection can be detached by the MS or by the network (the WGSN OA&M). The message flow for MS initiated detach is illustrated in Figure 2-7, and the steps are described as follows.

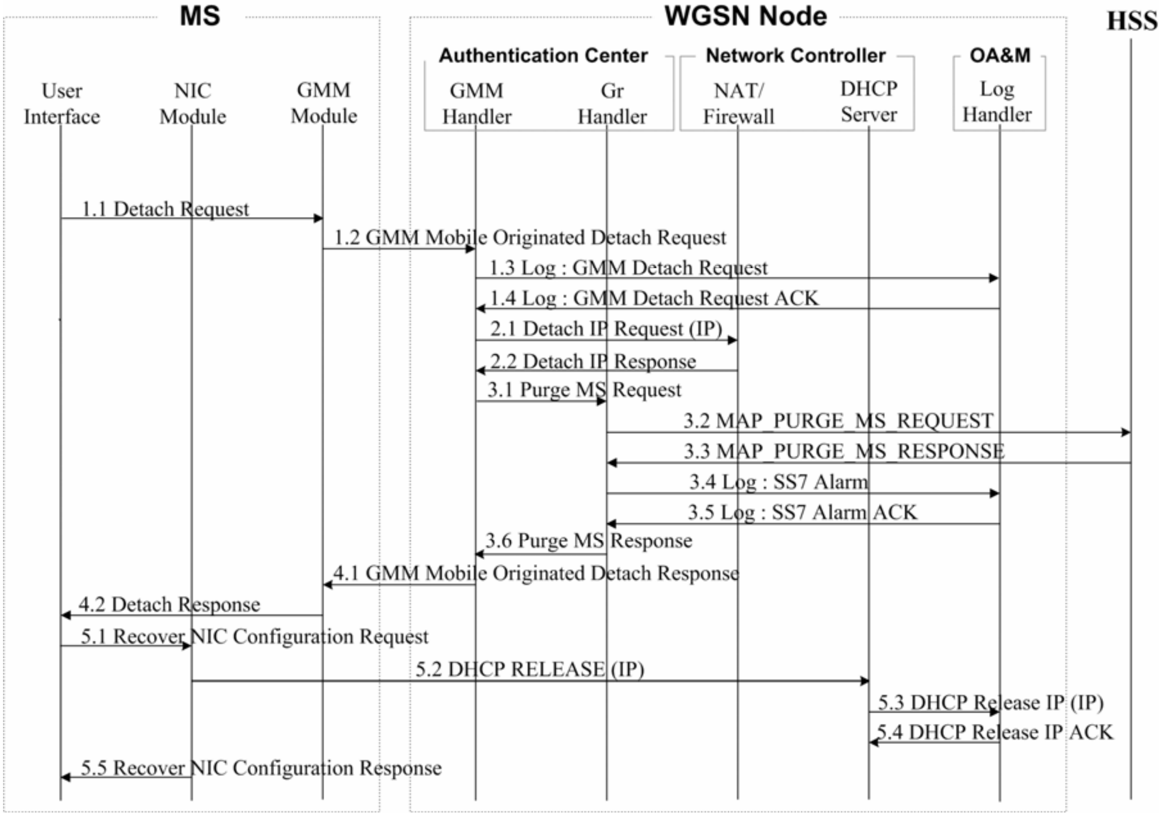


Figure 2-7. Message Flow for the MS-Initiated Detach Procedure

Step 1: When the user presses the detach button in the user interface, the GMM module is invoked to send the GMM Mobile Originated Detach Request (with parameters IMSI and IP address) to the GMM handler of the WGSN node. The GMM handler reports this detach event to the Log handler.

Step 2: The GMM handler sends the detach IP request to the firewall. From now on, the packets of this IP address will be filtered out by the firewall of the WGSN node.

Step 3: The GMM handler invokes the Gr handler to send the MAP_PURGE_MS Request (with the parameters IMSI and the SSN address of the WGSN node) to the HSS. The HSS updates the MS status in the database and replies the MAP_PURGE_MS Response to the Gr handler. The Gr handler reports this event to the Log handler.

Step 4: Through Mobile Originated Detach Response, the GMM handler informs the MS GMM module that the detach operation is complete.

Step 5: The MS NIC module is instructed to recover the original network configuration. It sends the DHCP RELEASE message to the DHCP server in the WGSN node. The DHCP server reclaims the IP address and reports this event to the Log handler. The NIC module then recovers the original network configuration (which was saved in Step 2 of the attach procedure).

The message flow for the network-initiated detach procedure is similar to that illustrated in Figure 2-7, and the details can be found in [23].

2.4 WGSN Push Mechanism

To reduce the power consumption and computation complexity of a WGSN MS, most WGSN applications are not activated at the MS until the user actually accesses them. This approach does not support "always-on" or MS terminated services such as incoming VoIP calls. To address this issue, a push mechanism called *Session Initiation Protocol* (SIP)-*based Push Center* (SPC) has been implemented in the WGSN node. In this approach, the mobile *Short Message Service* (SMS) mechanism, which consumes much less power than the WLAN modules, is always on.

Figure 2-8 illustrates the message flow of the push mechanism performed in WGSN. Suppose that a SIP VoIP caller in the external PDN issues a call request to a WGSN MS through SIP [46], the request is first sent to the WGSN node (path (a) in Figure 2-8). The

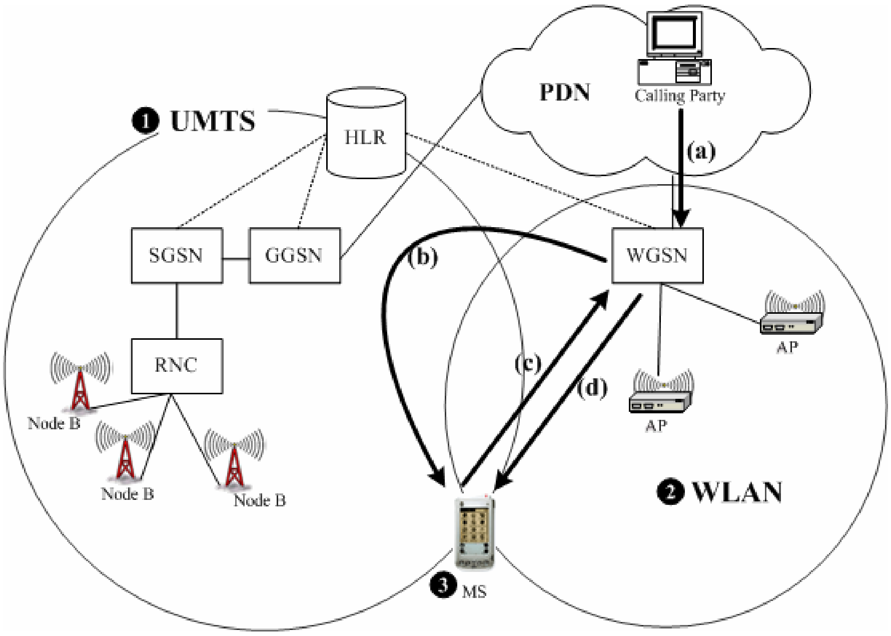


Figure 2-8. Message Flow of the SPC in WGSN

SPC checks if the SIP *User Agent* (UA) of the called MS is activated. If so, the request is directly forwarded to the called MS (path (d) in Figure 2-8). Otherwise, the request is suspended, and the SPC sends a GSM short message to the MS to activate the corresponding SIP UA (path (b) in Figure 2-8). After the SIP UA is activated, the MS informs the SPC (path (c) in Figure 2-8), and the call request from the caller is then delivered to the SIP UA following the standard SIP call setup procedure. We note that the VoIP call model is typically handled by the SIP UAs or a call server (or softswitch) that control the call setup process and indicate whether the called party is busy or idle [24]. The WGSN SPC is only responsible for pushing the SIP requests to the MSs where the SIP UAs are not activated.

For every SIP UA (an MS may have several SIP UAs for different applications), a status record is maintained in the SPC. A four-state *Finite State Machine* (FSM) is associated with the record. These states are

State 0: The SPC has not initiated the activation process.

State 1: The SPC has initiated the activation process, and one incoming call is waiting for setup.

State 2: The SPC has initiated the activation process. No incoming call is waiting for setup.

State 3: The SIP UA is active.

The incoming call waiting for setup at State 1 is referred to as the *outstanding call*. There is at most one outstanding call during the SIP UA activation process. The state transition diagram for the FSM of a status record is illustrated in Figure 2-9 and the details are given below:

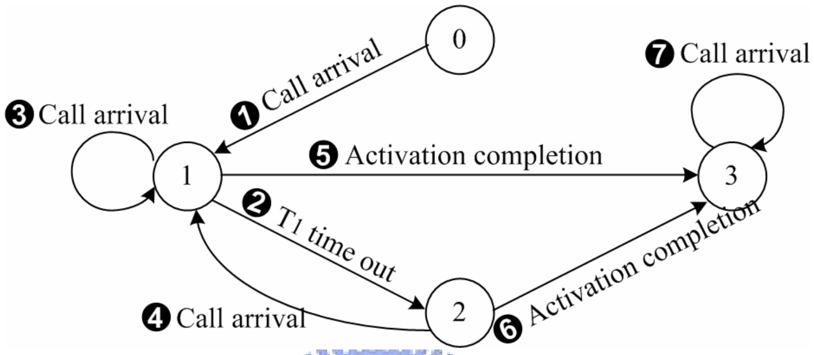


Figure 2-9 The FSM State Transition Diagram for the SPC Status Record

Transition 1: An incoming call request arrives at State 0. The SPC sends a message to activate the SIP UA. The SPC sets a timer *T1* and changes the state to “1”.

Transition 2: The timer *T1* expires at State 1. The SPC drops the current incoming call request by sending a timeout message to the caller. The state changes to “2”.

Transition 3: An incoming call arrives at State 1. The SPC drops this call request (because the called MS is already engaged in an outgoing call setup). The state remains in “1”.

Transition 4: An incoming call request arrives at State 2. This call becomes the outstanding call. The SPC sets the timer *T1* and changes the state to “1”.

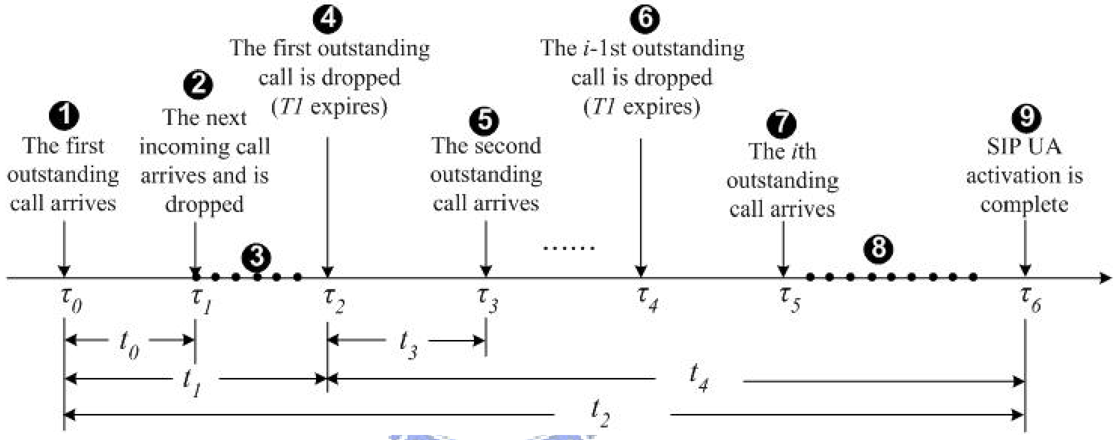
Transition 5: When the SPC receives the activation complete message from the called MS at State 1, the SPC forwards the outstanding call request to the SIP UA of the called MS following the standard SIP protocol. The state is changed to “3”.

Transition 6: When the SPC receives the activation complete message from the called MS at State 2, the SPC changes the state to “3”.

Transition 7: When the SPC receives a call request at State 3, it directly forwards the call request to the SIP UA of the called MS following the standard SIP protocol.

It is clear that for every SIP UA, the FSM eventually moves to State 3. There is exact one outstanding call at State 1, and there is no outstanding call at State 2. During the state transition, an incoming call is “lost” if either Transition 2 or 3 occurs.

2.4.1 Performance Analysis



❶ The first outstanding call arrives
❷ The next incoming call arrives and is dropped
❸
❹ The first outstanding call is dropped (T1 expires)
❺ The second outstanding call arrives
......
❻ The i-1st outstanding call is dropped (T1 expires)
❼ The ith outstanding call arrives
❽
❾ SIP UA activation is complete
τ0
τ1
τ2
τ3
τ4
τ5
τ6
t0
t1
t3
t4
t2

Figure 2-10. The Timing Diagram

(A dot “**.**” represents dropping of an incoming call immediately after it arrives at the SPC)

Figure 2-10 illustrates the timing diagram for the execution of SIP UA activation procedure in the SPC mechanism. This procedure is initiated by the first incoming call arriving at time τ_0 (Figure 2-10 (1)). Suppose that the SPC detects that the SIP UA of the destination MS is not activated. This incoming call is suspended at the SPC. The SPC sends a GSM short message to activate the destination MS (see Figure 2-8 (b)) and sets a timer *T1* for this outstanding call. If the activation procedure is not complete before *T1* expires, the call is dropped. In Figure 2-10, the timer *T1* for the first outstanding call expires at time τ_2 (Figure 2-10 (4)), and the SPC receives the activation complete message from the called MS at time τ_6 (Figure 2-10 (9)), where $\tau_6 > \tau_2$. During SIP UA activation, new incoming calls for the destination MS may arrive. If the outstanding call has not been dropped when a new incoming call arrives, then this new incoming call is dropped (see Figure 2-10 (2), (3), and (8)). Otherwise, this incoming call becomes the next outstanding call (see Figure 2-10 (5) and (7)).

In the following sub-sections, we investigate the performance of SIP push mechanism, where the expected number of lost calls during SIP UA activation is computed. The lost calls include the dropped outstanding calls due to *T1* expiration (see Figure 2-10 (4) and (6)) and the incoming calls arriving when an outstanding call exists (see Figure 2-10 (2), (3), and (8)).

2.4.2 Analytic Model

In this section, we present an analytic model to investigate the performance of push mechanism. We make the following assumptions:

1. The incoming call arrivals are a Poisson process with rate λ; therefore, the inter call arrival time t_0 is Exponentially distributed with the density function $f_{t_0}(t_0) = \lambda e^{-\lambda t_0}$. In Figure 2-10, $t_0 = \tau_1 - \tau_0$.
2. The *T1* timeout period (denoted as t_1) has the density function $f_{t_1}(t_1)$. In Figure 2-10, $t_1 = \tau_2 - \tau_0$. We consider *T1* with fixed interval 1/μ.
3. The SIP UA activation time is denoted as $t_2 = \tau_6 - \tau_0$. In this section, we assume t_2 to be Exponentially distributed with the mean 1/γ, and the density function $f_{t_2}(t_2) = \gamma e^{-\gamma t_2}$. We will also consider Gamma distributed t_2 in the simulation model.

In our study, the output measure is the expected number $E[N]$ of the lost calls during the activation period.

Consider the following two cases.

Case 1) The first outstanding call is successfully set up; i.e., the activation time t_2 of SIP UA is shorter than 1/μ.

Case 2) The first outstanding call is dropped; i.e., the activation time t_2 of SIP UA is longer than 1/μ.

Let P_i be the probability that Case *i* occurs, and N_i be the expected number of lost calls in Case *i*. P_i can be expressed as

$$\begin{aligned} P_1 &= \Pr[t_2 < 1/\mu] \\ &= \int_{t_2=0}^{1/\mu} f_{t_2}(t_2)dt_2 \\ &= 1-e^{-\left(\frac{\gamma}{\mu}\right)} \end{aligned}$$

and

$$P_2 = 1-P_1 = e^{-\left(\frac{\gamma}{\mu}\right)} \tag{2.1}$$

In Case 1, all incoming calls arrive during the SIP UA activation period t_2 are lost. Since the incoming calls are a Poisson stream, the expected number of lost calls during t_2 is λt_2. Therefore

$$\begin{aligned} P_1 \cdot N_1 &= \int_{t_2=0}^{1/\mu} \lambda t_2 \times f_{t_2}(t_2)dt_2 \\ &= \frac{\lambda}{\gamma} - \left(\frac{\lambda}{\gamma}\right)e^{-\left(\frac{\gamma}{\mu}\right)} - \left(\frac{\lambda}{\mu}\right)e^{-\left(\frac{\gamma}{\mu}\right)} \end{aligned} \tag{2.2}$$

In Case 2, the first outstanding call and all incoming calls during the waiting time of the first outstanding call are dropped. That is, the expected number of lost calls before *T1* expires is $1+\dfrac{\lambda}{\mu}$. We further analyze Case 2 by two sub-cases in terms of the next event after *T1* expires.

Case 2-1) The next event after the *T1* expiration is the completion of SIP UA activation (i.e., $t_4 < t_3$ in Figure 2-10). The expected number of lost calls after the drop of first outstanding call is 0.

Case 2-2) The next event after the *T1* expiration is an incoming call request (i.e., $t_3 < t_4$ in Figure 2-10), and this incoming call becomes the second outstanding call. From the residual life theorem and the memoryless property of the Exponential distribution

[47], t_3 has the same distribution as that for the inter call arrival time t_0. That is, $f_{t_3}(t) = f_{t_0}(t) = \lambda e^{-\lambda t}$. Similarly, we have $f_{t_4}(t) = f_{t_2}(t) = \gamma e^{-\gamma t}$. Since t_3 and t_4 have the same distributions as those for t_0 and t_2, respectively, the situation seen by this outstanding call is the same as that seen by the first outstanding call. Therefore, the expected number of lost calls after the arrival of this new outstanding call is $E[N]$.

Let $P_{2\text{-}i}$ denote the probability that Case 2-*i* occurs, we have

$$
\begin{aligned}
N_2 &= \left(1+\frac{\lambda}{\mu}\right) + \left(P_{2-1}\times 0 + P_{2-2}\times E[N]\right) \\
&= \left(1+\frac{\lambda}{\mu}\right) + \frac{\lambda E[N]}{\lambda+\gamma} \qquad (2.3)
\end{aligned}
$$

From Equations (2.1), (2.2), and (2.3)

$$
\begin{aligned}
E[N] &= \sum_{i=1}^{2} P_i \times N_i \\
&= \left[\frac{\lambda}{\gamma} - \left(\frac{\lambda}{\gamma}\right)e^{-\left(\frac{\gamma}{\mu}\right)} - \left(\frac{\lambda}{\mu}\right)e^{-\left(\frac{\gamma}{\mu}\right)}\right] + e^{-\left(\frac{\gamma}{\mu}\right)} \times \left[\left(1+\frac{\lambda}{\mu}\right) + \frac{\lambda E[N]}{\lambda+\gamma}\right] \qquad (2.4)
\end{aligned}
$$

By re-arranging Equation (2.4), we have

$$
E[N] = \frac{\dfrac{\lambda}{\gamma} + \left(1-\dfrac{\lambda}{\gamma}\right)\times e^{-\left(\frac{\gamma}{\mu}\right)}}{1-\left(\dfrac{\lambda}{\lambda+\gamma}\right)\times e^{-\left(\frac{\gamma}{\mu}\right)}} \qquad (2.5)
$$

2.4.3 Simulation Model

We utilize discrete event simulation experiments to validate the analytic model described in Section 2.4.2. In the simulation, three types of events are defined: **CallArrival** event represents the arrival of incoming call to an MS, **TimerExpiration** event represents that *T1* of an outstanding call expires, and **ActivationComplete** event indicates that the SIP UA of the called MS is successfully activated. Every event is associated with a timestamp representing when the event occurs. The type of an event *e* is denoted as *e*.type, and the timestamp of *e* is denoted as *e*.timestamp.

In the simulation, all events are first inserted into an event list. Then they are deleted from the event list and processed in the non-decreasing timestamp order. A simulation clock *clk* is maintained to indicate the progress of the simulation. The clock value is the timestamp of the event being processed. The inter call arrival times are produced from an Exponential random number generator with mean 1/λ. The SIP UA activation times are drawn from an Exponential random number generator with mean 1/γ. The *T1* timeout periods are drawn from an Exponential random number generator with mean 1/μ. The flag **OutstandingCall** is used in the simulation to indicate if there exists an outstanding call.

To ensure that the simulation results are stable, the SIP UA activation procedure is processed K=10^7 iterations. At the *i*th iteration ($i \le K$), the number n_i of lost calls during the SIP UA activation procedure is computed. Finally, the expected number $E[N]$ of lost calls is computed as

$$E[N] = \frac{\sum_{i=1}^{K} n_i}{K} \tag{2.6}$$

The flowchart of the simulation is shown in Figure 2-11, and the details are described as follows.

Step 1. The simulation is initialized (i.e., K=10^7, i=0).

Step 2. Check if K simulation iterations have been complete. If so, go to Step 11. If not, go to Step 3.

Step 3. The *i*th iteration is initialized. Specifically, the event list is reset, *clk* is set as 0, and the number of lost calls n_i is initialized to 0. Since the activation procedure is triggered by the first outstanding call, initially, **OutstandingCall**=*TRUE*. Then, three events are generated and inserted into the event list. The **ActivationComplete** event indicates the completion of the *i*th iteration. The **TimerExpiration** event shows that

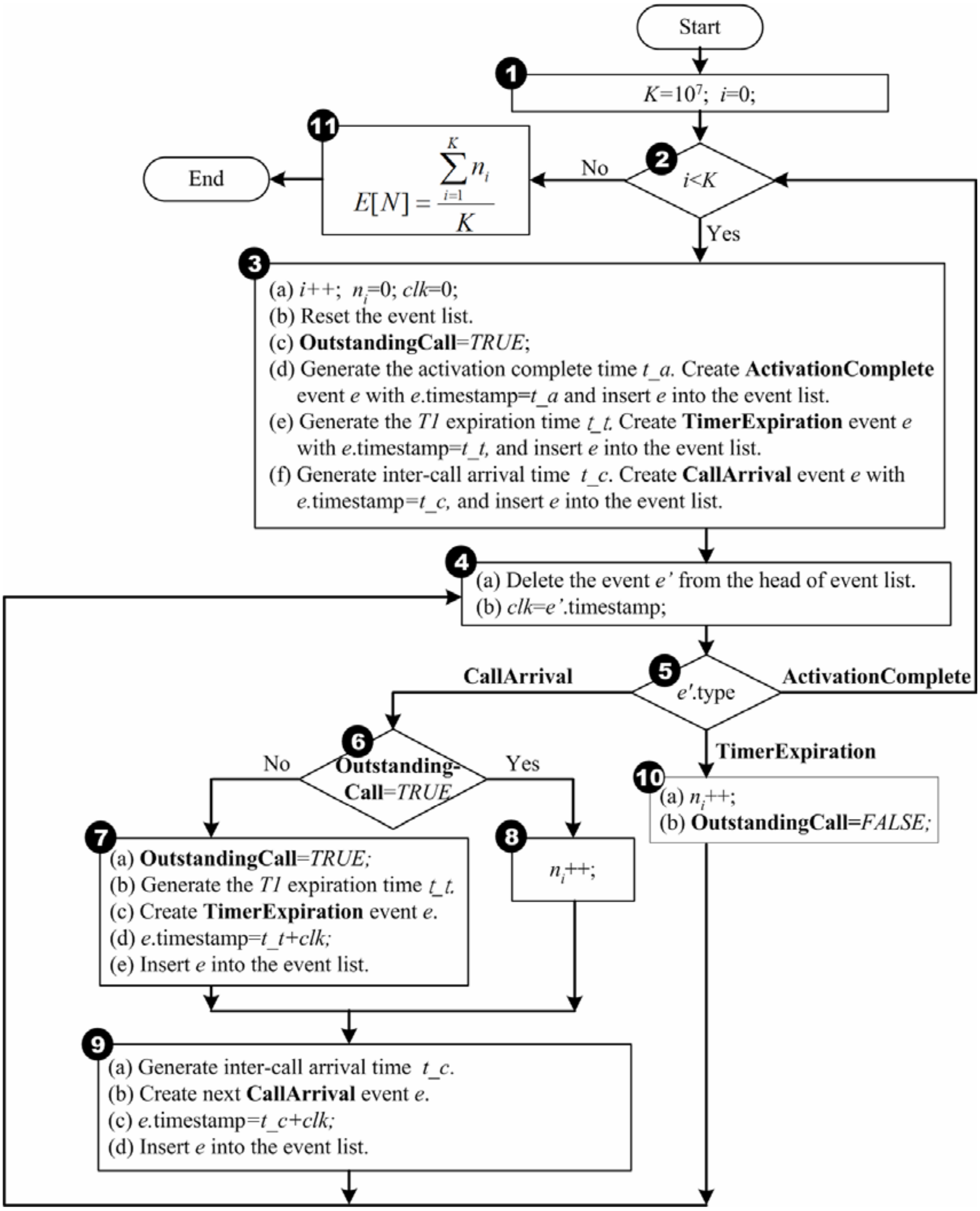

Start
1
K=10^7; i=0;
2
i<K
No
Yes
11
$E[N]=\frac{\sum_{i=1}^{K} n_i}{K}$
End
3
(a) i++; n_i=0; clk=0;
(b) Reset the event list.
(c) **OutstandingCall**=*TRUE*;
(d) Generate the activation complete time t_a. Create **ActivationComplete** event e with e.timestamp=t_a and insert e into the event list.
(e) Generate the *T1* expiration time t_t. Create **TimerExpiration** event e with e.timestamp=t_t, and insert e into the event list.
(f) Generate inter-call arrival time t_c. Create **CallArrival** event e with e.timestamp=t_c, and insert e into the event list.
4
(a) Delete the event e' from the head of event list.
(b) clk=e'.timestamp;
5
e'.type
CallArrival
ActivationComplete
TimerExpiration
6
Outstanding-Call=*TRUE*
No
Yes
7
(a) **OutstandingCall**=*TRUE;*
(b) Generate the *T1* expiration time t_t.
(c) Create **TimerExpiration** event e.
(d) e.timestamp=t_t+clk;
(e) Insert e into the event list.
8
n_i++;
9
(a) Generate inter-call arrival time t_c.
(b) Create next **CallArrival** event e.
(c) e.timestamp=t_c+clk;
(d) Insert e into the event list.
10
(a) n_i++;
(b) **OutstandingCall**=*FALSE;*

Figure 2-11. The Simulation Flowchart of Performance

the *T1* for the first outstanding call expires. The **CallArrival** event presents the arrival of second incoming call.

Steps 4 and 5. The next event *e'* is deleted from the head of the event list. The simulation clock *clk*=*e'*.timestamp. If *e'*.type=**CallArrival**, the simulation flow proceeds to Step

6. If *e'*.type=**TimerExpiration**, Step 10 is executed. If *e'*.type=**ActivationComplete**, the SIP UA of the called MS is successfully activated, and the flow goes to Step 2 to initiate another simulation iteration.

Step 6. If **OutstandingCall**=*TRUE* (i.e., there is no outstanding call in the system), go to Step 8. Otherwise, go to Step 7.

Step 7. The new incoming call becomes an outstanding call, and the flag **OutstandingCall** is set to *TRUE*. Then we generate a **TimerExpiration** event *e* to indicate the *T1* expiration for the new outstanding call. *e*.timestamp is set to *t_t+clk* where *t_t* is the *T1* expiration time. Finally, *e* is inserted into the event list.

Step 8. If **OutstandingCall**=*TRUE* at Step 6 (i.e., there exists an outstanding call in the system), the new incoming call is dropped. n_i is incremented by 1.

Step 9. Generate the next incoming call event *e*. Specifically, the inter-call arrival time *t_c* is generated, and *e*.timestamp=*t_c+clk*. The event type of *e* is **CallArrival**. Finally, *e* is inserted into the event list.

Step 10. If *e'*.type=**TimerExpiration** at Step 5, it means that the outstanding call is dropped. Therefore, **OutstandingCall** is set to *FALSE*, and n_i is incremented by 1.

Step 11. If K iterations of the simulation have been complete, *E*[*N*] is computed by using (7), and the simulation is terminated

Some simulation results are shown in Table 2-1. In all scenarios we considered, the discrepancies between the analytic model and simulation are small (less than 0.5% in all cases considered in the tables). Therefore, the accuracy of analytic model is proved.

2.4.4 Numerical Results

Based on the analytic and simulation models in the previous section, we use numerical examples to investigate the SPC performance. Based on Equation (2.5), Figure 2-12 plots *E*[*N*] against μ and λ. *E*[*N*] is an increasing function of μ and/or λ. Consider two extreme

Table 2-1. The *E*[*N*] Values (Analytic vs. Simulation)

(a) λ=0.8 γ and 0.2 γ

λ=0.8 γ				λ=0.2 γ			
μ/γ	Analytic	Simulation	Errors (%)	μ/γ	Analytic	Simulation	Errors (%)
2^{-8}	0.8000	0.7977	-0.2883	2^{-8}	0.2000	0.1994	-0.3080
2^{-7}	0.8000	0.7993	-0.0820	2^{-7}	0.2000	0.1999	-0.0360
2^{-6}	0.8000	0.8015	0.1880	2^{-6}	0.2000	0.1999	-0.0630
2^{-5}	0.8000	0.8003	0.0415	2^{-5}	0.2000	0.2002	0.1000
2^{-4}	0.8000	0.7997	-0.0328	2^{-4}	0.2000	0.2009	0.4400
2^{-3}	0.8002	0.7989	-0.1573	2^{-3}	0.2003	0.2010	0.3477
2^{-2}	0.8103	0.8115	0.1554	2^{-2}	0.2153	0.2146	-0.3445
2^{-1}	0.8800	0.8833	0.3748	2^{-1}	0.3154	0.3149	-0.1566
2^{-0}	1.0443	1.0438	-0.0522	2^{-0}	0.5266	0.5248	-0.3377
2^{1}	1.2613	1.2641	0.2243	2^{1}	0.7623	0.7617	-0.0793
2^{2}	1.4617	1.4591	-0.1802	2^{2}	0.9458	0.9448	-0.1075
2^{3}	1.6067	1.6045	-0.1338	2^{3}	1.0622	1.0612	-0.0966
2^{4}	1.6960	1.6975	0.0896	2^{4}	1.1282	1.1280	-0.0227
2^{5}	1.7460	1.7462	0.0160	2^{5}	1.1633	1.1625	-0.0724
2^{6}	1.7724	1.7712	-0.0691	2^{6}	1.1815	1.1804	-0.0910
2^{7}	1.7861	1.7868	0.0384	2^{7}	1.1907	1.1909	0.0217
2^{8}	1.7930	1.7956	0.1421	2^{8}	1.1953	1.1945	-0.0679

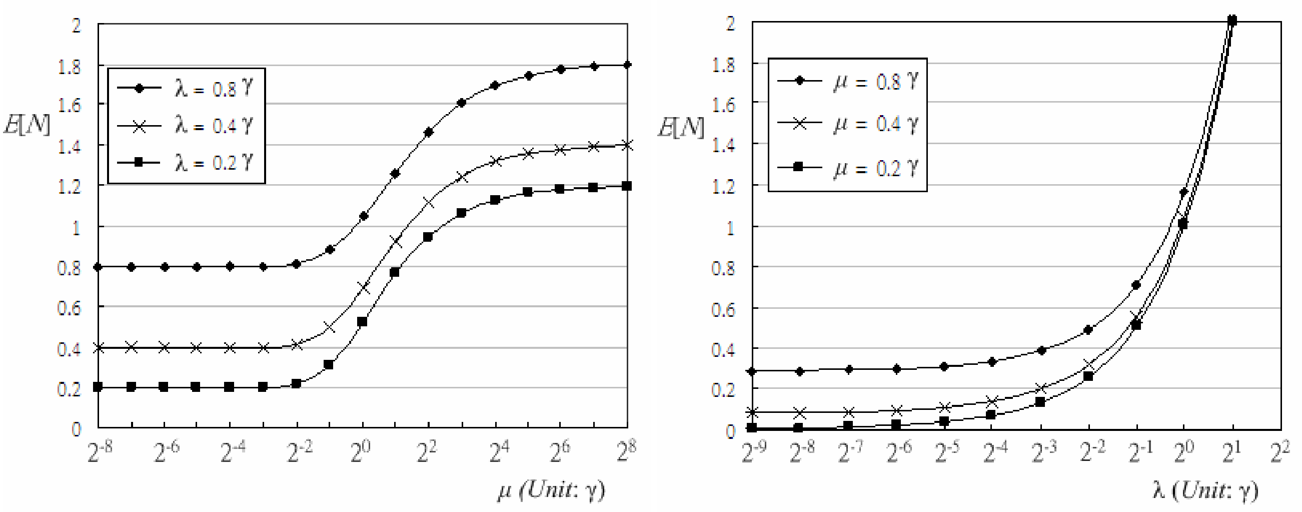
(b) μ=0.8 γ and 0.2 γ

μ=0.8 γ				μ=0.2 γ			
λ/γ	Analytic	Simulation	Errors (%)	λ/γ	Analytic	Simulation	Errors (%)
2^{-9}	0.2881	0.2875	-0.1934	2^{-9}	0.0087	0.0087	0.2763
2^{-8}	0.2896	0.2895	-0.0272	2^{-8}	0.0106	0.0106	-0.2557
2^{-7}	0.2927	0.2924	-0.1152	2^{-7}	0.0145	0.0144	-0.5971
2^{-6}	0.2990	0.2995	0.1656	2^{-6}	0.0223	0.0222	-0.1302
2^{-5}	0.3115	0.3117	0.0687	2^{-5}	0.0378	0.0376	-0.4237
2^{-4}	0.3368	0.3377	0.2744	2^{-4}	0.0688	0.0686	-0.3343
2^{-3}	0.3880	0.3884	0.0921	2^{-3}	0.1310	0.1308	-0.1128
2^{-2}	0.4931	0.4929	-0.0511	2^{-2}	0.2554	0.2560	0.2186
2^{-1}	0.7112	0.7096	-0.2239	2^{-1}	0.5045	0.5046	0.0216
2^{-0}	1.1672	1.1670	-0.0179	2^{-0}	1.0034	1.0030	-0.0383
2^{1}	2.1180	2.1207	0.1241	2^{1}	2.0023	2.0036	0.0689
2^{2}	4.0743	4.0820	0.1870	2^{2}	4.0014	4.0025	0.0275

cases where $\mu \to 0$ and $\mu \to \infty$. We have

$$\lim_{\mu\to0} E[N] = \lim_{\mu\to0}\left[\frac{\frac{\lambda}{\gamma}+\left(1-\frac{\lambda}{\gamma}\right)\times e^{-\frac{\gamma}{\mu}}}{1-\left(\frac{\lambda}{\lambda+\gamma}\right)\times e^{-\frac{\gamma}{\mu}}}\right]=\frac{\lambda}{\gamma} \quad \text{and} \quad \lim_{\mu\to\infty} E[N] = \lim_{\mu\to\infty}\left[\frac{\frac{\lambda}{\gamma}+\left(1-\frac{\lambda}{\gamma}\right)\times e^{-\frac{\gamma}{\mu}}}{1-\left(\frac{\lambda}{\lambda+\gamma}\right)\times e^{-\frac{\gamma}{\mu}}}\right]=1+\frac{\lambda}{\gamma}$$

$\mu \to 0$ indicates that the length of *T1* approaches infinite, and the first outstanding call keeps on waiting until the SIP UA activation completes. In this case, the outstanding call is always connected, and *E*[*N*] approximates to the number of subsequent incoming calls arriving during the SIP UA activation period, which is λ/γ. On the other hand, if $\mu \to \infty$, we have *T1*=0, and *T1* expires immediately after an outstanding call arrives. Therefore, the first outstanding call and all incoming calls arriving during the SIP UA activation are lost. The expected number of such calls is 1+λ/γ. These results are clearly observed in Figure 2-12 (a).



(a) Effect of μ

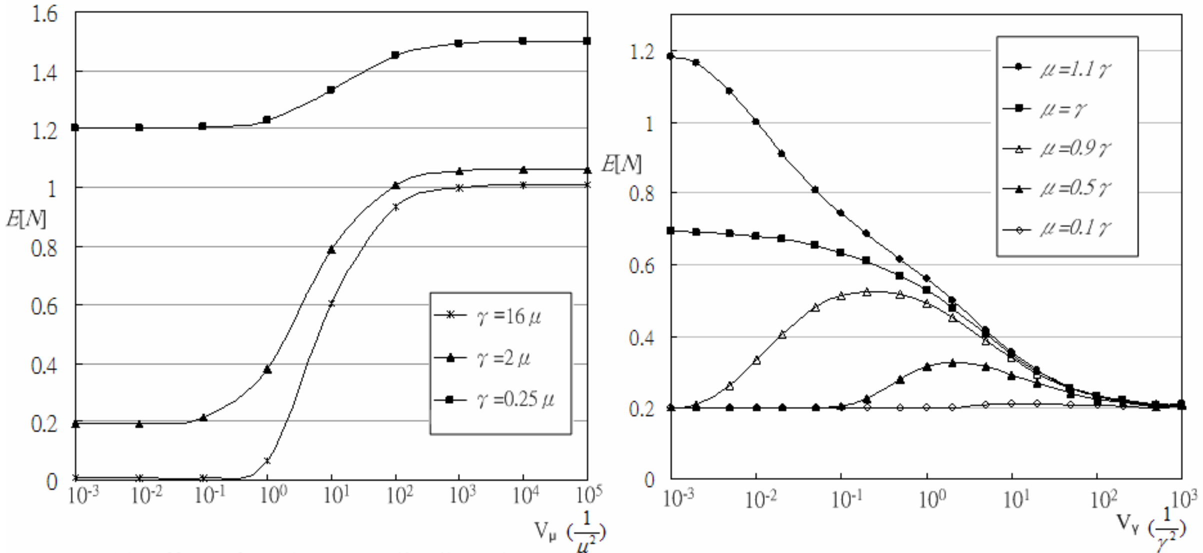
(b) Effect of λ

Figure 2-12. Effects of μ and λ on the Expected Number of Lost Calls

Figure 2-12 (a) also indicates that good $E[N]$ performance can be obtained when *T1* is set as 4/γ. Any timeout period longer than 4/γ has insignificantly improvement on the $E[N]$ performance. Figure 2-12 (b) indicates that $E[N]$ is insignificantly affected by the incoming call arrival rate λ when λ<γ/32. This result implies that to ensure good $E[N]$ performance, the SIP UA activation mechanism must be designed such that the activation time is shorter than 0.03125 times of the inter-call arrival time.

To extend the observations on the above examples, we use the discrete event simulation model to investigate the SIP UA activation performance with Gamma distributions. It has been shown that the distribution of any positive random variable can be approximated by a mixture of Gamma distributions (see Lemma 3.9 in [32]). One may also measure time periods in a real mobile network, and the measured data can be approximated by a Gamma distribution as the input to our simulation model. It suffices to use the Gamma distribution with different shape and scale parameters to represent different time distributions.

Suppose that *T1* has the Gamma distribution with mean $1/\mu$ and variance V_μ. Figure 2-13 (a) shows the effect of V_μ on $E[N]$ where λ=μ/8. The figure indicates that $E[N]$ is an increasing function of V_μ . When V_μ is very large, $E[N]$ is approximate to the $E[N]$ value when $\mu \to \infty$. This phenomenon is explained as follows. When V_μ increases, we will see significantly increasing number of short *T1* periods and insignificantly increasing number



(a) Effect of V$_{\mu}$ (Gamma distributed *T1*; $\lambda = \mu/8$)

(b) Effect of V$_{\gamma}$ (Gamma distributed activation time; $\lambda = 0.2\gamma$)

Figure 2-13. Effects of Variances V$_{\mu}$ and V$_{\gamma}$

of long *T1* periods, and the result is similar to that when *T1* is set to a small value. On the other hand, when V$_{\mu}$ is small, *T1* approaches to the fixed value $1/\mu$. Therefore, *E*[*N*] is approximate to that when $T1=1/\mu$. Figure 2-13 (a) also shows that *E*[*N*] is insignificantly affected by V$_{\mu}$ when $V_{\mu}<10^{-1}/\mu^2$ or $V_{\mu}>10^3/\mu^2$.

Consider the case when the activation time has the Gamma distribution with mean $1/\gamma$ and variance V$_{\gamma}$, and *T1* is of fixed length $1/\mu$. Figure 2-13 (b) shows the effect of V$_{\gamma}$ on *E*[*N*] where $\lambda=0.2\gamma$. As V$_{\gamma}$ increases, *E*[*N*] approaches to λ/γ. When V$_{\gamma}$ is small (e.g., $V_{\gamma}<10^{-3}/\gamma^2$), the activation time approaches to a fixed length $1/\gamma$. In this case, *E*[*N*] is insignificantly affected by V$_{\gamma}$ and is determined by the activation time and *T1*. When V$_{\gamma}$ is small, the impact of the activation time and *T1* on *E*[*N*] is described as follows.

(1) When $\mu<\gamma$ (i.e., *T1* is very likely to be longer than the activation time), there is high possibility that the first outstanding call would be connected, and all incoming calls arriving during the activation period are lost. Therefore, $E[N] = \lambda/\gamma$, which is 0.2 as correctly shown in Figure 2-13 (b).

(2) When $\mu=\gamma$ (i.e., *T1* is roughly the same as the activation time), the first outstanding call has 50% probability to be dropped. Therefore, $E[N] = 0.5+\lambda/\gamma = 0.7$ in Figure 2-13 (b).

(3) When $\mu<\gamma$ (i.e., *T1* is very likely to be shorter than the activation time), the first outstanding call is highly probably dropped, and $E[N] = 1+\lambda/\gamma = 1.2$ in Figure 2-13 (b).

When $\mu<\gamma$, $E[N]$ increases and then decrease as V_γ increases. This phenomenon is explained as follows. When V_γ increases, more long and short activation times are observed. Long activation times result in more lost calls, while short activation times cause fewer lost calls. When V_γ is small (e.g., $V_\gamma<10^{-1}/\gamma^2$ in the curve for $\mu=0.9\gamma$), the impact of long activation times is more significant. Therefore, $E[N]$ increases as V_γ increases. As V_γ becomes sufficiently large, the increase of the number of short activation times is more significant than that of long activation times. Hence, $E[N]$ is dominated by the effect of short activation times, and as V_γ increases, $E[N]$ decreases and finally approaches to λ/γ. For $\mu>\gamma$, when V_γ increases, the increase of the short activation times is more significant than that of long activation times. Therefore, $E[N]$ is a decreasing function of V_γ.

2.5 Summary

This chapter proposed WGSN, a solution for integrating 3G and WLAN services. We described how the 3G mechanisms are re-used for WLAN user authentication and network access without introducing new procedures and without modifying the existing 3G network components. We described the WGSN features and showed how they are designed and implemented. A WGSN prototype has been implemented in Industrial Technology Research Institute (ITRI) and National Chiao Tung University (NCTU) Joint Research Center. We focused on the WGSN push mechanism. An analytic model was proposed to investigate the performance of SIP push center (SPC). We measure the SPC performance by the expected number $E[N]$ of lost calls during SIP UA activation. Our study indicates that *T1* significantly affects the SPC performance. To obtain good $E[N]$ performance, the fixed *T1* period should be longer than 4 times of that for SIP UA activation, and SIP UA activation should be completed within 0.03125 times of the inter-call arrival time. We note that most callers would not await the call setup for a long time, and they would abort calls before a long *T1* expires. Therefore, it is suggested that a call should be connected within 10 seconds; that is, the SIP UA activation should be completed within 10 seconds. Or the SPC should send a message to notify the caller that the call setup will take longer time and ask the caller to be patient.

The SPC utilizes GSM short message service mechanism. According to [30], it takes 20 seconds to transmit a short message to an MS. This transmission delay significantly contributes to the SIP UA activation time and may not be acceptable. Therefore, we may further shorten the SIP UA activation time by using high priority short messages service. Short messages with high priority would be transmitted to the destination within e.g., 3-5 seconds. This improvement allows SIP UA activation to be complete within an acceptable delay.

Chapter 3

Authentication Vector Management for UMTS

In UMTS, the security function provides mutual authenticity and key agreement between the core network and the MS. Specifically, the SGSN in the core network obtains an array of *Authentication Vectors* (AVs) from the HSS/AuC, and consumes one AV for each mutual authentication. After the departure of the MS, the SGSN may keep the unused AVs for a time interval called the *Reservation Timeout* (RT) period. If the MS returns within the RT period, the SGSN uses the stored AVs for mutual authentication instead of obtaining new AVs from the HSS/AuC. Note that a long RT period results in fewer accesses to the HSS/AuC at the cost of extra AV storage in the SGSN. In this chapter, we propose an analytic model to investigate the impact of the RT period on the system performance. Our study provides the guidelines for the mobile operators to select an appropriate RT period.

3.1 UMTS Authentication Vector Management

When an MS enters an SGSN area, the UMTS authentication procedure is performed. The SGSN obtains an array of AVs from the HSS/AuC, and uses one AV for each of authentication procedure. Detailed message flow of the UMTS authentication procedure is described in Section 1.3. Note that Steps G.2 and G.3 in Figure 1-4 are called the *Authentication Data Request & Response* (ADR) operation. Steps G.4 and G.5 in Figure 1-4 are called the *User Authentication Request & Response* (UAR) operation. In the following subsections, we will elaborate two mechanisms of managing AVs defined in UMTS, and also present the motivation of this chapter.

3.1.1 Sequence Number Mechanism

To ensure that one AV is only used for one authentication and key agreement, the sequence number mechanism is used in AV generation. In HSS/AuC, a counter $SQN_{HE,}$ which is individual for each user, records the sequence number used in the previous AV generation. In the MS, the counter SQN_{MS} denotes the highest sequence number the MS has accepted. When the HSS/AuC generates an AV for the MS, it first generates the next sequence number SQN based on $SQN_{HE,}$ and uses SQN as an input parameter to generate the AV. Then the HSS/AuC set SQN_{HE}=SQN. In one ADR operation, the HSS/AuC may send the SGSN a batch of AVs, each of which has different sequence number. The SGSN utilizes the AVs in the sequence number order. When an AV is selected to perform the UMTS authentication, the SQN is concealed in AUTN and sent to the MS in Step G.4 of Figure 1-4. The MS retrieves SQN by RAND and the preshared key, and then verify if SQN is in the valid range. If SQN > SQN_{MS} and (SQN-SQN_{MS}) < Δ, where Δ is a valid difference defined in 3GPP TS 33.102, the authentication procedure proceeds and SQN_{MS}=SQN. Otherwise, the MS abandons the UMTS authentication procedure. The re-synchronization procedure is then activated to synchronize SQN_{MS} and SQN_{HE} and request the HSS/AuC to generate new AVs for the MS.

3.1.2 Array Mechanism

As defined in 3GPP TS 33.102 [10], when the MS moves from one SGSN area to another, the unused AVs are forwarded from the previously visited SGSN to the newly visited SGSN. This procedure requires secure links between SGSNs. However, it may not be practical in reality, especially when SGSNs are owned by different service providers or even different countries. In this case, the AVs are not forwarded to the new SGSN. Instead, the old SGSN stores the unused AVs for a time period called *Reservation Timeout* (RT) period. If the MS returns to the SGSN area within the RT period, the SGSN will utilize these stored AVs for authentication instead of obtaining new AVs from the HSS/AuC. Therefore, the signaling traffic for accessing the HSS/AuC is reduced. In this dissertation, this method is referred to as AV usage mechanism.

The AV usage mechanism may not succeed owing to the sequence number mechanism described in Section 3.1.1. For example, the MS originally resides in SGSN area 1 (Figure 1-2(13)), and then moves to SGSN area 2 (Figure 1-2 (14)). After the MS leaves the SGSN area 1, the unused AVs are stored in SGSN1. Then, SGSN2 executes the ADR operation to obtain new AVs from the HSS/AuC, and authenticates the MS. After a successful authentication, the MS sets SQN_{MS} as the sequence number of the new AV from SGSN2. Since the sequence numbers of the new AVs are greater than those of the AVs stored in SGSN1, SQN_{MS} becomes greater than the sequence numbers of AVs stored in SGSN1. Therefore, if the MS moves back to SGSN area 1, the authentication procedure with the unused AVs will be abandoned by the MS due to the failure of the sequence number validation.

In 3GPP TS 33.102, an array mechanism is utilized to solve the problem described above. In the array mechanism, the sequence number SQN is divided into two parts: SEQ is the most significant bits of SQN, and IND is the last 5 bits of SQN. In one ADR operation, the HSS/AuC uses the same IND value IND1 but different SEQ values to generate a batch of AVs for SGSN1. For SGSN2, the HSS/AuC uses another IND value IND2 to generate AVs. The MS has 32 sequence number counters, which have the same length as SEQ, and

$SEQ_{MS}(i)$ is denoted as the i-th sequence number counter in the MS. After a successful authentication with SGSN1, the MS records the sequence number in SEQ_{MS}(IND1). Similarly, when the MS moves to SGSN area 2, the highest sequence number of the AVs sent from SGSN2 is recorded in SEQ_{MS}(IND2). When the MS moves back to SGSN area 1, the SGSN1 authenticate the MS by using the stored AVs with sequence number SQN = SEQ | IND1. The MS validates SQN by checking if SEQ > SEQ_{MS}(IND1) and (SEQ-SEQ_{MS}(IND1)) > Δ. The validation will succeed because the largest sequence numbers of AVs from different SGSNs are recorded in different counters.

3.1.3 Motivation

Although the AV usage mechanism reduces the access traffic to the HSS/AuC, it also consumes extra storage to keep the unused AVs at the SGSN, and the HSS/AuC needs to maintain the relation of an IND value and its correspondent SGSN area. Note that a long RT period results in fewer accesses to the HSS/AuC at the cost of more AV storage required at an SGSN. Therefore, it is desirable to selecting an appropriate RT period to reduce the access to the HSS/AuC without consuming too much extra storage in the SGSN. In the following sections, we will investigate the effect of the RT period on the system performance. In Section 3.2, the AV usage mechanism is described. Section 3.3 presents an analytic model for measuring system performance. Section 3.4 shows the simulation model to validate against the analytic results. Numerical examples are given in Section 3.5.

3.2 The AV Usage Mechanism

Consider an SGSN area L_0. When an MS resides at L_0, the authentication activities are shown in Figure 3-1.

In this figure, the MS enters L_0 at time τ_1 (Figure 3-1 (1)), and sends a registration request to L_0 at time $\tau_{1,1,1}$. This registration request activates a UAR for mutual authentication

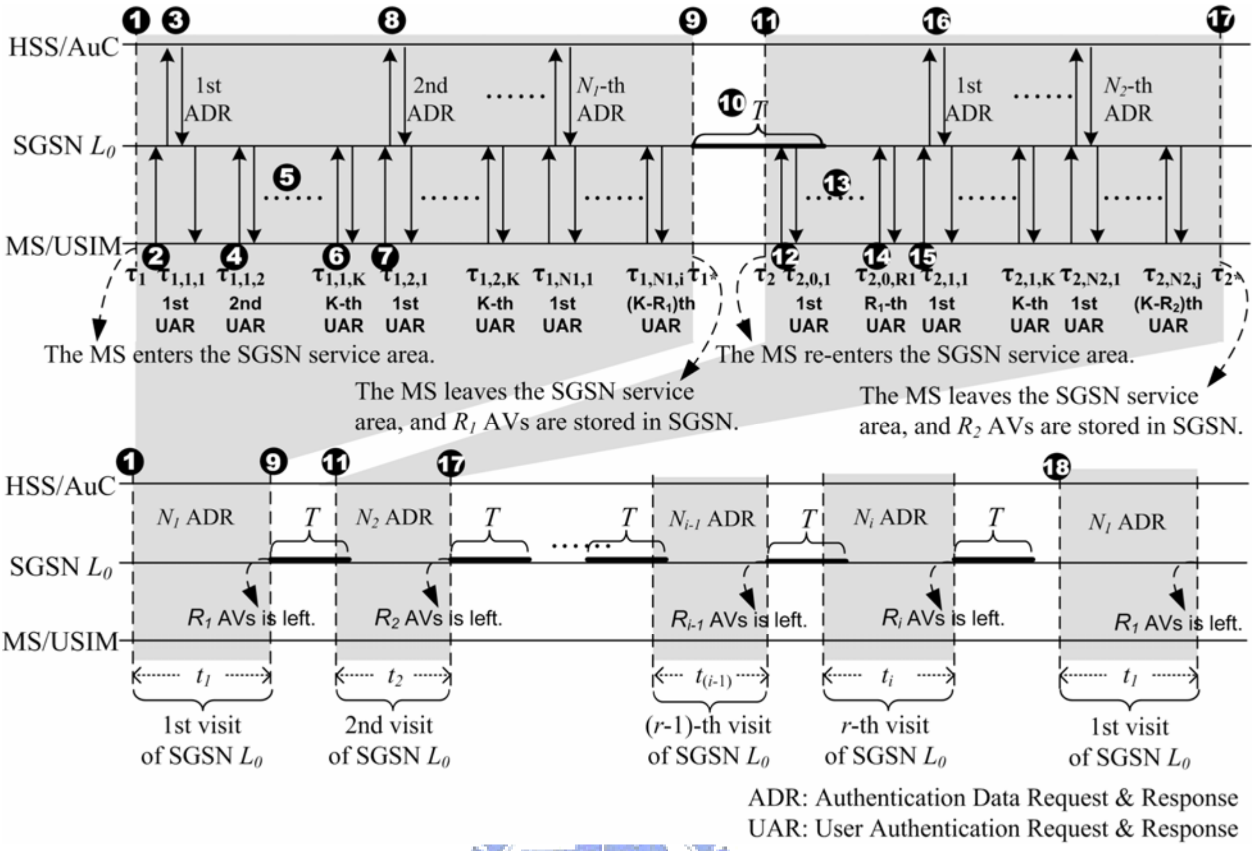

HSS/AuC
SGSN L_0
MS/USIM
1st ADR
2nd ADR
N_1-th ADR
T
1st ADR
N_2-th ADR
τ_1 $\tau_{1,1,1}$ $\tau_{1,1,2}$ $\tau_{1,1,K}$ $\tau_{1,2,1}$ $\tau_{1,2,K}$ $\tau_{1,N1,1}$ $\tau_{1,N1,i}$ τ_1^*
1st UAR
2nd UAR
K-th UAR
1st UAR
K-th UAR
1st UAR
(K-R_1)th UAR
τ_2 $\tau_{2,0,1}$ $\tau_{2,0,R1}$ $\tau_{2,1,1}$ $\tau_{2,1,K}$ $\tau_{2,N2,1}$ $\tau_{2,N2,j}$ τ_2^*
1st UAR
R_1-th UAR
1st UAR
K-th UAR
1st UAR
(K-R_2)th UAR
The MS enters the SGSN service area.
The MS re-enters the SGSN service area.
The MS leaves the SGSN service area, and R_1 AVs are stored in SGSN.
The MS leaves the SGSN service area, and R_2 AVs are stored in SGSN.
HSS/AuC
SGSN L_0
MS/USIM
N_1 ADR
T
N_2 ADR
T
T
N_{i-1} ADR
T
N_i ADR
T
N_1 ADR
R_1 AVs is left.
R_2 AVs is left.
R_{i-1} AVs is left.
R_i AVs is left.
R_1 AVs is left.
t_1
t_2
$t_{(i-1)}$
t_i
t_1
1st visit of SGSN L_0
2nd visit of SGSN L_0
(r-1)-th visit of SGSN L_0
r-th visit of SGSN L_0
1st visit of SGSN L_0
ADR: Authentication Data Request & Response
UAR: User Authentication Request & Response

Figure 3-1. MS Authentication Activities at an SGSN

between the MS and SGSN L_0 (Figure 3-1 (2)). Since L_0 does not have authentication information of the MS at its first visit, SGSN L_0 obtains an array of K AVs from the HSS/AuC through an ADR (Figure 3-1 (3)), and utilizes the first AV for the UAR request [1]. Subsequently, more UARs may be issued by the MS, and SGSN L_0 utilizes the next unused AVs in the array to perform the following UARs (Figure 3-1 (4), (5), and (6)). After $\tau_{1,1,k}$, all AVs have been consumed for UARs (Figure 3-1 (6)). Therefore, when a UAR arrives at $\tau_{1,2,1}$ (Figure 3-1 (7)), SGSN L_0 issues the second ADR (Figure 3-1 (8)) to obtain the next AV array from the HSS/AuC and uses the first AV in the array to perform the UAR. At time τ_1^* (Figure 3-1 (9)), the MS leaves L_0 with R_1 unused AVs. Then SGSN L_0 starts the RT timer of length T (Figure 3-1 (10)), and keeps the R_1 unused AVs in its storage during the RT period. If the MS returns to L_0 before the RT timer expires (Figure 3-1 (11)), SGSN L_0 will utilize these stored AVs for the next R_1 UARs (Figure 3-1 (12), (13), and (14)). After the R_1 AVs are consumed, a UAR occurs at time $\tau_{2,1,1}$ (Figure 3-1 (15)), and SGSN L_0 issues an ADR to obtains a new AV array (Figure 3-1 (16)). At time τ_2^*, the MS leaves L_0 again

(Figure 3-1 (17)). Let the residence time of the i-th visit to L_0 be t_i. During t_i, N_i ADRs are executed. When the MS leaves L_0, there are R_i unused AVs, which will be subsequently used at the (i+1)-th visit of the MS. Note that if the RT timer expires before the MS returns to L_0 (Figure 3-1 (18)), then these unused AVs are discarded; that is, $R_i = 0$ for the (i+1)-th visit.

3.3 Analytic Model

This section investigates the effect of the RT period T on the performance of the AV management. The following parameters and assumptions are made.

- The UAR arrivals are a Poisson process with rate λ .
- The SGSN residence time is exponentially distributed with rate μ (this exponential assumption will be relaxed in the simulation experiments).

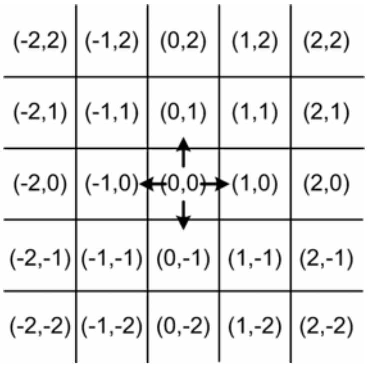
Three output measures are evaluated in our study.

- α: the probability that the MS re-enters L_0 within the RT period T
- β: the expected AV storage consumed when T>0, which is normalized by the expected AV storage consumed when T=0
- δ: the number of ADRs performed in one visit to SGSN L_0 as comparing with that when K=1. Let $E[N|K]$ be the expected number of ADRs performed in one visit to $L_{0,}$ where K AVs are obtained in one ADR. Then

$$\delta = \frac{E[N \mid K]}{E[N \mid K = 1]} . \tag{3.1}$$

In the following sub-sections, we derive the above output measures.

3.3.1 Derivation of Probability α



(-2,2)	(-1,2)	(0,2)	(1,2)	(2,2)
(-2,1)	(-1,1)	(0,1)	(1,1)	(2,1)
(-2,0)	(-1,0)	(0,0)	(1,0)	(2,0)
(-2,-1)	(-1,-1)	(0,-1)	(1,-1)	(2,-1)
(-2,-2)	(-1,-2)	(0,-2)	(1,-2)	(2,-2)

Figure 3-2. Two-dimensional SGSN Layout for the Random Walk Model

We utilize a two-dimensional random walk to model the MS movement. Figure 3-2 shows the layout of the SGSN areas, where a coordinate (x,y) specifies the location of an SGSN area. An MS resides in an SGSN area for a period, then moves to one of its four neighbors with the same probability 1/4. Let $L_j = (x_j, y_j)$ presents the coordinate of the SGSN where the MS resides after j movements. Initially, the MS resides at $L_0 = (0,0)$.

Let P_j be the probability that the MS returns to L_0 at the j-th movement. That is, $P_j = \Pr[\ L_j = L_0\ ]$. Following the two-dimensional random walk model, $P_j = 0$ if j is odd. Consider the even movements where $j = 2n$. Assume that there are a movements to the east and the same number of movements to the west. Similarly, there are b movements to the north and the same number of movements to the south. Then $L_{2n} = L_0$ if $2a+2b = 2n$. For $a \geq 0$ and $b \geq 0$, we have

$$P_{2n} = \sum_{a,b,a+b=n} \left(\frac{1}{4}\right)^{a+a+b+b} \frac{(2n)!}{a!a!b!b!}$$

$$= \left(\frac{1}{4}\right)^{2n} \binom{2n}{n}^2 \qquad (3.2)$$

Let Q_{2n} be the probability that the MS first returns to L_0 at the $2n$-th movement. In other words, $Q_{2n} = \Pr[\ L_{2n} = L_0, L_{2l} \neq L_0$ for $0<l<n\ ]$. It is obvious that for n=1, $Q_2 = P_2$.

Suppose that the MS enters L_0 at the $2n$-th movement, and the prior visit to L_0 occurs at the

$2m$-th movement, where $0 \leq m < n$. Then we have

$$\begin{aligned} P_{2n} &= \Pr[L_{2n} = L_0] \\ &= \sum_{m=0}^{n-1} \Pr[L_{2m} = L_0] \times \Pr[L_{2n} = L_{2m} = L_0, L_{2l} \neq L_0 \text{ for } m < l < n] \\ &= \sum_{m=0}^{n-1} P_{2m} \times Q_{2(n-m)} \end{aligned} \tag{3.3}$$

By rearranging (3.3), we have

$$Q_{2n} \quad = \quad P_{2n} - \sum_{m=1}^{n-1} P_{2m} Q_{2(n-m)} \tag{3.4}$$

Since the MS returns to L_0 at the $2n$-th movement, the MS moves across $2n$-1 SGSN areas before it returns. For j=1,2,…,$2n$-1, let the residence time at L_j be t^*_j with the density function $f(t^*_j) = \mu e^{-\mu t^*_j}$. Let t_{r} be the period between when the MS leaves L_0 and when it returns. That is $t_{\mathrm{r}} = t^*_1 + t^*_2 + \ldots + t^*_{2n-1}$, where $L_{2n} = L_0$, and $L_{2l} \neq L_0$ for $0<l<n$. Let $F(2n, t_{\mathrm{r}})$ be the cumulative distribution function that the MS returns to L_0 at $2n$-th movement at time t_{r}. It is clear that the $F(2n, t_{\mathrm{r}})$ is an Erlang distribution

$$F(2n, t_r) = 1 - \sum_{j=0}^{2n-2} \left[\frac{(\mu t_r)^j}{j!} \right] e^{-\mu t_r} \tag{3.5}$$

From (3.4) and (3.5), α is derived as follows.

$$\begin{aligned} \alpha &= \sum_{n=1}^{\infty} \{ \Pr[\text{the MS first returns to } L_0 \text{ at the } 2n\text{ - th movement}] \\ &\qquad \times \Pr[\text{the MS moves } 2n \text{ steps within time } T] \} \\ &= \sum_{n=1}^{\infty} Q_{2n} \times F(2n, T) \\ &= \sum_{n=1}^{\infty} Q_{2n} \times \left\{ 1 - \sum_{j=0}^{2n-2} \left[\frac{(\mu T)^j}{j!} \right] e^{-\mu T} \right\} \end{aligned} \tag{3.6}$$

3.3.2 Derivation for *β*

Consider the Markov chain illustrated in Figure 3-3, where state S_k represents that there are

k AVs stored in SGSN L_0. In this figure, the transition probability in a short observation interval Δs is considered. The descriptions of the transitions are given below.

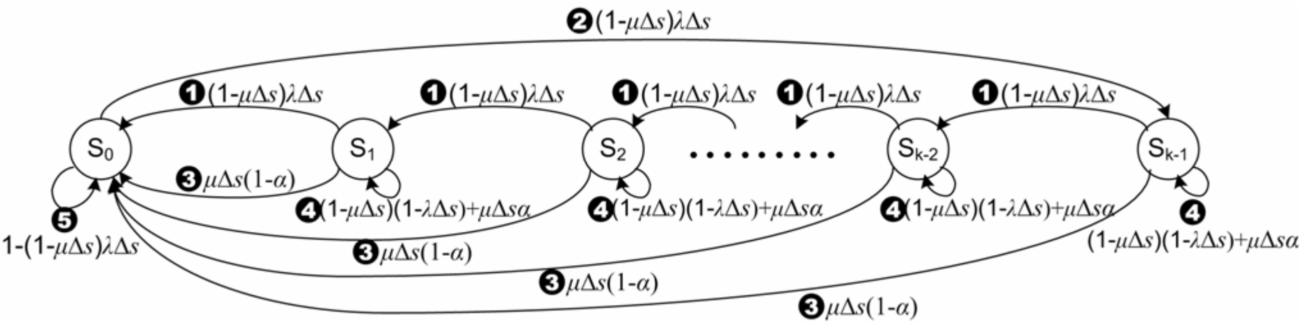


Figure 3-3. The State Transition Diagram for the AVs Size at SGSN L_0

Transition 1: At state S_k (k>0), the MS resides in L_0 with probability (1-$\mu\Delta s$). A UAR arrives with probability $\lambda\Delta s$, which decrements the number of AVs by one. That is, the Markov chain moves from S_k to S_{k-1} with probability (1-$\mu\Delta s$)$\lambda\Delta s$.

Transition 2: At S_0, the MS resides at L_0 with probability (1-$\mu\Delta s$), and a UAR occurs with probability $\lambda\Delta s$. Since k=0, SGSN L_0 issues an ADR to obtain K AVs from the HSS/AuC, and uses the first AV to perform the UAR. The remaining K-1 AVs are stored in SGSN L_0. Therefore, the Markov chain moves from S_0 to S_{K-1} with probability (1-$\mu\Delta s$)$\lambda\Delta s$.

Transition 3: At state S_k (k>0), the MS leaves L_0 with probability $\mu\Delta s$, and it does not return to L_0 within T with probability (1-α). In this case, all AVs stored in SGSN L_0 are discarded. That is, the Markov chain moves from S_k to S_0 with probability $\mu\Delta s$(1-α).

Transition 4: At state S_k (k>0), the MS stays in L_0 with probability (1-$\mu\Delta s$), and the probability that no UAR occurs during Δs is (1-$\lambda\Delta s$). In this case, the number of AVs stored in SGSN L_0 is not changed. Also, if the MS leaves L_0 with probability $\mu\Delta s$ and returns before RT expires with probability α, all unused AVs are still stored in SGSN $L_{0.}$ Thus, the state remains in S_k with probability (1-$\mu\Delta s$)(1-$\lambda\Delta s$)+ $\mu\Delta s\alpha$.

Transition 5: At S_0, SGSN L_0 does not keep any AVs, and the state remains in S_0 with probability 1-(1-$\mu\Delta s$)$\lambda\Delta s$.

Based on the above transitions, the transition probability matrix $H(\Delta s)$ is expressed as

$$H(\Delta s)=\begin{bmatrix}
1-(1-\mu\Delta s)\lambda\Delta s & 0 & 0 & \cdots & 0 & (1-\mu\Delta s)\lambda\Delta s \\
\begin{array}{c}(1-\mu\Delta s)\lambda\Delta s\\+\mu\Delta s(1-\alpha)\end{array} & \begin{array}{c}(1-\mu\Delta s)(1-\lambda\Delta s)\\+\mu\Delta s\alpha\end{array} & 0 & & 0 & 0 \\
\mu\Delta s(1-\alpha) & (1-\mu\Delta s)\lambda\Delta s & \begin{array}{c}(1-\mu\Delta s)(1-\lambda\Delta s)\\+\mu\Delta s\alpha\end{array} & & 0 & 0 \\
\mu\Delta s(1-\alpha) & 0 & (1-\mu\Delta s)\lambda\Delta s & & 0 & 0 \\
\mu\Delta s(1-\alpha) & 0 & 0 & & 0 & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
\mu\Delta s(1-\alpha) & 0 & 0 & & \begin{array}{c}(1-\mu\Delta s)(1-\lambda\Delta s)\\+\mu\Delta s\alpha\end{array} & 0 \\
\mu\Delta s(1-\alpha) & 0 & 0 & \cdots & (1-\mu\Delta s)\lambda\Delta s & \begin{array}{c}(1-\mu\Delta s)(1-\lambda\Delta s)\\+\mu\Delta s\alpha\end{array}
\end{bmatrix}_{K\times K}$$

The transition rate matrix $H=\lim_{\Delta s\to 0}\dfrac{H(\Delta s)-I}{\Delta s}$, where I=[$q_{x,y}$] is an identity matrix (i.e., $q_{x,y}$ = 1 if $x = y$, otherwise $q_{x,y}$ = 0). That is,

$$H=\begin{bmatrix}
-\lambda & 0 & 0 & \cdots & 0 & \lambda \\
\lambda+\mu-\mu\alpha & \mu\alpha-\mu-\lambda & 0 & & 0 & 0 \\
\mu(1-\alpha) & \lambda & \mu\alpha-\mu-\lambda & & 0 & 0 \\
\mu(1-\alpha) & 0 & \lambda & & 0 & 0 \\
\mu(1-\alpha) & 0 & 0 & & 0 & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
\mu(1-\alpha) & 0 & 0 & & \mu\alpha-\mu-\lambda & 0 \\
\mu(1-\alpha) & 0 & 0 & \cdots & \lambda & \mu\alpha-\mu-\lambda
\end{bmatrix}_{K\times K}$$

Let π = (π_0, π_1, π_2, …, π_{K-1}) be the probability matrix where π_k is the probability that k AVs are stored in SGSN L_0 at the steady state. Since π $H=0$, we have

$$\left.\begin{array}{r}
-\lambda\pi_0+\lambda\pi_1+\mu(1-\alpha)(\pi_1+\pi_2+\cdots+\pi_{K-1})=0\\
(\mu\alpha-\mu-\lambda)\pi_1+\lambda\pi_2=0\\
(\mu\alpha-\mu-\lambda)\pi_2+\lambda\pi_3=0\\
\vdots\qquad\qquad\\
(\mu\alpha-\mu-\lambda)\pi_{K-2}+\lambda\pi_{K-1}=0\\
(\mu\alpha-\mu-\lambda)\pi_{K-1}+\lambda\pi_0=0
\end{array}\right\}\qquad(3.7)$$

By rearranging (3.7), we have

$$\pi_k = A^{K-k}\pi_0 \quad \text{for } 1 \le k \le K\text{-1, where } A = \frac{\lambda}{\mu + \lambda - \mu\alpha} \tag{3.8}$$

By solving (3.8) with $\sum_{i=0}^{K-1} \pi_i = 1$, we have

$$\left.\begin{aligned} \pi_0 &= \frac{1-A}{1-A^K} \\ \pi_k &= A^{K-k}\left(\frac{1-A}{1-A^K}\right) \quad \text{for } 1 \le k \le K-1 \end{aligned}\right\} \tag{3.9}$$

Let ψ_T be the expected AV storage consumed in one visit to L_0 when $T \geq 0$. Then

$$\begin{aligned} \psi_T = {} & E[\text{the number of AVs stored in SGSN } L_0 \text{ at any time}] \\ & \times E[\text{the time period that SGSN } L_0 \text{ stores AVs}] \end{aligned}$$

where

$$\begin{aligned} & E[\text{the number of AVs stored in SGSN } L_0 \text{ at any time}] \\ & = \sum_{k=0}^{K-1} k\pi_k \\ & = A^K \pi_0 \left(\sum_{k=1}^{K-1} kA^{-k} \right) \\ & = \left(\frac{1-A}{1-A^K}\right)\left[\frac{A^{K-1}-1}{\left(1-A^{-1}\right)^2} + \frac{1-K}{1-A^{-1}}\right] \end{aligned} \tag{3.10}$$

As previously defined, t_r is the period between when the MS leaves L_0 and when it returns. We have

$$\begin{aligned} & E[\text{the period that SGSN } L_0 \text{ stores AVs}] \\ & = E[\text{the period that the MS resides in } L_0] \\ & \quad + E[\text{the period that SGSN } L_0 \text{ stores the unused AVs after the MS leaves } L_0] \\ & = E[\text{the period that the MS resides in } L_0] + \Pr[t_r > \text{T}] \times E[T \mid t_r > T] + \Pr[t_r \le \text{T}] \times E[t_r \mid t_r \le T] \\ & = \frac{1}{\mu} + (1-\alpha)T + \int_{t=0}^{T} t \left[\frac{d\left(\sum_{n=1}^{\infty} Q_{2n} \times F(2n,t) \right)}{dt} \right] dt \\ & = \frac{1}{\mu} + (1-\alpha)T + \int_{t=0}^{T} t \left\{ \sum_{n=1}^{\infty} Q_{2n} \times \left[\frac{\mu^{2n-1} t^{2n-2}}{(2n-2)!} \right] e^{-\mu t} \right\} dt \\ & = \frac{1}{\mu} + (1-\alpha)T + \sum_{n=1}^{\infty} \left\{ \left[\frac{Q_{2n}(2n-1)}{\mu} \right] \left\{ 1 - e^{-\mu T} \left[1 + \sum_{r=0}^{2n-2} \frac{(\mu T)^{2n-r-1}}{(2n-r-1)!} \right] \right\} \right\} \end{aligned} \tag{3.11}$$

From (3.10) and (3.11),

$$\psi_T = \Bigg\{\left(\frac{1-A}{1-A^K}\right)\Bigg[\frac{A^{K-1}-1}{\left(1-A^{-1}\right)^2}+\frac{1-K}{1-A^{-1}}\Bigg]\Bigg]\Bigg\}$$

$$\times\Bigg\{\frac{1}{\mu}+\left(1-\alpha\right)T+\sum_{n=1}^{\infty}\Bigg\{\Bigg[\frac{Q_{2n}(2n-1)}{\mu}\Bigg]\Bigg\{1-e^{-\mu T}\Bigg[1+\sum_{r=0}^{2n-2}\frac{(\mu T)^{2n-r-1}}{(2n-r-1)!}\Bigg]\Bigg]\Bigg\}\Bigg\}\quad\Bigg\} \tag{3.12}$$

From (3.12), β is derived as

$$\beta = \frac{\psi_T}{\psi_0} \tag{3.13}$$

3.3.3 Derivation for δ

Suppose that K AVs are obtained in one ADR. The expected number $E[N|K]$ of ADRs performed in one visit to L_0 is derived as follows. For $i \geq 1$, let $\Theta(N_i, R_{i-1}, R_i, t_i)$ be the probability that

(i) at the i-th visit to L_0, R_{i-1} unused AVs are stored in SGSN L_0, where R_0=0 and $0 \leq R_{i-1} < K$,

(ii) the residence time of the i-th visit to L_0 is t_i,

(iii) during t_i, N_i ADRs occur, and

(iv) there are R_i unused AVs when the MS leaves L_0, where $0 \leq R_i < K$.

Since N_i =0 has no effect on $E[N_i]$, it suffices to consider N_i >0 in the derivation. In this case, $N_iK+R_{i-1}- R_i$ UARs are performed in the period t_i, and $N_iK+R_{i-1}- R_i$>0. Therefore

$$\theta(N_i, R_{i-1}, R_i, t_i) = e^{-\lambda t_i}\left[\frac{(\lambda t_i)^{N_iK+R_{i-1}-R_1}}{(N_iK+R_{i-1}-R_i)!}\right]$$

Let $\varphi(N_i, R_{i-1})$ be the probability that when the MS enters L_0 at the i-th visit, R_{i-1} unused AVs are stored in SGSN L_0, and N_i ADRs are performed during the residence time of the i-th visit. For $N_i > 0$, $\varphi(N_i,R_{i-1})$ is derived as

$$
\begin{aligned}
&\varphi(N_i, R_{i-1}) \\
&= \sum_{R_i=0}^{K-1}\left[\int_{t_i=0}^{\infty} \theta(N_i, R_{i-1}, R_i, t_i) \times f(t_i)\, dt_i\right] \\
&= \sum_{R_i=0}^{K-1}\left\{\int_{t_i=0}^{\infty} e^{-\lambda t_i}\left[\frac{(\lambda t_i)^{N_i K + R_{i-1} - R_i}}{(N_i K + R_{i-1} - R_i)!}\right] f(t_i)\, dt_i\right\} \\
&= \left(\frac{\lambda}{\lambda+\mu}\right)^{N_i K + R_{i-1} + 1}\left[\left(\frac{\lambda+\mu}{\lambda}\right)^{K} - 1\right] \qquad (3.14)
\end{aligned}
$$

Let $\Gamma(N_i, T)$ be the probability that N_i ADRs are performed at the i-th visit to L_0, where $i \geq 1$ and $N_i > 0$, and the length of the RT period is T. Consider the following two cases:

Case 1: The MS re-enters L_0 within T. Thus, SGSN L_0 still stores R_{i-1} unused AVs, where $0 \leq R_{i-1} < K$. The probability of Case 1 is α.

Case 2: The MS re-enters L_0 after the RT timer expires, and $R_{i-1} = 0$. The probability of Case 2 is (1-α).

Then we have

$$
\Gamma(N_i, T) = \alpha\left[\sum_{k=0}^{K-1} \pi_k \varphi(N_i, k)\right] + (1-\alpha)\,\varphi(N_i, 0) \qquad (3.15)
$$

The first term of the right hand side in (3.15) is derived as follows. From (3.9) and (3.14)

$$
\begin{aligned}
&\alpha\left[\sum_{k=0}^{K-1} \pi_k \varphi(N_i, k)\right] \\
&= \alpha\left(\frac{1-A}{1-A^K}\right)\left(\frac{\lambda}{\lambda+\mu}\right)^{N_i K+1}\left[\left(\frac{\lambda+\mu}{\lambda}\right)^{K} - 1\right]\left\{1 + A^K\left[\frac{\lambda}{A(\lambda+\mu)-\lambda}\right]\left[1 - \left(\frac{\lambda}{A(\lambda+\mu)}\right)^{K-1}\right]\right\} \\
&\qquad (3.16)
\end{aligned}
$$

The second term of the right hand side in (3.15) is derived as follows.

$$
(1-\alpha)\,\varphi(N_i, 0) = (1-\alpha)\left(\frac{\lambda}{\lambda+\mu}\right)^{N_i K+1}\left[\left(\frac{\lambda+\mu}{\lambda}\right)^{K} - 1\right] \qquad (3.17)
$$

Let $E[N|K]=\lim_{i\to\infty} E[N_i \mid K]$ be the expected number of ADRs performed during one visit to SGSN L_0 at the steady state (i.e., when $i\to\infty$). From (3.15), (3.16), and (3.17), $E[N|K]$ is expressed as:

$$
\begin{aligned}
E[N \mid K] &= \sum_{N=1}^{\infty} N \times \Gamma(N,T) \\
&= \left\{ \frac{\alpha\left(\frac{\lambda}{\lambda+\mu}\right)\left(\frac{1-A}{1-A^K}\right)}{1-\left(\frac{\lambda}{\lambda+\mu}\right)^K} \right\} \left\{ 1+\left[\frac{\lambda A^K}{A(\lambda+\mu)-\lambda}\right]\left\{1-\left[\frac{\lambda}{A(\lambda+\mu)}\right]^{K-1}\right\}\right\} \\
&\quad + \frac{1}{\left[1-\left(\frac{\lambda}{\lambda+\mu}\right)^K\right]^2}\left\{(1-\alpha)\left(\frac{\lambda}{\lambda+\mu}\right)\left[1-\left(\frac{\lambda}{\lambda+\mu}\right)^K\right]\right\} \qquad (3.18)
\end{aligned}
$$

Finally, δ is derived from (3.1) and (3.18).

3.4 Simulation Model

In this section, we utilize discrete event simulation experiments to validate the analytic model described in the previous section. In the simulation, four types of events are defined: **MSUAR** represents that a UAR occurs, **MSLeave** indicates that the MS leave L_0, **RTExpire** represents that the RT timer expires, and **MSMove** indicates that the MS moves to a new SGSN area after leaving L_0. The type of an event *e* is denoted by *e*.type, and the timestamp of *e* is denoted by *e*.timestamp. All events are first inserted into an event list, and then processed in the non-decreasing timestamp order. The inter UAR arrival times are produced from a random number generator with mean $1/\lambda$. The SGSN residence times are drawn from a random number generator with mean $1/\mu$. We first use exponential random number to validate against the analytic model. Then we use Gamma random number to investigate the impact of general SGSN residence times.

To ensure that the results are stable, we simulate *MaxVisit*=5x10^7 visits to L_0. In the simulation process, k counts the number of AVs stored in SGSN L_0, and *AVCreateTime* records the timestamp when the currently stored AVs are obtained from the HSS/AuC. The counter *UARCount* records the number of UARs performed, *ADRCount* records the number of ADRs performed, *VisitCount* records the number of visits that the MS returns to L_0 before the RT timer expires, and *StorePeriod* calculates the total time periods in which AVs are stored in SGSN L_0. The above variables are used to compute the final measures as follows.

$$\left.\begin{aligned}
&\alpha = \frac{VisitCount}{MaxVisit} \\
&\psi_T = \frac{StorePeriod}{MaxVisit} \\
&E[N \mid K] = \frac{ADRCount}{MaxVisit}
\end{aligned}\right\} \tag{3.19}$$

The flowchart of the simulation is shown in Figure 3-4, and the details are described as follows.

Step 1. The variables *m*, *clk*, *k*, *StorePeriod*, *ADRCount*, *UARCount*, *VisitCount* are initialized.

Step 2. Check if *MaxVisit* visits to SGSN L_0 have been completed. If so, go to Step 17. Otherwise, go to Step 3.

Step 3. The *m*-th visit to SGSN L_0 is executed. The event list is reset. Then two events are generated and inserted into the event list. The **MSLeave** event represents the time when the MS leaves L_0 at the *m*-th visit to SGSN L_0. The **MSUSR** event represents the arrival of the first UAR.

Steps 4 and 5. The next event *e* is deleted from the head of the event list. The simulation clock *clk*=*e*.timestamp. If *e*.type=**MSUSR**, the simulation flow proceeds to Step 6. If *e*.type=**MSLeave**, Step 11 is executed. If *e*.type=**MSMove**, the flow proceeds to Step 12. If *e*.type=**RTExpire**, Step 16 is executed.

Step 6. For a UAR arrival, the counter *UARCount* is incremented by 1.

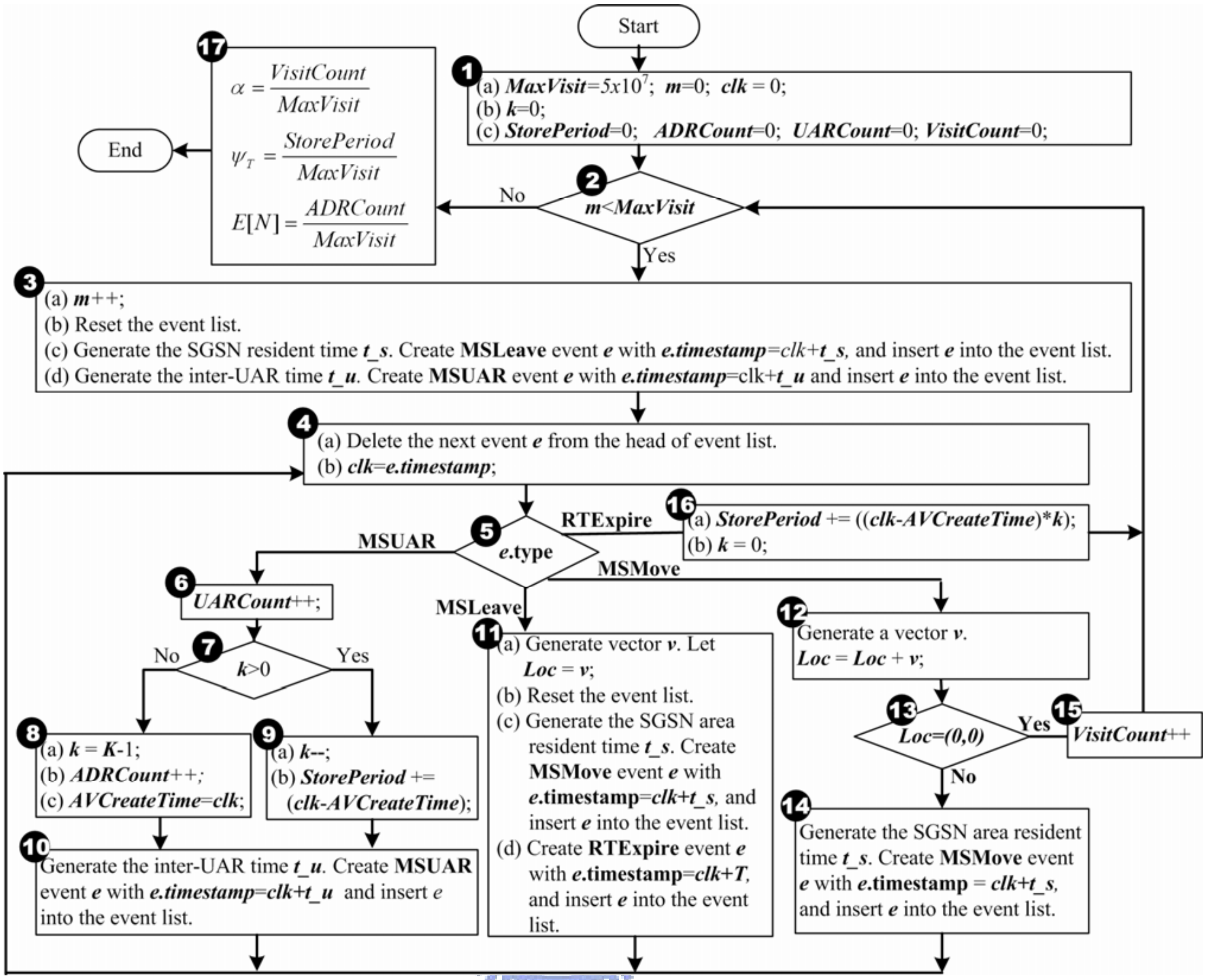

Start
1
(a) MaxVisit=5x10^7; m=0; clk = 0;
(b) k=0;
(c) StorePeriod=0; ADRCount=0; UARCount=0; VisitCount=0;
2
m<MaxVisit
No
Yes
17
α = VisitCount / MaxVisit
ψ_T = StorePeriod / MaxVisit
E[N] = ADRCount / MaxVisit
End
3
(a) m++;
(b) Reset the event list.
(c) Generate the SGSN resident time t_s. Create MSLeave event e with e.timestamp=clk+t_s, and insert e into the event list.
(d) Generate the inter-UAR time t_u. Create MSUAR event e with e.timestamp=clk+t_u and insert e into the event list.
4
(a) Delete the next event e from the head of event list.
(b) clk=e.timestamp;
5
e.type
MSUAR
RTExpire
MSMove
MSLeave
16
(a) StorePeriod += ((clk-AVCreateTime)*k);
(b) k = 0;
6
UARCount++;
7
k>0
No
Yes
8
(a) k = K-1;
(b) ADRCount++;
(c) AVCreateTime=clk;
9
(a) k--;
(b) StorePeriod += (clk-AVCreateTime);
10
Generate the inter-UAR time t_u. Create MSUAR event e with e.timestamp=clk+t_u and insert e into the event list.
11
(a) Generate vector v. Let Loc = v;
(b) Reset the event list.
(c) Generate the SGSN area resident time t_s. Create MSMove event e with e.timestamp=clk+t_s, and insert e into the event list.
(d) Create RTExpire event e with e.timestamp=clk+T, and insert e into the event list.
12
Generate a vector v.
Loc = Loc + v;
13
Loc=(0,0)
Yes
No
15
VisitCount++
14
Generate the SGSN area resident time t_s. Create MSMove event e with e.timestamp = clk+t_s, and insert e into the event list.

Figure 3-4. The Simulation Flowchart

Step 7. If $k > 0$, it means that there are AVs stored in SGSN L_0, and the flow goes to Step 9. Otherwise, go to Step 8.

Step 8. Since the SGSN L_0 does not have any AVs, the SGSN L_0 performs an ADR to obtain K AVs from the HSS/AuC, and uses the first AV for the UAR. Therefore, the ADR counter *ADRCount* is incremented by 1, k is set as K-1, and the current timestamp is recorded in *AVCreateTime*.

Step 9. One AV is used for the UAR, and k is incremented by 1.The time period when the used AV stored in SGSN L_0 is computed as *clk-AVCreateTime*, and is added to *StorePeriod*.

Step 10. The next **MSUSR** event is generated and inserted into the event list.

Step 11. The MS leaves $L_{0.}$ Generate a direction vector $v \in \{(0,1),(0,-1),(1,0),(-1,0)\}$, which indicates one of the four directions that the MS may move. The routing probability is 1/4 for each direction. The location where the MS resides after movement is recorded in *Loc* = *v.* Then the event list is reset, and two events are generated. The **MSMove** event indicates when the MS moves to the next SGSN area after leaving $L_{0.}$ The **RTExpire** event shows when the RT timer expires.

Step 12. Generate the next direction vector *v*. The MS location is updated by *Loc* = *Loc* + *v.*

Step 13. Check if the MS moves back to L_0. If so, the flow goes to Step 15. Otherwise, go to Step 14.

Step 14. Generate an **MSMove** event for the next MS movement.

Step 15. The MS moves back to L_0 for the $(m+1)$-th visit to L_0. *VisitCount* is incremented by 1, and the flow goes back to Step 2 for the next visit to L_0.

Step 16. The RT timer expires, and all the AVs stored in the SGSN L_0 are discarded. Therefore, k is set as 0, and the storage periods for these discarded AVs are added to *StorePeriod*. Then the flow proceeds to Step 2 for the next visit to L_0.

Step 17. If *MaxVisit* visits to L_0 have been complete, the output measures are computed by (3.19), and the simulation is terminated

The simulation experiments are validated against the analytic analysis in Section 3.3. As shown in Figure 3-5, Figure 3-6, and Figure 3-7, the discrepancies between the analytic model and simulation model are less than 1%. Therefore the analytic and simulation models are consistent..

3.5 Numerical Examples

Based on the analytic and simulation models, we use numerical examples to investigate how the RT period T affects the performance of AV management. These numerical examples also validate the simulation model (in Section 3.4) against the analytic analysis in Section 3.3.

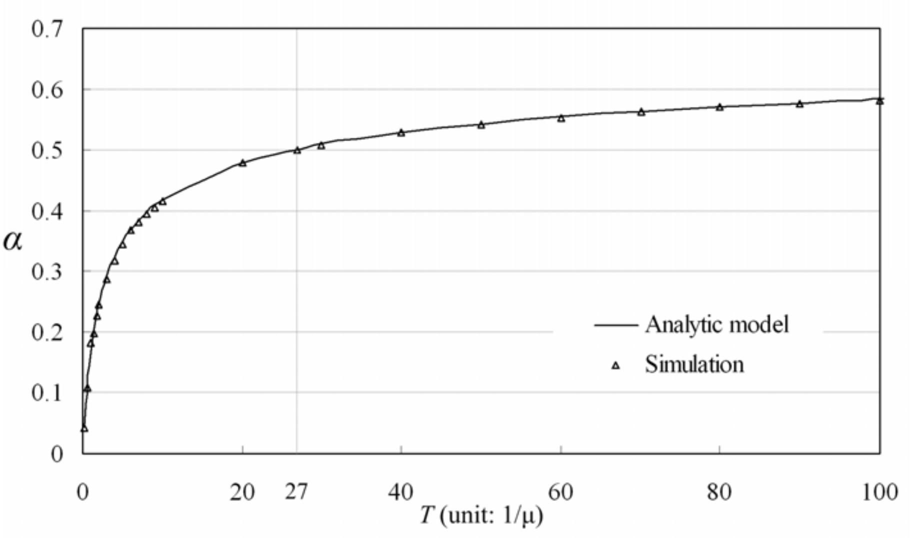


Figure 3-5. Effects of T on α

Based on (3.6), Figure 3-5 plots the probability α of re-entrance to L_0 against the RT period T. The figure indicates that for $T < 27/\mu$, α significantly increases as T increases. For $T \geq 27/\mu$, the impact of T on α becomes less significant. We note that the α curve is determined by the probabilities of the movement directions. In the two-dimensional random walk, if the routing probabilities of the movement directions are not the same, then it is very likely that the MS will never return to L_0. In the real world, the MS movement may exhibit locality, and the MS eventually moves back to L_0.

According to (3.12) and (3.13), Figure 3-6 plots the normalized AV storage β against T and K. The figure indicates that β is an almost linearly increasing function of T. When T=27/μ, SGSN L_0 consumes 17 times as much AV storage as that when T=0.

Based on (3.1) and (3.18), Figure 3-7 plots δ against T and K. We observe that δ decreases as T increases. For T>27/μ, the effect of T on δ is negligible. When $T \rightarrow \infty$, all AVs are utilized for the UARs, and δ=1/K. For the same T value, it is obvious that δ increases as K decreases. Consider the case K=30 and T=27/μ, the ADR traffic decreases 24.9% as

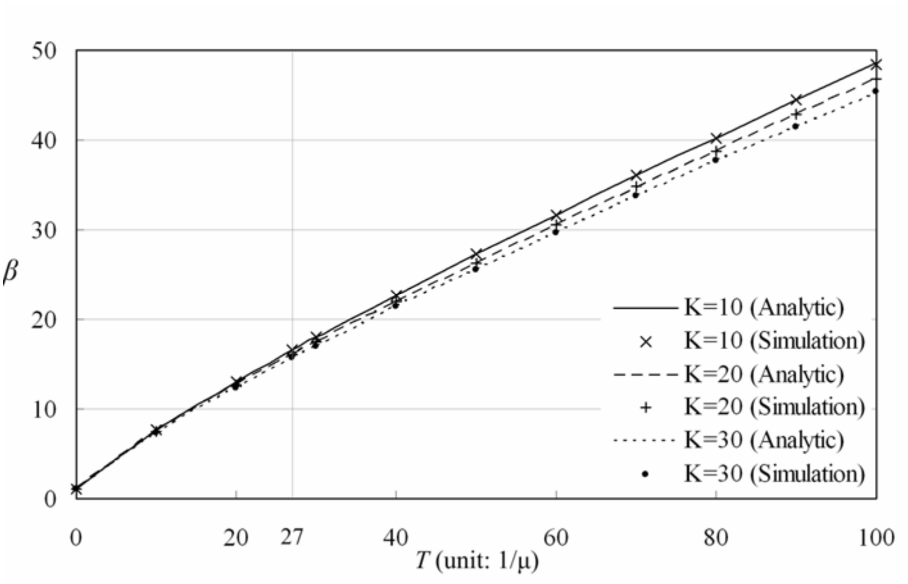


Figure 3-6. Effects of *T* and *K* on *β* (*λ*=20*μ*)

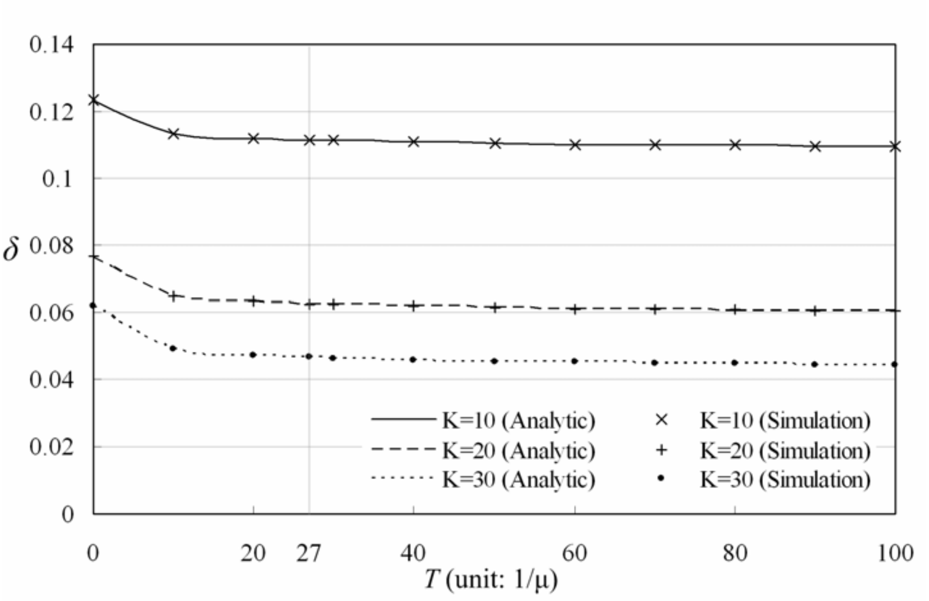


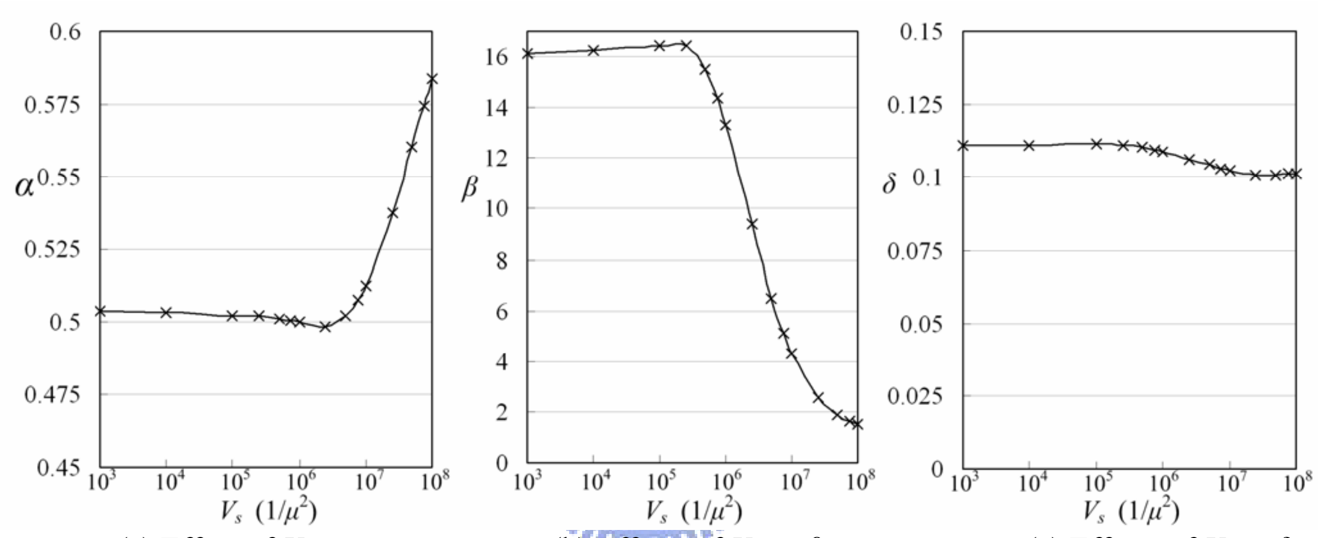
Figure 3-7. Effects of *T* and *K* on *δ* (*λ*=20*μ*)

compared with *K*=30 and *T*=0.

Figure 3-6 and Figure 3-7 indicate the relation between the storage usage and the ADR traffic, and provide the guidelines for the mobile operators to configure the RT timer. For example, if the operator sets *K*=10 and wants to reduce 88.86% of the ADR traffic (as compared with when *K*=1), the RT period *T*=27/μ should be selected. In this case, the

SGSN utilizes 17 times the AV storage as that when T=0.

Figure 3-8 shows the effects for the variance of the SGSN residence times. The Gamma distribution with mean $1/\mu$ and variance V_s is considered for SGSN residence times because it has been shown that the distribution of any positive random variable can be approximated by a mixture of Gamma distributions (see Lemma 3.9 in [32]). Following the past experience [16][22][53], we can measure the SGSN residence times in a real mobile network, and the measured data can be approximated by a Gamma distribution as the input to our simulation model.



(a) Effect of V_s on α (b) Effects of V_s on β (c) Effects of V_s on δ

Figure 3-8. Effects of V_s (λ=20μ, T=27/μ, and K=10)

Figure 3-8 shows the effect of variance V_s for the SGSN residence time distribution on the system performance. When $V_s < 2.5\text{x}10^5/\mu^2$, the impact of V_s on α and δ is insignificant, and β increases as V_s increases. For $V_s > 2.5\text{x}10^5/\mu^2$, as V_s increases, α significantly increases, β significantly decreases, and δ insignificantly decreases. This phenomenon is explained as follows. As V_s increases, more short and long SGSN residence times are observed, and the increase of the number of short SGSN residence times is more significant than that of long SGSN residence times. Since t_r is composed of SGSN residence times, the increase of short t_r is also more significant than that of long t_r. Short t_r results in large α value, and the SGSN consumes less AV storage after the MS leaves the SGSN area (i.e. small β value is expected). Moreover, as α increases, more stored AVs are used for UARs, and the number of ADRs decreases. Therefore, the AV usage mechanism has better performance when the variance of SGSN residence times becomes large.

3.6 Summary

In UMTS, when an MS leaves a SGSN area, the SGSN may keep the unused AVs for an interval called the RT period. If the MS returns to the SGSN area within the RT period, the SGSN uses these stored AVs for mutual authentication instead of obtaining new AVs from the HSS/AuC. This AV usage mechanism reduces the signaling traffic between the SGSN and the HSS/AuC. On the other hand, this mechanism results in extra AV storage at the SGSN. In this chapter, we proposed an analytic model to investigate the impact of the RT period on the system performance. Three output measures are considered: the re-entrance probability, the extra AV storage, and the ADRs traffic between the SGSN and the HSS/AuC. The analytic results were validated against the simulation experiments. Our study provides the guidelines for the mobile operators to implement the AV usage mechanism.

Chapter 4

One-Pass Authentication Procedure for UMTS and IMS

UMTS supports IP multimedia services through IP multimedia core network subsystem (IMS). Since the IMS information is delivered through the GPRS transport network, a UMTS MS must activate GPRS PDP context before it can register to the IMS network. In the 3GPP specifications, authentication is performed at both the GPRS and the IMS networks before an MS can access the IMS services. We observe that many steps in this 3GPP "two-pass" authentication procedure are identical. Based on our observation, this chapter proposes an one-pass authentication procedure that only needs to perform GPRS authentication. At the IMS level, authentication is implicitly performed in IMS registration. Our approach may save up to 50% of the IMS registration/authentication traffic, as compared with the 3GPP two-pass procedure.

4.1 Fraudulent IMS Usage

When an MS attaches to the UMTS network, the GPRS authentication procedure is performed for mutual authentication between the MS and the UMTS core network. Detailed message flow of the GPRS authentication procedure is given in Figure 1-4. In addition to GPRS authentication, before the MS accesses the IMS services, the IMS authentication is performed. The IMS authentication is basically the same as the GPRS authentication. Detailed message flow of the IMS authentication procedure is illustrated in Figure 1-5.

Although GPRS authentication is implemented by GMM and SS7 MAP, and IMS authentication is implemented by SIP and Cx, many steps of these two authentication procedures are duplicated (see Table 4-1). Unfortunately, these redundant steps are required. That is, after GPRS authentication, it is necessary to authenticate the MSs again at the IMS level. Without IMS authentication, an IMS user may pretend to be another IMS user. Consider the example in Figure 4-1, where there are two MSs. MS-A has the IMSI value *imsi-A* and the IMPI value *impi-A*. MS-B has the IMSI value *imsi-B* and the IMPI value *impi-B*. Suppose that MS-B is a legal GPRS user and has passed the GPRS authentication (by using *imsi-B*) to obtain GPRS network access. If no IMS authentication is required, MS-B may perform IMS registration by sending the CSCF a Register request that includes the MS-A's IMPI value *impi-A* as a parameter. The CSCF will consider this IMS registration as a legal action activated by MS-A. Therefore, MS-B can illegally access the IMS services of MS-A. The above example shows that IMS-level authentication is required to prevent illegal access to the IMS services.

In the next section, we describe an one-pass authentication procedure for both GPRS and IMS authentications. Our approach significantly reduces the number of accesses to the HSS/AuC. We also formally prove that the one-pass procedure correctly authenticate the IMS users.

Table 4-1. Identical Steps in GPRS and IMS Authentications

GPRS authentication (SS7 MAP)	IMS authentication (SIP/Cx)
G.2: MAP_SEND_ AUTHENTICATION_INFO Request Parameter: IMSI	**I.2:** Multimedia Authentication Request Parameter: IMPI
G.3: MAP_SEND_ AUTHENTICATION_INFO Response Parameter: AV[1..n]	**I.3:** Multimedia Authentication Answer Parameter: AV[1..n]
G.4: User Authentication Request Parameter: **RAND**||**AUTN**	**I.4:** 401 Unauthorized Parameter: **RAND**||**AUTN**
G.5: User Authentication Response Parameter: **RES**	**I.5:** Register Parameter: **RES**
G.6: GMM Attach Accept	**I.8:** 200 Ok

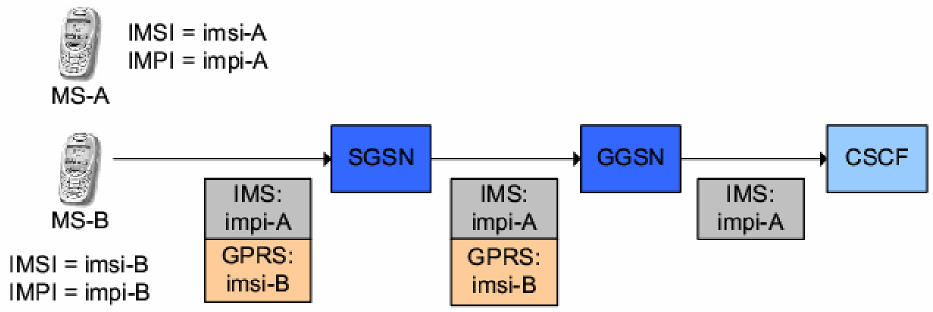


Figure 4-1. Illegal IMS Registration

4.2 One-Pass Authentication Procedure

This section proposes an one-pass authentication (performed at the GPRS level) that can authenticate an IMS user without explicitly performing the IMS-level authentication. In our approach, the SGSN implements a SIP application level gateway (ALG) [15] that modifies the format of SIP messages (to be elaborated). We first describe the SIP message flow of the one-pass procedure. Then, we provide a brief cost comparison between the one-pass and the two-pass procedures.

4.2.1 SIP Message Flow

After GPRS authentication (Steps G.1–G.6 in Figure 1-4) the MS performs PDP context activation to obtain GPRS access. Then, the MS registers to the IMS through Steps I*.1–I*.4 illustrated in Figure 4-2.

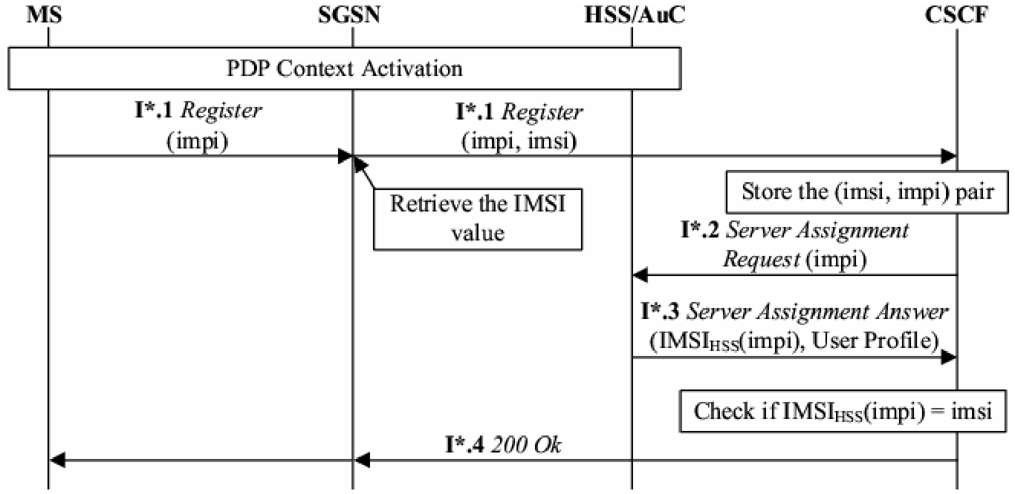

MS
SGSN
HSS/AuC
CSCF
PDP Context Activation
I*.1 Register (impi)
I*.1 Register (impi, imsi)
Retrieve the IMSI value
Store the (imsi, impi) pair
I*.2 Server Assignment Request (impi)
I*.3 Server Assignment Answer (IMSIHSS(impi), User Profile)
Check if IMSIHSS(impi) = imsi
I*.4 200 Ok

Figure 4-2. IMS Registration (One-Pass Authentication)

Step I*.1. The MS sends a SIP Register message to the SGSN with the parameter IMPI = *impi*. Note that after PDP context activation, the SGSN can identify the IMSI of the MS that transmits the GPRS packets [5]. The SIP ALG in the SGSN adds the IMSI value (i.e., *imsi*) of the MS in the Register message and forwards it to the CSCF. Details of a possible SIP ALG implementation can be found in [15].

Step I*.2. The CSCF stores the (*imsi*, *impi*) pair in the MS record, and sends a Cx Server Assignment Request message to the HSS/AuC with the parameter IMPI = *impi*. We note that if the CSCF has stored the (*imsi*, *impi*) pair before, then **Steps I*.2** and **I*.3** are skipped.

Step I*.3. The HSS/AuC uses the received IMPI value *impi* as an index to retrieve the IMSI and the user profile of the MS. We denote $\mathrm{IMSI}_{\mathrm{HSS}}(impi)$ as the IMSI value retrieved from the HSS/AuC. The HSS/AuC stores the CSCF name and sends a Cx Server Assignment Answer to the CSCF (with the parameters $\mathrm{IMSI}_{\mathrm{HSS}}(impi)$ and user profile).

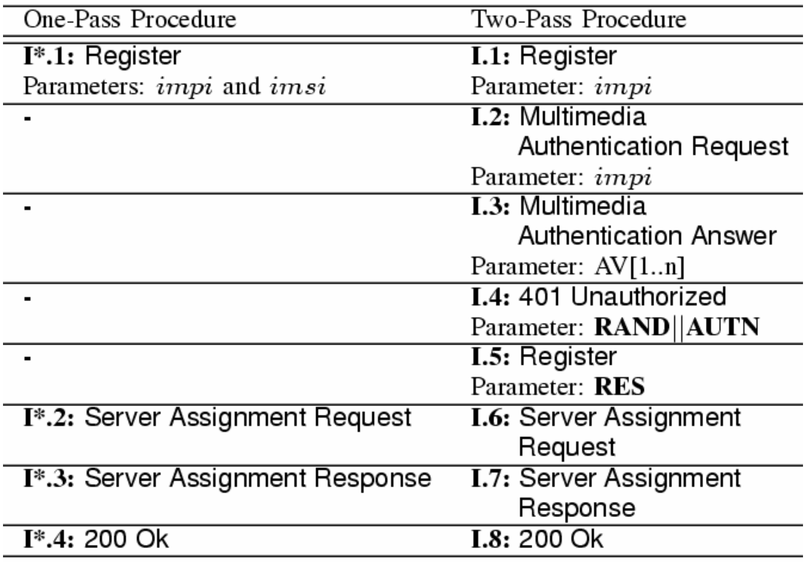
Step I*.4. The CSCF checks whether the value *imsi* and $\mathrm{IMSI}_{\mathrm{HSS}}(impi)$ are the same. If so, the CSCF sends a SIP 200 OK message to the SGSN and the authentication is considered successful. If $\text{IMSI}_{\text{HSS}}(impi) \neq imsi$, then it implies that the registration is illegal (i.e., the scenario illustrated in Figure 4-1 occurs). Suppose that $\text{IMSI}_{\text{HSS}}(impi) = imsi$. The SGSN forwards the 200 OK message to the MS, and the IMS registration procedure is successfully completed.

4.2.2 Cost Analysis

Table 4-2 compares the steps executed in the one-pass and the two-pass authentication procedures. Suppose that the expected SIP message delivery cost between an MS and the CSCF is one unit, and the expected Cx message delivery cost between the CSCF and the HSS/AuC is γ units. It is anticipated that $\gamma < 1$ for the following two reasons.

- The CSCF and the HSS/AuC exchange the Cx messages through IP network. On the other hand, besides the IP network overhead, SIP communications between the MS and the CSCF involves GPRS core network and UTRAN radio network.
- The CSCF and the AuC/HSS are typically located at the same location, while the MS is likely to reside at a remote location.

Table 4-2. Comparing the One-Pass and the Two-Pass Authentication Procedures in IMS Registration



One-Pass Procedure	Two-Pass Procedure
I*.1: Register Parameters: *impi* and *imsi*	**I.1:** Register Parameter: *impi*
-	**I.2:** Multimedia Authentication Request Parameter: *impi*
-	**I.3:** Multimedia Authentication Answer Parameter: AV[1..n]
-	**I.4:** 401 Unauthorized Parameter: **RAND**||**AUTN**
-	**I.5:** Register Parameter: **RES**
I*.2: Server Assignment Request	**I.6:** Server Assignment Request
I*.3: Server Assignment Response	**I.7:** Server Assignment Response
I*.4: 200 Ok	**I.8:** 200 Ok

It is clear that the expected IMS registration C_1 for the one-pass procedure (see Figure 4-2) is

$$C_1 = 2 + 2\gamma \tag{4.1}$$

Note that Step I*.1 needs to trigger SIP ALG for SIP message analysis. Since this action is executed in micro kernel of the SGSN, the overhead can be ignored as compared with SIP message exchange. Similarly, the extra cost of $\mathrm{IMSI}_{\mathrm{HSS}}$(*impi*) and *imsi* comparison at Step I*.4 can be ignored. Our analysis assumes that the (*imsi*, *impi*) pair does not exist at Step I*.1. Therefore Steps I*.2 and I*.3 are always executed. This assumption favors the two-pass procedure.

In the two-pass procedure, if the distribution of authentication vectors from the HSS/AuC to the SGSN (Steps I.1–I.4 in Figure 1-5) is performed, then the expected IMS registration cost $C_{2,1}$ is expressed as

$$C_{2,1} = 4 + 4\gamma \tag{4.2}$$

If the authentication vector distribution is not executed in the two-pass procedure, then the expected IMS registration cost $C_{2,2}$ is expressed as

$$C_{2,2} = 4 + 2\gamma \tag{4.3}$$

Like periodic location update in UMTS [38], IMS registration is periodically performed. In Steps I.2 and I.3 of the two-pass procedure, an AV array of size n (where $n \geq 1$) is sent from the HSS/AuC to the CSCF. Therefore, one out of the n IMS registrations incurs execution of Steps I.2 and I.3. Therefore, from (4.2) and (4.3), the expected IMS registration cost C_2 for the two-pass procedure is

$$C_2 = \left(\frac{1}{n}\right)C_{2,1} + \left(\frac{n-1}{n}\right)C_{2,2} = 4 + \left(\frac{n+1}{n}\right)2\gamma \tag{4.4}$$

From (4.1) and (4.4), the improvement S of the one-pass procedure over the two-pass procedure is

$$S = \frac{C_2 - C_1}{C_2} = \frac{n+\gamma}{2n + (n+1)\gamma} \tag{4.5}$$

Figure 4-3 plots S as a function of n and γ. The figure indicates that the one-pass procedure can save up to 50% of the SIP/Cx traffic for IMS registration/authentication, as compared with the two-pass procedure. Another significant advantage of the one-pass procedure is that it consumes much less AVs (about 50% less) than the two-pass procedure.

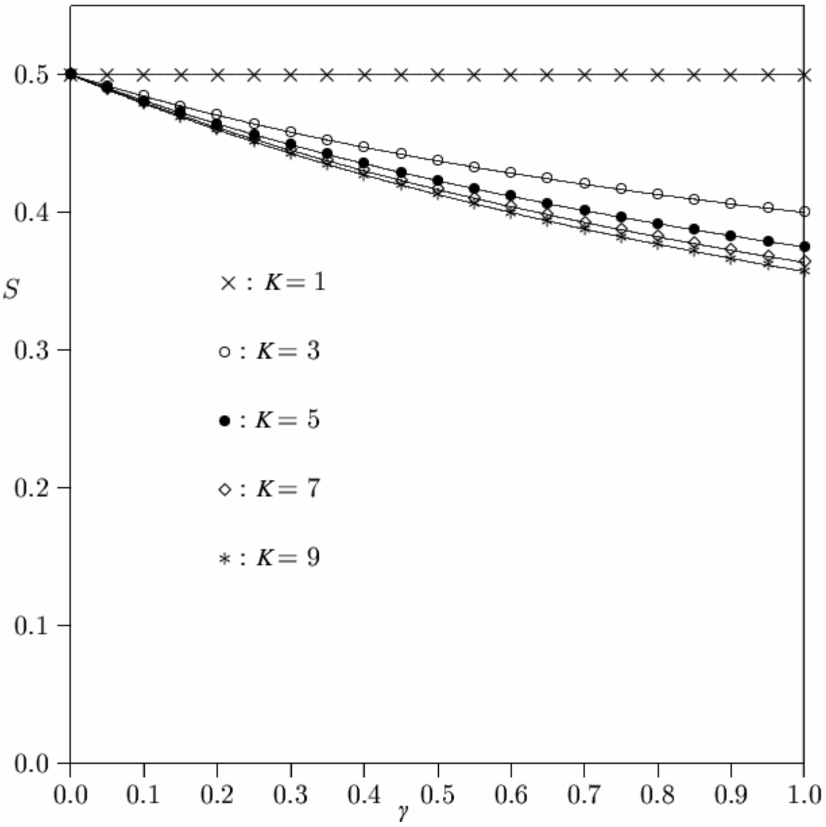


Figure 4-3. Improvement of the One-Pass Procedure over the Two-Pass Procedure

One may argue that implementation of a SIP ALG is required in the one-pass procedure. Since IMS is based on SIP, a SIP ALG is required for other purposes (see an example in [25]). Therefore, the one-pass procedure will not incur extra cost for implementing SIP ALG.

4.3 Correctness of The One-Pass Procedure

In this section, we prove that the one-pass authentication procedure correctly authenticates the IMS users. In UMTS, every MS maintains the attributes IMSI, IMPI, and the preshared secret key **K** in its SIM card. Consider an MS with IMSI = *imsi*, IMPI = *impi*, and **K** = *k*. To simplify our discussion, we assume that these parameters are grouped into a set $R_{MS} = \{imsi, impi, k\}$ in the SIM card of the MS. Define functions IMSI_{MS}, IMPI_{MS}, and K_{MS} such that for any $x \in R_{MS}$

$\text{IMSI}_{MS}(x) = imsi$, where *imsi* is the IMSI value in R_{MS}. (4.6)

$\text{IMPI}_{MS}(x) = impi$, where *impi* is the IMPI value in R_{MS}. (4.7)

$K_{MS}(x) = k$, where *k* is the **K** value in R_{MS}. (4.8)

Based on the above definitions, it is clear that, for example

$\text{IMSI}_{MS}(impi) = \text{IMSI}_{MS}(k) = imsi.$

Similarly, for every MS, the HSS/AuC maintains a record R_{HSS} that consists of attributes IMSI, IMPI, and **K**. That is, for an MS who has legal GPRS and IMS accesses

$R_{HSS} = \{imsi, impi, k\} = R_{MS}$.

Like (4.6)-(4.8), we define functions IMSI_{HSS}, IMPI_{HSS}, and K_{HSS} such that for any $x \in R_{HSS,}$

$\text{IMSI}_{HSS}(x) = imsi$, where *imsi* is the IMSI value in R_{HSS}. (4.9)

$\text{IMPI}_{HSS}(x) = impi$, where *impi* is the IMPI value in R_{HSS}. (4.10)

$K_{HSS}(x) = k$, where *k* is the **K** value in R_{HSS}. (4.11)

In 3G 23.060 [5] and 3G 33.203 [11], MS authentication at the GPRS and the IMS levels are based on the following Theorem.

Theorem 1: Suppose that an MS claims that it has the IMSI value *imsi* and the IMPI value *impi*. Then,

a) The MS is a legal GPRS user if $K_{MS}(imsi) = K_{HSS}(imsi)$.

b) The MS is a legal IMS user if $K_{MS}(impi) = K_{HSS}(impi)$.

Note that Theorem 1 does not hold if an illegal user already possesses the SIM information of a legal user (e.g., by duplicating the SIM card through the SIM card reader [25]). This issue was addressed in [33]. In this chapter, we assume that such fraudulent usage does not occur. 3GPP GPRS authentication procedure (i.e., Steps G.1–G.6) checks if both a GPRS user and the HSS/AuC have the same preshared secret key **K** using Theorem 1 and Fact 1a below. Similarly, 3GPP IMS authentication procedure (i.e., Steps I.1–I.8) checks if both an IMS user and the HSS/AuC have the same preshared secret key using Theorem 1 and Fact 1b.

Fact 1:

a) For an MS claiming IMSI = *imsi*, if XRES = RES, then $\mathrm{K}_{\mathrm{MS}}(imsi) = \mathrm{K}_{\mathrm{HSS}}(imsi)$.

b) For an MS claiming IMPI = *impi*, if XRES = RES, then $\mathrm{K}_{\mathrm{MS}}(impi) = \mathrm{K}_{\mathrm{HSS}}(impi)$.

Now, we prove that the one-pass authentication correctly authenticates the IMS users (i.e., the one-pass procedure checks if $\mathrm{K}_{\mathrm{MS}}(impi) = \mathrm{K}_{\mathrm{HSS}}(impi)$). From the definitions of the $\mathrm{IMSI}_{\mathrm{HSS}}$ and $\mathrm{K}_{\mathrm{HSS}}$ functions [i.e., (4.9) and (4.11)], it is trivial to have the following fact.

Fact 2:

For any IMPI value *impi*, if $\mathrm{IMSI}_{\mathrm{HSS}}(impi) = imsi$, then $\mathrm{K}_{\mathrm{HSS}}(impi) = \mathrm{K}_{\mathrm{HSS}}(imsi)$.

With Fact 2, correctness of the one-pass authentication procedure is guaranteed according to the following two theorems.

Theorem 2: Suppose that

a) an MS with the IMSI value *imsi* has passed the GPRS authentication; that is

$$\mathrm{K}_{\mathrm{MS}}(imsi) = \mathrm{K}_{\mathrm{HSS}}(imsi). \tag{4.12}$$

b) The MS claims that its IMPI value is *impi*.

c) The network maps *impi* to the IMSI value *imsi*; that is

$$\mathrm{IMSI}_{\mathrm{HSS}}(impi) = imsi. \tag{4.13}$$

Then, the MS is a legal IMS user. In other words

$$\mathrm{K}_{\mathrm{MS}}(impi) = \mathrm{K}_{\mathrm{HSS}}(impi). \tag{4.14}$$

Proof:

From hypothesis a, $imsi \in \mathrm{R}_{\mathrm{MS}}$. In hypothesis b, the MS claims that it has the IMPI value *impi*, which implies that $impi \in R_{MS}$. From (4.8)

$$K_{MS}(imsi) = K_{MS}(impi). \tag{4.15}$$

From Fact 2 and (4.13) in hypothesis c, we have

$$K_{HSS}(impi) = K_{HSS}(imsi). \tag{4.16}$$

From (4.12) in hypothesis a and (4.16), we have

$$K_{MS}(imsi) = K_{HSS}(impi). \tag{4.17}$$

From (4.15) and (4.17), we have

$K_{MS}(impi) = K_{HSS}(impi)$.

In other words, if hypotheses a–c hold, an MS is a legal IMS user with IMPI = *impi*.

Q.E.D.

Theorem 3: The one-pass authentication procedure correctly authenticates the IMS users; that is, for an MS claiming the IMPI value *impi*, the one-pass procedure recognizes the MS as a legal IMS user if $K_{MS}(impi) = K_{HSS}(impi)$.

Proof:

After Steps G.1–G.6 have been executed, the network verifies that $K_{MS}(imsi)= K_{HSS}(imsi)$; i.e., (4.12) in Theorem 2 is satisfied.

At Step I*.1, the MS claims that its IMPI value is *impi* and, therefore, the network assumes that $K_{MS}(imsi) = K_{MS}(impi)$; i.e., (4.15) in Theorem 2 is satisfied.

At Step I*.4, the one-pass authentication checks if $IMSI_{HSS}(impi) = imsi$ [i.e., (4.13) in Theorem 2 is checked]. If so, $K_{MS}(impi) = K_{HSS}(impi)$ as a direct consequence of Theorem 2, and the authentication procedure recognizes the MS as a legal user (according to Theorem 1). Otherwise, the authentication fails.

In other words, the one-pass procedure follows Theorem 1 to authenticate an MS.

Q.E.D.

4.4 Summary

This chapter proposed an efficient IMS registration procedure without explicitly performing tedious authentication steps. As specified by the 3GPP, after a UMTS mobile user has obtained GPRS network access through GPRS authentication, the "same" authentication procedure must be executed again at the IMS level (during IMS registration) before it can receive the IP multimedia services. This chapter described an one-pass authentication procedure, which only needs to perform GPRS authentication. At the IMS registration, the one-pass procedure performs several simple operations to verify if a user is legal. We prove that the one-pass procedure correctly authenticates the IMS users. Compared with the eight-step two-pass authentication, the four-step one-pass authentication saves two to four SIP/Cx message exchanges among the MS, the SGSN, the CSCF, and the HSS/AuC. Our study indicates that this new approach can save up to 50% of the network traffic generated by the IMS registration. This approach also saves 50% of the storage for buffering the authentication vectors.

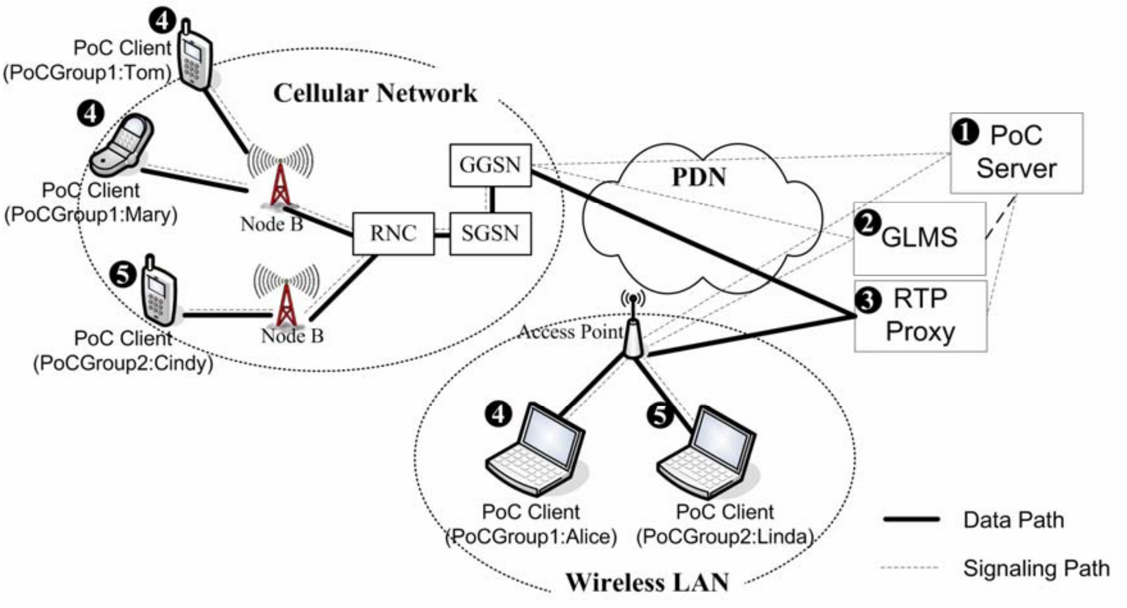
Chapter 5

A Client-Side Design for PoC Service

This chapter proposes a client architecture for the *Push to Talk over Cellular* (PoC) service based on the *Open Mobile Alliance* (OMA) PoC specifications v1.0 release. We show that most standard VoIP modules can be reused for the PoC client, and the VoIP software can be easily extended to support PoC service. Then we present the detailed message flows between the PoC client and other network entities in the PoC system. A PoC client prototype has been implemented in the *Industrial Technology Research Institute* (ITRI) and *National Chiao-Tung University* (NCTU) Joint Research Center.

5.1 Introduction to PoC Service

Push to Talk over Cellular (PoC) service provides a walkie-talkie like service in the cellular communication infrastructure. In this service, several predefined PoC group members participate in one PoC session. Since the PoC session is half-duplex, only one group member speaks at a time, and the others listen. Therefore, a user must ask for the floor (the permission to speak) by pressing the push-to-talk button before he/she starts to talk. In this chapter, we describe the implementation of a PoC client in the WLAN environment [25] with some variation from the Open Mobile Alliance (OMA) PoC specifications v1.0 release [42]. The PoC architecture is illustrated in Figure 5-1.



4
PoC Client (PoCGroup1:Tom)
4
PoC Client (PoCGroup1:Mary)
5
PoC Client (PoCGroup2:Cindy)
Cellular Network
Node B
Node B
RNC
SGSN
GGSN
PDN
1 PoC Server
2 GLMS
3 RTP Proxy
Access Point
4
5
PoC Client (PoCGroup1:Alice)
PoC Client (PoCGroup2:Linda)
Wireless LAN
Data Path
Signaling Path

Figure 5-1. PoC Architecture

In our design, Session Initiation Protocol (SIP) [46] is utilized to implement the PoC service, where a PoC group includes a predefined set of group members, and the SIP Universal Resource Identifier (URI) of each PoC group member is maintained in a group member list. The PoC group is identified by a Telephone URI (TEL URI; e.g., tel: +88635131350) or a SIP URI (e.g., sip:PoCGroup1@pcs1.csie.nctu.edu.tw). The PoC group information and its group member list are stored in the Group and List Management

Server (GLMS; Figure 5-1 (2)). When the PoC clients (Figure 5-1 (4) and (5)) join in the PoC system or when the PoC server (Figure 5-1 (1)) handles a call invitation, they obtain the group information and its group member list from the GLMS. The PoC server is responsible for handling PoC session management (create or delete a PoC session) and floor control. Floor control permits a PoC client to talk at a certain time. A PoC client uses SIP to transmit the session management requests and floor control requests to the PoC server. After the PoC session is established, each PoC group member (PoC client) builds a Real-time Transport Protocol (RTP) [49] session with the RTP Proxy (Figure 5-1 (3)). If a PoC group member obtains the floor, his/her voice is sent to the RTP proxy through the RTP session. The RTP proxy then forwards the voice packets to all group members.

Note that in the OMA PoC specification v1.0 release, the communication between the PoC client and the aggregation proxy (a GLMS-like component) is based on HTTP [26], and the floor control messages between the PoC client and the PoC server is carried by RTCP [49]. Our design uses SIP instead of HTTP and RTCP, so that we can develop the PoC client directly from the standard VoIP modules.

This chapter proposes a PoC client architecture for mobile terminal targeted at the cellular network [35] or WLAN [25]. An OSA-based PoC server architecture is described in [29][40]. The chapter is organized as follows. Section 5.2 represents the design and implementation of the PoC client. Section 5.3 shows the message flows for PoC client operations.

5.2 Implementation of PoC Client

The PoC client architecture proposed in this chapter is illustrated in Figure 5-2. The details are given in the following subsections.

5.2.1 User Interface Module

A user interacts with the PoC system through the User Interface module (Figure 5-2 (1)).

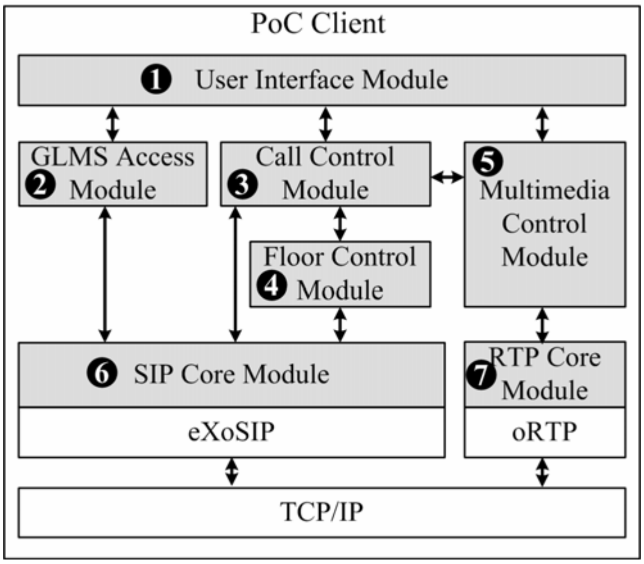


Figure 5-2. PoC Client Architecture

This module consists of four main window dialogs. The Login dialog (Figure 5-3(1) and Figure 5-4 (a)) is popped up after the PoC client program is executed. It waits for the user to input the SIP URI (for the PoC client) and the password, and then executes the SIP registration procedure. After the user is authenticated, the PoC client enters the standby mode, and the Phone dialog (Figure 5-3 (2) and Figure 5-4 (b)) is activated. The Phone dialog alerts the user when an incoming call arrives. If the user misses the call, the Phone dialog shows the missed call information on the user interface. The user can change the presence status through the Phone dialog. After closing the Phone dialog, the user can select to open the Main dialog (Figure 5-3 (3) and Figure 5-4 (d)) or the Taskbar Status Area icon (Figure 5-3 (4) and Figure 5-4 (c)). In the Taskbar Status Area icon mode, when a call arrives or a call is missed, a message window is popped up to notify the user. The Main dialog contains 5 pages. The Personal Information page (Figure 5-3 (5)) presents the user's personal information. The Phone Book page (Figure 5-3 (6) and Figure 5-4 (d)) shows the PoC group information and VoIP contact list. The Configuration page (Figure 5-3 (7)) allows the user to set the ringing tone, the IP address of GLMS, and network configuration. The Call Record page (Figure 5-3 (8)) provides the call history information. The PoC/VoIP Call page (Figure 5-3 (9)) is created when a call is established. This page presents the call information, and allows the PoC user to request the floor.

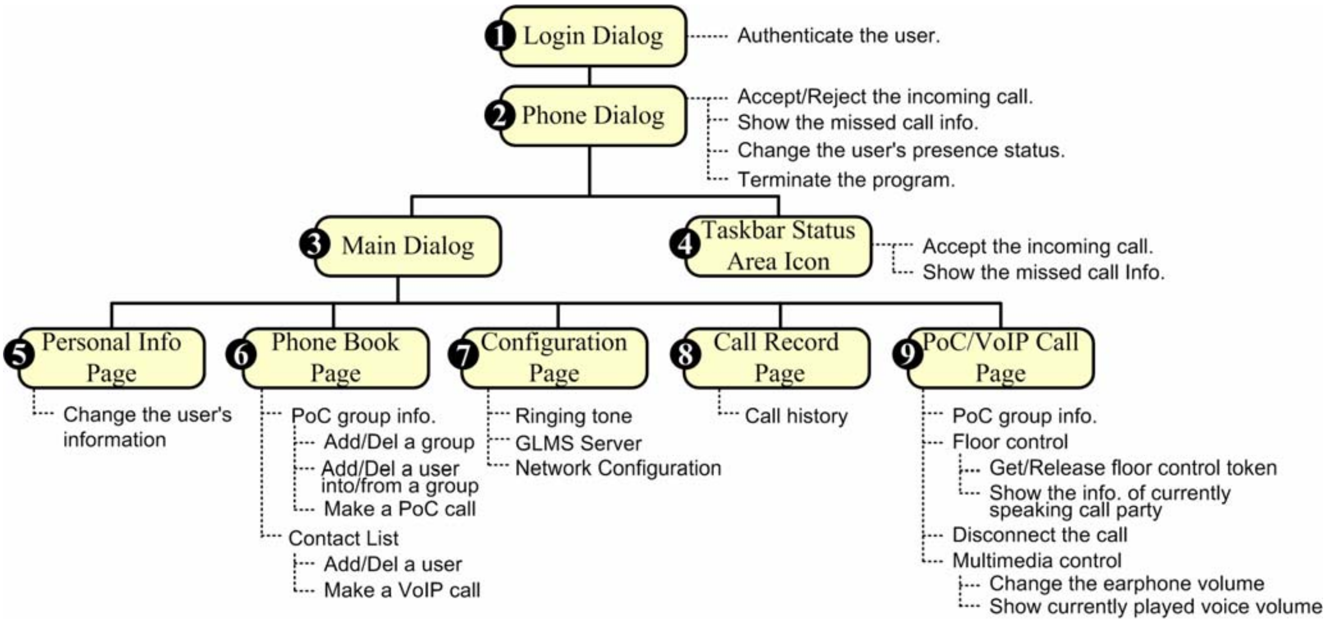
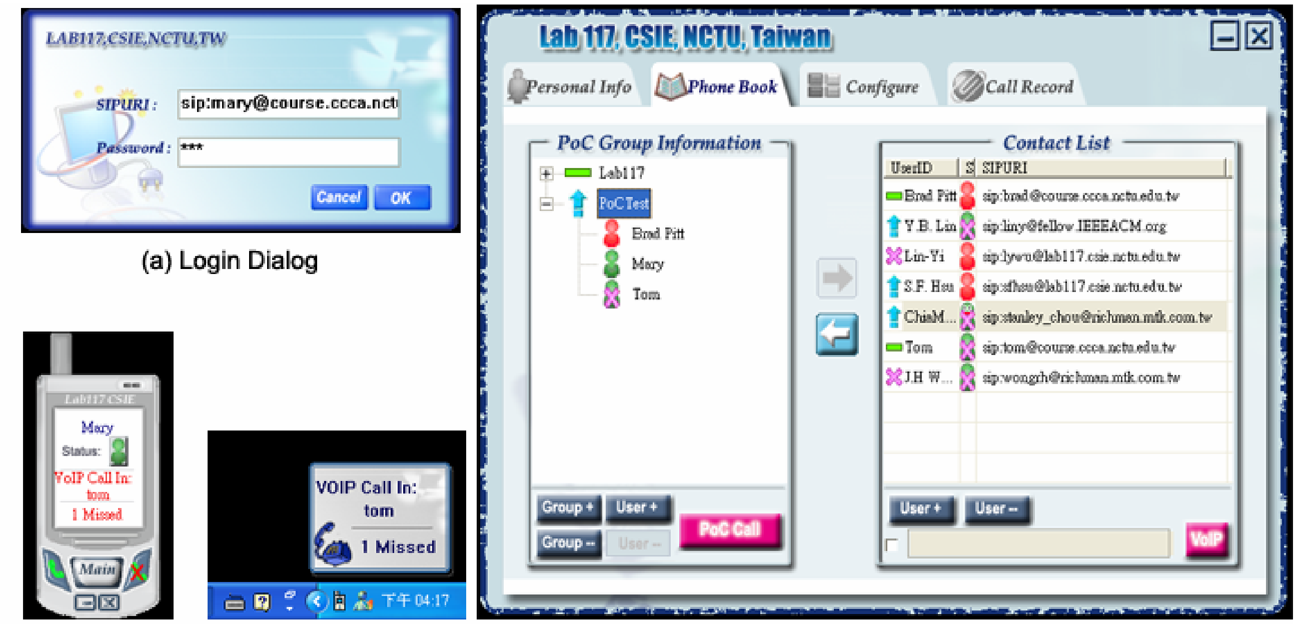


Figure 5-3. Functionalities of the User Interface



SIPURI : sip:mary@course.ccca.nct

Password : ***

(a) Login Dialog

Mary
Status:
VoIP Call In:
tom
1 Missed

VOIP Call In:
tom
1 Missed

UserID	S	SIPURI
Brad Pitt		sip:brad@course.ccca.nctu.edu.tw
Y.B. Lin		sip:liny@fellow.IEEEACM.org
Lin-Yi		sip:lywu@lab117.csie.nctu.edu.tw
S.F. Hsu		sip:sfhsu@lab117.csie.nctu.edu.tw
ChiaM...		sip:stanley_chou@richman.mtk.com.tw
Tom		sip:tom@course.ccca.nctu.edu.tw
J.H W...		sip:wongjh@richman.mtk.com.tw

(b) Phone Dialog (c) Taskbar Status Area Icon (d) Main Dialog

Figure 5-4. The User Interface

5.2.2 Call Control Module

The Call Control module (Figure 5-2 (3)) instructs other PoC client modules to handle the call related activities according to the presence status of the user and the preference types of other call parties. Figure 5-5 illustrates the state diagram for the call control finite state machine (FSM) maintained by the Call Control module. The states are described below (where the term "user" represents this PoC client).

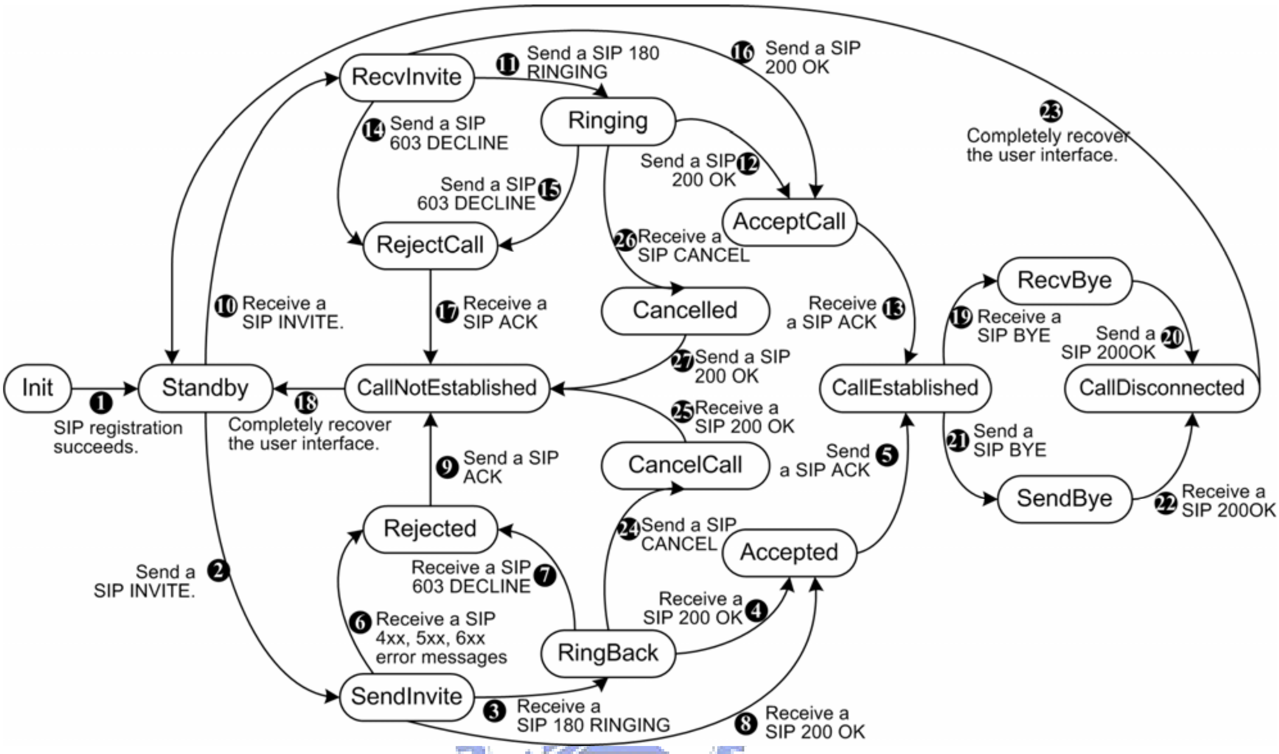


Figure 5-5. State Diagram of the Call Control FSM

Init: The FSM waits for SIP registration.

Standby: The FSM waits for an incoming/outgoing call.

SendInvite: The user requests an outgoing call.

RingBack: The other call party is ringing.

CancelCall: The user cancels the call.

Accepted: The outgoing call is accepted by the other call party.

Rejected: The outgoing call is rejected by the other call party.

RecvInvite: An incoming call arrives.

Ringing: The PoC client is playing the ringing tone to notify the user of the incoming call.

Cancelled: The incoming call is cancelled by the other call party.

AcceptCall: The user accepts the incoming call.

RejectCall: The user rejects the incoming call.

CallNotEstablished: The call is not established.

CallEstablished: The call is established, and the conversation begins.

SendBye: The user requests to disconnect the call.

RecvBye: The other call party requests to disconnect the call.

CallDisconnected: The call is disconnected.

Details of the state transitions will be given in Section 5.3.

5.2.3 Floor Control Module

When a PoC call party obtains the "floor", he/she has the right to speak at that moment. To obtain the floor, the PoC client needs to send a floor request to the PoC server and wait for a positive response. In addition, the PoC server broadcasts the floor status to the call parties (e.g., someone has obtained the floor). The floor-related functionalities are implemented in the Floor Control module (Figure 5-2 (4)). This module maintains an FSM with the state transition diagram illustrated in Figure 5-6. When a PoC call is established, the Call Control module issues a StartFloorReq command to activate the floor control FSM (transition 1 in Figure 5-6). Details of other state transitions are described in Section 5.3. The states are described as follows.

Init: A PoC call is activated. The user is waiting for a floor control signal from the PoC server.

Free: No call party has the floor. The user is free to request the floor.

ReqPending: Floor request has been sent to the PoC server. The user is waiting for the response.

Owner: The user obtains the floor.

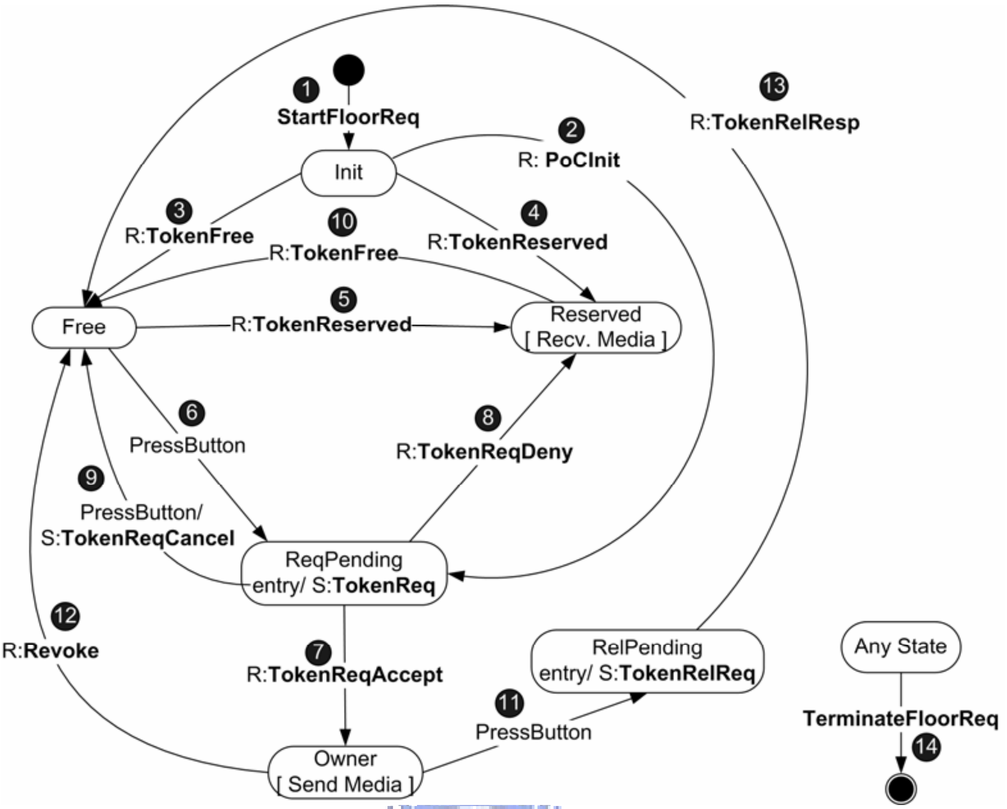


Figure 5-6. State Diagram of the Floor Control FSM

Reserved: Some other call party obtains the floor.

RelPending: Floor release request has been sent to the PoC server. The user is waiting for the response.

5.2.4 Other PoC Client Modules

The GLMS Access, the Multimedia Control, the SIP Core, and the RTP Core modules are described in this subsection.

- The GLMS Access module (Figure 5-2 (2)) is responsible for retrieving the user's information from the GLMS.

- The Multimedia Control module (Figure 5-2 (5)) plays the ringing tone, the ringback tone, or the busy tone to notify the user of various call states. After a call is established, this module plays the received voice data from other call parties and records the user's voice when the user is permitted to speak.

- The SIP Core module (Figure 5-2 (6)) supports SIP communication with other network entities in the PoC system (e.g., GLMS and PoC servers). This module is invoked by three PoC client modules for SIP communication.

  (1) The GLMS Access module interacts with the GLMS through the SIP MESSAGE method.

  (2) The Floor Control module (Figure 5-2 (4)) exchanges the floor control signals with the PoC server by using the SIP INFO method.

  (3) The Call Control module (Figure 5-2 (3)) executes the call setup or disconnection procedures following the standard SIP VoIP protocol.

  In our implementation, the eXosip library [51][52] is utilized for SIP protocol support, and newly added functionalities (e.g., SIP mobility) are also incorporated.

- The RTP Core module (Figure 5-2 (7)) builds a RTP session between the PoC client and the RTP proxy (for a PoC call), or another call party (for a VoIP call). The RTP Core module is implemented based on the oRTP library [43].

In the proposed PoC client architecture, the GLMS Access module and the Floor Control module are specifically designed for PoC service. Other modules are used in both PoC and VoIP services.

5.3 PoC Message Flow

Based on the PoC client architecture described in the previous section, we represent the message flow between the PoC client and other network entities in the PoC system, including Outgoing Call Setup, Incoming Call Setup, Floor Reservation, Floor Release, and Call Disconnection.

5.3.1 Outgoing Call Setup Procedure

When the user requests to make a PoC call, the Outgoing Call Setup procedure is executed as illustrated in Figure 5-7.

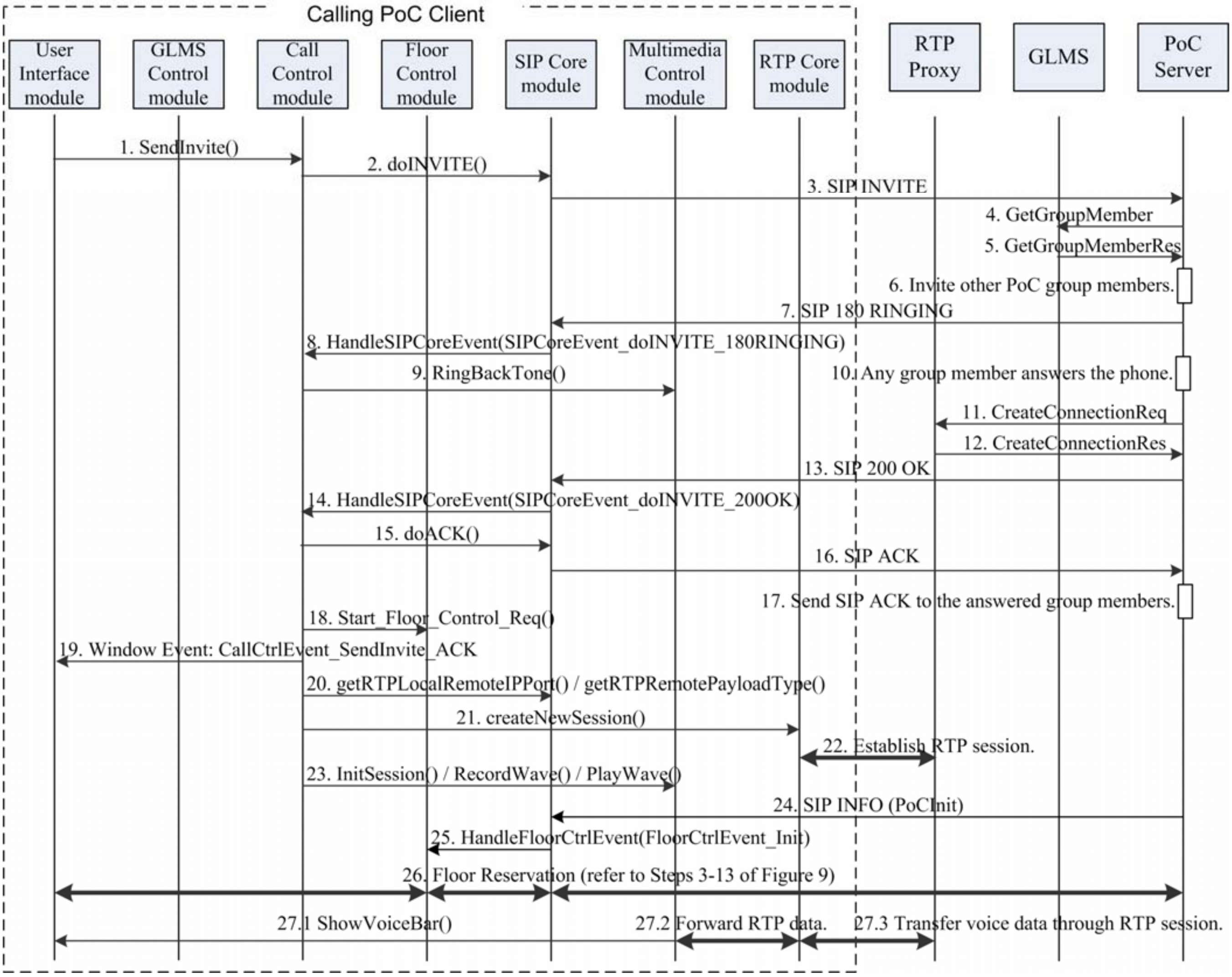

Calling PoC Client
User Interface module
GLMS Control module
Call Control module
Floor Control module
SIP Core module
Multimedia Control module
RTP Core module
RTP Proxy
GLMS
PoC Server
1. SendInvite()
2. doINVITE()
3. SIP INVITE
4. GetGroupMember
5. GetGroupMemberRes
6. Invite other PoC group members.
7. SIP 180 RINGING
8. HandleSIPCoreEvent(SIPCoreEvent_doINVITE_180RINGING)
9. RingBackTone()
10. Any group member answers the phone.
11. CreateConnectionReq
12. CreateConnectionRes
13. SIP 200 OK
14. HandleSIPCoreEvent(SIPCoreEvent_doINVITE_200OK)
15. doACK()
16. SIP ACK
17. Send SIP ACK to the answered group members.
18. Start_Floor_Control_Req()
19. Window Event: CallCtrlEvent_SendInvite_ACK
20. getRTPLocalRemoteIPPort() / getRTPRemotePayloadType()
21. createNewSession()
22. Establish RTP session.
23. InitSession() / RecordWave() / PlayWave()
24. SIP INFO (PoCInit)
25. HandleFloorCtrlEvent(FloorCtrlEvent_Init)
26. Floor Reservation (refer to Steps 3-13 of Figure 9)
27.1 ShowVoiceBar()
27.2 Forward RTP data.
27.3 Transfer voice data through RTP session.

Figure 5-7. Outgoing Call Setup Procedure

Steps 1-3: Based on the PoC group SIP URI specified by the user, the User Interface module requests the Call Control module to initiate a PoC call. The Call Control module instructs the SIP Core module to issue a SIP INVITE, and the call control FSM moves from **Standby** to **SendInvite** (transition 2 in Figure 5-5). The SIP INVITE is delivered to the PoC server where the PoC group's SIP URI is registered.

Steps 4-6: The PoC server queries the member data of the designated PoC group from the GLMS and dispatches the SIP INVITE to each of the group members.

If an error occurs during the PoC call invitation, the PoC server returns a SIP 4xx, 5xx, or 6xx error message to the calling PoC client. Then the call control FSM of the calling PoC client moves from **SendInvite** to **Rejected** (transition 6 in Figure 5-5), and the calling PoC client replies a SIP ACK to the PoC server. Finally, the call control FSM moves to the **Standby** state (through transition 9 and 18 in Figure 5-5), and the PoC client waits for next user instruction or incoming call. Suppose that no error occurs. Step 7 and the subsequent steps are executed.

Steps 7-9: If a called PoC client receives the call invitation, this client plays the ringing tone and replies a SIP 180 RINGING to the calling PoC client through the PoC server. The Multimedia Control module of the calling PoC client plays the ringback tone to indicate the user that the Outgoing Call Setup procedure is in progress, and the call control FSM moves from **SendInvite** to **RingBack** (transition 3 in Figure 5-5).

Steps 10-12: If any of the called PoC group members picks up the phone (accepts the call invitation), the member replies a SIP 200 OK to the PoC Server. The PoC server chooses the audio codec used in the RTP session between the calling PoC client and the RTP proxy. The PoC Server forwards the codec information to the RTP proxy, and also instructs the RTP proxy to reserve ports for the RTP session (Steps 11 and 12).

Steps 13 and 14: The PoC server sends a SIP 200 OK to the calling PoC client, which includes the audio codec information, the IP address of the RTP proxy, and the reserved ports in the RTP proxy. These parameters are used in establishing the RTP session at Step 22. The calling PoC client call control FSM moves from **RingBack** to **Accepted** (transition 4 in Figure 5-5).

Steps 15-17: The SIP Core module responds a SIP ACK, and the call control FSM moves from **Accepted** to **CallEstablished** (transition 5 in Figure 5-5).The PoC server forwards the SIP ACK to the called PoC group members who have accepted the call.

Step 18: The Call Control module of the calling PoC client instructs the Floor Control module to enable the floor control function. The floor control FSM moves to **Init (**transition 1 in Figure 5-6).

Step 19: The User Interface module shows the PoC group information on the Call page of the Main dialog. Note that the user is only allowed to participate in one PoC or VoIP call.

Steps 20-22: The calling PoC client creates the RTP connection to the RTP proxy. Specifically, the Call Control module inquires the SIP Core module about the negotiated audio codec and the IP address/port number of the remote endpoint of RTP session (i.e., the RTP proxy), which are obtained from the PoC server at Step 13. The information is passed to the RTP Core module to create the RTP session following the standard RTP protocol.

Step 23: The Multimedia Control module activates the audio device and generates two processes for recording and playing the voice data.

Steps 24-26: The calling PoC client is the first call party who is permitted to speak. The PoC server sends a PoCInit message carried by the SIP INFO method to the calling PoC client. The floor control FSM moves from **Init** to **ReqPending** (transition 2 in Figure 5-6). The Floor Reservation procedure is invoked, and the message flow is described in Section 5.3.3.

Step 27: At this point, the conversion begins. The calling PoC client sends the voice to the RTP proxy through the Multimedia Control module and the RTP Core module (Step 27.2 and Step 27.3). During the call, the voices from other group members are forwarded to the Multimedia Control module (Step 27.3 and Step 27.2), and are played through the audio device. The volume of the currently played voice is reported to the User Interface module and shown on the Call page of the Main dialog (Step 27.1).

Note that when other called PoC clients receive the call invitation at Step 6 of Figure 5-7, they may automatically accept the call invitation without playing the ringing tone. If so, Steps 7-9 are skipped, and the PoC group members directly reply a SIP 200 OK to the calling PoC client through the PoC Server. Then the call control FSM of the calling PoC client moves from **SendInvite** to **Accepted** (transition 8 in Figure 5-5).

When the called PoC clients receive the call invitation, and the ringing tone is played at Step 7 of Figure 5-7, two other situations may occur.

(1) All PoC group members reject the call invitation. The PoC server replies a SIP 603 DECLINE to the calling PoC client and the Outgoing Call Setup procedure exits. The call control FSM of the calling PoC client moves from **RingBack** to **Rejected** (transition 7 in Figure 5-5).

(2) The calling party presses the cancel button to cancel the outgoing call before any of the PoC group members replies a message. The call control FSM of the calling PoC client moves from **RingBack** to **CancelCall** (transition 24 in Figure 5-5), and waits for the PoC server to reply a SIP 200 OK. Upon receipt of the SIP 200 OK, the call control FSM moves from **CancelCall** to **CallNotEstablished** (transition 25 in Figure 5-5).

5.3.2 Incoming Call Setup Procedure

For an incoming PoC call, the called PoC client is invited by the PoC server to join in the PoC call through the Incoming Call Setup procedure as illustrated in Figure 5-8, and the detailed steps are described as follows.

Steps 1-4: The PoC server receives a PoC call invitation from the calling party. It dispatches the PoC call invitation to each of the PoC group members.

Steps 5-9: After receiving the SIP INVITE, the call control FSM of the called PoC client moves from **Standby** to **RecvInvite** (transition 10 in Figure 5-5). The PoC client processes the incoming call according to the user's presence status (i.e., Online, Offline, Busy, or NoDisturb) and the preference type of the calling party (i.e., Auto-Answer, Manual-Answer, or Reject). Suppose that the user's presence status is not set as NoDisturb and the preference type of the calling party is set as

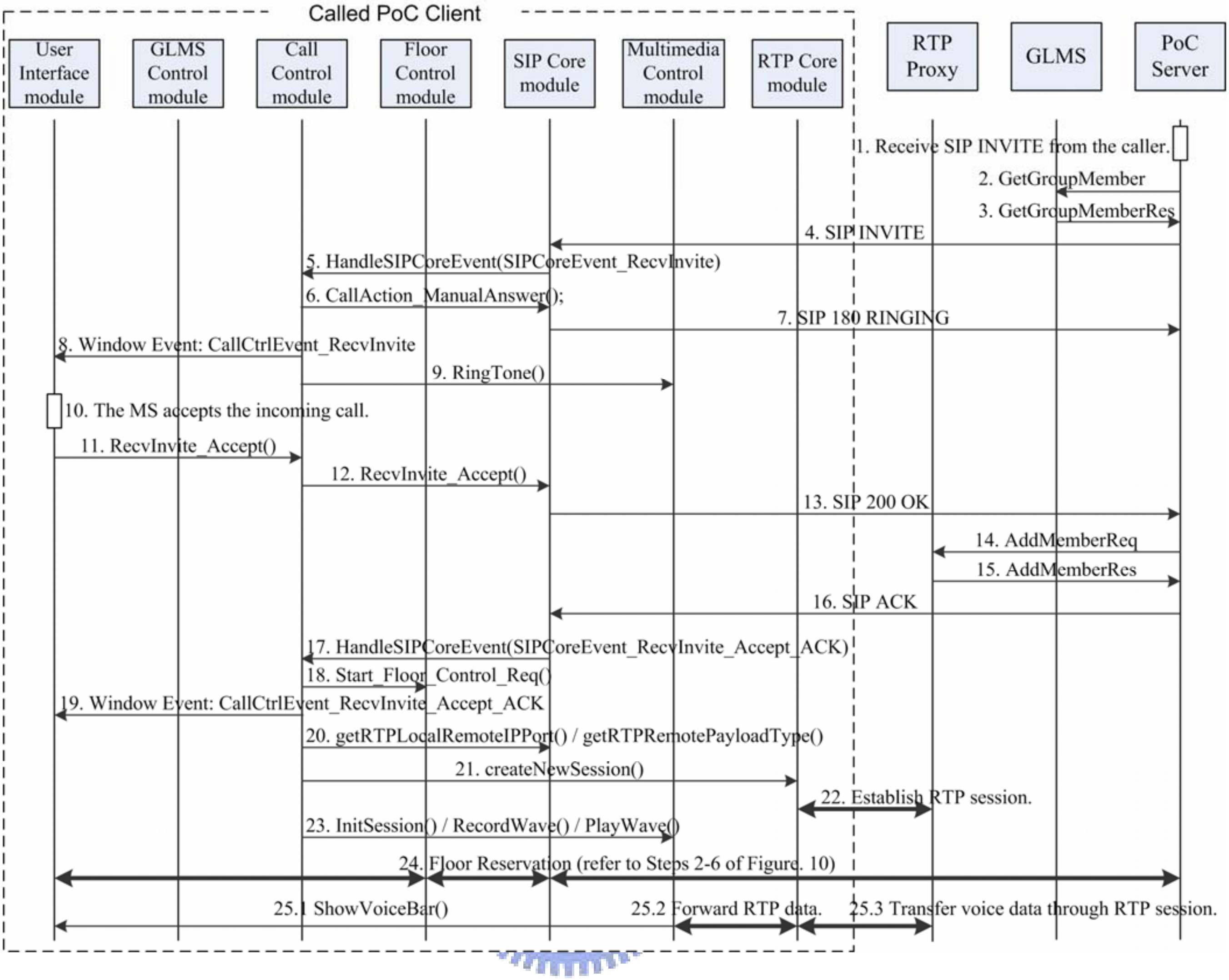

Called PoC Client
User Interface module
GLMS Control module
Call Control module
Floor Control module
SIP Core module
Multimedia Control module
RTP Core module
RTP Proxy
GLMS
PoC Server
1. Receive SIP INVITE from the caller.
2. GetGroupMember
3. GetGroupMemberRes
4. SIP INVITE
5. HandleSIPCoreEvent(SIPCoreEvent_RecvInvite)
6. CallAction_ManualAnswer();
7. SIP 180 RINGING
8. Window Event: CallCtrlEvent_RecvInvite
9. RingTone()
10. The MS accepts the incoming call.
11. RecvInvite_Accept()
12. RecvInvite_Accept()
13. SIP 200 OK
14. AddMemberReq
15. AddMemberRes
16. SIP ACK
17. HandleSIPCoreEvent(SIPCoreEvent_RecvInvite_Accept_ACK)
18. Start_Floor_Control_Req()
19. Window Event: CallCtrlEvent_RecvInvite_Accept_ACK
20. getRTPLocalRemoteIPPort() / getRTPRemotePayloadType()
21. createNewSession()
22. Establish RTP session.
23. InitSession() / RecordWave() / PlayWave()
24. Floor Reservation (refer to Steps 2-6 of Figure. 10)
25.1 ShowVoiceBar()
25.2 Forward RTP data.
25.3 Transfer voice data through RTP session.

Figure 5-8. Incoming Call Setup Procedure

Manual-Answer. The called PoC client sends a SIP 180 RINGING to the PoC server. Then the User Interface module shows the incoming call notification message, and the Multimedia Control module plays the ringing tone. The called PoC clients call control FSM moves from **RecvInvite** to **Ringing** (transition 11 in Figure 5-5).

Steps 10-13: The called PoC client accepts the incoming call and replies a SIP 200 OK to the PoC server. The call control FSM moves from **Ringing** to **AcceptCall** (transition 12 in Figure 5-5).

Steps 14 and 15: The PoC server chooses the audio codec to be used in the RTP session between the called PoC client and the RTP proxy. Then it forwards the codec information to the RTP proxy, and also instructs the RTP proxy to reserve ports for the

RTP session.

Steps 16 and 17: The PoC server sends a SIP ACK to the called PoC client, which includes the audio codec information, the IP address of the RTP proxy, and the reserved ports in the RTP proxy. These parameters are used in establishing the RTP session at Step 22. When the called PoC client receives the SIP ACK, the call control FSM moves from **AcceptCall** to **CallEstablished** (transition 13 in Figure 5-5).

Steps 18-23: These steps are the same as Steps 18-23 in Figure 5-7.

Step 24: After the PoC call is established, the PoC server invokes the Floor Reservation procedure to inform the called PoC client of the current speaker. The detailed steps are described at Steps 2-6 in Figure 5-10. The called PoC client floor control FSM moves from **Init** to **Reserved** (transition 4 in Figure 5-6).

Step 25: This step is the same as Step 27 in Figure 5-7.

After receiving the SIP INVITE at Step 4, two other scenarios may occur to the called PoC client.

(1) If the user's presence status is set as NoDisturb or the preference type of the calling party is set as Reject, the called PoC client replies a SIP 603 DECLINE to reject the incoming call. The called PoC client call control FSM moves from **RecvInvite** to **RejectCall** (transition 14 in Figure 5-5), and the Incoming Call Setup procedure exits.

(2) If the user's presence status is not set as NoDisturb and the preference type of the calling party is set as Auto-Answer, Steps 6-11 in Figure 5-8 are skipped, and the called PoC client directly replies a SIP 200 OK to accept the incoming call. Then the call control FSM moves from **RecvInvite** to **AcceptCall** (transition 16 in Figure 5-5).

After the called PoC client plays the ringing tone at Step 9, other two scenarios may occur.

(1) The called PoC client rejects the incoming call. The call control FSM moves from **Ringing** to **RejectCall** (transition 15 in Figure 5-5).

(2) The calling PoC client cancels the call before the called PoC client makes the decision.

The called PoC client receives a SIP CANCEL, and the call control FSM moves from **Ringing** to **Cancelled** (transition 26 in Figure 5-5). Then the called PoC client replies a SIP 200 OK, and the call control FSM moves from **Cancelled** to **CallNotEstablished** (transition 27 in Figure 5-5).

In Step 24, the PoC server informs the called PoC client of the floor status. It is also possible that no PoC member attempts to talk while the called PoC client joins in the PoC call (i.e., the floor is idle). In this case, the PoC server sends a TokenFree message to the called PoC client through the SIP INFO method. The floor control FSM of the called PoC client moves from **Init** to **Free** (transition 3 in Figure 5-6).

5.3.3 Floor Reservation Procedure

After a PoC call is established, the user presses the floor request button to request for the permission to talk. Figure 5-9 illustrates the Floor Reservation procedure between the requesting PoC client and the PoC server, and Figure 5-10 illustrates the message flow between the PoC server and other group members.

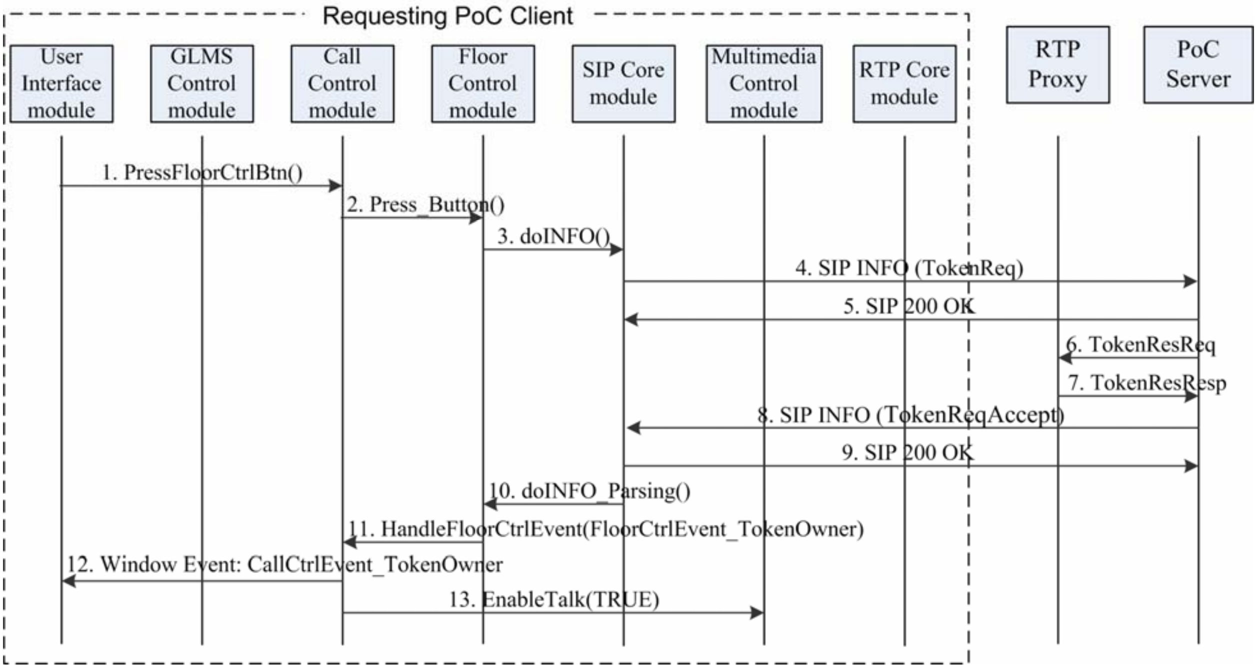


Figure 5-9. Floor Reservation Procedure (the requesting PoC client)

Steps 1-5: The requesting PoC client issues a TokenReq message to request for the floor. The request is delivered to the PoC server through the SIP INFO method. Then the PoC server replies a SIP 200 OK. The floor control FSM moves from **Free** to **ReqPending** (transition 6 in Figure 5-6).

Steps 6-11: If the PoC server grants the floor to the requesting PoC client, it instructs the RTP proxy to dispatch the voice from the requesting PoC client to the other PoC group members. Then the PoC server sends a TokenReqAccept message to the requesting PoC client. The requesting PoC client replies a SIP 200 OK, and the floor control FSM moves from **ReqPending** to **Owner** (transition 7 in Figure 5-6).

Steps 12 and 13: The Call Control module instructs the User Interface module to show that the user is the currently speaking call party, and also instructs the Multimedia Control module to record the user's voice.

After the PoC server receives the floor reservation request at Steps 4 and 5, other two situations may occur.

(1) More than one PoC group members request for the floor at the same time, and the requesting PoC client fails to obtain the floor. The PoC server sends a TokenReqDeny message to the requesting PoC client, and the floor control FSM moves from **ReqPending** to **Reserved** (transition 8 in Figure 5-6).

(2) Before the PoC server makes the decision, the requesting PoC client cancels the request. The requesting PoC client sends a TokenReqCancel message to the PoC server, and the floor control FSM moves from **ReqPending** to **Free** (transition 9 in Figure 5-6).

In the Floor Reservation procedure, the interaction between the PoC server and a listening call party is illustrated in Figure 5-10.

Step 1: The PoC server receives a TokenReq from one of the PoC group members. This step is the same as Step 4 in Figure 5-9.

Steps 2-5: The PoC server sends a TokenReserved message to each of the other PoC group

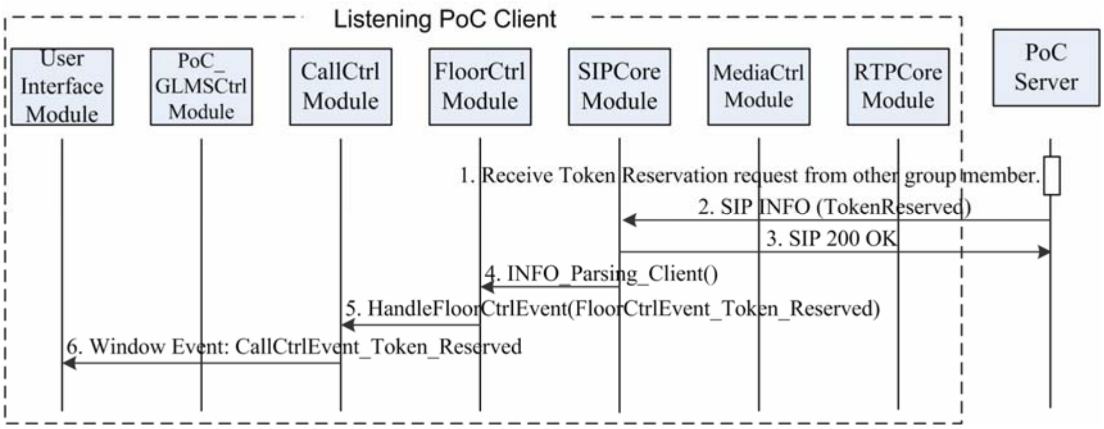

Listening PoC Client
User Interface Module
PoC_ GLMSCtrl Module
CallCtrl Module
FloorCtrl Module
SIPCore Module
MediaCtrl Module
RTPCore Module
PoC Server
1. Receive Token Reservation request from other group member.
2. SIP INFO (TokenReserved)
3. SIP 200 OK
4. INFO_Parsing_Client()
5. HandleFloorCtrlEvent(FloorCtrlEvent_Token_Reserved)
6. Window Event: CallCtrlEvent_Token_Reserved

Figure 5-10. Floor Reservation Procedure (a listening PoC client)

members (i.e., the listening clients) to announce that the floor is obtained by someone. The message is delivered to a listening PoC client through the SIP INFO method. The listening PoC client replies a SIP 200 OK, and its floor control FSM moves from **Free** to **Reserved** (transition 5 in Figure 5-6).

Step 6: The Call Control module instructs the User Interface module to show the information of the currently speaking call party and disable the floor request button.

5.3.4 Floor Release Procedure

After a PoC group member obtains the floor, he/she can press the floor release button to release the floor. If the floor is kept by a user after a long period, the PoC server revokes the floor to guarantee fairness. Figure 5-11 shows the message flow that the speaking PoC client releases the floor.

Steps 1-5: The releasing PoC client issues a TokenRelReq message to release the floor. The request is delivered to the PoC server through the SIP INFO method. The floor control FSM moves from **Owner** to **RelPending** (transition 11 in Figure 5-6). Then the PoC server replies a SIP 200 OK.

Steps 6 and 7: The PoC server instructs the RTP proxy to stop dispatching the voice from the releasing PoC client.

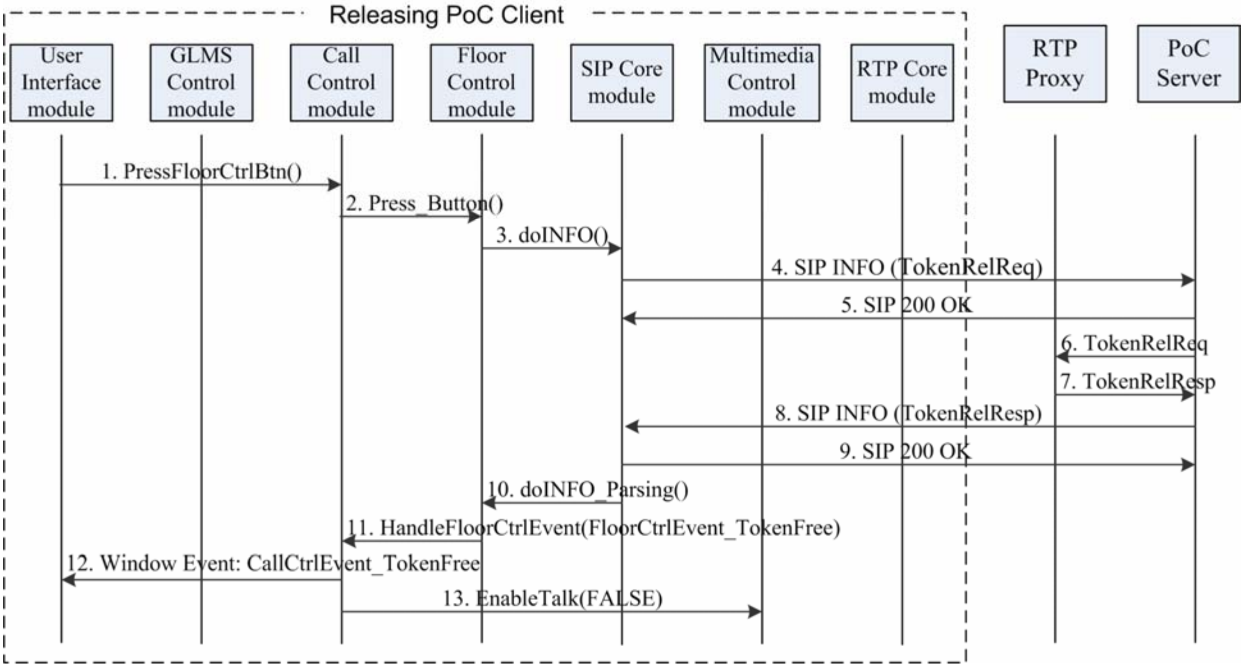


Figure 5-11. Floor Release Procedure (the releasing PoC client)

Steps 8-11: The PoC server sends a TokenRelResp message to the releasing PoC client. The PoC client replies a SIP 200 OK. Then the Floor Control module moves the floor control FSM from **RelPending** to **Free** (transition 13 in Figure 5-6).

Steps 12 and 13: The Call Control module instructs the User Interface module to show that the floor is idle, and instructs the Multimedia Control module to stop recording the user’s voice.

Figure 5-12 shows the message flow where the PoC server notifies the other listening PoC clients of floor releasing.

Step 1: The PoC server receives a TokenRelReq message from the currently speaking call party. This step is the same as Step 4 in Figure 5-11.

Steps 2-5: The PoC server sends a TokenFree message to a listening PoC client through the SIP INFO method. Then the listening PoC client replies a SIP 200 OK, and the floor control FSM moves from **Reserved** to **Free** (transition 10 in Figure 5-6).

Step 6: The Call Control module instructs the User Interface module to show that the floor is idle and enable the floor request button on the Call page of the Main dialog.

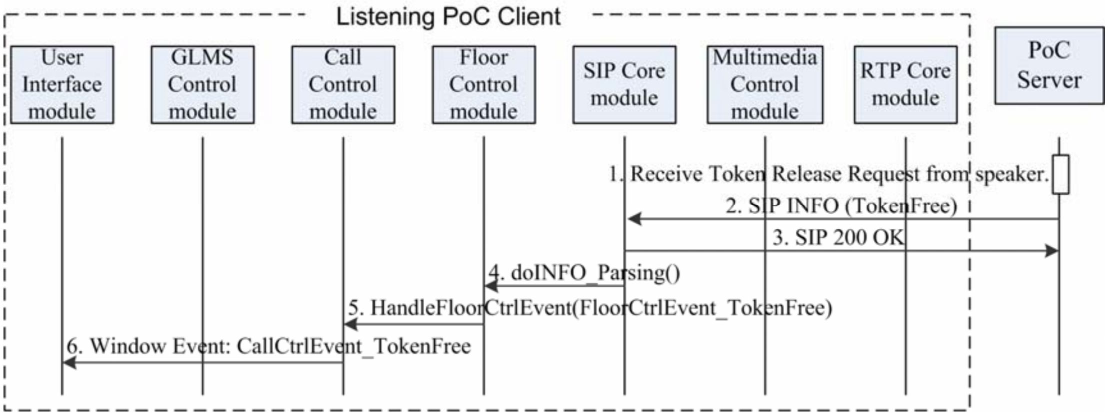


Figure 5-12. Floor Release Procedure (a listening PoC client)

Figure 5-13 illustrates the message flow that the PoC server revokes the floor from the speaking PoC client.

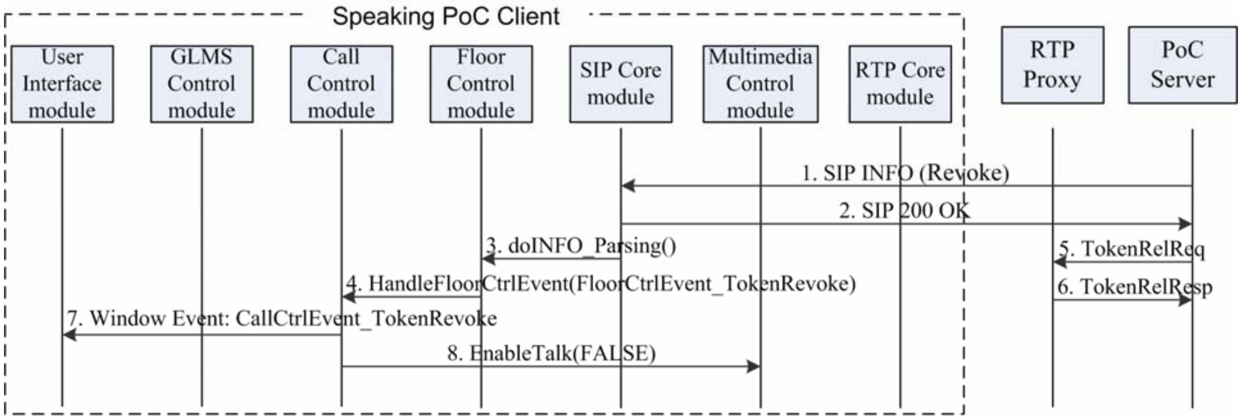


Figure 5-13. Floor Revoking Procedure

Steps 1-4: The PoC server sends a Revoke message to a speaking PoC client. The speaking PoC client replies a SIP 200 OK, and the floor control FSM moves from **Owner** to **Free** (transition 12 in Figure 5-6).

Steps 5 and 6: These steps are the same as Steps 6 and 7 in Figure 5-11.

Steps 7 and 8: These steps are the same as Steps 12-13 in Figure 5-11.

5.3.5 Call Disconnection Procedure

During a PoC call, a PoC client may disconnect by invoking the Call Disconnection procedure. The remaining PoC group members continue the conversation. The message flow of the Call Disconnection procedure is illustrated in Figure 5-14, and the detailed steps are described as follows.

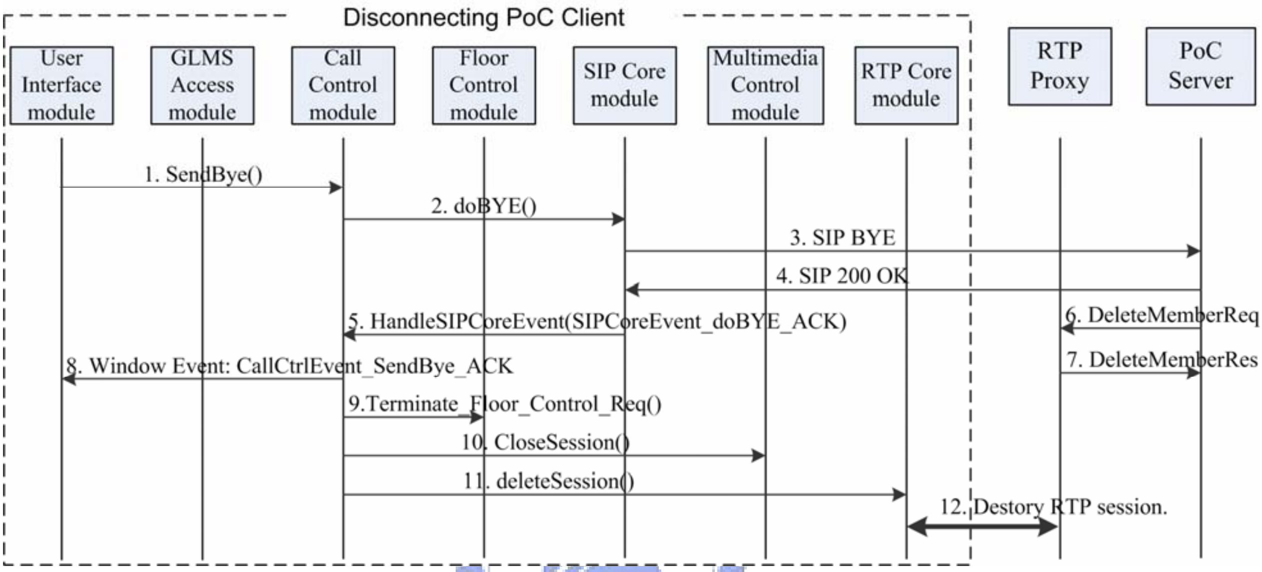


Figure 5-14. Call Disconnection Procedure

Steps 1-3: The PoC client sends a SIP BYE to the PoC server. The call control FSM moves from **CallEstablished** to **SendBye** (transition 21 in Figure 5-5).

Steps 4 and 5: The PoC server replies a SIP 200 OK. The call control FSM moves from **SendBye** to **CallDisconnected** (transition 22 in Figure 5-5).

Steps 6 and 7: The PoC server instructs the RTP proxy to close the RTP session between the RTP proxy and the PoC client.

Step 8: The Call Control module instructs the User Interface module to close the Call page of the Main dialog.

Step 9: The Call Control module instructs the Floor Control module to disable the floor control function. The floor control FSM moves to **End** (transition 14 in Figure 5-6).

Step 10: The Multimedia Control module stops playing and recording voice.

Steps 11 and 12: The Call Control module instructs the RTP Core module to close the RTP session following the standard RTP protocol.

After the above steps, the call control FSM moves back to **Standby** (transition 23 in Figure 5-5), and the PoC client waits for another user instruction or incoming call.

5.4 Summary

In this chapter, we described a design and implementation of PoC client in cellular network or WLAN. The detailed message flows for Outgoing Call Setup, Incoming Call Setup, Floor Reservation, and Floor Release procedures are presented. In the proposed PoC client architecture, most standard VoIP modules are reused for the PoC service, and the VoIP software can be easily extend to support PoC service by adding the GLMS Access module and the Floor Control module. A PoC client prototype has been implemented in the Industrial Technology Research Institute (ITRI) and National Chiao Tung University (NCTU) Joint Research Center.

Chapter 6

Conclusions and Future Works

In this dissertation, we investigated four design issues in the UMTS protocol layers. This chapter summarizes our study and contributions, and briefly discusses the future directions.

6.1 Conclusions

In this dissertation, we discussed four UMTS design issues. In Chapter 2, a WLAN and UMTS interworking solution called *WLAN GPRS Support Node* (WGSN) was proposed. The WGSN network is a loosely-coupled architecture that satisfies the Scenario 3 features defined in the 3GPP TS 22.934. We showed that the UMTS mechanisms can be re-used for the WLAN user authentication and access control without modifying the existing UMTS components. Then we focused on the performance of the WGSN push mechanism. Our study indicates that the SIP UA activation should be completed within 0.03125 times of the inter-call arrival time. To shorten the SIP UA activation time, we also suggested that the push operation should be implemented by high priority short message service.

In Chapter 3, we studied the *Authentication Vector* (AV) usage mechanism in the UMTS core network. The AV usage mechanism reduces the number of access to the HSS/AuC at the cost of extra AV storage in the SGSN. We proposed analytic and simulation models to investigate its performance. Our study indicates that the *Reservation Timeout* (RT) period significantly affects the re-entrance probability, extra AV storage, and the number of accesses to the HSS/AuC. These analytic results provide guidelines for the mobile operators to implement the AV usage mechanism.

In Chapter 4, we proposed an one-pass authentication, in which the redundant steps in the GPRS authentication and IMS authentication procedures are removed. The one-pass authentication saves up to 50% of the network traffic generated by the IMS authentication, and saves 50% of the storage for buffering the AVs. We also formally proved that the IMS user is correctly authenticated in the one-pass authentication.

In Chapter 6, we described the design and implementation of the client for OSA-based PoC service, which provides a walkie-talkie like service in the UMTS network. In the proposed PoC client architecture, most standard VoIP modules are reused for the PoC service, and the VoIP software can be easily extended to support PoC service.

6.2 Future Works

Based on the research results in this dissertation, the following design issues on the UMTS network can be investigated further.

UMTS and WLAN interworking: Depending on the business strategies, the Scenario 4 features defined in 3GPP TS 22.934 are required in long-term commercial operation. In Scenario 4, the service continuity of the PS domain is defined. Specifically, when handover occurs between WLAN and UMTS, the user might experience a data stream interruption, but he/she needs not to involve in the re-establishment of the sessions. To satisfy the Scenario 4 features, WGSN needs to be enhanced in the following three aspects.

(1) The MS must have the capability for detecting the radio strength of the access network, and a decision mechanism is required to select the appropriate access technology. When handover occurs, the MS automatically switches between the UMTS module and the WLAN module.

(2) The interwork is responsible for re-establishing the session without user involvement. One possible solution is to utilize Mobile IP, which has the advantage of minimal modification to the WGSN. The WGSN plays as a foreign

agent, and the MS needs to support Mobile IP client. Home agent is implemented as part of the operator's home network.

(3) *Quality of Service* (QoS) is an essential issue for service continuity. Since 3G and WLAN networks have different capabilities and characteristics, the user would gain different QoS grades in different networks. Therefore, QoS adaptation is required during system switching.

Application level authentication: The OSA platform provides a flexible service creation and execution environment for the third party to run their own applications on the UMTS network. Before the third party provides the services, authentication procedure may be performed between the application server and the MS. However, the MS has been verified at the GPRS level and maybe also be verified in the IMS level. To reduce the redundant authentication, we can further discuss the integration of the authentication procedures in the GPRS, IMS, and application levels. The integrated solution has to correctly authenticate the user in each of the levels.

Appendix A

Notation

This appendix lists the notation used in this dissertation.

A.1 Notation for Chapter 2

- t_0: the inter call arrival time
- t_1: the *T1* timeout period
- t_2: the SIP UA activation time
- $f_{t_0}(t_0)$: the density function for t_0
- $f_{t_2}(t_2)$: the density function for t_2
- λ: the call arrival rate for an MS
- 1/μ: the expected *T1* timeout period
- 1/γ: the expected SIP UA activation time
- $E[N]$: the expected number of the lost calls during the activation period
- V_μ: the variance of Gamma distributed SIP UA activation times

A.2 Notation for Chapter 3

- $L_j=(x_j,y_j)$: the coordinate of the SGSN where the MS resides after *j* movements
- *K*: the number of AVs obtained from the HSS/AuC in an ADR

- T: the reservation timeout period
- t_i: the SGSN residence time of the i-th visit to L_0
- N_i: the number of ADRs performed at the i-th visit to L_0
- R_i: the number of unused AVs left when the MS leaves L_0 at the i-th visit
- λ: the arrival rate of UAR
- 1/μ: the expected SGSN residence time
- $P_{j:}$ the probability that the MS returns to L_0 at the j-th movement (i.e., $P_j = \Pr[L_j = L_0]$)
- Q_{2n}: the probability that the MS first returns to L_0 at the $2n$-th movement

  (i.e., $Q_{2n} = \Pr[L_{2n} = L_{0,}\ L_{2l} \neq L_0$ for $0<l<n$])
- t^*_j: the residence time at SGSN L_j
- t_{r}: the period between when the MS leaves L_0 and when it returns.

  That is, $t_{\mathrm{r}} = t^*_1 + t^*_2 + \ldots + t^*_{(2n-1)}$, where $L_{2n} = L_{0,}$ and $L_{2l} \neq L_0$ for $0<l<n$
- $F(2n,\ t_{\mathrm{r}})$: the cumulative distribution function that the MS returns to L_0 at $2n$-th movement at time t_{r}
- S_k: a state indicating that there are k AVs stored in SGSN L_0
- $\pi_{k:}$ the probability that k AVs are stored in SGSN L_0 while the system is at the steady state. That is, $\pi_k = \Pr[\text{the process is at state } S_k]$
- $H(\Delta s) = (p_{i,j})$: transition probability matrix, where $p_{i,j}$ presents the transition probability from S_i to S_j
- H: transition rate matrix
- ψ_T: the expected AV storage consumed when T>0
- $\Theta(N_i, R_{i-1}, R_i, t_i)$: the probability that at the i-th visit to L_0, R_{i-1} unused AVs are stored in SGSN L_0, where R_0=0. The residence time of the i-th visit to L_0 is t_i. During t_i, N_i ADRs occur, and R_i unused AVs are left when the MS leaves L_0

- $\varphi(N_i, R_{i-1})$: the probability that when the MS enters L_0 at the i-th visit, R_{i-1} unused AVs is stored in SGSN L_0, and N_i ADRs are performed during the residence time of i-th visit
- $\Gamma(N_i, T)$: the probability that N_i ADRs occur at the i-th visit to L_0, where $i \geq 1$, N_i,>0, and the length of the RT period is T
- $E[N|K]$: the expected number of ADRs performed during one visit to SGSN L_0, were K AVs are obtained in one ADR
- V_S: the variance of Gamma distributed SGSN residence times
- α: the probability that the MS re-enters L_0 within timeout period T
- β: the expected AV storage consumed when T>0, which is normalized by the expected AV storage consumed when T=0
- δ: the number of ADRs performed in one visit to SGSN L_0 as comparing with that when K=1. That is $\delta = \dfrac{E[N \mid K]}{E[N \mid K = 1]}$

Reference

[1] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Messaging Service (MMS); Stage 1. 3GPP TS 22.140, v6.7.0, 2005.

[2] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service requirements for the Internet Protocol (IP) multimedia core network subsystem; Stage 1. 3GPP TS 22.228, v7.3.0, 2005.

[3] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Feasibility Study on 3GPP System to Wireless Local Area Network (WLAN) Interworking. 3GPP TR 22.934, v. 6.2.0, 2003.

[4] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Technical realization of the Short Message Service (SMS). 3GPP TS 23.040, v6.7.0, 2006.

[5] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service Description; Stage 2. 3GPP TS 23.060, v 7.0.0, 2006.

[6] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; Virtual Home Environment (VHE)/Open Service Access (OSA). 3GPP TS 23.127 V6.1.0, 2004.

[7] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; IP Multimedia Subsystem (IMS); Stage 2. 3GPP TS 23.228 V7.3.0, 2003.

[8] 3GPP. 3rd Generation Partnership Project; Technical Specification Core Network

and Terminals; IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signaling flows and message contents. 3GPP TS 29.228 V7.1.0, 2006.

[9] 3GPP. 3rd Generation Partnership Project; Technical Specification Core Network and Terminals; Cx and Dx interfaces based on the Diameter protocol; Protocol details. 3GPP TS 29.229 V7.1.0, 2006.

[10] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; 3G Security; Security Architecture. 3GPP TS 33.102, V7.0.0, 2005.

[11] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; 3G security; Access security for IP-based services. 3GPP TS 33.203 V7.1.0, 2006.

[12] Aboba, B., and Beadle, M. The Network Access Identifier. Internet Engineering Task Force, RFC 2486, 1999.

[13] Biggs, B. A SIP Application Level Gateway for Network Address Translation. Internet Engineering Task Force, Draft draft-biggs-sip-nat-00, 2000.

[14] Blunk, L., and Vollbrecht, J. PPP Extensible Authentication Protocol (EAP). Internet Engineering Task Force, RFC 2284, 2003.

[15] Chen, W.-E., Wu, Q.-C., Pang, A.-C., and Lin, Y.-B. Design of SIP application level gateway for UMTS. *Design and Analysis of Wireless Networks*, Y. Pan and Y. Xiao (editors), Nova Science Publishers, Hardbound, 2004.

[16] Chlamtac, I., Fang, Y., and Zeng, H. Call Blocking Analysis for PCS Networks under General Cell Residence Time. *IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, September 1999.

[17] Chou, C.-M., Hsu, S.-F., Lee, H.-Y., Lin, Y.-C., Lin, Y.-B., and Yang, R.S. CCL OSA: A CORBA-based Open Service Access System. Accepted and to appear in *International Journal of Wireless and Mobile Computing*.

[18] Droms, R. Dynamic Host Configuration Protocol. Internet Engineering Task Force, RFC 2131,1997.

[19] de Laat, C., Gross, G., Gommans, L., Vollbrecht, J., and Spence, J. Generic AAA Architecture. Internet Engineering Task Force, RFC 2903, 2000.

[20] ETSI. Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Requirements and Architectures for Interworking between HIPERLAN/2 and 3rd Generation Cellular Systems. ETSI TR 101 957, 2001.

[21] ETSI. Digital cellular telecommunications system (Phase 2+); Mobile Application Part (MAP) specification. GSM 09.02, v. 7.5.0, 2000.

[22] FarEasTone Telecom. *Private communication*. 2003.

[23] Feng, V., et al. System Description Document of WLAN-based GPRS Support Node (WGSN). ITRI and NCTU Joint Research Center, Technical Report, 2003.

[24] Feng, V., Lin, Y.-B., and Chou, S.-L. Design and Implementation of A Softswitch for Third Generation Mobile All-IP Network. *Wireless Communications and Mobile Computing Journal*, 359-381, 2004.

[25] Feng, W.-S., Wu, L.-Y., Lin, Y.-B., and Chen, W.-E. WGSN: WLAN-based GPRS Support Node with Push Mechanism. *The Computer Journal*, 47(4):405-417, 2004.

[26] Fielding, R. et al. Hypertext Transfer Protocol - HTTP/1.1. Internet Engineering Task Force, RFC 2616, 1999.

[27] Haverinen, H., Mikkonen, J., and Takamaki, T. Cellular access control and charging for mobile operator wireless local area networks. *IEEE Wireless Communications*, 9(6): 52 -60, 2002.

[28] Holma, H., and Toskala, A. *WCDMA for UMTS*. John Wiley & Sons, 2002.

[29] Hsu, S.-F., Lin, Y.-C., Lin, Y.-B., and Yang, J.-S. An OSA Application Server for Mobile Services. ITRI and NCTU Joint Research Center, Technical Report, 2005.

[30] Hung, H.-N., Lin, Y.-B., Lu, M.-K., and Peng, N.-F. A Statistic Approach for Deriving the Short Message Transmission Delay Distributions. *IEEE Transactions on Wireless Communications*, 3(6):2345-2352, 2004.

[31] Institute of Electrical and Electronics Engineers. IEEE Standard for Local and Metropolitan Area Networks-Port-Based Network Access Control. IEEE Std 802.1X-2001, 2001.

[32] Kelly, F.P. *Reversibility and Stochastic Networks*. John Wiley & Sons, Location, 1979.

[33] Lin, Y.-B., Chen, M.-F., and Rao, H. C.-H. Potential fraudulent usage in mobile telecommunications networks. *IEEE Transactions on Mobile Computing*, 1(2): 123-131, 2002.

[34] Lin, Y.-B., and Chen, Y.-K. Reducing Authentication Signaling Traffic in Third Generation Mobile Network. *IEEE Transaction on Wireless Communication*, 2(3): 491-501, 2003

[35] Lin, Y.-B., and Chlamtac, I. *Wireless and Mobile Network Architectures*. Wiley, London, 2001.

[36] Lin, Y.-B., Huang, Y.-R., Chen, Y.-K., and Chlamtac, I. Mobility management: From GPRS to UMTS. *Wireless Communication Mobile Computing*, 1(4): 339–360, 2001.

[37] Lin, Y.-B., Haung, Y.-R., Pang, A.-C., and Chlamtac, I. All-IP Approach for UMTS Third Generation Mobile Networks. *IEEE Network*, 16(5): 8-19, 2002.

[38] Lin, Y.-B., Lee, P.-J., and Chlamtac, I. Dynamic periodic location area update in mobile networks. *IEEE Transactions on Vehicular. Technology*, 51(6):1494–1501, 2002.

[39] Lin, Y.-B., Lo, Y.-C., and Rao, C.-H. A Push Mechanism for GPRS Supporting Private IP Addresses. *IEEE Communications Letters*, 7(1): 24-26, 2003.

[40] Lin, Y.-B. and Pang, A.-C. *Wireless and Mobile All-IP Networks*. John Wiley & Sons, 2005.

[41] MRTG: Multi Router Traffic Grapher. http://people.ee.ethz.ch/~oetiker/webtools/mrtg/

[42] Open Mobile Alliance. Push to Talk over Cellular Architecture. OMA-AD-PoC_V1_0-20050428-C Candidate Version 1.0, April 2005.

[43] oRTP - a Real-time Transport Protocol stack under LGPL. http://www.linphone.org/ortp/

[44] Perkins, C. IP Mobility Support for IPv4. Internet Engineering Task Force, RFC 3344, 2002.

[45] Rao, C.H., Chang, D.-F., and Lin, Y.-B. iSMS: An Integration Platform for Short Message Service and IP Networks. *IEEE Network*, 15(2): 48-55, 2001.

[46] Rosenberg, J., et al. SIP: Session Initiation Protocol. Internet Engineering Task Force, RFC 3261, 2002.

[47] Ross, S.M. *Stochastic Processes*. John Wiley & Sons, Location, 1996.

[48] Salkintzis, A.K., Fors, C., and Pazhyannur, R. WLAN-GPRS integration for next-generation mobile data networks. *IEEE Wireless Communications*, 9(5): 112 -124, 2002.

[49] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. RTP: A Transport Protocol for Real-Time Applications. Internet Engineering Task Force, RFC 3550, 2003.

[50] Stewart, R., et al. Stream Control Transmission Protocol. Internet Engineering Task Force, RFC 2960, 2000.

[51] The eXtended osip library. http://savannah.nongnu.org/projects/exosip/

[52] The GNU oSIP library. http://www.gnu.org/software/osip/osip.html

[53] Zeng, H., and Chlamtac, I. Handoff Traffic Distribution in Cellular Networks. *IEEE*

Wireless Communications and Networking Conference (WCNC), New Orleans, September 1999.

Curriculum Vita

Lin-Yi Wu was born in Taipei, Taiwan, R.O.C., in 1976. She received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1999 and 2001, respectively. Her current research interests include heterogeneous networks integration, personal communications services, and mobile computing.

Publication Lists

Journal Papers

1. **Wu, L.-Y.**, and Lin, Y.-B. Authentication Vector Management for UMTS. submitted to *IEEE Transactions on Wireless Communications*.
2. **Wu, L.-Y.**, Tsai, M.-H., Lin, Y.-B., and Yang, J.-S. A Client-Side Design and Implementation for Push to Talk over Cellular Service. Accepted and to appear in *Wireless Communications & Mobile Computing Journal*.
3. Chang, M.-F., **Wu, L.-Y.**, and Lin, Y.-B. Performance Evaluation of a Push Mechanism for WLAN and Mobile Network Integration. *IEEE Transactions on Vehicular Technology*, 55(1): 380-383, 2006.
4. Lin, Y.-B., Chang, M.-F., Hsu M.-T., and **Wu, L.-Y.** One-Pass GPRS and IMS Authentication Procedure for UMTS. *IEEE Journal on Selected Areas in Communications*, 23(6): 1233-1239, 2005.
5. **Wu, L.-Y.**, Chang, M.-F., and Lin, Y.-B. Corrigendum: WGSN: WLAN-based GPRS Support Node with Push Mechanism. *The Computer Journal*, 47(5): 624, 2004.
6. Feng, W.-S., **Wu, L.-Y.**, Lin, Y.-B., and Chen, W.-E. WGSN: WLAN-based GPRS Support Node with Push Mechanism. *The Computer Journal*, 47(4): 405-417, 2004.

Conference Papers

1. **Wu, L.-Y.**, Chang, M.-F., and Lin, Y.-B. Modeling the Push Mechanism for WGSN. *Internal Workshop on Broadband Wireless Services and Applications 2004* (BroadWISE 2004), San Jose, U.S., Oct. 2004.
2. Feng, W.-S., **Wu, L.-Y.**, Lin, Y.-B., and Chen, W.-E. WGSN: Wireless LAN based GPRS Support Node. *National Computer Symposium* 2003, TaiChung, Taiwan, Dec. 2003.

Awards

1. Best Paper Award, Computer Society of the Republic of China, 2003.
2. Best Paper Award, National Computer Symposium 2003 (NCS2003).