國立交通大學

資訊工程系

碩士論文

第七號信令系統信令在MTP與SCTP實作上的比較與分析

Comparison of MTP and SCTP for SS7 Signaling

研 究 生：黃偉哲

指導教授：林一平 教授

中華民國九十三年七月

第七號信令系統信令在 MTP 與 SCTP 實作上的比較與分析

Comparison of MTP and SCTP for SS7 Signaling

研究生：黃偉哲　　Student：Wei-Che Huang

指導教授：林一平 博士　　Advisor：Dr. Yi-Bing Lin

國立交通大學
資訊工程系
碩士論文

A Thesis
Submitted to Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science and Information Engineering

July 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年七月

第七號信令系統信令在MTP與SCTP實作上的比較與分析

學生：黃偉哲　　指導教授：林一平 博士

國立交通大學資訊工程學系碩士班

摘要

第七號信令系統信令提供行動電信通訊網路控制與管理的功能。在現今的行動網路，第七號信令系統信令是實作在以 *Message Transfer Part* 為基礎的網路上。但在新的 *Universal Mobile Telecommunication System* (UMTS) all-IP 的架構下，第七號信令系統信令將會被應用傳輸在 IP 網路。本論文即實作第七號信令系統信令在傳統以 *Message Transfer Part* 為基礎的網路上的傳輸與在 UMTS all-IP 架構下 *Stream Control Transmission Protocol* 上的傳輸。我們並從訊息格式、連線準備與資料傳輸/回報這三個觀點來比較這兩種不同的方法，並以 3GPP TS 23.060 所定義的 Send Authentication Info 程序來測量分析這兩種方法實作的效能。

Comparison of MTP and SCTP for SS7 Signaling

Student: Wei-Che Huang Advisor: Prof. Yi-Bing Lin

Department of Computer Science and Information Engineering
National Chiao Tung University

Abstract

Signaling System Number 7 (SS7) signaling provides control and management functions in the mobile telecommunications network. Traditional SS7 signaling is implemented in *Message Transfer Part-based* network, which is utilized in the existing mobile networks. In *Universal Mobile Telecommunication System* (UMTS) all-IP architecture, the SS7 signaling transport will be carried by IP-based network. In this paper, we design and implement *Message Transfer Part-based* SS7 signaling transport and *Stream Control Transmission Protocol-based* SS7 signaling transport for UMTS all-IP network. We compare of these two approaches in three perspectives: message format, connection setup, and data transmission/ack. We also illustrate the performance of our implementations by using the Send Authentication Info procedure defined in 3GPP Technical Specification 23.060.

Acknowledgement

I would like to express my sincere thanks to my advisor, Prof. Yi-Bing Lin. Without his supervision and perspicacious advice, I can not complete this thesis.

Thanks also to the colleagues in the Laboratory 117. I especially like to express my thanks to Meng-Hsun Tsai for his helpful comments and suggestions. Furthermore, thanks to all my friends in National Chiao Tung University for our happy reminiscences.

Finally, I am grateful to my dear parents, my sister, my brother and all whom I love and who love me for their unfailing love and firm support in these years.

Contents

List of Tables

List of Figures

Acronym List

A-link	Access Link
A_rwnd	Advertised Receiver Window Credit
ACK	Acknowledgment
ASE	Application Service Element
ASN.1	Abstract Syntax Notation One
B-link	Bridge Link
BIB	Backward Indicator Bit
BSN	Backward Sequence Number
C-link	Cross Link
CCS	Common Channel Signaling
CK	Check Bits
CPCI	Compact Peripheral Component Interconnect
D-link	Diagonal Link
DoS	Denial-of-Service
DPC	Destination Point Code
DSM	Dialogue State Machine
E-link	Extended Link
F-link	Fully-associated Link
FIB	Forward Indicator Bit
FISU	Fill-in Signal Unit
FSM	Finite State Machine
FSN	Forward Sequence Number
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
GSN	GPRS Support Node
GTT	Global Title Translation
HLR	Home Location Register
HOL	Head of the Line
IETF	Internet Engineering Task Force
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
ISM	Invoke State Machine
ISUP	Integrated Services Digital Network User Part
LI	Length Indicator

LSSU	Link Status Signal Unit
MAP	Mobile Application Part
MP	Message Priority
MSU	Message Signal Unit
MTP	Message Transfer Part
MTP1	Message Transfer Part Level 1
MTP2	Message Transfer Part Level 2
MTP3	Message Transfer Part Level 3
MTU	Maximum Transmission Unit
M3UA	MTP3 User Adaptation Layer
NI	Network Indicator
OMAP	Operations, Maintenance, and Administration Part
OPC	Originating Point Code
OS	Operating System
PCI	Peripheral Component Interconnect
PCR	Preventive Cyclic Retransmission
PDN	Packet Data Network
POTS	Plain Old Telephone Service
PROM	Programmable Read Only Memory
PS	Packet Switched
PSTN	Public Switched Telephone Network
SACK	Selective Acknowledgement
SAL	Service Access Layer
SAP	Service Access Point
SCCP	Signaling Connection Control Part
SCLC	SCCP Connectionless Control
SCMG	SCCP Management
SCOC	SCCP Connection-oriented Control
SCP	Service Control Point
SCRC	SCCP Routing Control
SCTP	Stream Control Transmission Protocol
SGSN	Serving GPRS Support Node
SI	Service Indicator
SIF	Signaling Information Field
SIGTRAN	Signaling Transport
SIO	Service Indicator Octet
SLC	Signaling Link Code
SLS	Signaling Link Selection

SS7	Signaling System Number 7
SSN	Stream Sequence Number
SSP	Service Switching Point
STP	Signal Transfer Point
SU	Signal Unit
TCAP	Transaction Capabilities Application Part
TCB	Transmission Control Block
TCP	Transmission Control Protocol
TSM	Transaction State Machine
TSN	Transmission Sequence Number
UA	User Adaptation
UMTS	Universal Mobile Telecommunication System

Chapter 1 Introduction

The evolution of radio and mobile core network technologies over the last two decades has enabled the development of the mobile telecommunications services to provide mobile users with voice, data, and multimedia services at any time, any place, and in any format. Business opportunities for such services are tremendous and several mobile telecommunications networks have been developed to meet rapid growth prompted by heavy market demand. Most of them are connected to the Public Switched Telephone Network (PSTN) to provide access to wireline telephones. In the broadest sense, mobile telecommunications services include all forms of radio-telephone communication that are interconnected to the PSTN, including cellular radio and aeronautical public correspondence. Global System for Mobile Communication (GSM) is the one of the most famous examples of high-tier digital cellular systems (mobile phone systems) for widespread vehicular and pedestrian services. Furthermore, wideband wireless systems (also called third-generation system) have been developed to accommodate Internet and multimedia services. [3]

Not until recently, only a small portion of GSM subscribers used data services, because existing GSM systems do not support easy access, high data rate, and attractive prices. GSM operators must offer better services to stimulate the demand. The solution is the General Packet Radio Service (GPRS) [1,3]. GPRS reuses the existing GSM infrastructure to provide end-to-end packet-switched services and the GPRS core network is anticipated to evolve as the core network for the third-generation mobile systems as well. The GPRS service had been well deployed now and bridges second-generation systems (i.e., GSM) into third-generation system. In the following sections, we describe the protocols and management software that make mobile telecommunications network run, and introduce GPRS network signaling.

1.1 An Signaling System Number 7 Overview

Common Channel Signaling (CCS) [4] is a signaling method that provides control and management functions in the telephone network. CCS consists of supervisory functions, addressing, and call information provisioning. A CCS channel conveys messages to initiate and terminate calls, determines the status of some part of the network, and controls the amount of traffic allowed. CCS uses a separate out-of-band signaling network to carry signaling messages. Signaling System Number 7 (SS7) [3,5] is a CCS system for an improvement to the earlier signaling systems, which lacked the sophistication required to deliver much more than Plain Old Telephone Service (POTS). Figure 1.1.1 shows the typical SS7 network where the dashed lines represent the signaling links and the solid line represents a trunk. The SS7 network consists of three distinct components:

Service Switching Point (SSP) is a telephony switch interconnected by SS7 links. The SSPs perform call processing on calls that originate, tandem, or terminate at that node.

Signal Transfer Point (STP) is a switch that relays SS7 messages between network switches and databases. Based on the address fields of the SS7 message, the STPs route the messages to the correct outgoing signaling link. To meet the stringent reliability requirements, STPs are provisioned in mated pairs, as shown in Figure 1.1.1

Service Control Point (SCP) contains databases for providing enhanced services. An SCP accepts queries from an SSP and returns the requested information to the SSP.

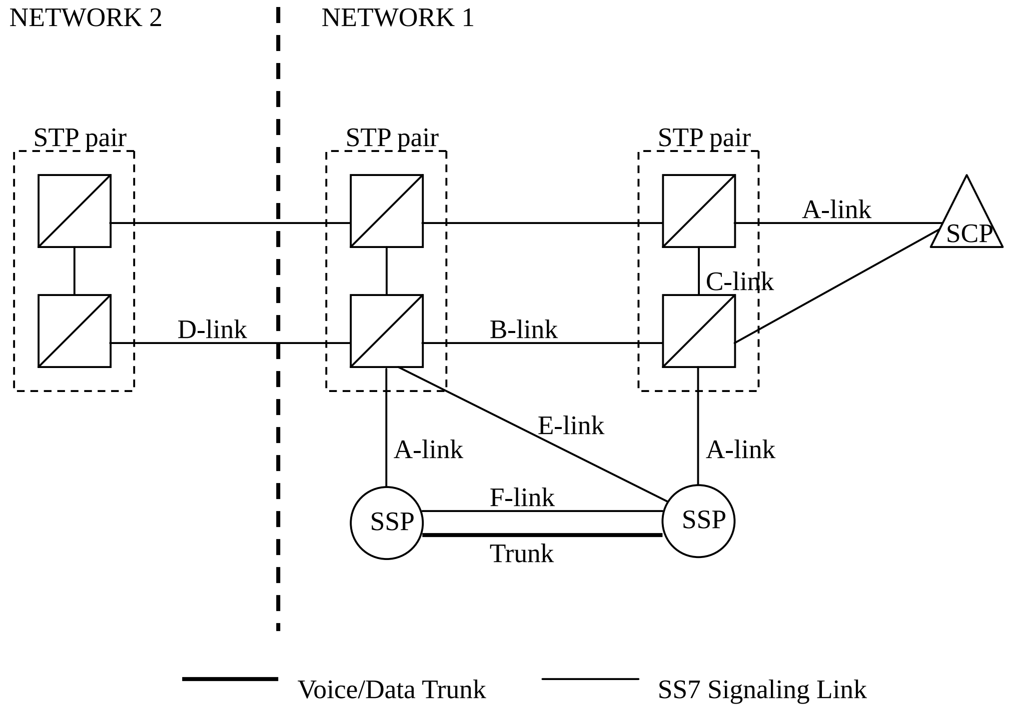


Figure 1.1.1: SS7 Architecture

In an SS7 network, the trunks (voice circuits) connect SSPs to carry user data/voice information. The signaling links connect SCPs to STPs, and STPs to SSPs. The SSPs and SCPs are connected indirectly through STPs. There are six types of SS7 signaling link as shown in Figure 1.1.1:

Access Links (A-links) connect the SSP/STP pairs or the SCP/STP pairs. Each SSP and SCP will have a minimum of one A-link to each STP pair.

Bridge Links (B-links) connect STPs in different paris. B-links are deployed in a quad arrangement with 3-way path diversity.

Cross Links (C-links) connect mated STPs in a pair.

Diagonal Links (D-links) are the same as the B-links except that the connected STPs belong to different SS7 networks, for example, one in the PSTN and one in mobile telecommunications network.

Extended Links (E-links) provide extra connectivity between an SSP to STPs other than its home STP.

Fully-associated Links (F-links) connect SSPs directly.

In the existing SS7 networks, E-links and F-links are not widely deployed. The signaling links are monitored such that the failure links are automatically detected and the traffic load is shared by the active links.

The SS7 protocol follows the OSI layered protocol, which was designed for ease of adding new features. The SS7 protocol layers and the corresponding OSI layers are shown in Figure 1.1.2. Theses layers are described as follows:

- The Message Transfer Part (MTP) [6] consists of three levels corresponding to the OSI physical layer, data link layer, and network layer, respectively. The MTP level 1 (MTP1) defines the physical, electrical, and functional characteristics of the signaling links connecting SS7 components. The MTP level 2 (MTP2) provides reliable transfer of signaling messages between two directly connected signaling points. The MTP level 3 (MTP3) provides the functions and procedures related to message routing and network management.
- The Signaling Connection Control Part (SCCP) [7] provides additional functions such as Global Title Translation (GTT) to the MTP. The MTP utilizes GTT to transfer noncircuit-related signaling information such as mobile telecommunications service registration and cancellation.
- The Transaction Capabilities Application Part (TCAP) [8] provides the capability to exchange information between applications using noncircuit-related signaling.

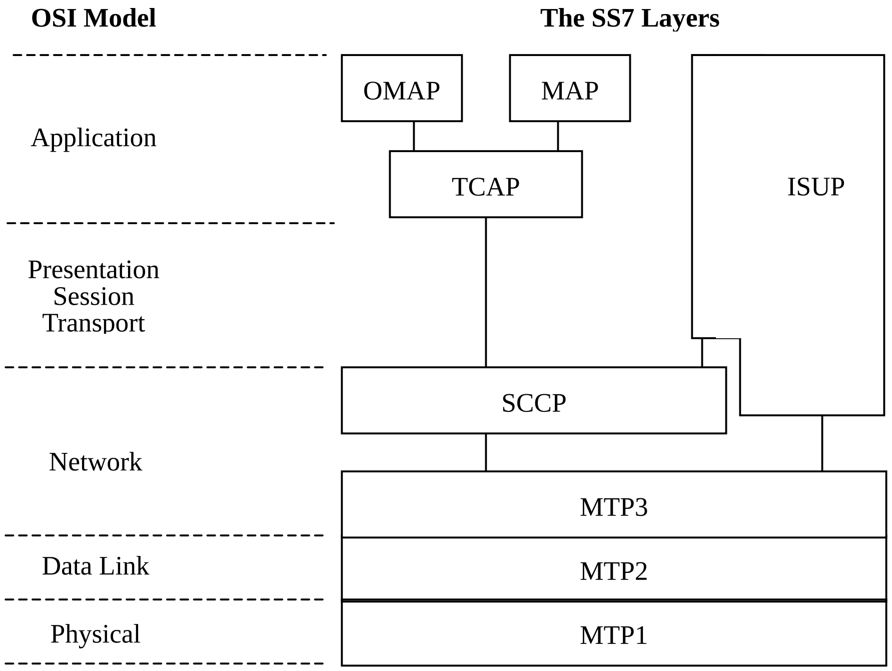


Figure 1.1.2: SS7 Signaling Protocol

- The Integrated Services Digital Network User Part (ISUP) [9] establishes circuit-switched network connections (e.g., for call setup). Passalong signaling service sends the signaling information to each switching point involved in a call connection.
- The Operations, Maintenance, and Administration Part (OMAP) [10] is an application of TCAP. Details of general OMAP are beyond the scope of this paper.
- The Mobile Application Part (MAP) is an application of TCAP to support mobile roaming management. GSM MAP [2] is implemented at this layer and is elaborated in Section 1.3.

The GSM MAP protocol is implemented in the MAP as an application of the TCAP. The wireless call setup/release are completed by using the ISUP. The MTP and the SCCP provide routing services between a mobile telecommunications network and the PSTN.

1.2 An SCTP Overview

Signaling Transport (SIGTRAN) is a working group of the Internet Engineering Task Force (IETF) [17]. The primary purpose of this working group is to address the issues regarding the transport of the packet-based PSTN signaling (i.e., the SS7 signaling) over IP networks. SIGTRAN defines not only the architecture but also the definition of a suite of protocols to carry SS7 messages over IP. This suite of protocols consists of a new transport protocol, the Stream Control Transmission Protocol (SCTP) [14], and a set of User Adaptation (UA) layers, which provides the same services of the lower layers of SS7. For an example, MTP3 User Adaptation Layer (M3UA) [15] provides the same MTP3 services to the MTP3-user (i.e., SCCP or ISUP).

Transmission Control Protocol (TCP) [16] provides reliable data transfer in IP networks. However, TCP is not an ideal protocol for SS7 signaling transport for the following reasons.

- TCP provides strict order-of-transmission. This feature is not required in SS7 transport, but will cause the Head of the Line (HOL) blocking problem. HOL blocking occurs when the messages are transferred over a single TCP connection. If a message gets lost, other in-sequence messages must wait until the lost message has been retransmitted.
- The TCP socket does not support multi-homing. A host is considered multi-homed if there is more than one IP address that can be used as a destination address to reach that host. The multi-homed host is often equipped with multiple network interfaces, and the IP addresses are allocated and processed in different network interfaces. A TCP application can only bind a single IP address to the TCP connection. When the network interface failed, the connection is lost and probably cannot be reestablished.
- TCP is vulnerable to blind Denial-of-Service (DoS) attacks such as flooding SYN

attacks.

Like TCP, SCTP provides reliable IP connection. SCTP employs TCP-friendly congestion control (including slow-start, congestion avoidance, and fast retransmit). Unlike TCP, SCTP provides selective acknowledgments for packet loss recovery, and message-oriented data delivery service. Moreover, SCTP offers new delivery options (ordered or unordered) and features that are particularly desirable for SS7 signaling. An example is multi-homing. In multi-homing, an SCTP association (i.e., a connection) allows the SCTP endpoints of the association to have multiple IP addresses for reliability enhancement (we will elaborate on this feature in Section 3.2). Another useful feature is multi-streaming. Independent delivery among data streams prevents the HOL blocking problem that causes additional delay (we will elaborate on this feature in Section 3.2). SCTP also features a four-way handshake to establish an association, which makes it resistant to blind DoS attacks and thus improves overall protocol security (we will elaborate on this feature in Section 3.2).

Figure 1.2.1 illustrates the SCTP packet format. The SCTP packet begins with an SCTP common header. This header includes Source Port Number (Figure 1.2.1 (1)), Destination Port Number (Figure 1.2.1 (2)), Verification Tag (Figure 1.2.1 (3)), and Checksum (Figure 1.2.1 (4)). Source Port Number is the SCTP sender’s port number used to identify the association to this SCTP packet. Destination Port Number is the SCTP port number where this packet is destined. Verification Tag is used to validate the sender of the SCTP packet. Checksum is used by the Adler32 algorithm [14] to maintain the packet’s integrity. The remaining part of the SCTP packet consists of one or more chunks which contain either control or data information. Each Chunk has a chunk header that includes Chunk Type (Figure 1.2.1 (5)), Chunk Flags (Figure 1.2.1 (6)), Chunk Length (Figure 1.2.1 (7)), and Chunk Value (Figure 1.2.1 (8)). The Chunk Type filed identifies the type of information contained in the Chunk Value field. Figure 1.2.2 lists the defined chunk types. These types determine the usage of the Chunk Flags field and the Chunk Value field. The Chunk Value field contains the actual information to be transferred in the chunk.

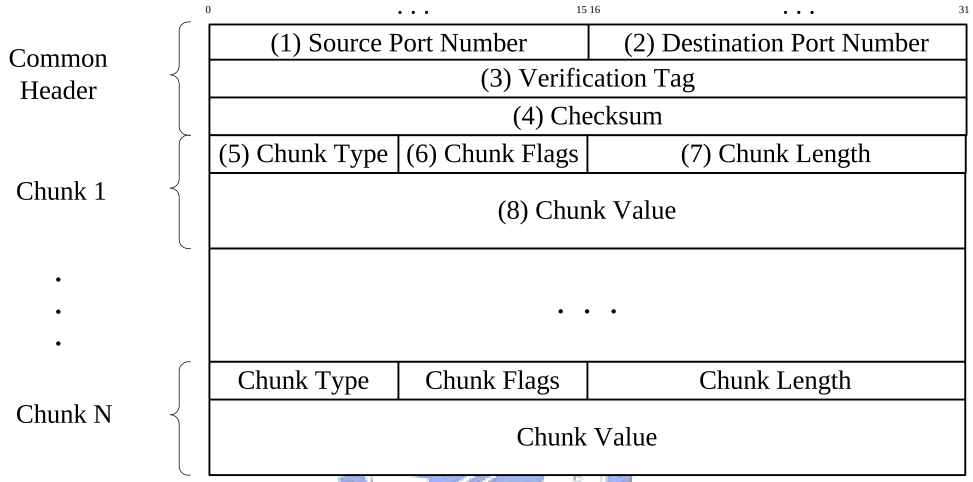
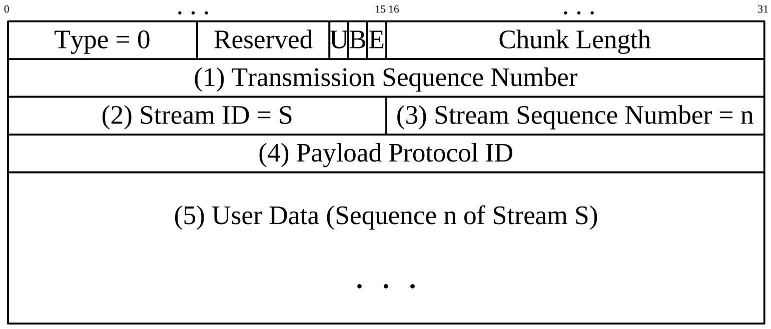


Figure 1.2.1: SCTP Packet Format

ID Value	Chunk Type
--------	---------------
0	- Payload Data (**DATA**)
1	- Initiation (**INIT**)
2	- Initiation Acknowledgement **(INIT ACK**)
3	- Selective Acknowledgement (**SACK**)
4	- Heartbeat Request (**HEARTBEAT**)
5	- Heartbeat Acknowledgement **(HEARTBEAT ACK**)
6	- Abort (**ABORT**)
7	- Shutdown (**SHUTDOWN**)
8	- Shutdown Acknowledgement **(SHUTDOWN ACK**)
9	- Operation Error (**ERROR**)
10	- State Cookie (**COOKIE ECHO**)
11	- Cookie Acknowledgement **(COOKIE ACK**)
12	- Reserved for Explicit Congestion Notification Echo (**ECNE**)
13	- Reserved for Congestion Window Reduced (**CWR**)
14	- Shutdown Complete (**SHUTDOWN COMPLETE**)

Figure 1.2.2: SCTP Chunk Types

The DATA chunk format is illustrated in Figure 1.2.3. The DATA chunk header includes Type, Unordered bit, Beginning fragment bit, Ending fragment bit, Chunk Length, Transmission Sequence Number (TSN; Figure 1.2.3 (1)), Stream Identifier (Stream ID; Figure 1.2.3 (2)), Stream Sequence Number (SSN; Figure 1.2.3 (3)), Payload Protocol Identifier (Payload ID; Figure 1.2.3 (4)), and User Data (Figure 1.2.3 (5)). The Unordered bit is used to indicate if the DATA chunk uses an ordered or unordered delivery service. In the ordered service, the receiver receives messages in the same order as they were sent by the sender. The unordered service delivers messages to the receiver as soon as they are received from the network, regardless of the order in which the sender produced them. The Beginning and Ending fragment bits are used for the SCTP message fragmentation when the message size is larger than the Maximum Transmission Unit (MTU) allowed in the transport path. Stream ID identifies the stream to which the following User Data belongs. The TSN and the SSN provide two separate sequence numbers on every DATA chunk. The TSN is used for per-association reliability. Each DATA chunk is assigned a TSN to permit the receiving SCTP endpoint to acknowledge the received packets and detect duplicate deliveries. The SSN is used for per-stream ordering. Payload Protocol ID is used to identify the type of information being carried in this DATA chunk. The other control chunks carry information needed for association functionality, such as association establishment, association termination, data acknowledgment (ACK), failure detection and recovery, etc.



Type = 0 | Reserved | U | B | E | Chunk Length

(1) Transmission Sequence Number

(2) Stream ID = S | (3) Stream Sequence Number = n

(4) Payload Protocol ID

(5) User Data (Sequence n of Stream S)

. . .

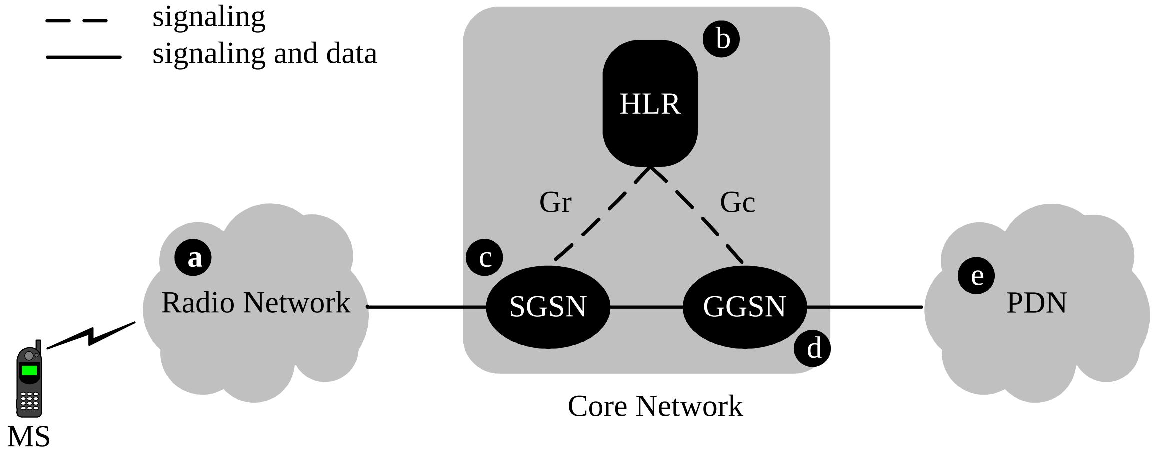
U: Unordered bit
B: Beginning fragment bit
E: Ending fragment bit

Figure 1.2.3: DATA Chunk Format

The lifetime of an SCTP association consists of three phases [14]: association establishment, data transfer, and association termination. SCTP uses a four-way handshake and cookie mechanism to establish an association that prevents blind SYN attacks (to be elaborated in Section 4.2). After the association is established, the two SCTP endpoints can transfer data by using the DATA chunk (to be elaborated in Section 4.3). The association termination can be initiated by either one of the endpoints engaged in the association and is a three-way handshake process.

1.3 GPRS Network Signaling

General Packet Radio Service (GPRS) [1] supports high-speed Packet Switched (PS) data for accessing versatile multimedia services anytime and anywhere. Figure 1.3.1 shows the architecture for the GPRS PS service domain. In this figure, the dashed lines represent signaling links, and the solid lines represent data and signaling links. The GPRS core network consists of GPRS Support Nodes (GSNs) such as Serving GPRS Support Node (SGSN; see Fig. 1.3.1 (c)) and Gateway GPRS Support Node (GGSN; see Fig. 1.3.1 (d)).

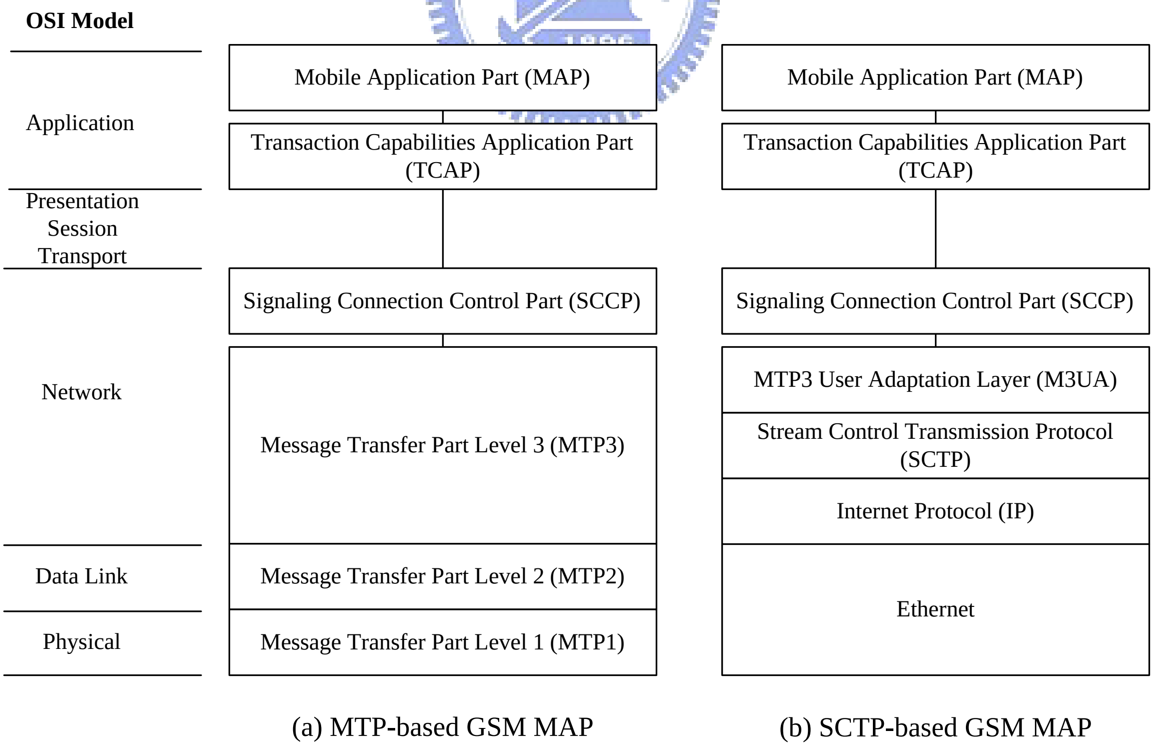


GGSN: Gateway GPRS Support Node
HLR: Home Location Register
MS: Mobile Station
PDN: Packet Data Network
SGSN: Serving GPRS Support Node

Figure 1.3.1: GPRS Network Architecture

An SGSN connecting to the Radio Network (see Fig. 1.3.1 (a)) plays a role in the PS service domain similar to a mobile switching center in the circuit switched service domain (i.e., the GSM core network). The GGSN interworks to the external Packet Data Network (PDN; see Fig. 1.3.1 (e)). The Home Location Register (HLR; see Fig. 1.3.1 (b)) is the master database containing all user-related subscriptions and location information. For mobility management and session management [3], the HLR communicates with both SGSN and GGSN through the Gr and Gc interfaces, respectively. These two interfaces are based on the GSM Mobile Application Part (MAP) [2].

As shown in Figure 1.3.2 (a), GSM MAP is an application of the SS7 protocol described in Section 1.1. GSM MAP uses SCCP classes 0 and 1 connectionless services to provide efficient routing with or without maintaining message sequencing. The network entities may consist of several application service elements (ASEs). The SCCP addresses these ASEs with subsystem numbers. The subsystem numbers for GSM MAP ASEs are listed in [11]. For intra-GSM network message delivery, the destination address of the message may be a simple destination point code (DPC) that can be used by the MTP for direct routing. For inter-GSM network message delivery, the originating node does not have enough knowledge to identify the actual address of the destination. In this case, the SCCP translates the actual destination address by Global Title Translation (GTT) [7]. Then, the SCCP layer invokes the MTP_TRANSFER primitive to provide the MTP layer message transfer services. The MTP layer generates the SS7 MTP message, and the message is sent to the destination over MTP-based SS7 network. The above descriptions assume the MTP-based protocol layers. These protocol layers can be replaced by the SCTP-based protocol hierarchy as shown in Figure 1.3.2 (b). When the SCCP layer invokes MTP_TRANSFER primitive to the M3UA layer, the M3UA layer generates the appropriate M3UA packet. Then the packet is sent to the destination through the SCTP/IP/Ethernet layers.


OSI Model
Application
Presentation
Session
Transport
Network
Data Link
Physical
Mobile Application Part (MAP)
Transaction Capabilities Application Part (TCAP)
Signaling Connection Control Part (SCCP)
Message Transfer Part Level 3 (MTP3)
Message Transfer Part Level 2 (MTP2)
Message Transfer Part Level 1 (MTP1)
Mobile Application Part (MAP)
Transaction Capabilities Application Part (TCAP)
Signaling Connection Control Part (SCCP)
MTP3 User Adaptation Layer (M3UA)
Stream Control Transmission Protocol (SCTP)
Internet Protocol (IP)
Ethernet

(a) MTP-based GSM MAP (b) SCTP-based GSM MAP

Figure 1.3.2: GSM MAP Protocol Hierarchy

The GPRS network entities (such as SGSN and HLR) communicate with each other through MAP dialogues by invoking MAP service primitives. A service primitive can be one of four types: Request, Indication, Response, and Confirm. The service primitive is initiated by a MAP service user of a network entity called the dialogue initiator, as shown in Figure 1.3.3 (1). The service type is Request. This service request is sent to the MAP service provider of the network entity.

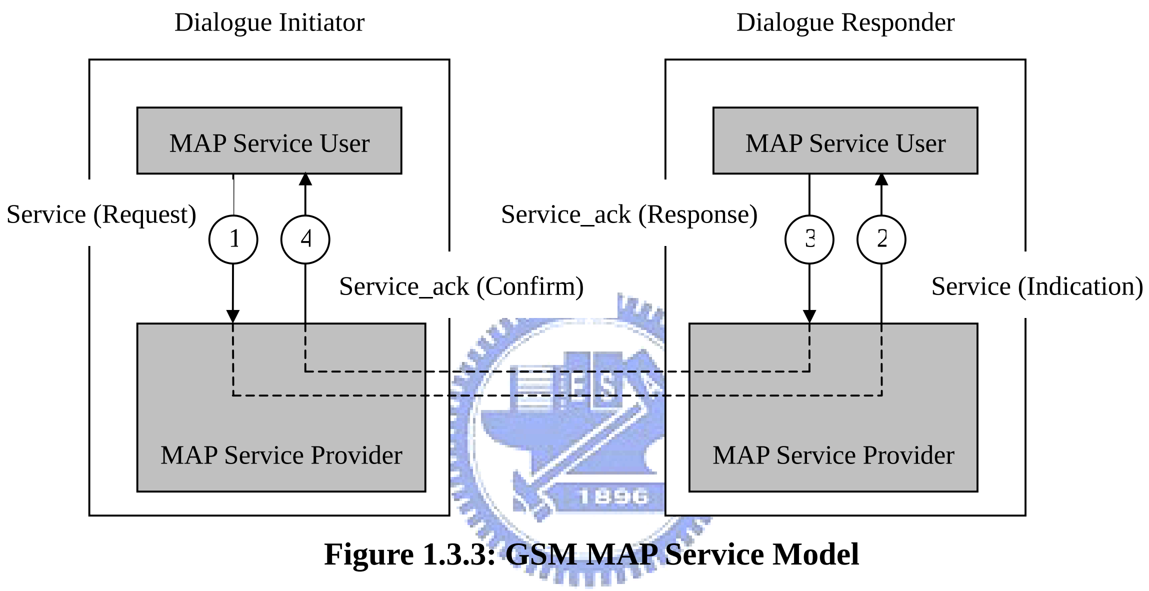


Figure 1.3.3: GSM MAP Service Model

The service provider delivers the request to the peer network entity (i.e., the dialogue responder) by TCAP. When the MAP service provider of the peer network entity receives the request, it invokes the same service primitive with type Indication to inform the destination MAP service user (see Figure 1.3.3 (2)). In most cases, the information (parameters) of the service with type Indication is identical to that with type Request. The primitive is typically a query, which asks the dialogue responder to perform some operations.

A corresponding service acknowledgment, with or without results, may be sent from the dialogue responder to the dialogue initiator. The same service primitive with type

Response is invoked by the MAP service user of the dialogue responder (see Figure 1.3.3 (3)). After the MAP service provider of the dialogue initiator receives this response, it invokes the same service primitive with type Confirm (see Figure 1.3.3 (4)). The parameters of the Confirm and the Response services are identical in most cases, except that the Confirm service may include an extra provider error parameter to indicate a protocol error.

A MAP dialogue consists of several MAP services to perform a common task. The services are either specific or common. The specific services include:

- Mobility services
- Operation and maintenance services
- Call-handling services
- Supplementary services
- Short message service management services

The common MAP services are used to establish and clear MAP dialogue between peer MAP service users. They invoke functions supported by TCAP and report abnormal situations. As defined in GSM MAP [2], six common MAP services are described as follows:

MAP_OPEN is used to establish a MAP dialogue. This service is confirmed by the service provider; That is, MAP_OPEN has the Request/Indication and Response/Confirm types.

MAP_CLOSE is used to clear a MAP dialogue. This service is not confirmed by the service provider; that is, the service primitive only has the Request/Indication types, but not the Response/Confirm types.

MAP_DELIMITER is used to explicitly request the TCAP to transfer the MAP protocol data units to the peer entities. This service does not have any parameters and is not confirmed by the service provider.

For the descriptions of the MAP_U_ABORT, MAP_P_ABORT, and MAP_NOTICE services, the readers are referred to [2,3].

1.4 Motivation

Traditionally, the SS7 signaling for mobile networks is implemented in MTP-based SS7 network. In Universal Mobile Telecommunication System (UMTS) all-IP architecture [18,19,20], the SS7 signaling transport will be carried out by IP-based SS7 network. Evolving from GSM and GPRS, the UMTS all-IP architecture integrates the IP and mobile technologies. Such integration results in lower cost of IP-based SS7 network. In this thesis, we implement and discuss two approaches for SS7 signaling transport; namely, the MTP-based approach and the SCTP-based approach. Chapter 2 describes our implementations of theses two approaches. Chapter 3 compares these two approaches in three perspectives: message format, connection setup, and data transmission/Ack. We also illustrate the performance of the SCTP-based and the MTP-based MAP approaches by using the Send Authentication Info procedure defined in 3GPP Technical Specification 23.060 [1] as an example. Finally, Chapter 4 concludes our research.

Chapter 2 The MTP-based and the SCTP-based Approaches

This chapter describes the MTP-based and the SCTP-based approaches. The MAP layer implementations for these two approaches are the same. The lower layer implementations for the MTP-based and the SCTP-based approaches are very different. However, Our modular design of the MAP layer allows quick porting to different implementations of the TCAP layer. In this chapter, we describe the MAP implementation for both the MTP-based and the SCTP-based approaches. Section 2.1 describes implementation of the MAP layer. Section 2.2 describes the lower layer implementation for the MTP-based approach. Section 2.3 describes the lower layer implementation for the SCTP-based approach. Section 2.4 shows an example of the SCTP-based MAP message delivery.

2.1 The MAP Layer

The MAP layer is implemented by modifying MAP version 1.4 software developed by Trillium Digital Systems Inc. [12]. Figure 2.1.1 illustrates the software architecture of the MAP layer, which includes the following modules.

GSM MAP Module (Fig 2.1.1 (1)) implements standard GSM MAP service primitives based on the GSM Recommendations 09.02 [2]. Examples of the primitives are MAP_OPEN, MAP_CLOSE, and MAP_DELIMETER as described in Section 1.3.

Layer Manager Interface Module (Fig 2.1.1 (2)) provides functions to control and monitor the status of the MAP layer. The functions include layer resources configuration, layer resources activation/deactivation, layer information tracing, and layer status indication.

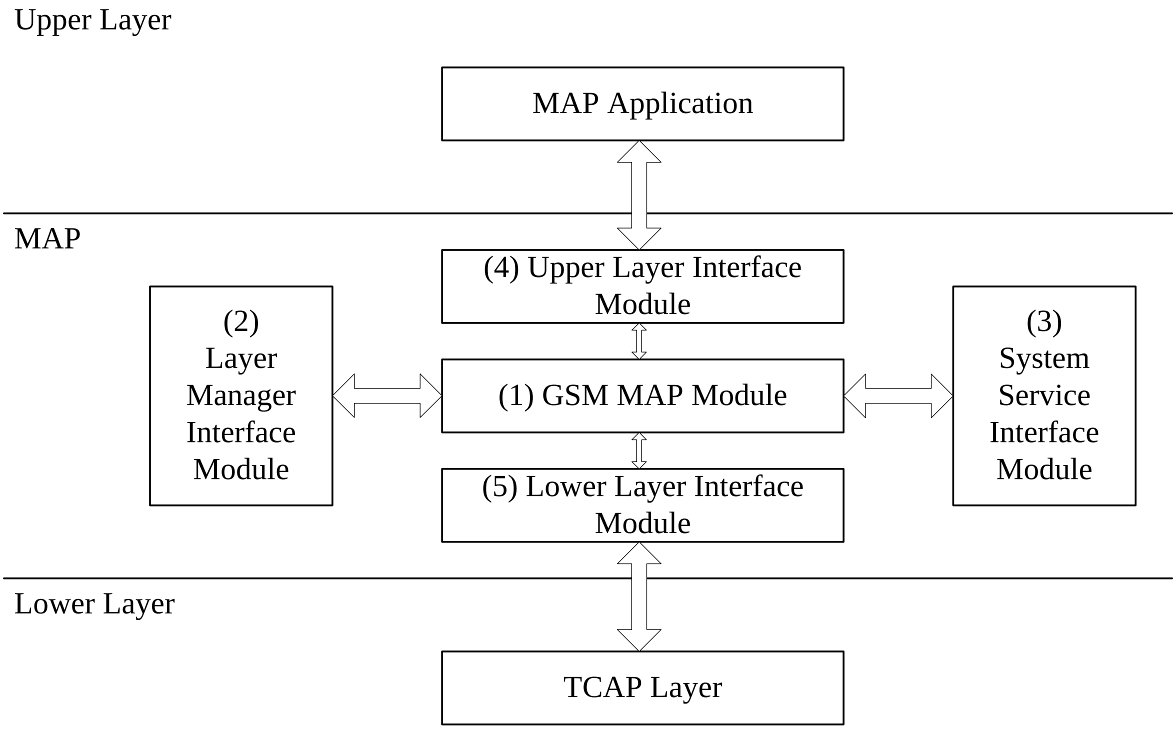

Upper Layer
MAP Application
MAP
(4) Upper Layer Interface Module
(2) Layer Manager Interface Module
(1) GSM MAP Module
(3) System Service Interface Module
(5) Lower Layer Interface Module
Lower Layer
TCAP Layer

Figure 2.1.1: Software Architecture of the MAP Layer

System Service Interface Module (Fig 2.1.1 (3)) provides functions required for the MAP layer buffer management, timer management, date and time management, resource checking, and initialization. All functions related to Operating System (OS) are also included in this module.

Upper Layer Interface Module (Fig 2.1.1 (4)) provides a function-based interface to interact with the upper layer (i.e., the application program) through the function calls.

Lower Layer Interface Module (Fig 2.1.1 (5)) provides an interface to communicate with the lower layer (i.e., TCAP).

The modular design of the MAP layer implementation provides flexibility and portability, which allows quick porting of our MAP layer to different TCAP implementations. We only need to modify the Lower Layer Interface Module for the specific TCAP layers implemented in the MTP-based and the SCTP-based approaches. We can also port our MAP software to various OSs by modifying the System Service Interface Module.

2.2 The TCAP Implementation of the MTP-based Approach

Performance Technologies Inc. 372 series SS7 card (referred to as the SS7 card in this thesis) is a host-independent SS7 controller board embedded with full SS7 functionality [6, 7, 8]. The SS7 card can be plugged into a host (i.e. a computer) on which applications (i.e., TCAP user) are developed by Peripheral Component Interconnect (PCI) or Compact Peripheral Component Interconnect (CPCI) interface. Figure 2.2.1 illustrates the MTP-based SS7 architecture consisting of a host and the SS7 card. The MAP layer is implemented in the host. The details are elaborated in Section 2.1. In the host, the configuration and protocol interface for the SS7 card are implemented in the Lower Layer Interface Module of the MAP layer (Figure 2.1.1 (5)). The MAP layer interacts with the TCAP layer in the SS7 card through the device driver (Figure 2.2.1 (1)). All other SS7 protocol layers (i.e., TCAP, SCCP, and MTP; see Figure 2.2.1(2)) necessary for communication with the SS7 network entity reside on the SS7 card. Each protocol layer is implemented as a software component on the SS7 card.

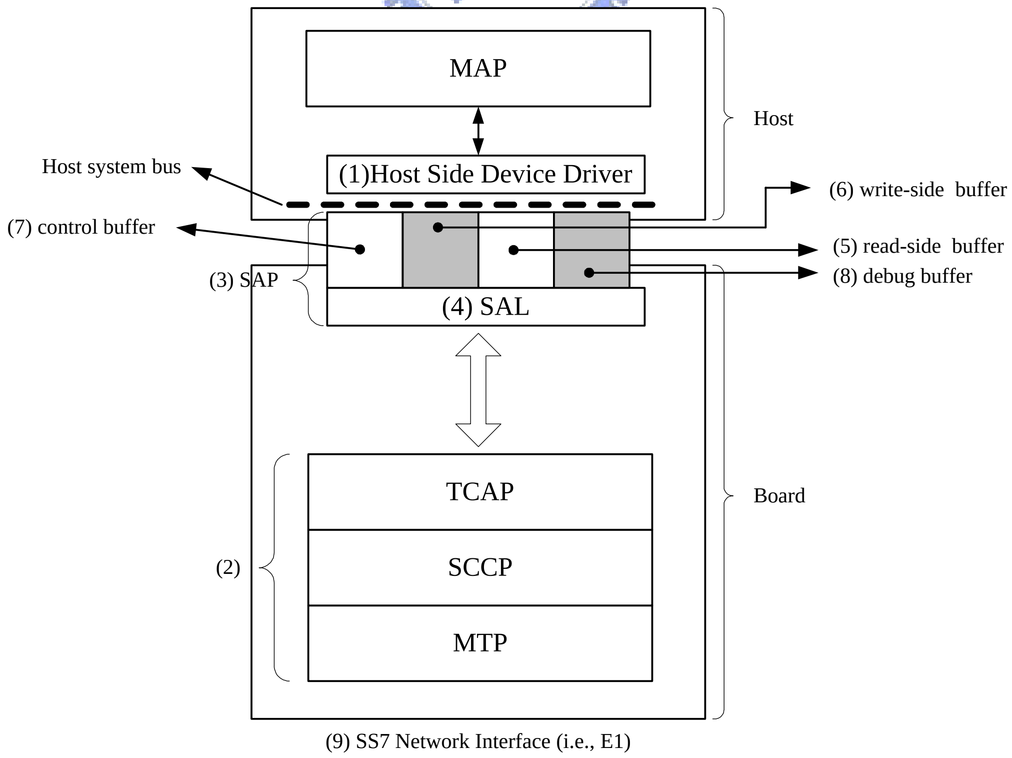


Figure 2.2.1: The MTP-based SS7 Architecture

The executable software components (i.e. TCAP, SCCP, and MTP) are installed on the host, and must be downloaded to the SS7 card by a host-resident download program (not shown in Fig 2.2.1) each time the SS7 card is reset. The host device driver communicates with the SS7 card through the Service Access Point (SAP; see Figure 2.2.1 (3)) on the SS7 card. The SAP consists of Service Access Layer (SAL; see Figure 2.2.1(4)) software and four buffers implemented in a shared memory region on the SS7 card. The SAL manages the license key and the shared memory, and distributes messages to the appropriate layers and buffers. The read-side buffer (Figure 2.2.1 (5)) and the write–side buffer (Figure 2.2.1 (6)) are used to exchange messages between the host and the SS7 card. The control buffer (Figure 2.2.1 (7)) is used by the SAL to manage the other buffers and to keep track of information, such as read-side and write-side buffer pointers and semaphores. The debug buffer (Figure 2.2.1 (8)) is used to transfer diagnostic information from the SS7 card to the host.

The SS7 card also includes an SS7 physical layer interface (i.e., E1; Figure 2.2.1 (9)) and a Programmable Read Only Memory (PROM; not shown in Figure 2.2.1) that is used to download the SAL software component from the host. To initialize the SS7 card, the PROM software is activated with a hardware reset and remains active until the first software component (i.e., the SAL) is downloaded. After the SAL is downloaded, it takes control from the SS7 card PROM and disables the PROM. Then the MTP2, MTP3, SCCP, and TCAP software components can be downloaded into the SS7 card, and the host can invoke the TCAP services through the device driver. Readers are referred to [13] for the details.

2.3 The TCAP Implementation of the SCTP-based Approach

The SCTP protocol is implemented on top of the Internet Protocol (IP). Figure 2.3.1 illustrates the software modules of the SCTP-based protocol stack. A *Stack Entity* (represented by a dotted rectangle) is a set of modules that implement the functionalities defined in the corresponding specifications (e.g., RFC 2960 [14]). We elaborate on the stack entities for each layer in the following sections. The SCTP-based architecture includes the following modules.

OS Interface Module provides OS-independent functions to be invoked by the stack entities. These function calls are mapped to actual function calls provided by the underlying OS. The functions include buffer management, timer management, and so on.

System Management Interface Module in each layer provides functions to control and monitor the specific stack entity. Examples of the functions include system initialization, statistic collection, and layer information tracing.

User Interface and the Transport Layer Interface Modules provide function-based interfaces to interact with upper and lower layers through the function calls. For example, the functions include transmission service invoking, incoming data indication, and so on. In our implementation, these modules are used for the communication between M3UA and SCTP.

Encapsulating Shell provides a message-based interface between upper and lower layers. The message-based interface is implemented by socket to allow the communications between two different processes (e.g., communication between the TCAP and the SCCP).

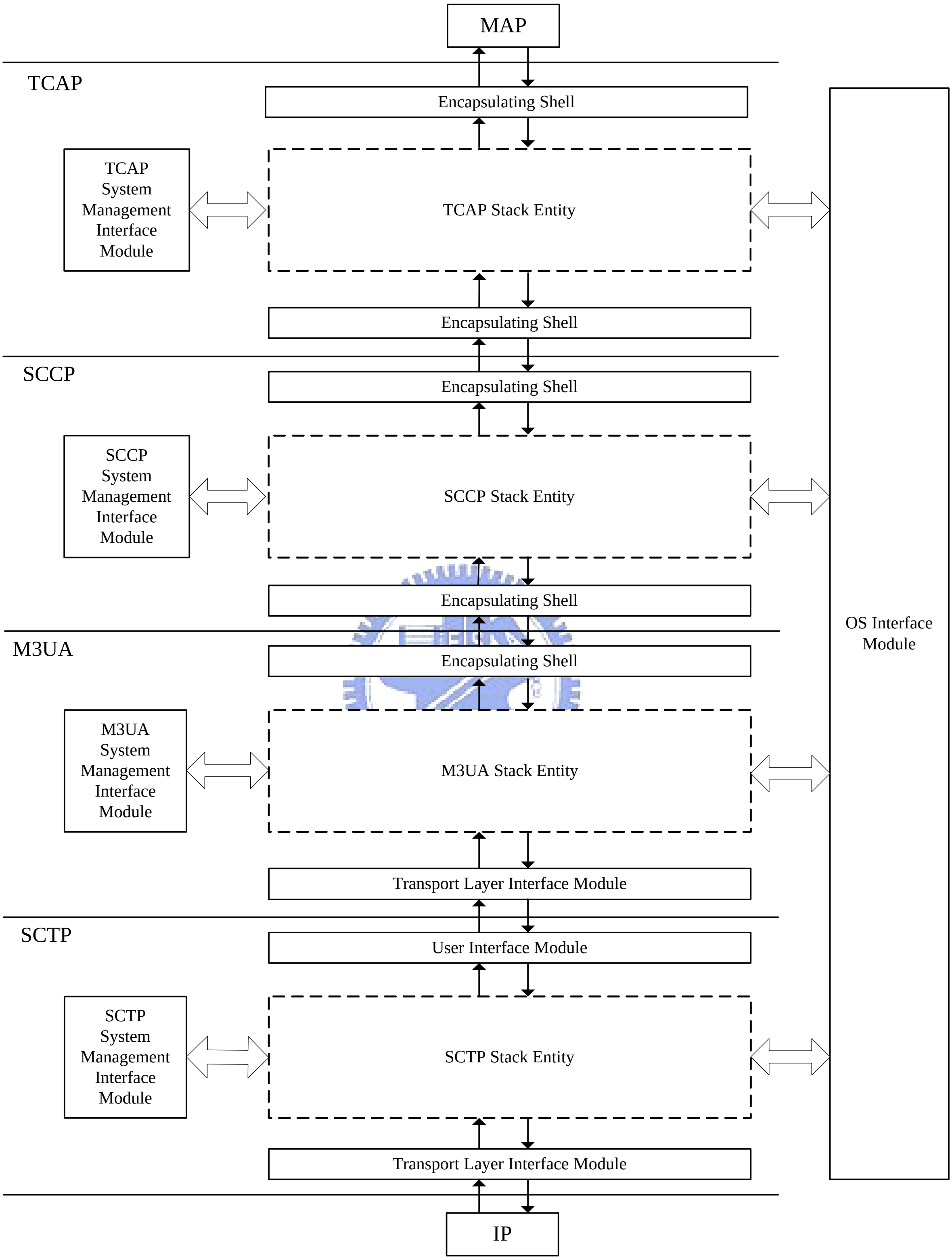


Figure 2.3.1: Software Architecture of the SCTP-based Approach

In our approach, TCAP, SCCP, and M3UA/SCTP are implemented as individual processes. These processes communicate with each other by the message-based interface (i.e., socket). With this process architecture, we can independently and flexibly deploy each layer. Also, modular design of our process structure is easy to debug, maintain, analysis and port to various underlying OSs.

2.3.1 SCTP and M3UA Stack Entities

Both SCTP and M3UA stack entities include the modules shown in Figure 2.3.2.

Message Handling Module (Figure 2.3.2 (1)) provides needed functions for message parsing and message building such as message parameter checking and message header building.

Finite State Machine (FSM) Management Module (Figure 2.3.2 (2)) implements the functionalities and maintains FSMs defined in the specific protocol recommendations. At the SCTP layer, the *association state machine* is implemented [14]. At the M3UA layer, the *adjacent signaling point state machine* is implemented [15].

Database Module (Figure 2.3.2 (3)) maintains information required in the specific protocol layer. For example, in the SCTP layer, the Database Module maintains the association status including the number of streams, IP addresses of the multi-homed SCTP endpoint, and so on.

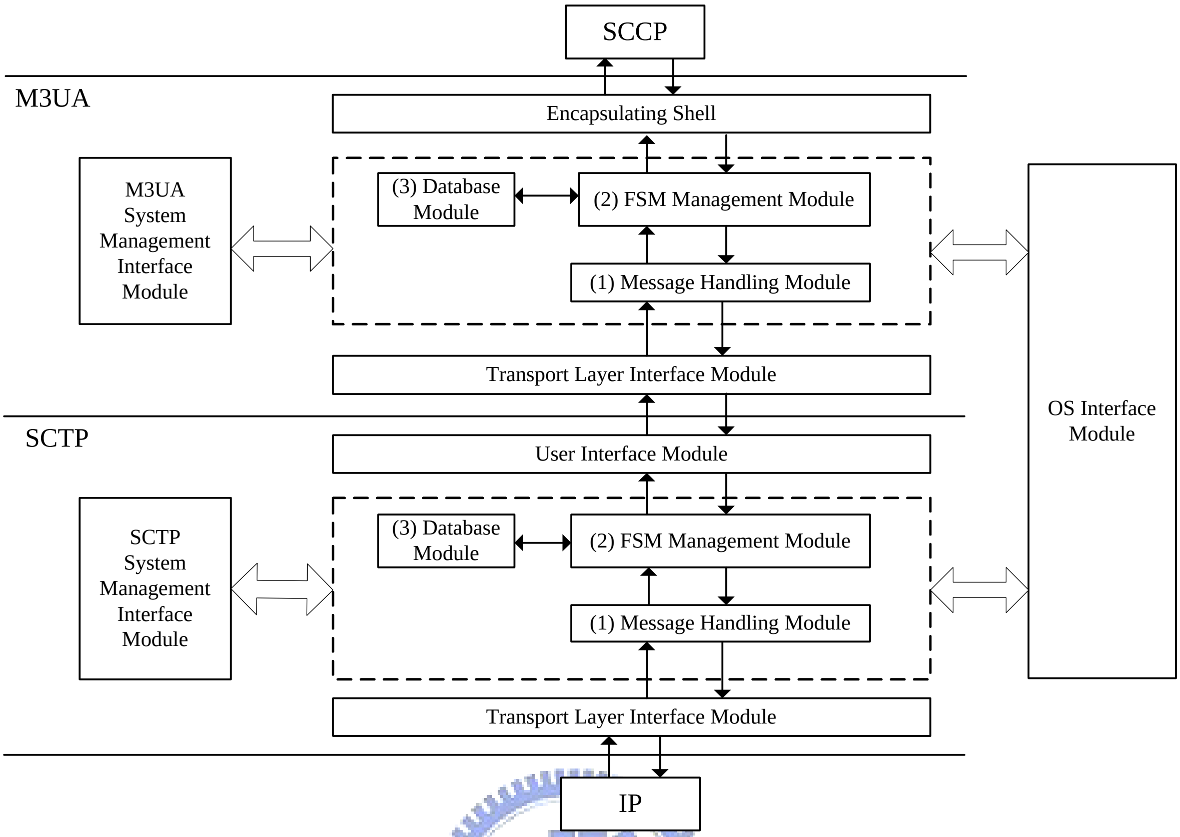

SCCP
M3UA
Encapsulating Shell
M3UA System Management Interface Module
(3) Database Module
(2) FSM Management Module
(1) Message Handling Module
Transport Layer Interface Module
OS Interface Module
SCTP
User Interface Module
SCTP System Management Interface Module
(3) Database Module
(2) FSM Management Module
(1) Message Handling Module
Transport Layer Interface Module
IP

Figure 2.3.2: Software Architecture of SCTP and M3UA

2.3.2 SCCP Stack Entity

The SCCP stack entity is designed based on SCCP functionalities [7] including the modules shown in Figure 2.3.3.

Message Processing Module (Figure 2.3.3 (1)) provides needed functions for message parsing, message building, and message dispatch (e.g., connectionless service request or connection-oriented service request from TCAP).

SCCP Connectionless Control (SCLC) Module (Figure 2.3.3 (2)) and SCCP Connection-oriented Control (SCOC) Module (Figure 2.3.3 (3)) handle the message transfer in the corresponding services.

SCCP Routing Control (SCRC) Module (Figure 2.3.3 (4)) determines the route of the outgoing messages and dispatches the incoming messages to SCLC or SCOC.

SCCP Management (SCMG) Module (Figure 2.3.3 (5)) provides the capabilities to handle the congestion or failure of SCCP.

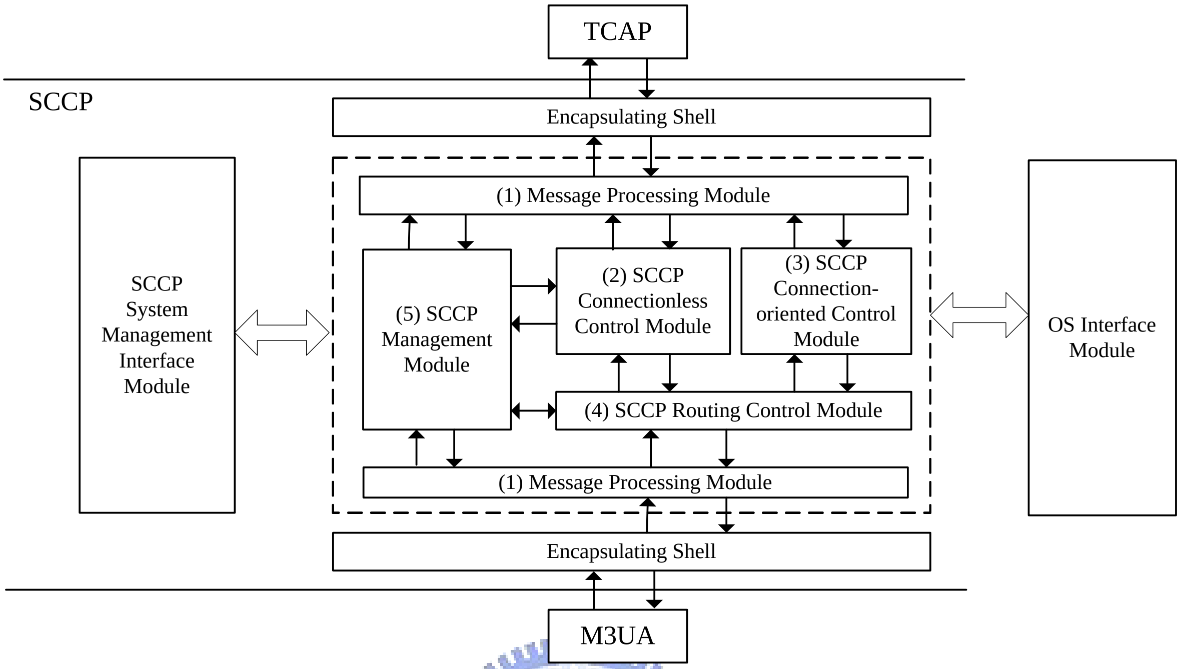

TCAP
SCCP
Encapsulating Shell
(1) Message Processing Module
SCCP System Management Interface Module
(5) SCCP Management Module
(2) SCCP Connectionless Control Module
(3) SCCP Connection-oriented Control Module
(4) SCCP Routing Control Module
(1) Message Processing Module
Encapsulating Shell
OS Interface Module
M3UA

Figure 2.3.3: Software Architecture of SCCP

2.3.3 TCAP Stack Entity

The TCAP implementation consists of two sub-layers; namely, component sub-layer (TC) and transaction sub-layer (TR) defined in the ITU Recommendation Q.771-Q.775 [8]. Figure 2.3.4 illustrates the relationship among the TCAP sub-layers and the upper and lower layers. Figure 2.3.5 shows the details of the modules implemented for the TCAP sub-layers.

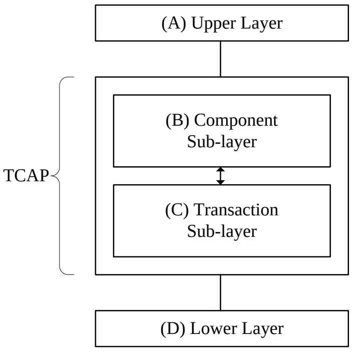


Figure 2.3.4: Abstract Software Architecture of TCAP

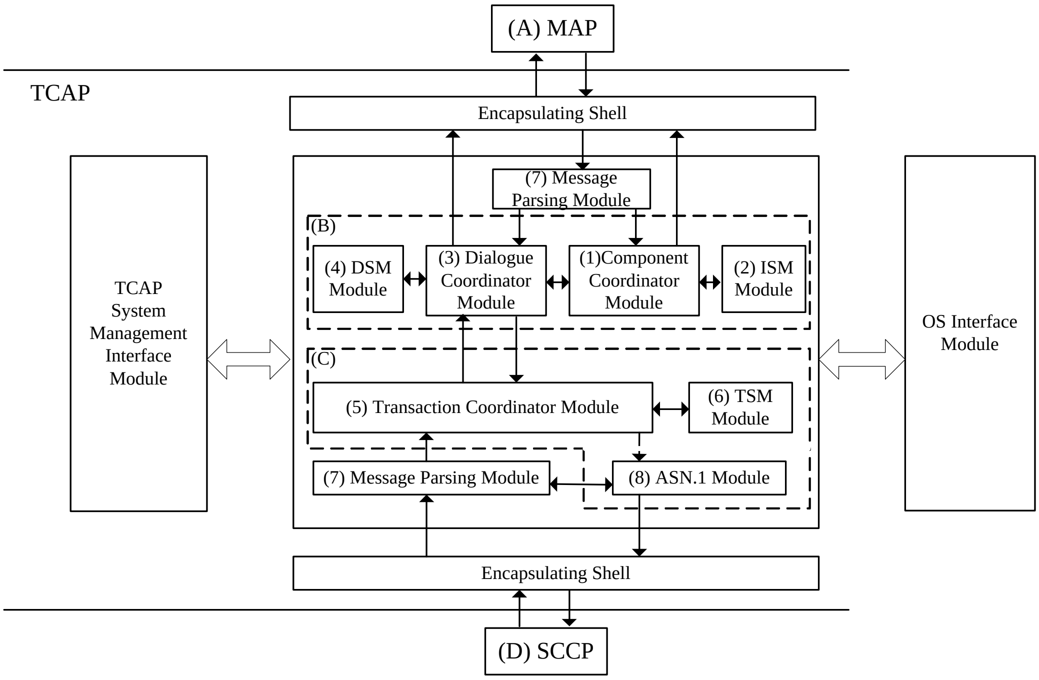

(A) MAP
TCAP
Encapsulating Shell
TCAP System Management Interface Module
(7) Message Parsing Module
(B)
(4) DSM Module
(3) Dialogue Coordinator Module
(1)Component Coordinator Module
(2) ISM Module
OS Interface Module
(C)
(5) Transaction Coordinator Module
(6) TSM Module
(7) Message Parsing Module
(8) ASN.1 Module
Encapsulating Shell
(D) SCCP

Figure 2.3.5: Detailed Software Architecture of TCAP

2.3.3.1 The Component Sub-layer

The component sub-layer is responsible for (a) component and (b) dialogue handling. Implementation of this sub-layer includes Component Coordinator Module (Figure 2.3.5 (1)), Invoke State Machine (ISM) Module (Figure 2.3.5 (2)), Dialogue Coordinator Module (Figure 2.3.5 (3)), and Dialogue State Machine (DSM) Module (Figure 2.3.5 (4)). In the implementation, Message Parsing Module (Figure 2.3.5 (7)) and Encapsulation Shell Module are used as interfaces between layers. They are not defined in the specifications, but are required in our implementation. Message Parsing Module parses and dispatches the message and Encapsulating Shell Module provides socket communications between two layers.

(a) Component handling is supported by Component Coordinator Module (Figure 2.3.5 (1)) to deal with TCAP components that convey operations (e.g., MAP_SEND_ AUTHENTICATION_INFO) and replies from the TC-user (i.e., the MAP layer). An operation is performed by the remote end. Invocation of an operation is identified by an Invoke ID. Simultaneous invocations of operations are distinguished by their

Invoke IDs. Four classes of operation are defined.

- Class 1: Both success and failure are reported.
- Class 2: Only failure is reported.
- Class 3: Only success is reported.
- Class 4: Neither success nor failure is reported.

Different types of state machines are defined for different classes of operations (referred to [8] for the details). The ISM Module (Figure 2.3.5 (2)) maintains the state machine for each invoked operation. There are four finite state machines for the four classes of operation. Every reply corresponds to one operation, and may be a return result indicating success, a return error indicating operation failure, or a reject indicating inability to perform the operation. Table 2.3.1 lists the primitives for component handling in the component sub-layer. The type "Request" indicates the primitive issued from the upper layer to the component sub-layer, and the type "Indication" indicates the primitive issued from the component sub-layer to the upper layer.

Table 2.3.1: Primitives for Component Handling

Name	Type	Description
TC_INVOKE	Request / Indication	Invocation of an operation.
TC_RESULT_L	Request / Indication	Only result or last part of the segmented result of a successfully executed operation.
TC_RESULT_NL	Request / Indication	Non-final part of the segmented result of a successfully executed operation.
TC_U_ERROR	Request / Indication	Reply to a previously invoked operation, indicating that the operation execution failed.

TC_L_CANCEL	Indication	Informs the TC-user locally that an operation invocation is terminated due to a timeout condition.
TC_U_CANCEL	Request	Causes local termination of an operation invocation, as a consequence of a TC-user decision.
TC_L_REJECT	Indication	Informs the local TC-user hat a component sub-layer detected invalid component was received.
TC_R_REJECT	Indication	Informs the local TC-user that a component was rejected by the remote component sub-layer.
TC_U_REJECT	Request / Indication	Rejection of a component by the TC-user, indicating a malformation that prevents the operation from being executed, or the reply from being understood.
TC_TIMER_RESET	Request	Allows the local TC-user to refresh a timer of an operation invocation.

(b) Dialogue handling facilities is supported by Dialogue Coordinator Module (Figure 2.3.5 (3)) to exchange TCAP components within a dialogue. The component sub-layer provides two kinds of dialogues; namely unstructured dialogue and structured dialogue. The unstructured dialogue is used to deliver components that do not require replies (i.e., the class 4 operations). An unstructured dialogue is also called a unidirectional dialogue, where the components are grouped in one single unidirectional message. On the other hand, in a structured dialogue, the TC-users (i.e., the MAP layers) indicate the beginning, the continuation, and the end of a dialogue. The MAP layer can exchange successive messages that contain components within this dialogue. Multiple structured dialogues can run simultaneously, and each of them is identified by a unique Dialogue ID. Dialogue State Machine (DSM) Module (Figure 2.3.5 (4)) maintains the state machine of each structured dialogue (referred to [8] for the details). Table 2.3.2 lists the primitives related to dialogue handling.

Table 2.3.2: Primitives for Dialogue Handling

Name	Type	Description
TC_UNI	Request / Indication	Requests/indicates an unstructured dialogue.
TC_BEGIN	Request / Indication	Begins a dialogue.
TC_CONTINUE	Request / Indication	Continues a dialogue.
TC_END	Request / Indication	Ends a dialogue.
TC_U_ABORT	Request / Indication	Allows a TC-user to terminate a dialogue abruptly, without transmitting any pending components.
TC_P_ABORT	Indication	Informs the TC-user that the dialogue has been terminated by the service provider (e.g., the transaction sub-layer).
TC_NOTICE	Indication	Informs the TC-user that the Network Service Provider (e.g., the SCCP layer) is unable to provide the requested service.

Primitives TC_UNI, TC_BEGIN, TC_CONTINUE and TC_END cause invoked component(s) to be delivered to the remote end. When one of those primitives is issued to the component sub-layer, Dialogue Coordinator Module will query Component Coordinator Module to obtain the TCAP components invoked by the TCAP-user, and then issues the corresponding primitive with the parameter “User Data” (i.e., the TCAP components) to the transaction sub-layer.

2.3.3.2 The Transaction Sub-layer

The transaction sub-layer provides primitives to deal with the exchange of TR-user messages. This sub-layer includes Transaction Coordinator Module (Figure 2.3.5 (5)), Transaction State Machine (TSM) Module (Figure 2.3.5 (6)), and Abstract Syntax Notation One (ASN.1) Module (Figure 2.3.5 (8)). Transaction Coordinator Module (Figure 2.3.5 (5)) deals with transaction handling primitives and manages message exchange within a transaction. Multiple transactions can be run simultaneously, and each of them is identified by a unique Transaction ID. The state machine of each transaction (referred to [8] for the details) is maintained by Transaction State Machine (TSM) Module. ASN.1 Module provides ASN.1-related encoding/decoding functions [17] for encoding /decoding the TCAP message format. Table 2.3.3 shows the primitives provided by the transaction sub-layer.

Table 2.3.3: Primitives for the Transaction Sub-layer

Name	Type	Description
TR_UNI	Request / Indication	Information may be sent from one TR-user to another TR-user without their establishing an explicit association.
TR_BEGIN	Request / Indication	Starts a transaction with a particular

		Transaction ID and delivers any accompanying TR-user message.
TR_CONTINUE	Request / Indication	Allows two TR-users to exchange TR-user messages in both directions within a transaction.
TR_END	Request / Indication	Ends a transaction and causes the associated transaction ID to be released.
TR_U_ABORT	Request / Indication	Allows a TR-user to terminate a transaction abruptly.
TR_P_ABORT	Indication	Informs the TR-user that the transaction has been terminated in reaction to abnormal situations. The possible reasons for such a decision are indicated in Recommendation Q.774 [8].
TR_NOTICE	Indication	Notifies the TR-user if the requested service cannot be exercised (e.g., the SCCP layer cannot deliver the TCAP message to the remote node).

There is a one-to-one relationship between a dialogue and a transaction (see Table 2.3.4). When the transaction sub-layer is used to support a dialogue, dialogue handling primitives of the component sub-layer are mapped to transaction handling primitives of the transaction sub-layer with the similar generic names.

Table 2.3.4: Mapping of TC Dialogue Handling Primitives to TR Transaction Handling Primitives

TC primitive	TR primitive
TC_UNI	TR_UNI
TC_BEGIN	TR_BEGIN
TC_CONTINUE	TR_CONTINUE
TC_END	TR_END
TC_U_ABORT	TR_U_ABORT
TC_P_ABORT	TR_P_ABORT
TC_NOTICE	TR_NOTICE

As an example, when the MAP layer issues the TC_BEGIN request primitive to the component sub-layer, the component sub-layer groups the previous invoked TCAP components (e.g., the component conveying the SEND_AUTHENTICATION_INFO operation) of the same dialogue, and issues the TR_BEGIN request primitive with the parameter "User Data" (i.e., the grouped TCAP components) to the transaction sub-layer. Then the transaction sub-layer generates the appropriate TCAP message (including the components) and delivers this message to the remote end.

2.4 An Example of the SCTP-based MAP Message Delivery

This section illustrates MAP message delivery for the SCTP-based approach. Consider a scenario where the SGSN attempts to invoke the MAP_SEND_AUTHENTICATION_INFO operation in the HLR. Figure 2.4.1 shows the interaction among the layers for our example. Note that in the implementation, Message Parsing Module, Message Processing Module and Encapsulation Shell Module are used as interfaces between layers. They are not defined in the specifications, but are required in implementation. We will bypass the descriptions of these modules, and readers are referred to Section 2.4 for the details.

Step 1: The SGSN application initiates a MAP dialogue by invoking the MAP_OPEN request primitive of the MAP layer. Then the SGSN application invokes the MAP_SEND_AUTHENTICATION_INFO request primitive of the MAP layer to request remote operation.

Steps 2-4: Based on the request primitive from the SGSN application, the MAP Layer invokes the TC_INVOKE request primitive of the TCAP layer to set the operation code (i.e., SEND_AUTHENTICATION_INFO) and parameters of the TCAP component.

Step 5: The Component Coordinator Module creates the TCAP component for the invocation of the operation (i.e., SEND_AUTHENTICATION_INFO), and the corresponding state machine of the operation is maintained by the ISM Module.

Step 6: The SGSN application invokes the MAP_DELIMETER request primitive of the MAP layer to explicitly request the TCAP layer to transfer the MAP message.

Steps 7- 9: The MAP Layer invokes the TC_BEGIN request primitive of the TCAP layer to trigger TCAP component transmission.

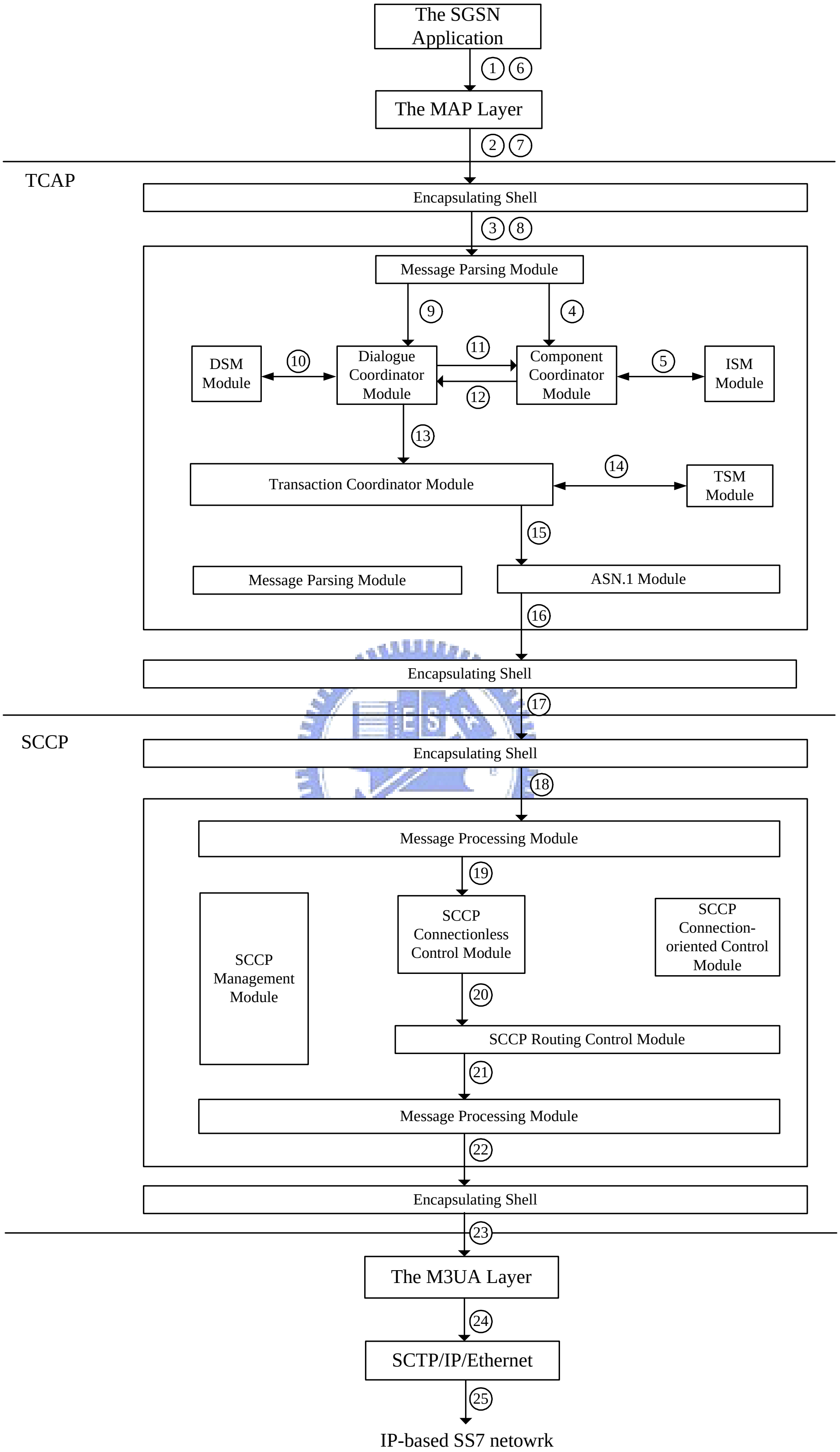


Figure 2.4.1: An Example of the SCTP-based MAP Message Delivery

Steps 10-13: The Dialogue Coordinator Module in the TCAP layer starts a dialogue, and the state machine of the dialogue is maintained by the DSM Module. The Dialogue Coordinator Module then queries the Component Coordinator Module to obtain the invoked TCAP component (i.e., the component conveys the MAP_SEND_AUTHENTICATION_INFO operation), and issues the TR_BEGIN request primitive with parameter “User Data” (i.e., the TCAP component) to the Transaction Coordinator Module.

Steps 14-19: The Transaction Coordinator Module starts a transaction, and the state machine of the transaction is maintained by the TSM Module. The Transaction Coordinator Module then utilizes the ASN.1 Module to encode the TCAP message, and invokes the SCCP_N_UNITDATA request primitive of the SCCP layer to use SCCP Connectionless Service for delivering the message to the remote end.

Steps 20-23: The SCCP Connectionless Control Module handles the requested connectionless service. The SCCP Routing Control Module determines the route of the message, and invokes the MTP_TRANSFER request primitive of the M3UA layer to use the MTP message transfer services.

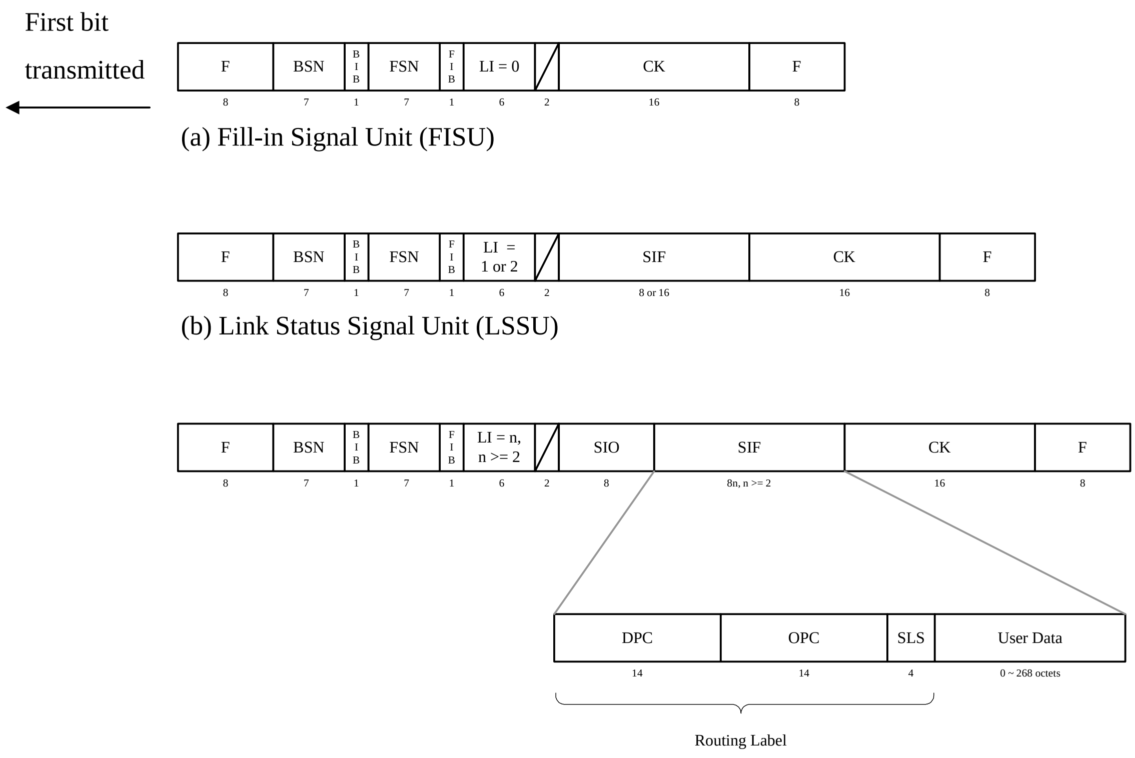
Steps 24-25: The M3UA Layer checks if the adjacent signaling point in the route is active. If so, the M3UA Layer generates the M3UA packet, and the packet is sent to the destination through the SCTP/IP/Ethernet layers and the IP-based SS7 network.

Chapter 3 Comparing the MTP-based and SCTP-based SS7 Implementations

This chapter compares MTP-based and SCTP-based implementations in three perspectives; namely, message format, connection setup, and data transmission/ack. Then, we use the Send Authentication Info procedure defined in 3GPP Technical Specification 23.060 [1] as an example to illustrate the performance of the SCTP-based and the MTP-based MAP mechanisms.

3.1 Message Format

When SS7 signaling messages are transferred over MTP-based SS7 network, the MTP2 and MTP1 layers provide reliable transfer of the messages between two adjacent signaling points. The Signal Unit (SU) used by MTP2 is shown in Figure 3.1.1. There are three types of the SU, Fill-in Signal Unit (FISU), Link Status Signal Unit (LSSU) and Message Signal Unit (MSU). The FISU provides signaling link failure detection, and is transmitted when no MSUs or LSSUs are transmitted. The LSSU is used to notify adjacent signaling point of the link status. The MSU is used to carry signaling information of upper layer protocol (i.e. MTP3) in the Signaling Information Field (SIF) that is limited to 272 octets.



(a) Fill-in Signal Unit (FISU)

(b) Link Status Signal Unit (LSSU)

(c) Message Signal Unit (MSU)

BIB: Backward Indicator Bit
BSN: Backward Sequence Number
CK: Check Bits
DPC: Destination Point Code
F: Flag
FIB: Forward Indicator Bit

FSN: Forward Sequence Number
LI: Length Indicator
OPC: Originating Point Code
SIF: Signaling Information Field
SIO: Service Information Octet
SLS: Signaling Link Selection

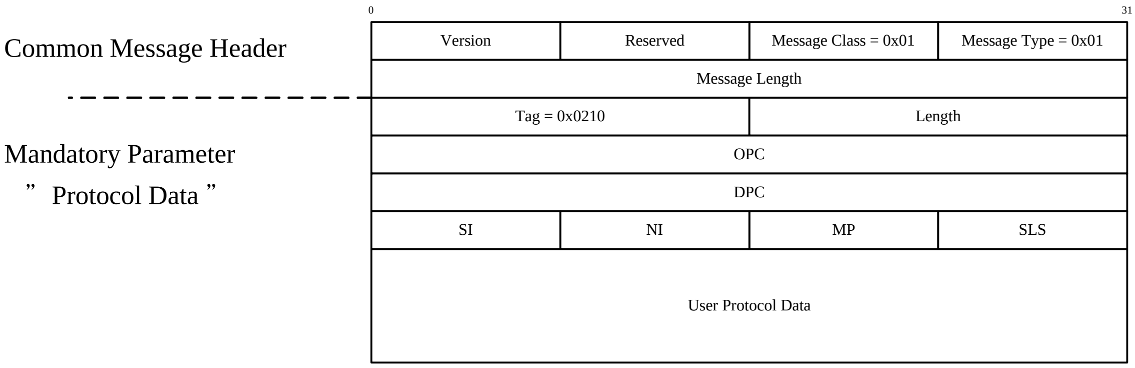
Figure 3.1.1: MTP Signal Unit Formats

In the MSU, Service Indicator octet (SIO) consists of Service Indicator (SI), Network Indicator (NI) and two spare bits. The SI is used for message distribution to determine the MTP-user part (i.e., SCCP or ISUP). The NI is used for discrimination between international and national messages. For national message, the two spare bits may be used to indicate Message Priority (MP) for the optional procedure in national applications. The Length Indicator (LI) field is used by MTP2 to determine the SU type, and indicates the number of octets, which follows the LI field and precedes the Check Bits (CK) field. The values of the LI can be

- LI = 0 (FISU)
- LI = 1 or 2 (LSSU)
- LI = 3 or more (MSU)

Descriptions of other fields can be found in [6]. The MTP3 provides the functions and procedures related to message routing and network management. MTP3 uses Routing Label (in SIF; see Figure 3.1.1 (c)) of the MSU to determine how to route the messages. Routing Label consists of Destination Point Code (DPC), Originating Point Code (OPC), and Signaling Link Selection (SLS). The DPC and OPC fields are used to indicate the signaling point addresses. The SLS field exactly corresponds to the Signaling Link Code (SLC), which indicates the signaling link between the destination point and originating point to which the message refers. The User Data field (i.e., SCCP or ISUP) is up to 268 octets in length. Figure 3.1.1 (c) illustrates the MTP message format. The message header is about 13 bytes.

Just like MTP-based approach, SCTP-based implementation provides the same set of primitives and services to the MTP3-User (i.e., SCCP or ISUP). Figures 3.1.2 illustrates the M3UA packet format for transferring signaling data of the upper layer (i.e. SCCP or ISUP). The packet includes a common message header and a mandatory parameter



Version	Reserved	Message Class = 0x01	Message Type = 0x01
Message Length			
Tag = 0x0210		Length	
OPC			
DPC			
SI	NI	MP	SLS
User Protocol Data			

DPC: Destination Point Code	OPC: Originating Point Code
NI: Network Indicator	SI: Service Indicator
MP: Message Priority	SLS: Signaling Link Selection Code

Figure 3.1.2: M3UA Packet Format for Transferring Messages

“Protocol Data”. The “Protocol Data” parameter contains information of the SS7 MTP message, including the SIO and Routing Label (see Figure 3.1.1 (c)). The OPC, DPC, SI, NI, MP and SLS fields are similar to those in the SS7 MTP message, respectively. The User Protocol Data field contains the MTP-User information (i.e., SCCP or ISUP), and is similar to the User Data field in the SS7 MTP message (see Figure 3.1.1 (c)). The M3UA layer does not impose the 272-octet SIF length limit as specified by the SS7 MTP2. Larger information blocks can be accommodated directly by M3UA/SCTP. However, the maximum 272-octet block size must be followed when the signaling point interworks with an MTP-based SS7 network that does not support the transfer of larger information blocks. Figure 3.1.3 illustrates the packet format including M3UA, SCTP, IP and Ethernet for SS7 signaling transport over IP-based SS7 network. The message headers are about 86 bytes. The message header overhead for M3UA and the lower layers is about 6.5 times of that for MTP.

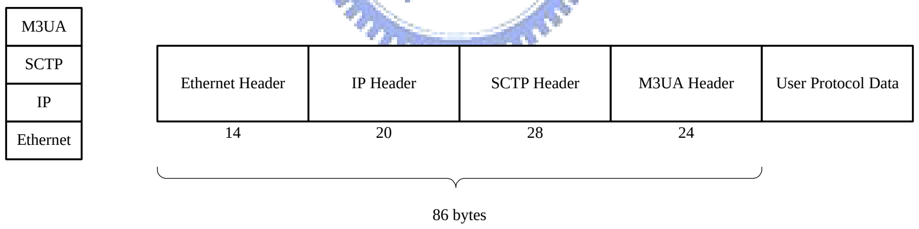


Figure 3.1.3: Message Headers of M3UA and Lower Layers

3.2 Connection Setup

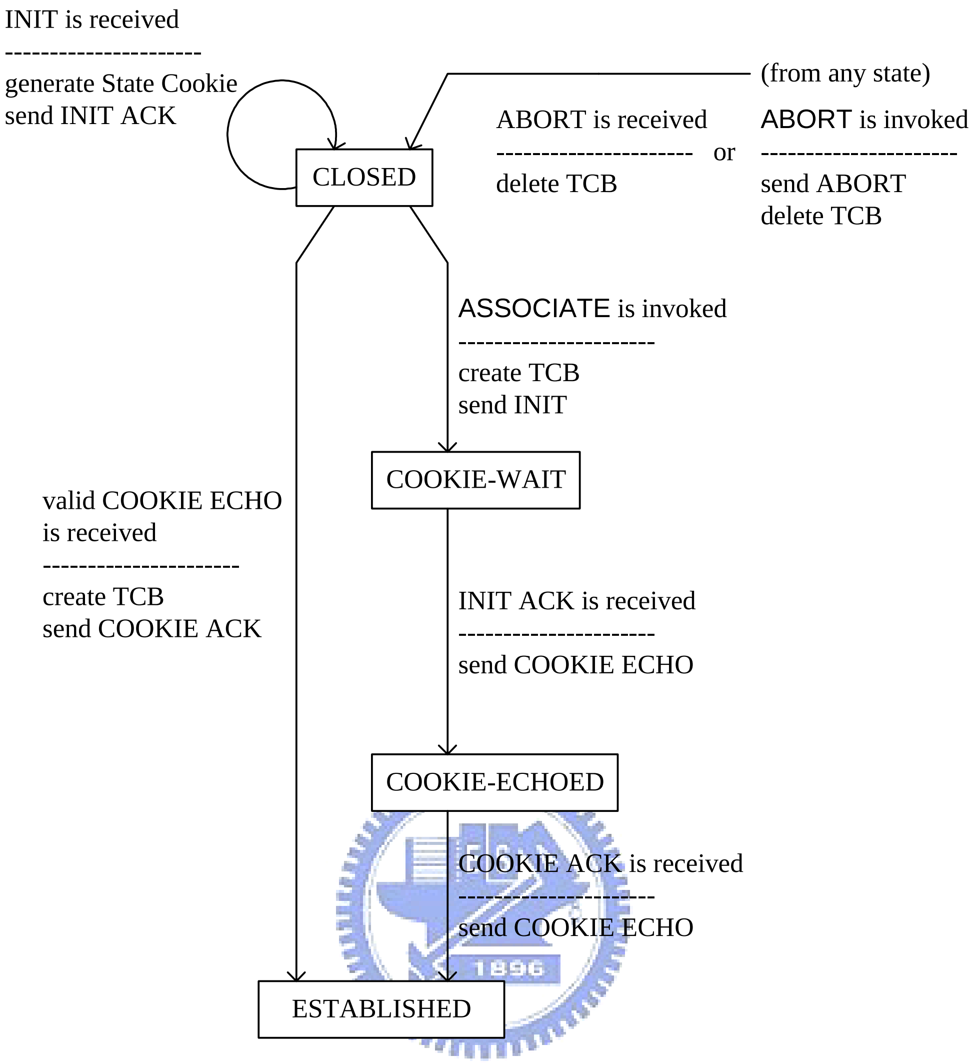
In our MTP-based SS7 implementation, a signaling data link is a bidirectional transmission path for signaling, which consists of standard E1 64 Kb/s channels connected to the SS7 card. A signaling point is typically equipped with multiple SS7 network interfaces (i.e., E1 interfaces) to an adjacent signaling point so that more than one signaling link can be used to carry the SS7 messages with enhanced availability. In this approach, MTP2 provides reliable transfer within a signaling link, and uses FISUs (see Figure 3.1.1 (a)) as a keep-alive signal to detect if the signaling link is available for carrying messages. MTP3 provides functions and procedures related to message routing and network management. MTP3 defines a link set that consists of the signaling links connected to the same adjacent signaling point. In addition to the concept of link set, MTP3 also defines a route as a collection of the link sets used to reach a particular destination. A link set can belong to more than one route. MTP3 uses the routing label of MSU (see Figure 3.1.1 (c)) to determine how to route the messages. The routing label of MSU consists of OPC, DPC and SLS. If the DPC of the received MSU is the local signaling point, then the message is processed by MTP3. If the DPC identifies another signaling point, MTP3 selects an appropriate route according to the information stored in its routing tables to transfer message. The selection of the particular signaling link is determined by the SLS field of MSU. If the MTP-user messages are transferred in sequence, the SLS field should be coded with the same value for the messages. If the MTP-user messages do not need in-sequence delivery service, the messages may be assigned to any SLS or may be assigned to a default SLS such as 0000 to allow load sharing of the message delivery. MTP3 will route the MTP-user messages through the appropriate signaling link based on the loads of these signaling links. If a signaling link fails in that path or a signaling point becomes congested, the messages can be alternatively rerouted by the MTP3 network management.

Each signaling point in an MTP-based SS7 network configures the signaling links, the link sets, the routes and the routing tables in advance. MTP3 does not need to conduct additional connection setup, and the MTP3-user (i.e., SCCP) can use MTP-TRANSFER service provided by MTP3 to transfer messages to the destination.

In the SCTP-based SS7 network, M3UA/SCTP/IP provide the same MTP functions to the M3UA-user (i.e., SCCP or ISUP). In this approach, Internet Protocol provides functions for packet routing in the IP network. SCTP provides reliable transfer with two features that are particularly desirable for SS7 signaling; namely, multi-streaming and multi-homing. An SCTP stream is a unidirectional logical channel established from one SCTP endpoint to another SCTP endpoint, and can be considered a signaling link in the MTP-based approach. The SCTP stream is identified by the Stream ID field in the DATA chunk (see Figure 1.2.3). The Stream Sequence Number (SSN) field in the DATA chunk is used to preserve the data order within a stream. Each stream independently deliveries messages so that the Head of the Line blocking problem in TCP can be avoided. While one stream is blocked and is waiting for the next in-sequence message, message deliveries of other streams are not affected. The M3UA may use the SLS value to select the SCTP stream. The messages that need to be transferred in sequence are assigned to the same SLS value. SCTP also supports multi-homed endpoint that has more than one IP addresses. The IP addresses are typically assigned to different network interfaces of the endpoint. The multi-homed SCTP endpoint specifies available IP addresses during the SCTP association establishment (to be elaborated later), and selects a primary path (i.e. a primary destination address). To ensure reachability, an endpoint sends the HEARTBEAT chunk to its peer endpoint to probe a particular destination address (i.e. IP address) defined in the present association. This mechanism is equivalent to the keep-alive signal (i.e., FISU) in the MTP-SS7 based network. Each endpoint sends DATA chunks through the primary path for normal transmission. Retransmitted DATA chunks use an alternate path. Continued failures of the primary path result in the decision to transmit all chunks to the alternate destination until the primary destination becomes available again. Note that, SCTP does not support load sharing of the multi-homed endpoint by simultaneously using the multiple paths. M3UA provides the MTP3 functions to the M3UA-user (i.e., SCCP). M3UA also provides management of SCTP associations and address mapping from SS7 point codes to IP addresses. Because the MTP3 routing is based on OPC, DPC, and SLS, these parameters are used to determine the IP addresses of SCTP endpoints and specific stream of the association between the endpoints.

In an IP-based SS7 network, the routing tables of IP are configured in advance. However, to support multi-streaming and multi-homing features, SCTP needs to use a four-way handshake procedure to exchange information and allocate resources to establish connection (i.e., SCTP association) between the peer multi-streaming (multi-homing) endpoints. The M3UA-user (i.e., SCCP) invokes MTP-TRANSFER service of M3UA to transfer messages after the SCTP association is established.

The SCTP association is established by a 4-way handshake procedure. A finite state machine for association establishment is implemented in each of the endpoints. The events that drive the finite state machine include the primitives invoked by the SCTP-user (i.e. ASSOCIATE and ABORT in Figure 3.2.1), reception of the SCTP chunks (i.e., INIT, INIT ACK, COOKIE ECHO, COOKIE ACK, and ABORT in Figure 3.2.1), and expiry of the timers. The chunk described in Section 1.2 is the basic structure to carry information in the SCTP packet. Figure 3.2.1 illustrates a simplified state transition diagram for association establishment where the error conditions and timeout events are omitted. The omitted details can be found in [14].



TCB: Transmission Control Block

Figure 3.2.1: State Transition Diagram for SCTP Association Establishment

Suppose that SCTP endpoint A attempts to set up an association with SCTP endpoint Z. The SCTP association establishment process consists of the following steps (shown in Figure 3.2.2).

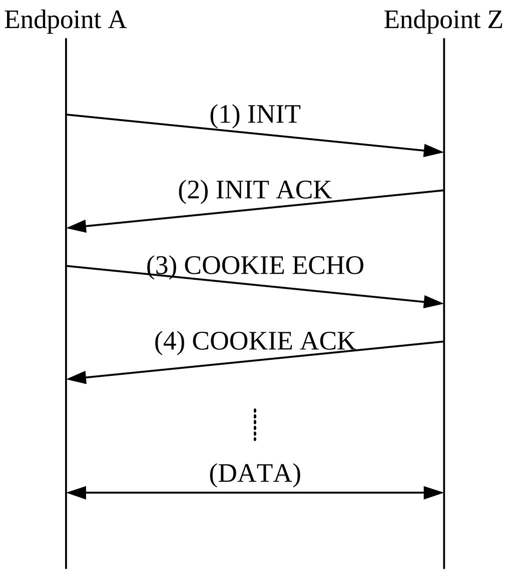


Figure 3.2.2: SCTP Association Establishment by Four-way Handshake

Step 1: Initially, the finite state machines of both endpoints are in the CLOSED state. The SCTP-user in endpoint A invokes the ASSOCIATE primitive to initiate an SCTP association. A Transmission Control Block (TCB) is created to contain all statuses and operational information for the endpoint to maintain and manage this association. Then endpoint A sends an INIT chunk to endpoint Z. In the INIT chunk, endpoint A specifies the number of the streams that it can support in this association. More than one IP address can be included in this INIT chunk if endpoint A is multi-homed. After sending the SCTP packet, endpoint A moves to the COOKIE-WAIT state.

Step 2: Upon receipt of the INIT chunk, endpoint Z shall respond immediately with an INIT ACK chunk. In this chunk, endpoint Z specifies the number of the streams that it can support in this association, and multiple IP addresses are included if endpoint Z is multi-homed. This chunk also includes a State Cookie [14] that is generated by endpoint Z to contain all information necessary to establish the association. The State Cookie is private and only useful to the generator (i.e., endpoint Z) for TCB creation later. The action of sending the INIT ACK chunk does not enable endpoint Z to allocate any resources or keep any states for this new association. Endpoint Z remains in the CLOSED state. In this way, denial of service attacks (such as SYN attacks presented in TCP) can be avoided.

Step 3: Upon receipt of the INIT ACK chunk, endpoint A sends the COOKIE ECHO chunk that includes the same State Cookie received in the INIT ACK chunk. The received State Cookie is private and only useful to the sender of the INIT ACK chunk (i.e., endpoint Z). Endpoint A does not use this State Cookie and must not modify this State Cookie. At the end of this step, endpoint A moves to the COOKIE-ECHOED state.

Step 4: Upon receipt of the COOKIE ECHO chunk, endpoint Z authenticates the State Cookie from the received COOKIE ECHO chunk to insure that the State Cookie was previously generated by endpoint Z. If the authentication succeeds, endpoint Z uses this State Cookie to create a TCB for this association. Endpoint Z replies with a COOKIE ACK chunk, and its finite state machine moves to the ESTABLISHED state. Upon receipt of the COOKIE ACK chunk from endpoint Z, endpoint A moves to the ESTABLISHED state, and the association is established.

3.3 Data Transmission/Ack

In the MTP-based approach, MTP2 provides reliable in-sequence transfer for a signaling link directly connecting two signaling points. There are two error correction methods; namely, basic error correction and Preventive Cyclic Retransmission (PCR). PCR is used whenever satellite transmission is required for signaling links, which will not be elaborated in this thesis. Basic error correction method ensures correct and in-sequence transfer of MSU (see Figure 3.1.1 (c)) over the signaling link, and operates independently in two transmission directions. Basic error correction utilizes positive acknowledgment, negative acknowledgment and retransmission of MSU. Several sequence numbers and indication bits are included in these messages. Forward Sequence Number (FSN) and

Forward Indicator Bit (FIB) of MSUs (see Figure 3.1.1 (c)) in one direction are associated with Backward Sequence Number (BSN) and Backward Indicator Bit (BIB) of SUs (see Figure 3.1.1) in the other direction. FSN is the sequence of an MSU. FIB is an one-bit field of MSU to indicate retransmission start. If an MSU has an FIB value different from the previous MSU, it represents the first retransmitted MSU. It makes no difference whether the FIB value is 0 or 1. BSN is used to acknowledge the last correctly received MSU. BIB is an one-bit field of SU with the following usage.

- If the BIB value of an SU is equal to the BIB value of the previous SU, it positively acknowledges the MSU whose FSN is equal to the BSN of the SU.
- If an SU has a BIB value different from the previous SU, it represents a negative acknowledgment that results in retransmission of the corresponding MSUs.

Like FIB, it makes no difference whether the BIB value is 0 or 1. Note that under normal operation, the FIB included in the transmitted MSU is equal to the BIB value of the received SU to indicate that the sender has received the last SU sent by the receiver. Suppose that signaling point A sends MSUs to signaling point Z. When signaling point Z receives an MSU, it will first check if the FSN of the received MSU exceeds the FSN of the last correctly received MSU by 1 to ensure in-sequence delivery. Secondly, signaling point Z will check the FIB of the received MSU. If the FIB of the received MSU is equal to the BIB of the last sent SU, the MSU is forwarded to the MTP3. If the received MSU is not in the FSN sequence, and the FIB of the received MSU is equal to the BIB of the last sent SU, signaling point Z sends a negative acknowledgement (i.e., the SU whose BIB value differs from the BIB value of the last SU) to signaling point A to request retransmission. Upon receipt of the negative acknowledgment from signaling point Z, signaling point A checks the BIB of the received SU. If the BIB of the received SU is not equal to the FIB of the last MSU, all un-received MSUs are transmitted in sequence starting with the MSU whose FSN exceeds the FSN of the most recently acknowledged

MSU by 1. Signaling point A can send new MSUs only when the last un-received MSU has been transmitted. At the beginning of a retransmission, the FIB value of the first retransmitted MSU is inverted, which equals to the BIB value of the last received SU (i.e., the negative acknowledgement). The new FIB value is maintained in subsequently transmitted MSU until a new retransmission is started. Thus, under normal operations, the FIB included in the transmitted MSU is equal to the BIB value of the received SU.

Figure 3.3.1 shows an example of basic error correction for MSUs sent from signaling point A to signaling point Z. Assuming that before the transmission of MSU(FIB=0, FSN=14), the BIB value of the last received SU in signaling point A is 0, and signaling point Z has received MSUs with FSNs up to 13. In this scenario, signaling point A sends MSU(FIB=0, FSN=14), and signaling point Z positively acknowledges MSUs with FSNs up to 14 by SU(BIB=0, BSN=14). Signaling point A continues to send MSU(FIB=0, FSN=15) and MSU(FIB=0, FSN=16). Signaling point Z receives MSU(FIB=0, FSN=15) with error and discards MSU(FIB=0, FSN=15). Later, MSU(FIB=0, FSN=16) is arrived. MSU(FIB=0, FSN=16) will fail the “sequence test” since the FSN of MSU(FIB=0, FSN=16) does not exceed the FSN of the last received MSU (i.e., MSU(FIB=0, FSN=14)) by 1. Signaling point Z discards MSU(FIB=0, FSN=16) and negatively acknowledges MSUs with FSNs up to 14 by SU(BIB=1, BSN=14). Note that the BIB of SU(BIB=1, BSN=14) is not equal to the BIB of the previous SU (i.e., SU(BIB=0, BSN=14)). Upon receipt of the negative acknowledgment SU(BIB=1, BSN=14), signaling point A immediately starts retransmission of MSU(FIB=1, FSN=15) and MSU(FIB=1, FSN=16). Note that the FIB of MSU(FIB=1, FSN=15) is not equal to the FIB of the previous MSU (i.e., MSU(FIB=0, FSN=16)) which indicates starting of retransmission. Upon receipt of MSU(FIB=1, FSN=15) and MSU(FIB=1, FSN=16) correctly, signaling point Z positive acknowledges MSUs with

FSNs up to 16 by SU(BIB=1, BSN=16).

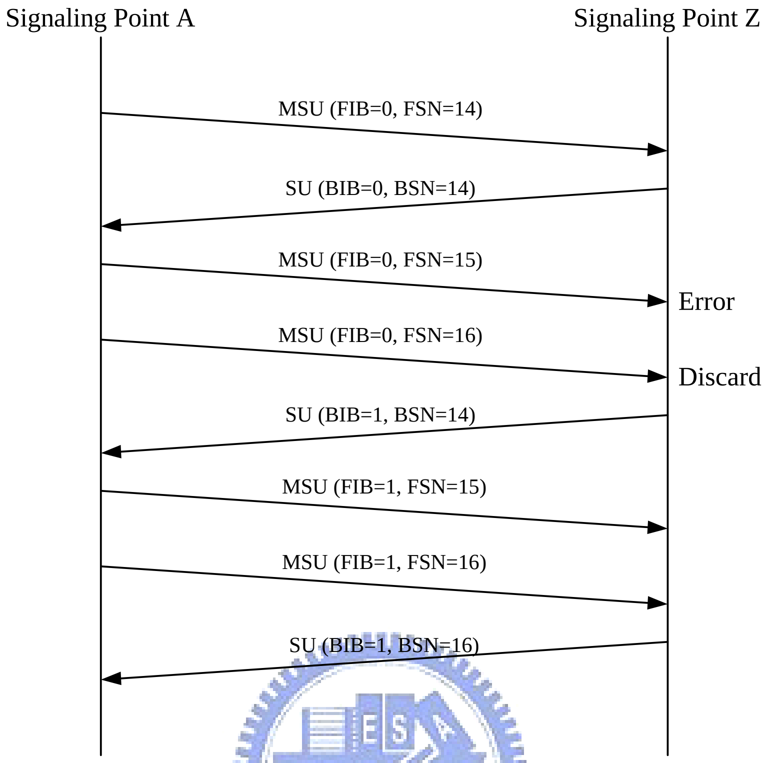
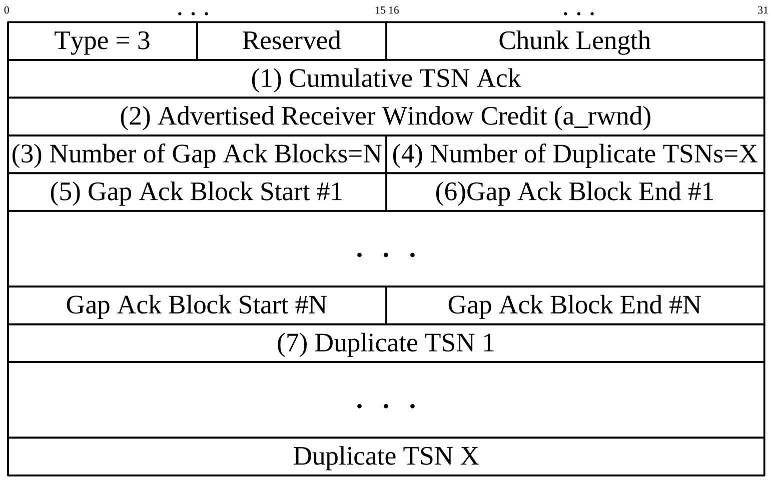


Figure 3.3.1: An Example of Basic Error Correction

For the IP-based SS7 network, an SCTP stream can be considered as a one-way signaling link in the MTP-based SS7 network. Like MTP, SCTP also provides reliable in-sequence transfer within a stream over the IP-based SS7 network. As described in Section 1.2, an SCTP packet (see Figure 1.2.1) is composed of an SCTP common header and one or more SCTP chunks which contain control or data information. The SCTP-user (i.e., M3UA) message passed to the SCTP layer for transmission will be carried in the DATA chunk (see Figure 1.2.3). The in-sequence delivery service utilizes Stream ID (see Figure 1.2.3 (2)) and Stream Sequence Number (SSN; see Figure 1.2.3 (3)) in the DATA chunk. The SCTP layer forwards the received messages (i.e., the DATA chunks) associated with the specific Stream ID to the SCTP-user (i.e., M3UA) in the SSN order. SSN is associated with a particular stream. Unlike MTP2, the DATA chunks that are not received in the SSN sequence are stored in the receive buffer allocated by the SCTP layer. The receive buffer stores all received DATA chunks that have not yet been forwarded to the SCTP-user (e.g., if the DATA chunks are not received in the SSN sequence or the SCTP-user is busy). The DATA chunk in the receive buffer is released until it has been forwarded to the SCTP-user. The SCTP uses Selective Acknowledgement [14] to ensure reliable transfer. In this approach, each DATA chunk is assigned a sequence number TSN (see Figure 1.2.3 (1)) that is used to acknowledge the received DATA chunks and to detect duplicate deliveries. TSN is used within an association that may contain more than one stream for either ordered delivery service or unordered delivery service, and is independent of any SSN assigned at the stream level. Note that SSN cannot be used for the purpose of TSN because SSN is only associated with a particular stream. The receiver uses the Selective Acknowledgement (SACK) chunk to acknowledge the received DATA chunks, to inform the sender of the gaps found in a received TSN sequence, and to provide the available receive buffer size of the receiver. (A "gap" represents not-yet-received messages of consecutive TSNs.) The sender may retransmit the DATA chunks based on the SACK chunk sent from the receiver.

Figure 3.3.2 illustrates the SACK chunk format, which includes Type, Chunk Length, Cumulative TSN Ack, Advertised Receiver Window Credit (a_rwnd), Number of Gap Ack Blocks, Number of Duplicate TSNs and the optional fields (i.e., Gap Ack Block and Duplicate TSN). Cumulative TSN Ack (Figure 3.3.2 (1)) contains the TSN of the last in-sequence DATA chunk received before a gap. A_rwnd (Figure 3.3.2 (2)) indicates the available receive buffer size for the sender of this SACK chunk. Number of Gap Ack Blocks (Figure 3.3.2 (3)) indicates the number of Gap Ack Blocks included in this SACK chunk. Number of Duplicate TSNs (Figure 3.3.2 (4)) indicates the number of duplicate TSNs received by the endpoint. Each Gap Ack Block



<table>
<tr><td>Type = 3</td><td>Reserved</td><td colspan="2">Chunk Length</td></tr>
<tr><td colspan="4">(1) Cumulative TSN Ack</td></tr>
<tr><td colspan="4">(2) Advertised Receiver Window Credit (a_rwnd)</td></tr>
<tr><td colspan="2">(3) Number of Gap Ack Blocks=N</td><td colspan="2">(4) Number of Duplicate TSNs=X</td></tr>
<tr><td colspan="2">(5) Gap Ack Block Start #1</td><td colspan="2">(6)Gap Ack Block End #1</td></tr>
<tr><td colspan="4">. . .</td></tr>
<tr><td colspan="2">Gap Ack Block Start #N</td><td colspan="2">Gap Ack Block End #N</td></tr>
<tr><td colspan="4">(7) Duplicate TSN 1</td></tr>
<tr><td colspan="4">. . .</td></tr>
<tr><td colspan="4">Duplicate TSN X</td></tr>
</table>

TSN: Transmission Sequence Number

Figure 3.3.2: SACK Chunk Format

contains Gap Ack Block Start offset (Figure 3.3.2 (5)) and Gap Ack Block End offset (Figure 3.3.2 (6)) that indicate the range of TSNs been received. In order to calculate the actual TSNs, these offsets (i.e., Gap Ack Block Start and Gap Ack Block End) are added to the Cumulative TSN Ack (Figure 3.3.2 (1)). Duplicate TSN (Figure 3.3.2 (7)) indicates that the chunk of the TSN has been received more than once. Further usage of Duplicate TSN has not yet been defined in the SCTP specification.

Figure 3.3.3 shows an example of the SACK chunk. Consider a scenario where endpoint A sends 11 DATA chunks to endpoint Z in an SCTP stream. The size of each DATA chunk is 200 bytes. Each SCTP packet contains one DATA chunk, and the receive buffer size of endpoint Z is 5600 bytes. Suppose that endpoint Z has received the DATA chunks with TSNs 1-4, 7 and 8. Since endpoint Z has not received DATA chunks with TSNs 5 and 6, the DATA chunks with TSNs 7 and 8 cannot be forwarded to the SCTP-user and are stored in the receive buffer. At this point, the available receive buffer size becomes 5200 bytes. When the SCTP packet containing the DATA chunk with TSN 11 is received, endpoint Z detects that the DATA chunks with TSNs 5, 6, 9 and 10 has not been received, and the DATA chunk with TSN 11 is stored in the receive buffer. At this point, the available receive buffer size becomes 5000 bytes. After processing the SCTP packet containing the DATA chunk with TSN 11, endpoint Z immediately sends a SACK chunk to endpoint A as the acknowledgement. The format of the SACK chunk is illustrated in Figure 3.3.3. Upon receipt of the SACK chunk from endpoint Z, endpoint A will retransmit the DATA chunks with TSNs 5, 6, 9, and 10.

0 . . . 15 16 . . . 31

<table>
<tr><td>Type = 3</td><td>Reserved</td><td>Chunk Length = 24</td></tr>
<tr><td colspan="3">Cumulative TSN Ack = 4</td></tr>
<tr><td colspan="3">a_rwnd = 5000</td></tr>
<tr><td colspan="2">Number of Gap Ack Blocks = 2</td><td>Number of Duplicate TSNs = 0</td></tr>
<tr><td colspan="2">Gap Ack Block Start #1 = 3</td><td>Gap Ack Block End #1 = 4</td></tr>
<tr><td colspan="2">Gap Ack Block Start #2 = 7</td><td>Gap Ack Block End #2 = 7</td></tr>
</table>

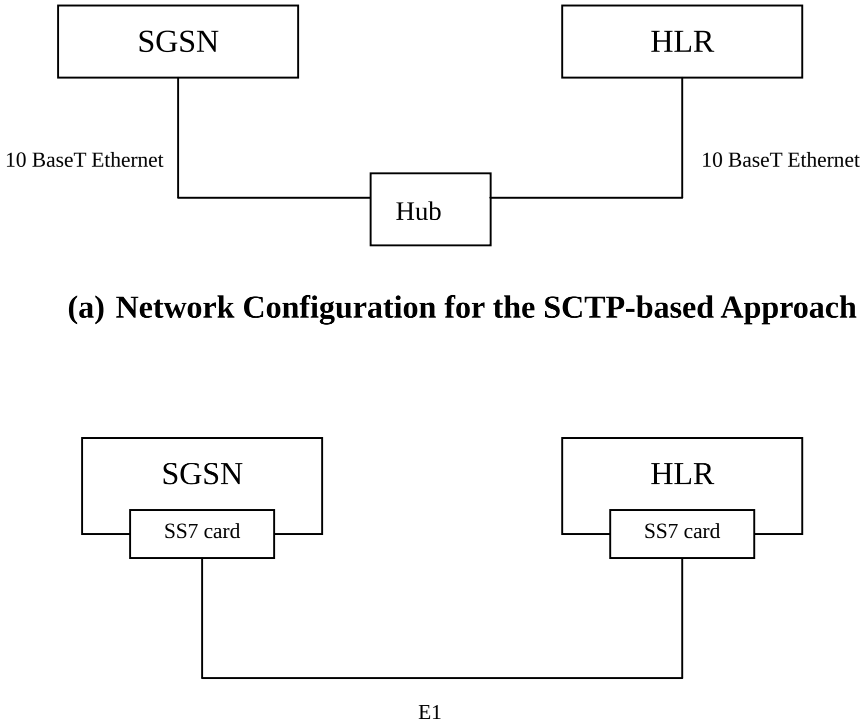
Figure 3.3.3: An Example of the SACK Chunk

3.4 Performance Measurement

In this section, we use the Send Authentication Info procedure defined in 3GPP Technical Specification 23.060 [1] as an example to illustrate the performance of the SCTP-based and the MTP-based MAP approaches.

3.4.1 Measurement Environment

Our experimental environment consists of two 3G network nodes, the SGSN and the HLR. The SGSN runs Red Hat Linux 7,1 on the i686 platform with a 600 MHz AMD-K7 processor, 256 MB of RAM, an Intel 82557 Ethernet card and a Performance Technologies PCI 372A SS7 card [13]. The HLR runs Solaris 8 on the sun4u sparc SUNW UltraSPARC-IIi-cEngine platform with a 440 MHz UltraSPARC-IIi processor, 512 MB of RAM, two Sun Happy Meal Fast Ethernet 100 Mb/s Ethernet cards and a Performance Technologies CPC372PQ SS7 card. Figure 3.4.1(a) illustrates the network configuration of SGSN and HLR for the SCTP-based MAP. In this configuration, the SGSN and the HLR are connected via Buffalo LGH-M5P 10 Mbps Ethernet hub. Figure 3.4.1(b) illustrates the network configuration of SGSN and HLR for the MTP-based MAP. In this configuration the SGSN and the HLR are directly connected by their SS7 interfaces and we configure both the SS7 interfaces of them to use one channel that provides 64 Kb/s bandwidth for unidirectional SS7 signaling transport over E1.



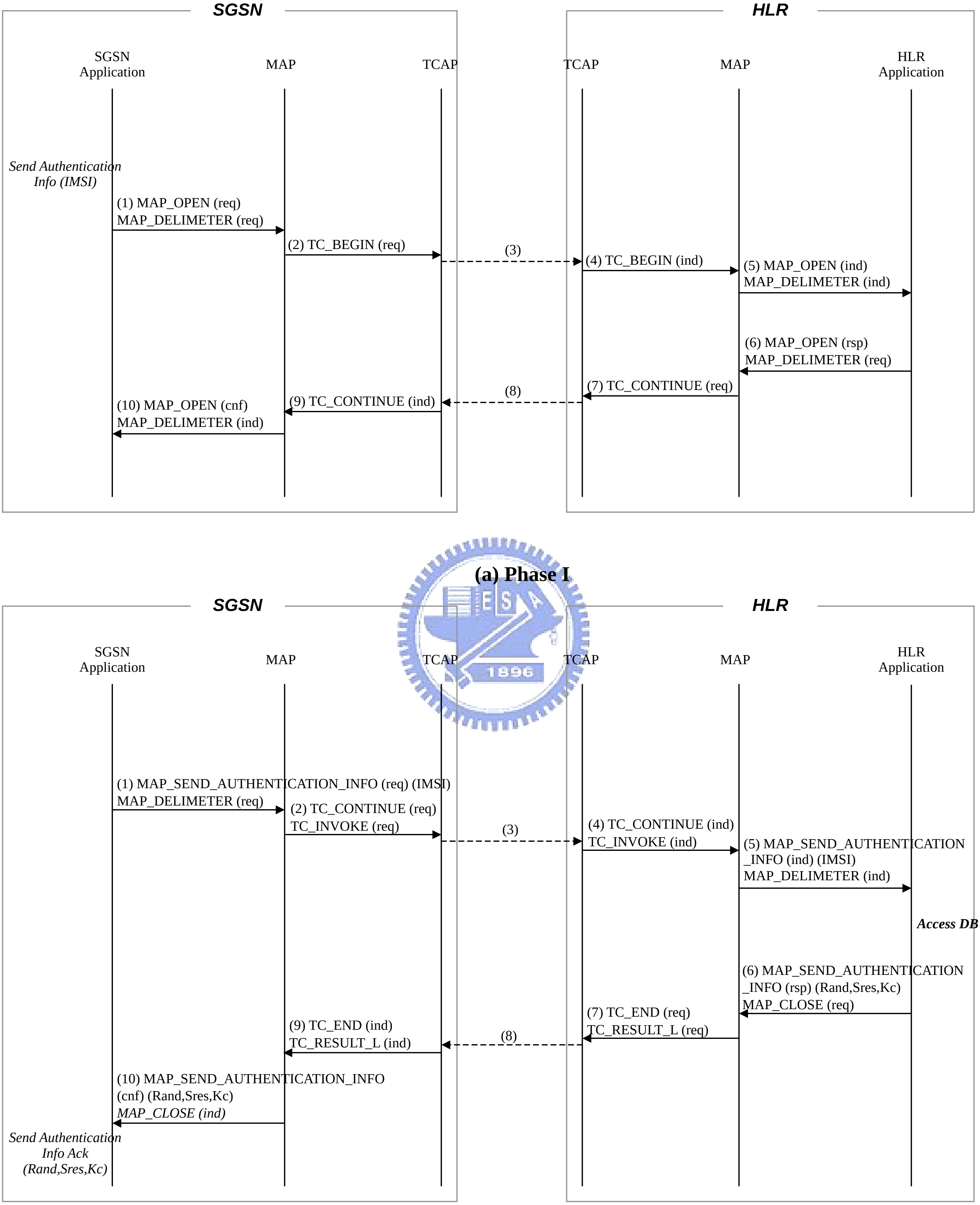
(a) Network Configuration for the SCTP-based Approach

(b) Network Configuration for the SS7-based Approach

Figure 3.4.1: Network Configuration for Our Experiments

3.4.2 Testing Scenarios

We use the Send Authentication Info procedure defined in 3GPP Technical Specification 23.060 [1] as an example to test the performance of the SCTP-based and the MTP-based MAP mechanisms. The procedure consists of two phases. Phase I includes a common MAP service MAP_OPEN for establishing a MAP dialogue between peer MAP service users. Phase II includes the mobility MAP service MAP_SEND_AUTHENTICATION_INFO for retrieving authentication information from the HLR. The SS7 message exchanges between the SGSN and the HLR for this procedure are illustrated in Figure 3.4.2, and the detailed steps are described as follows.



(a) Phase I

(b) Phase II

Figure 3.4.2: Send Authentication Info Message Flow

Phase I (Common MAP Service)

Step 1 [SGSN]: The SGSN application initiates a MAP dialogue by invoking the MAP_OPEN Request primitive. This primitive is followed by the MAP_DELIMETER Request primitive. The MAP_DELIMETER Request primitive is used to explicitly request the TCAP layer to transfer the MAP protocol data units (MAP_OPEN in this step) to the peer entity.

Step 2 [SGSN]: The MAP layer handles the MAP_OPEN Request primitive and enables the TC_BEGIN primitive to start a new dialogue at the TCAP layer. The TC_BEGIN primitive is of the type Request. The type "Request" indicates that the primitive is issued from the upper layer to the TCAP layer. Before the MAP layer receives the response from the TCAP layer, it refuses to accept new requests from the SGSN application.

Step 3 [SGSN -> HLR]: The TCAP layer generates the SS7 TCAP message with the appropriate package and the component types. The message is sent to the HLR. As described in Section 1.3, there are two lower-layer protocol mechanisms to deliver this message. We do not repeat here.

Step 4 [HLR]: When the TCAP message arrives at the HLR, the TCAP layer issues the TC_BEGIN primitive (Indication) to the MAP layer. The type "Indication" means that the primitive is issued from the TCAP layer to the upper layer.

Step 5 [HLR]: If the MAP layer does not identify any errors from the received TC_BEGIN Indication primitive and the system is not overloaded, the MAP layer issues the MAP_OPEN Indication primitive to the HLR application. This primitive is followed by the MAP_DELIMETER Indication primitive. The MAP layer then waits for the MAP_OPEN Response primitive from the HLR application.

Step 6 [HLR]: The HLR application checks the MAP_OPEN Indication primitive. Assuming that no error occurs, the HLR application accepts the dialogue by invoking the MAP_OPEN Response primitive, followed by the MAP_DELIMETER Request primitive to the MAP layer.

Step 7 [HLR]: The MAP layer indicates that it wants to continue the dialogue by issuing a TC_CONTINUE Request primitive. At this point, the MAP dialogue is considered established at the HLR side.

Step 8 [HLR -> SGSN]: The TCAP layer of the HLR generates the SS7 TCAP message with the appropriate package and the component types. The message is sent to the SGSN. As described in Section 1.3, there are two lower-layer protocol mechanisms to deliver this message. We do not repeat here.

Step 9 [SGSN]: When the TCAP message arrives at the SGSN, the TCAP layer issues TC_CONTINUE Indication primitive to the MAP layer.

Step 10 [SGSN]: The MAP layer processes the TC_CONTINUE Indication primitive and issues the MAP_OPEN Confirm primitive to the SGSN application. The SGSN application handles the MAP_OPEN Confirm primitive that indicates the success of the MAP dialogue establishment between the SGSN and the HLR. This primitive is followed by the MAP_DELIMETER Indication primitive.

After Phase I, the SGSN and the HLR are able to communicate with each other through the established MAP dialogue. The SGSN application then initiates Phase II to retrieve authentication information from the HLR. The detailed steps are described as follows.

Phase II (Mobility MAP Service)

Step 1 [SGSN]: The SGSN application retrieves authentication information from the HLR by invoking the MAP_SEND_AUTHENTICATION_INFO Request primitive (with the argument International Mobile Subscriber Identity (IMSI) that is used to identify the mobile user). This primitive is followed by the MAP_DELIMETER Request primitive.

Step 2 [SGSN]: Upon receipt of the MAP_SEND_AUTHENTICATION_INFO Request primitive from the SGSN application, the MAP layer uses the TC_INVOKE Request primitive to set the operation code and TCAP parameters, and issues the TC_CONTINUE Request primitive to continue the dialogue.

Steps 3 and 4 [SGSN -> HLR]: These steps are similar to Steps 3 and 4 in Phase I.

Step 5 [HLR]: The MAP layer checks the TC_CONTINUE Indication primitive as described in Step 5 in Phase I, and then encounters the TC_INVOKE Indication primitive. The MAP layer checks if the received arguments are correct, if the service can be identified and if the service parameters are available. Assuming that no error occurs, the MAP layer issues the MAP_SEND_AUTHENTICATION_INFO Indication primitive to the HLR application.

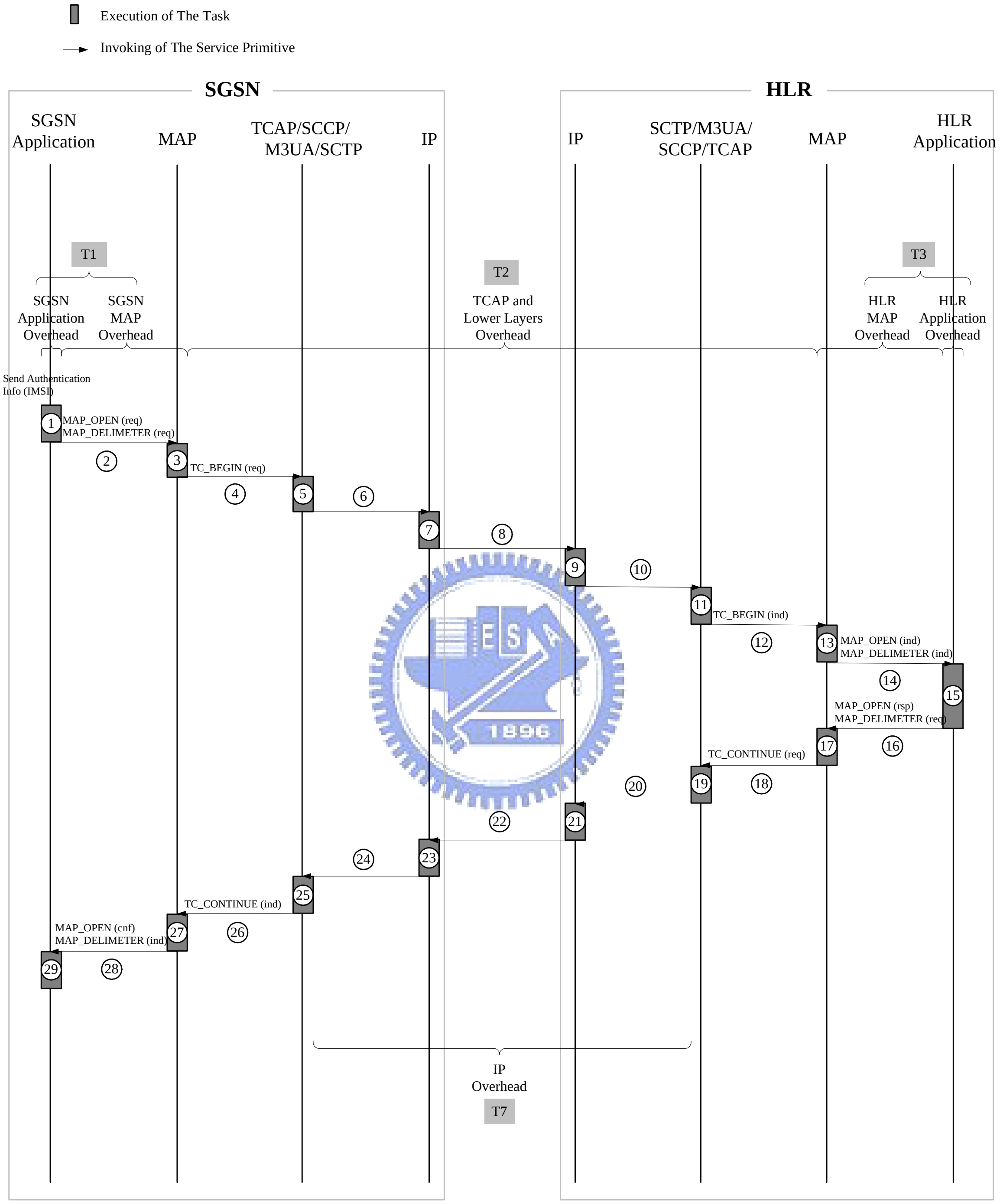
Step 6 [HLR]: Upon receipt of the MAP_SEND_AUTHENTICATION_INFO Indication primitive from the MAP layer, the HLR application first generates a 128-bit random number called Rand and accesses the authentication database to retrieve the secret key Ki of the IMSI. The HLR application then produces an encryption key Kc and a signed result SRES. The details can be found in [3]. The HLR application then returns the authentication vector (Rand, SRES, Kc) through the MAP_SEND_AUTHENTICATION_INFO Response primitive. This primitive is followed by the MAP_CLOSE Request primitive. The MAP_CLOSE Request primitive is used to clear the MAP dialogue.

Step 7 [HLR]: Upon receipt of the MAP_SEND_AUTHENTICATION_INFO Response primitive and the MAP_CLOSE Request primitive from the HLR application, the MAP layer issues the TC_RESULT_L Request primitive that returns the results of the invoked operation to the TCAP layer, and enables the TC_END Request primitive that indicates the end of the dialogue.

Steps 8 and 9 [HLR -> SGSN]: Delivered by the lower-layer protocols at Step 8, the TCAP message arrives at the SGSN. The TCAP layer issues the TC_END/TC_RESULT_L Indication primitives to the MAP layer.

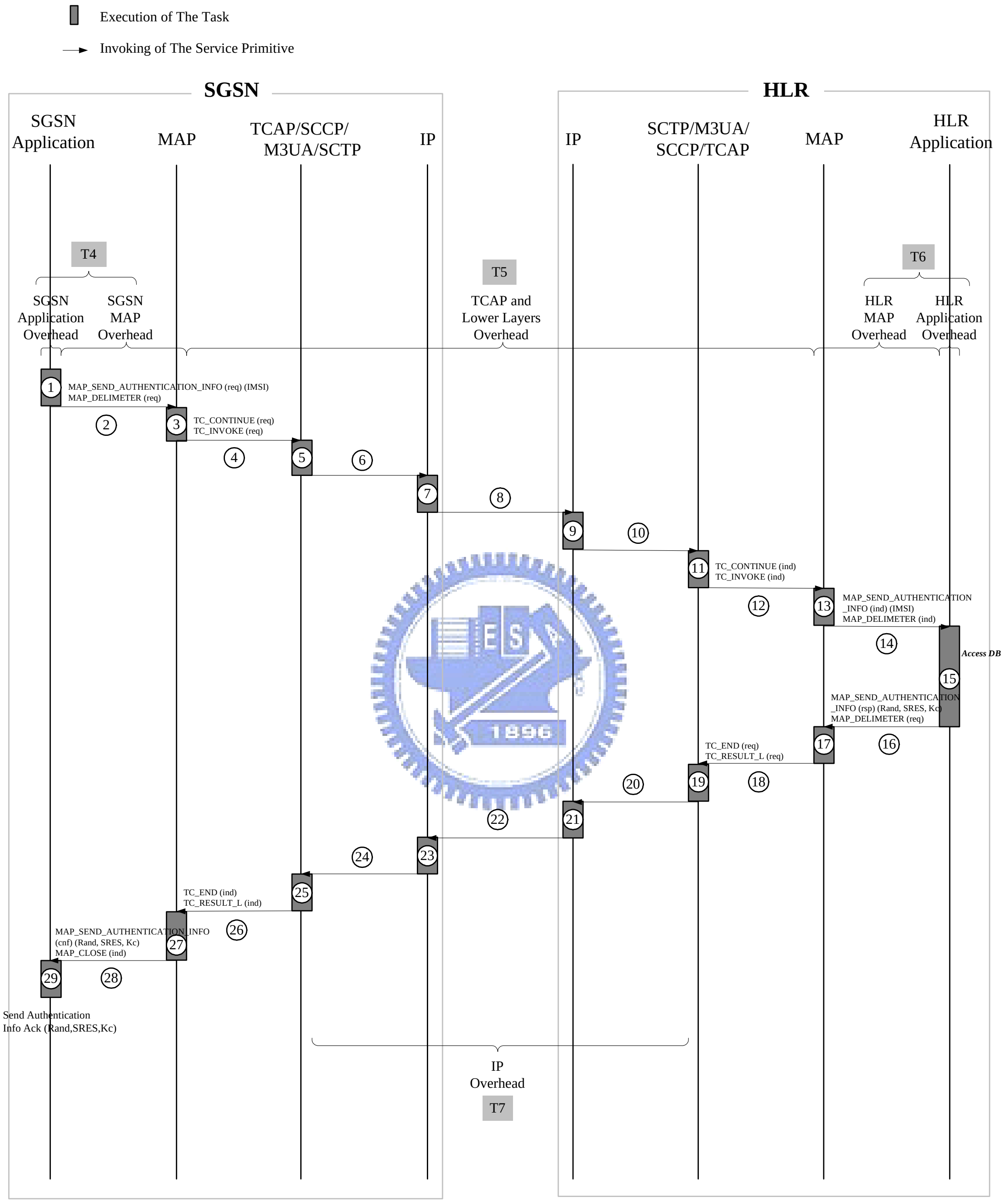
Step 10 [SGSN]: The MAP layer maps the TC_RESULT_L parameters to the MAP_SEND_AUTHENTICATION_INFO Confirm primitive and issues the MAP_CLOSE Indication primitive to the SGSN application. The SGSN application then handles the MAP_SEND_AUTHENTICATION Confirm primitive and finishes the Send Authentication Info procedure. The dialogue is terminated by the MAP_CLOSE and the TC_END Indication primitives.

The execution overhead for the MAP_OPEN service consists of five components. Figure 3.4.3 (a) shows the execution overhead for the SCTP-based approach. In this figure, the SGSN application overhead includes the execution of the application task at the SGSN (i.e., (1) and (29) in Figure 3.4.3 (a)). The HLR application overhead includes the execution of the application task at the HLR (i.e., (15) in Figure 3.4.3 (a)). The SGSN MAP overhead includes the execution of the MAP task and invoking of the MAP primitive at the SGSN (i.e., (2), (3), (27) and (28) in Figure 3.4.3 (a)). The HLR MAP overhead includes the execution of the MAP task and invoking of the MAP primitive at the HLR (i.e., (13), (14), (16), and (17) in Figure 3.4.3 (a)). The TCAP and the lower layers overhead includes (I) the executions of the TCAP, SCCP, M3UA and SCTP tasks and the invoking of the TCAP, SCCP, M3UA primitives and SCTP service (i.e., (4), (5), (11), (12), (18), (19), (25), and (26) in Figure 3.4.3 (a)), and (II) the IP overhead between both the SGSN and the HLR. The IP overhead includes (I) the execution of the IP task (i.e. (7), (9), (21), and (23) in Figure 3.4.3 (a)), (II) invoking of the IP service (i.e. (6), (10), (20), and (24) in Figure 3.4.3 (a)), and (III) the IP packet transmission times (i.e., (8) and (22) in Figure 3.4.3 (a)) between both network nodes.



(a) MAP_OPEN Service

Figure 3.4.3: Execution Overhead for the SCTP-based Approach

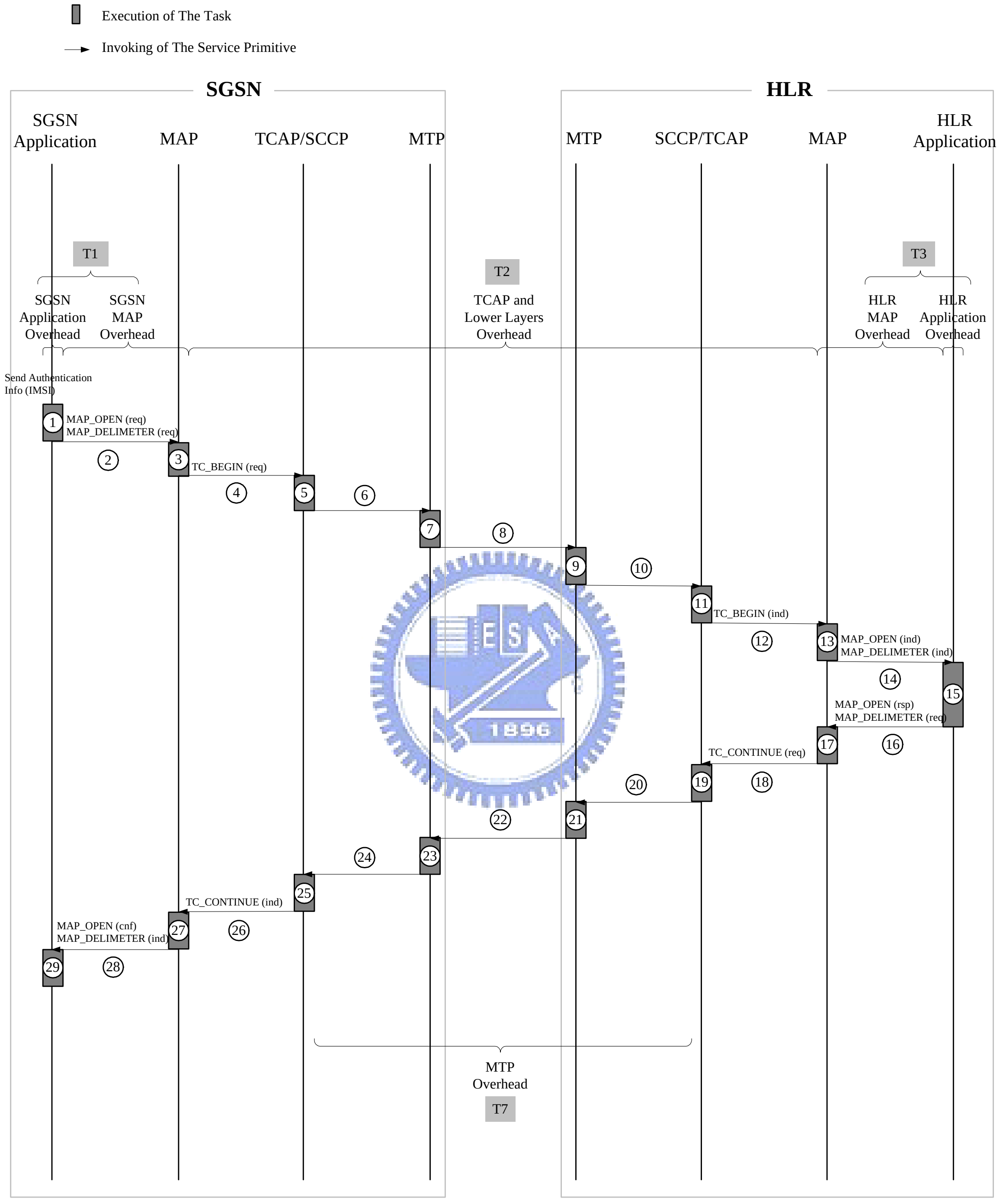


(b) MAP_SEND_AUTHENTICATION_INFO Service

Figure 3.4.3: Execution Overhead for the SCTP-based Approach

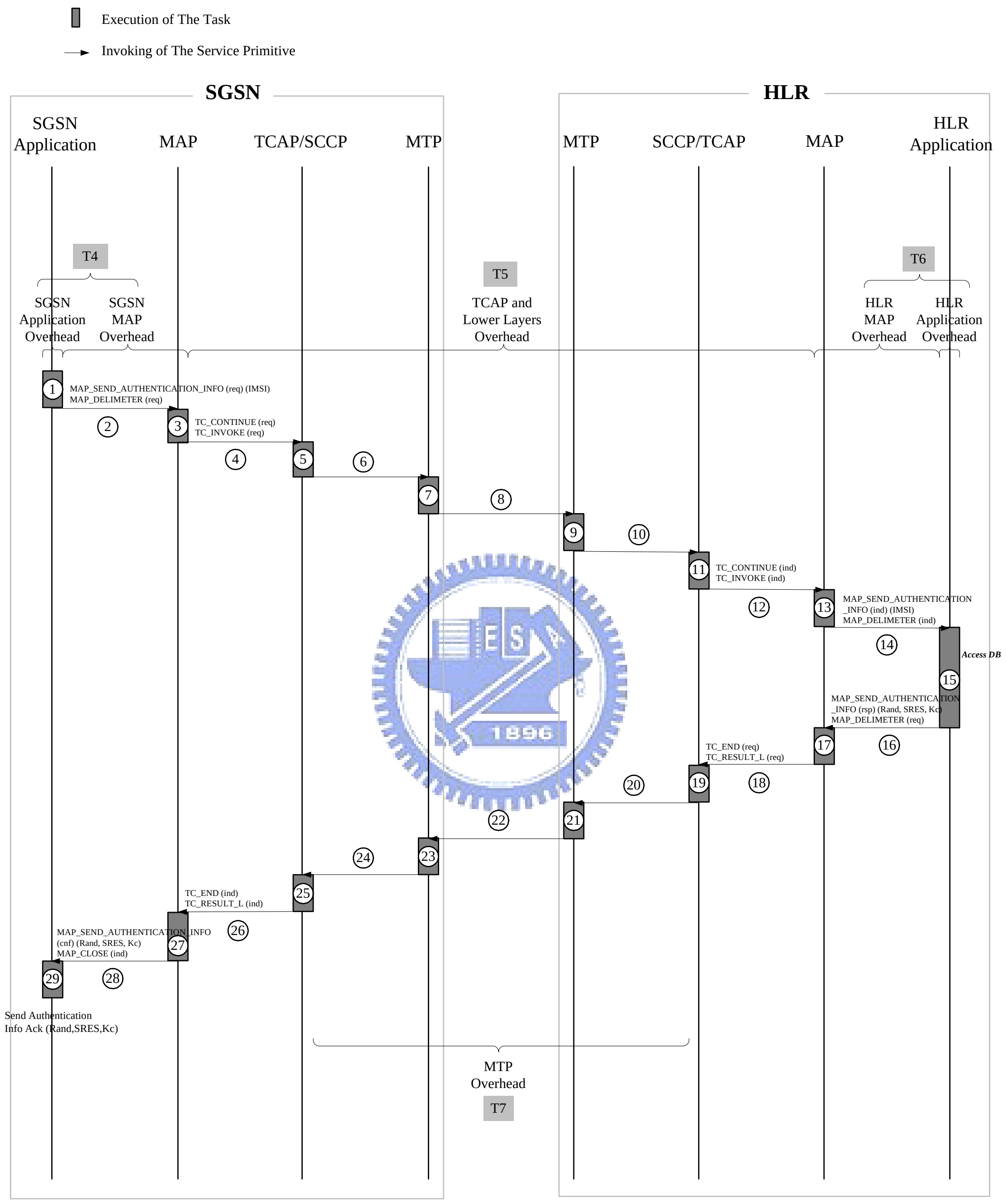
As shown in Figure 3.4.3 (b), the execution overhead for the MAP_SEND_AUTHENTICATION_INFO service is similar to that for the MAP_OPEN service except that the HLR application overheads are different. When the HLR application receives the MAP_SEND_AUTHENTICATION_INFO Indication primitive from the MAP layer, the HLR application will access the authentication database and generate authentication vector (Rand, SRES, Kc) as describe in Step 6 in Phase II.

Figure 3.4.4 (a) shows the MAP_OPEN execution overheads for the MTP-based approach. These overheads are similar to those for the SCTP-based approach except that the TCAP and the lower layers overheads are different. The TCAP and the lower layers overhead for the MTP-based approach includes (I) the executions of the TCAP and SCCP tasks and the invoking of the TCAP and SCCP primitives (i.e., (4), (5), (11), (12), (18), (19), (25), and (26) in Figure 3.4.4 (a)), and (II) the MTP overhead between both the SGSN and the HLR. The MTP overhead includes (I) the execution of the MTP task (i.e., (7), (9), (21), and (23) in Figure 3.4.4 (a)), (II) invoking of the MTP primitive (i.e., (6), (10), (20), and (24) in Figure 3.4.4 (a)), and (III) the MTP packet transmission times (i.e., (8) and (22) in Figure 3.4.4 (a)) between both network nodes. The execution overhead for the MAP_SEND_AUTHENTICATION_INFO service (see Figure 3.4.4 (b)) is similar to that for the MAP_OPEN service as described in the previous paragraph, which will not be repeated here.



(a) MAP_OPEN Service

Figure 3.4.4: Execution Overhead for the MTP-based Approach



(b) MAP_SEND_AUTHENTICATION_INFO Service

Figure 3.4.4: Execution Overhead for the MTP-based Approach

For the SCTP-based approach and the MTP-based approach, we measure the following elapsed times:

T1: The SGSN application overhead and the SGSN MAP overhead for the MAP_OPEN service

T2: The overheads of the TCAP and the lower layers for the MAP_OPEN service

T3: The HLR application overhead and the HLR MAP overhead for the MAP_OPEN service

T4: The SGSN application overhead and the SGSN MAP overhead for the MAP_SEND_AUTHENTICATION_INFO service

T5: The overheads of the TCAP and the lower layers for the MAP_SEND_AUTHENTICATION_INFO service

T6: The HLR application overhead and the HLR MAP overhead for the MAP_SEND_AUTHENTICATION_INFO service

T7: The IP overhead for both the MAP_OPEN and the MAP_AUTHENTICATION_INFO services. The elapsed time T7 is only measured when SS7 signaling message is transported over IP-based SS7 network.

The elapsed time for the MTP overhead is not measured. As described in Section 2.2, the TCAP, SCCP, and MTP layers for the MTP-based approach are implemented in the hardware (i.e., Performance Technologies SS7 cards [13]), we are not able to collect the detailed statistics (i.e., the MTP overhead) for the MTP-based approach.

3.4.3 Measurement Results

We collect T1, T2, T3, T4, T5, T6 and T7 statistics during execution of 50,000 separate Send Authentication Info procedures over IP-based SS7 network and over MTP-based SS7 network, respectively. Figures 3.4.5-3.4.11 show the histograms of the statistics where the solid curves represent the statistics over IP-based SS7 network and the dashed curves represent the statistics over MTP-based SS7 network. The details are elaborated as follows.

Analysis of the elapsed time T1 of the SGSN application overhead and the SGSN MAP overhead for the MAP_OPEN service (see Figures 3.4.3 (a) and 3.4.4 (a)): As described in Chapter 2, the SCTP-based approach and the MTP-based approach use the same implementation of the MAP layer. However, they have different implementations for communication with the TCAP layer. Therefore, Figure 3.4.5 shows that the elapsed times T1 measured for these two approaches have about 4.58% discrepancy. The same effect is also observed for the elapsed time T4 in Figure 3.4.8. For the SCTP-based approach, the T1 statistic has the mean 1.441 ms, and the variance 0.000973 ms^2. For the MTP-based approach, the T1 statistic has the mean 1.506 ms, and the variance 0.34582 ms^2.

Analysis of the TCAP and the lower layer overheads T2 for the MAP_OPEN service (see Figures 3.4.3 (a) and 3.4.4 (a)): Because the lower-layer protocols for the SCTP-based approach and the MTP-based approach are different (and therefore the implementations are different), the elapsed times T2 for these two approaches significantly differ. As shown in Figure 3.4.6, the elapsed time T2 for the MTP-based approach is about 6.5 times of that for the SCTP-based approach. Same effect is observed for the elapsed time T5 in Figure 3.4.9. For the SCTP-based

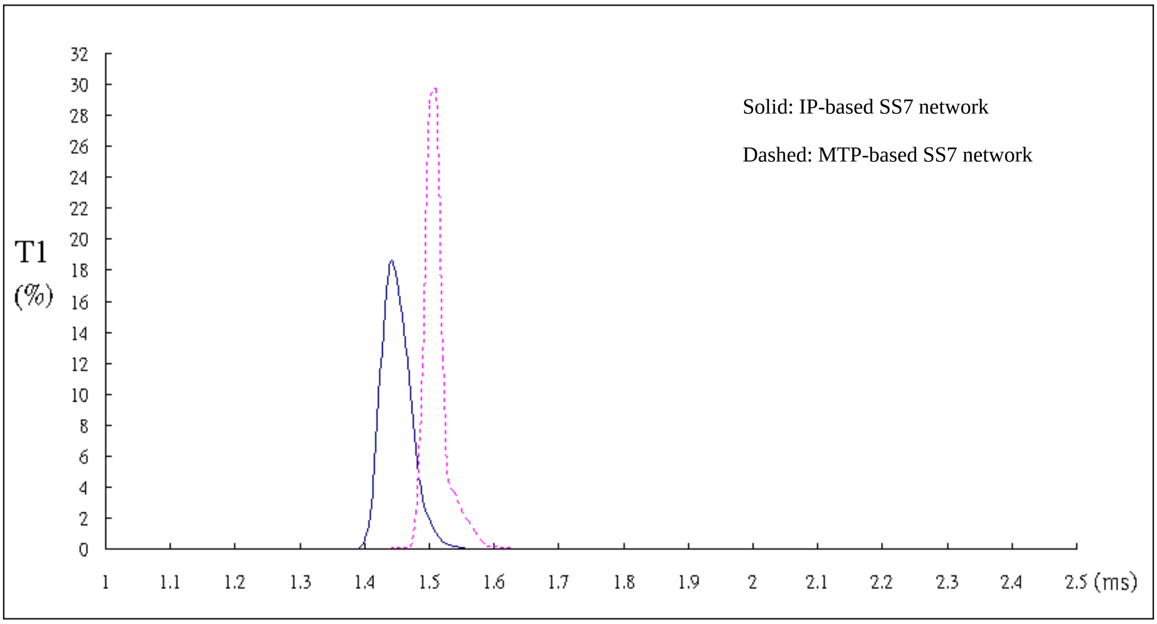
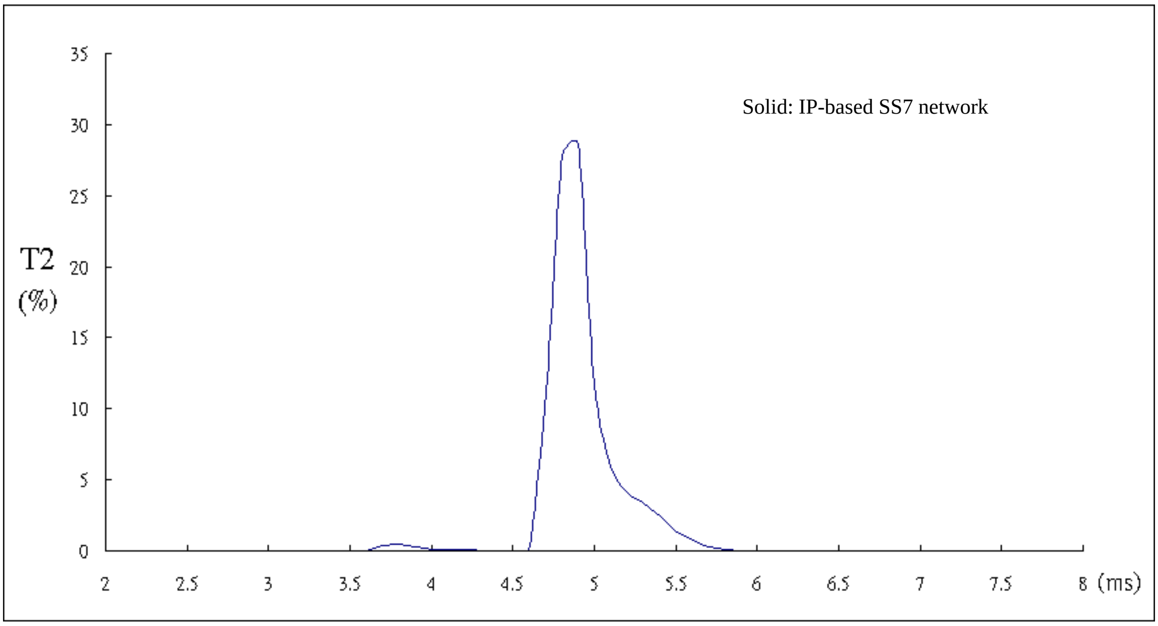
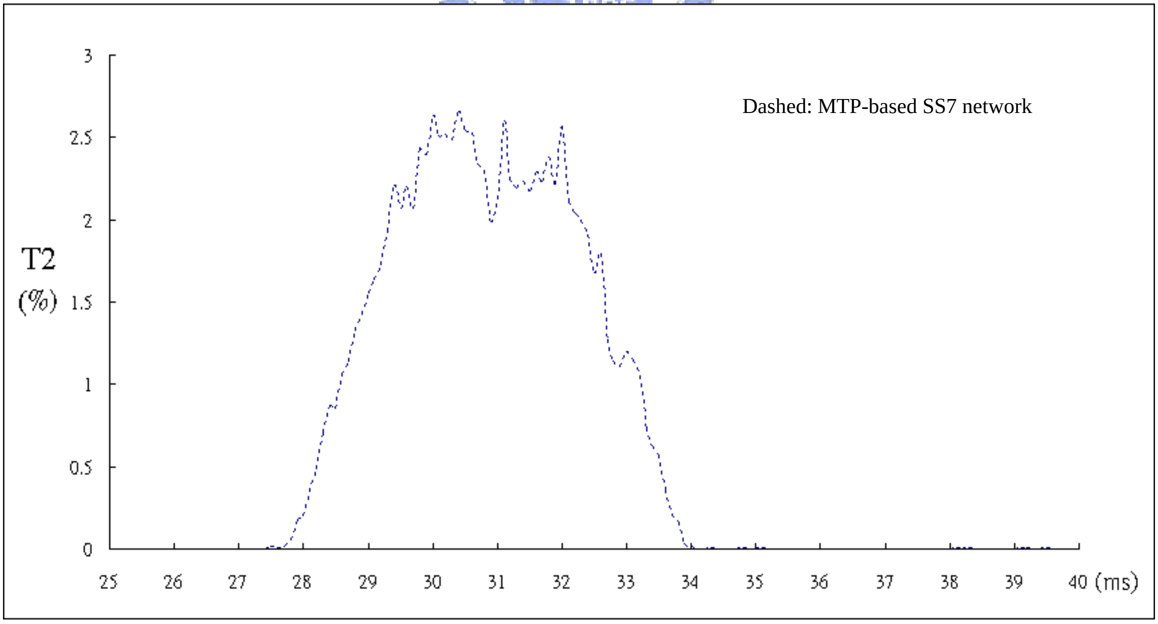


Figure 3.4.5: The T1 Histograms



(a) IP-based SS7 Network



(b) MTP-based SS7 Network

Figure 3.4.6: The T2 Histograms

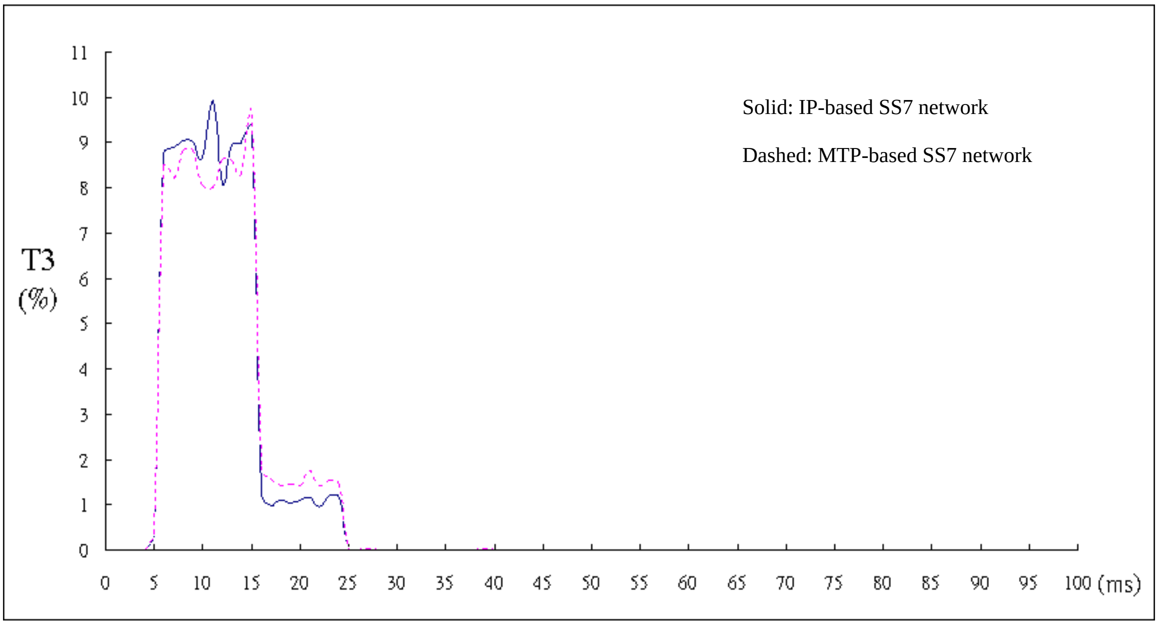


Figure 3.4.7: The T3 Histograms

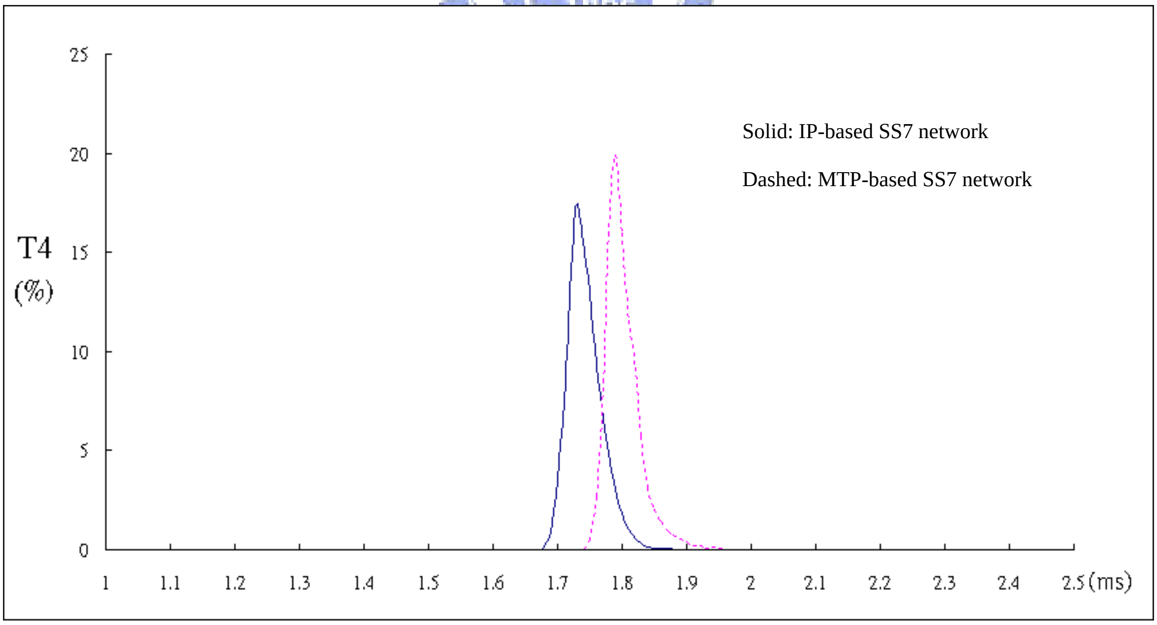
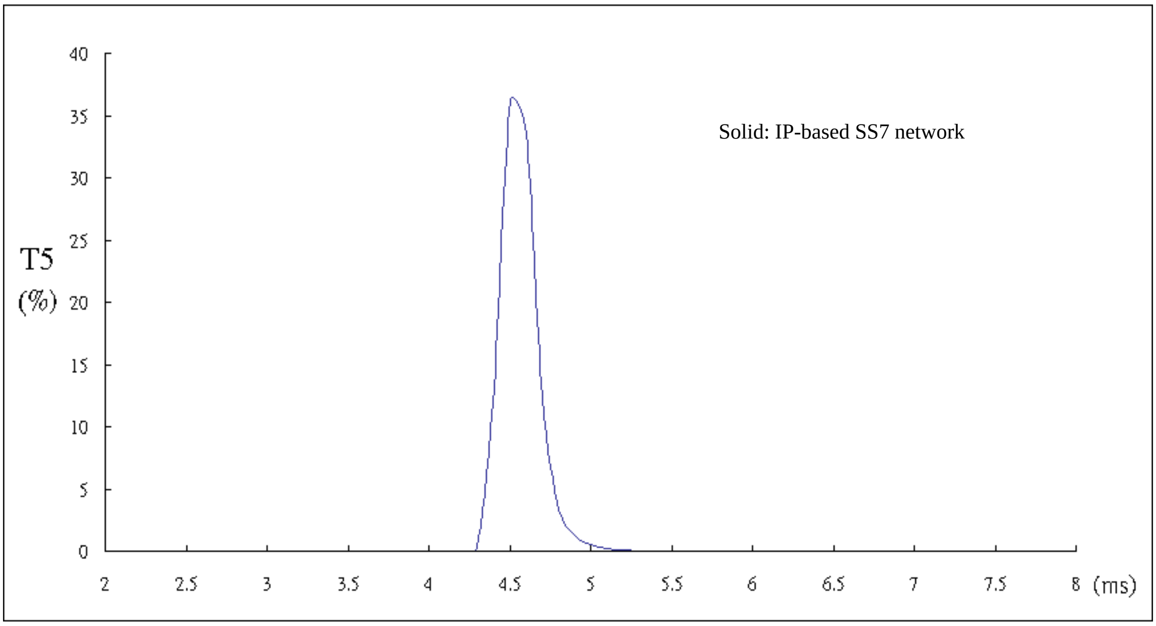
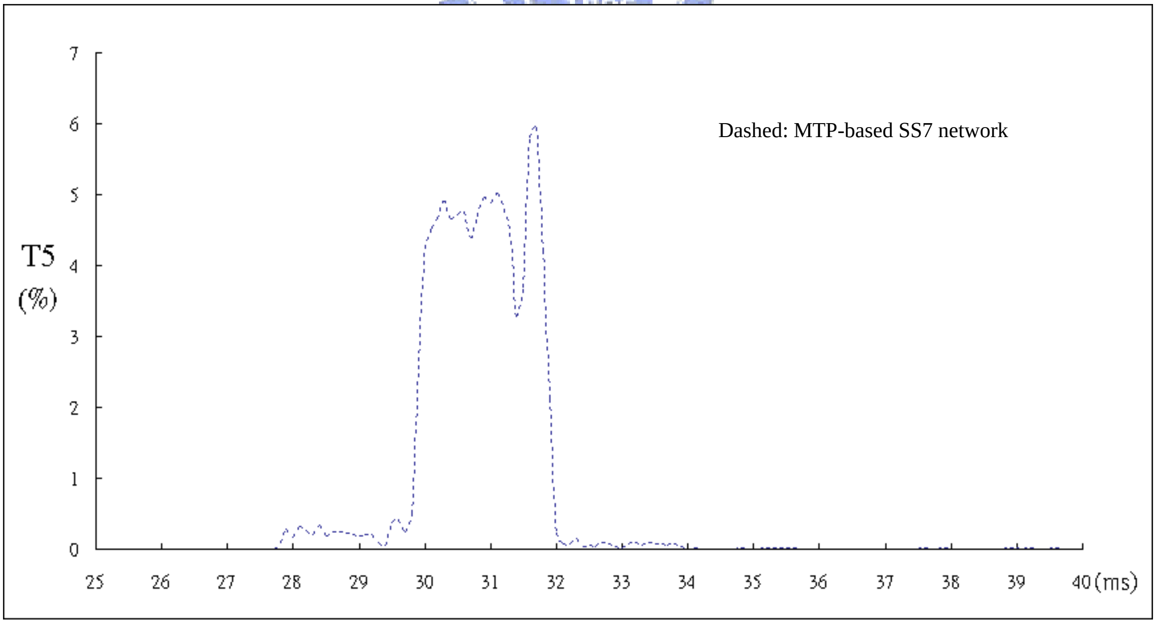


Figure 3.4.8: The T4 Histograms



(a) IP-based SS7 Network



(b) MTP-based SS7 Network

Figure 3.4.9: The T5 Histograms

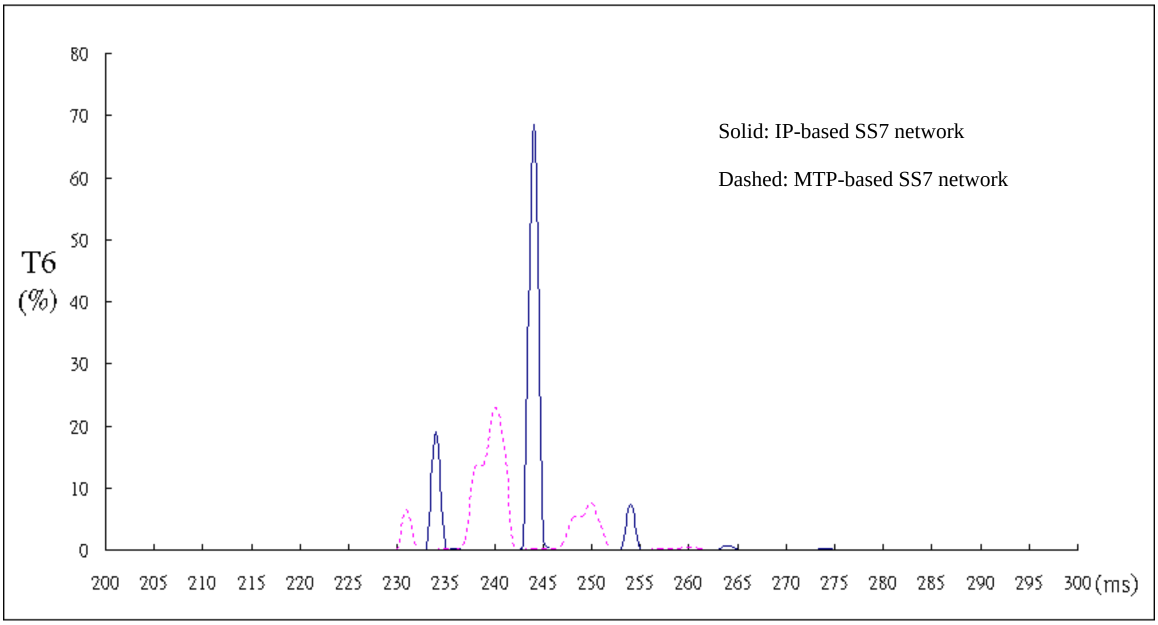


Figure 3.4.10: The T6 Histograms

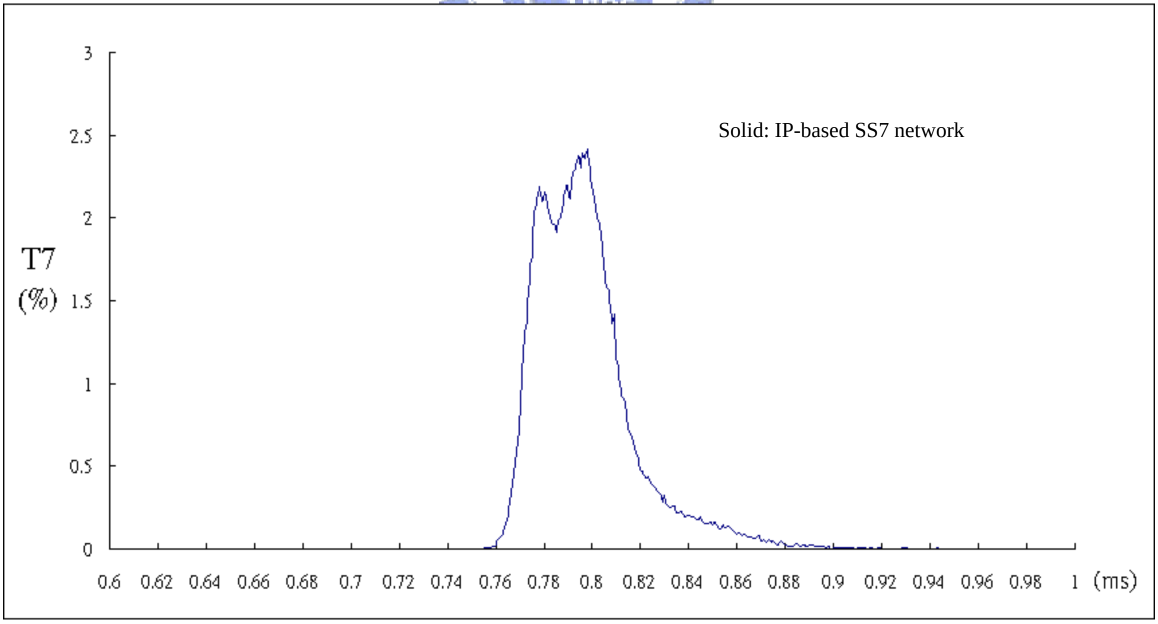


Figure 3.4.11: The T7 Histogram

approach, the T2 statistic has the mean 4.855 ms, and the variance 0.558509 ms^2. For the MTP-based approach, the T2 statistic has the mean 30.731 ms, and the variance 1.853327 ms^2.

Analysis of the elapsed time T3 of the HLR application overhead and the HLR MAP overhead for the MAP_OPEN service (see Figures 3.4.3 (a) and 3.4.4 (a)): As described in the analysis for T1, different implementations of communication between the TCAP layer and the MAP layer also affect T3. However, the effect is insignificant as compared with the total processing time of the MAP layer and the execution of the HLR application. Figure 3.4.7 shows that the histogram of T3 over IP-based SS7 network is similar to that for MTP-based SS7 network. For the SCTP-based approach, the T3 statistic has the mean 10.463 ms, and the variance 17.60297 ms^2. For the MTP-based approach, the T3 statistic has the mean 10.854 ms, and the variance 20.53695 ms^2.

Analysis of the elapsed time T4 of the SGSN application overhead and the SGSN MAP overhead for the MAP_SEND_AUTHENTICATION_INFO service (see Figures 3.4.3 (b) and 3.4.4 (b)): This analysis is similar to that for T1. For the SCTP-based approach, the T4 statistic has the mean 1.738 ms, and the variance 0.290116 ms^2. For the MTP-based approach, the T4 statistic has the mean 1.793 ms, and the variance 0.003693 ms^2.

Analysis of the TCAP and the lower layer overheads T5 for the MAP_SEND_AUTHENTICATION_INFO service (see Figures 3.4.3 (b) and 3.4.4 (b)): This analysis is similar to that for T2. For the SCTP-based approach, the T5 statistic has the mean 4.478 ms, and the variance 0.246326 ms^2. For the MTP-based approach, the T5 statistic has the mean 30.752 ms, and the variance 1.06369 ms^2.

Analysis of the elapsed time T6 of the executions for the HLR application and MAP for the MAP_SEND_AUTHENTICATION_INFO service (see Figures 3.4.3 (b) and

3.4.4 (b)): Figure 3.4.10 shows several peaks in the histograms of the elapsed times T6 for both approaches. This phenomenon is explained as follows. To simultaneously handle multiple dialogues, the HLR application is implemented using multiple programming technique, where the Round-Robin algorithm is used for CPU scheduling at the HLR. Depending on the number of outstanding dialogues seen at the HLR, the execution time of a dialogue will fall in one of the “peaks” in Figure 3.4.10. Therefore, the histogram of the delay time is represented by these peaks that are separated by about 10 ms. In Figure 3.4.10, we also see curve shift of the T6 histograms for the SCTP-based approach and the MTP-based approach. The discrepancy is negligible (about 1.7%). For the SCTP-based approach, the T6 statistic has the mean 241.986 ms, and the variance 32.29521 ms^2. For the MTP-based approach, the T6 statistic has the mean 240.646 ms, and the variance 45.27385 ms^2.

Analysis of the IP overhead T7 for both services (see Figures 3.4.3 (a) and 3.4.3 (b)) for the SCTP-based approach: Figure 3.4.11 shows that the measured elapsed time T7 is less than 1 ms, which contributes insignificantly to the total elapsed time of the MAP service. The T7 statistic has the mean 0.803 ms, and the variance 0.070963 ms^2.

Based on the measured statistics, the most significant performance difference between the two mechanisms of our implementations is the TCAP and the lower layer overheads including transmission time (see Figures 3.4.6 and 3.4.9).

Chapter 4 Conclusions

The SS7 signaling is used to provide control and management functions in the mobile network. This thesis presented the implementations of the two approaches for SS7 signaling transport; namely, the MTP-based approach and the SCTP-based approach. We described our implementations, and compared of these two approaches in three perspectives: message format, connection setup, and data transmission/ack. We used the Send Authentication Info procedure defined in 3GPP Technical Specification 23.060 [1] as an example to illustrate the performance of the SCTP-based and the MTP-based MAP approaches.

References

[1] 3GPP TS 23.060, v 4.7.0 (2002) Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service Description; Stage2. 3rd Generation Partnership Project.

[2] 3GPP TS 09.02, v. 7.14.0 (2003) Technical Specification Group Core Network; Mobile Application Part (MAP) Specification. 3rd Generation Partnership Project.

[3] Lin, Y.-B., and Chlamtac, I. (2001) Wireless and Mobile Network Architectures. Wiley, Location.

[4] Lin, Y.-B. (1999) Telephone Network and PBX Software. Wei-Keg Book Co., Ltd.

[5] Russell, Travis (2002) Signaling System #7. McGraw-Hill.

[6] ITU-T Recommendations Q.701-Q.704, Q.706, Q.707. Message Transfer Part of SS7.

[7] ITU-T Recommendations Q.711-Q.714, Q.716. Signaling Connection Control Part of SS7.

[8] ITU-T Recommendations Q.771-Q.775. Transaction Capabilities Application Part of SS7.

[9] ITU-T Recommendations Q.761-Q.764, Q.766. ISDN User Part of SS7.

[10] ITU-T Recommendations Q.750-Q.755. Operations Maintenance and Administration Part of SS7.

[11] 3GPP TS 03.03, v 7.8.0 (2003) Technical Specification Group Core Network; Numbering, addressing and identification; 3rd Generation Partnership Project.

[12] Trillium digital Systems Inc., http://www.trillium.com/.

[13] NewNet Connect7^{TM} User Manual, Part No. D-0116-US-214-000, ADC Telecommunication, Inc. (2000)

[14] IETF RFC 2960 (2000) SCTP: Stream Control Transmission Protocol. Internet

Engineering Task Force.

[15] IETF RFC 3332 (2002) M3UA: MTP3 User Adaptation Layer. Internet Engineering Task Force.

[16] IETF RFC 793 (1981) TCP: Transmission Control Protocol. Internet Engineering Task Force.

[17] ITU-T Recommendations Q.773. Transaction Capabilities Formats and Encoding.

[18] 3GPP TS 23.002 v 5.3.0 (2001) Specification Group Services and Systems Aspects; network Architecture. 3rd Generation Partnership Project.

[19] Bos, L., and Leroy, S. (2001) Toward an All-IP-based MTS System Architecture. IEEE Networks, vol. 15, no. 1, pp. 36-45.

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