Hannu Johansson

AVOIMEN LÄHDEKOODIN INTEGROINTI MOBIILIVERKKOON
OPEN SOURCE INTEGRATION TO MOBILE NETWORK

Tietotekniikan koulutusohjelma
Degree Programme in Information Technology
2013

samk
Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

AVOIMEN LÄHDEKOODIN INTEGROINTI MOBIILIVERKKOON

Johansson, Hannu
Satakunnan ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Kesäkuu 2013
Ohjaaja: Aromaa, Juha, DI
Sivumäärä: 33
Liitteitä: 3
URN:NBN:fi:amk-2013052210011

Asiasanat: Linux, avoin lähdekoodi, lyhytsanomakeskus, mobiiliverkko

---

Työn aiheena oli toteuttaa avoimen lähdekoodin kehitystyökaluilla ja kirjastoilla mobiiliverkkoon integroituva sovellus. Sovellukseksi valittiin lyhytsanomakeskus (SMSC), jolla voitaisiin todentaa eri protokollakerrosten toimivuus.

Toteutusta varten selvitettiin mobiiliverkon määrittelyistä tarvittavien protokollien ja palveluiden toiminnallisuudet sekä valittiin kehitykseen tarvittavat avoimen lähdekoodin työkalut.

Sovellus testattiin aidossa mobiiliverkossa, jonka perusteella todettiin, että käytetyt protokollatasot ja määrittelyjen toiminnallisuus voidaan toteuttaa saatavilla olevien avoimen lähdekoodin työkaluilla ja kirjastoilla.

OPEN SOURCE INTEGRATION TO MOBILE NETWORK

Johansson, Hannu
Satakunta University of Applied Sciences
Degree Programme in Information Technology
June 2013
Supervisor: Aromaa, Juha, M.Sc.
Number of pages: 33
Appendices: 3
URN:NBN:fi:amk-2013052210011

Keywords: Linux, open source, short message service center, mobile network

---

The subject of this thesis was to implement application integrating to the mobile network by using open source development tools and libraries. Chosen application was Short Message Service Center (SMSC) to prove that all related protocol layers can be implemented.

Related specifications and service descriptions were investigated and needed open source software evaluated for the implementation.

Application was tested in a real mobile network. As a result of testing the conclusion was that protocol layers and service functionalities can be fully implemented using open source software.

TABLE OF CONTENTS

ABBREVIATIONS

AS	Application Server
ASN1	Abstract Syntax Notation number 1
ASP	Application Server Process
ETSI	European Telecommunications Standards Institute
GSM	Global System for Mobile
GT	Global Title
HLR	Home Location Register
IETF	Internet Engineering Task Force
IMSI	International Mobile Subscription Identity
IP	Internet Protocol
LK	Linux kernel
LTE	Long-Term Evolution
MAP	Mobile Application Part
MSC	Mobile Switching Center
MTP	Message Transfer Part
OSI	Open System Interconnection model
OSS	Open Source Software
PC	Point Code
RFC	Request For Comments
SCCP	Signalling Connection Control Part
SCTP	Stream Control Transmission Protocol
SG	Signalling Gateway
SIGTRAN	Framework architecture for Signalling Transport
SMSC	Short Message Service Center
SUA	SCCP User Adaptation Layer
TCAP	Transmission Capabilities Application Part
TCP	Transmission Control Protocol
UMTS	Universal Mobile Telecommunications System
VLR	Visitor Location Register

1 INTRODUCTION

From the beginning of the second mobile network generation (GSM) to the latest 3rd (UMTS) and 4rd generations (LTE) there have been always open and public standards and specifications by the standard organizations. Public specifications mainly cover all functionalities from the mobile device to the individual network elements. There is some implementation details left to the hardware vendor, but for example all interworking functionalities and protocol layers are openly standardized. Therefore network elements are working together with each other independently of the device's actual manufacturer.

During the decades mobile industry related hardware and software have been traditionally commercial and therefore very expensive. In the software industry there has been rapid change from commercial software business models towards open source models. Telecom hardware vendors still keep many of their tools and internal APIs (Application Programming Interface) private. As stated all the mobile standards are public, so the motivation and scope of this thesis work was to make proof of concept mobile network product using fully open source software components and tools. The chosen functionality is SMSC (Short Message Service Center) with basic features: receive and deliver message from one mobile user to another. Goal is to show that all protocol layers can be implemented by freely available tools and product can be integrated to the real network according to the specifications.

2 RELATED SPECIFICATIONS

This chapter gives overview of the related specifications in this thesis. Specifications are divided in to the two domains. Mobile network specifications (chapter 2.1) are mainly circuit-switched signaling in lower OSI level layers. Second domain is Internet standards (chapter 2.2) which represent traditionally packet based protocols like IP and protocols carried mostly over the IP.

2.1 Mobile network specifications

Many of the basic international standards of the fixed land line and mobile network are provided by International Telecommunications Union (ITU-T). European family of the mobile network standards consists mainly of the three organization group per each generation. After the first analog generation, second generation specification group was GSM (GSM) eventually maintained by European Telecommunications Standards Institute (ETSI). ETSI is also member of currently most active group 3GPP (3GPP) which produced 3G and latest 4G level specifications. (Korhonen 2003, pages 1-8)

2.1.1 Mobile network architecture

This chapter gives a short overview of the typical mobile network elements that are relevant for this thesis. According to the mobile network specifications Figure 1 presents the basic configuration of the network.

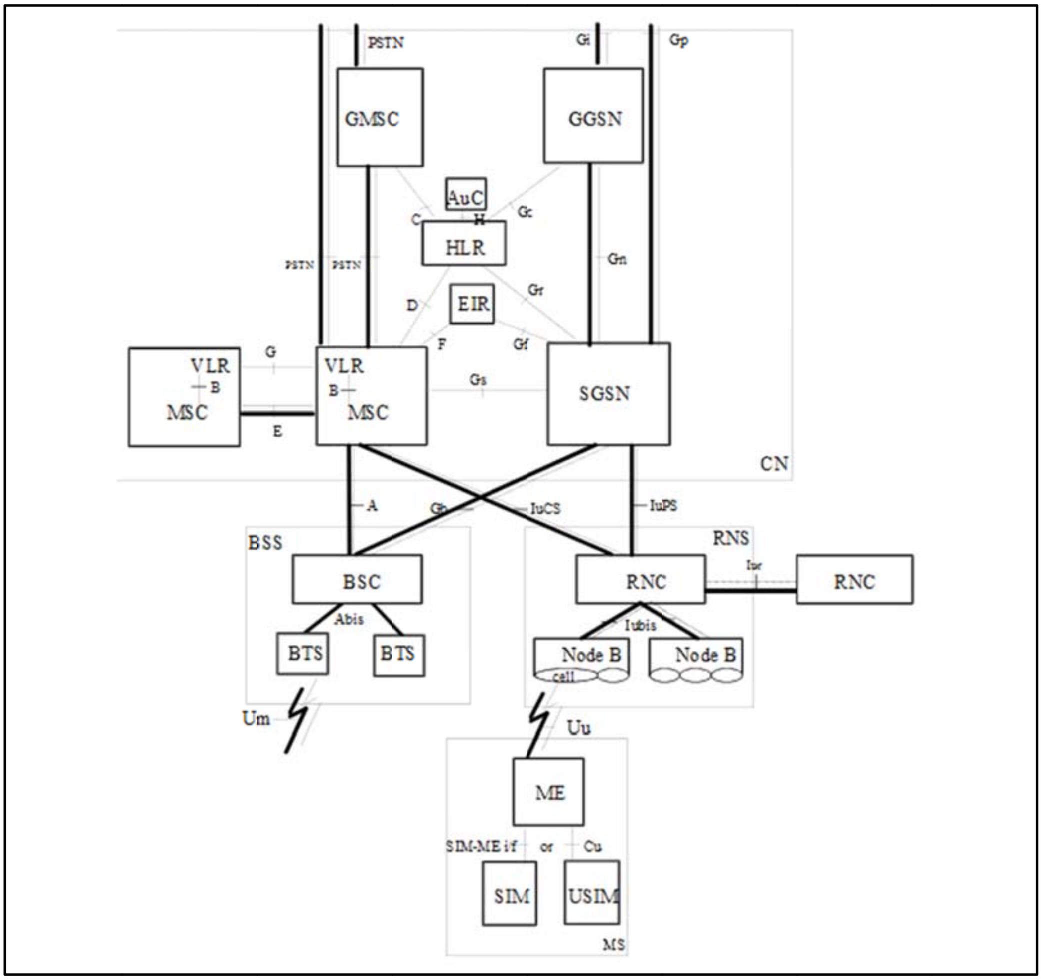


Figure 1 The basic configuration of mobile network (3GPP TS 23.002, page 22)

Mobile network is divided into Core network and Access network. Above in Figure 1 there are three boxes that represent the access network. Mobile Station (MS) which represents for example the user's mobile phone. It is drawn to be connected to the Radio Network System (RNS) via NodeB which is 3G UMTS radio part. In the left of the RNS rectangle, there is Base Station System (BSS) rectangle which represents the 2G GSM radio interface part.

Core Network is the bigger rectangle in the upper side of Figure 1. It has all network elements that handle most of the functionalities:

- Home Location Register (HLR), central database of the network which has defined available for services for the subscribers (for example, grant access to use Short Message Services)

- Visitor Location Register (VLR) and Mobile Switching Center (MSC) communicate with the radio access network part and handle the mobile station attaches and detaches to the network. Also, for example, short messages are transferred via the MSC/VLR to the subscriber.

All other interfaces and elements are fully described in specification (3GPP TS 23.002).

2.1.2 Mobile network protocol layers

From second to third generation, lower level of the mobile network specifications are mainly based on the circuit-switched protocols which fulfill the OSI model's first layers. Mobile network protocols are mapped to the OSI (Open System Interconnection model) model by the specification as shown in Figure 2. (ITU-T Q.1400)

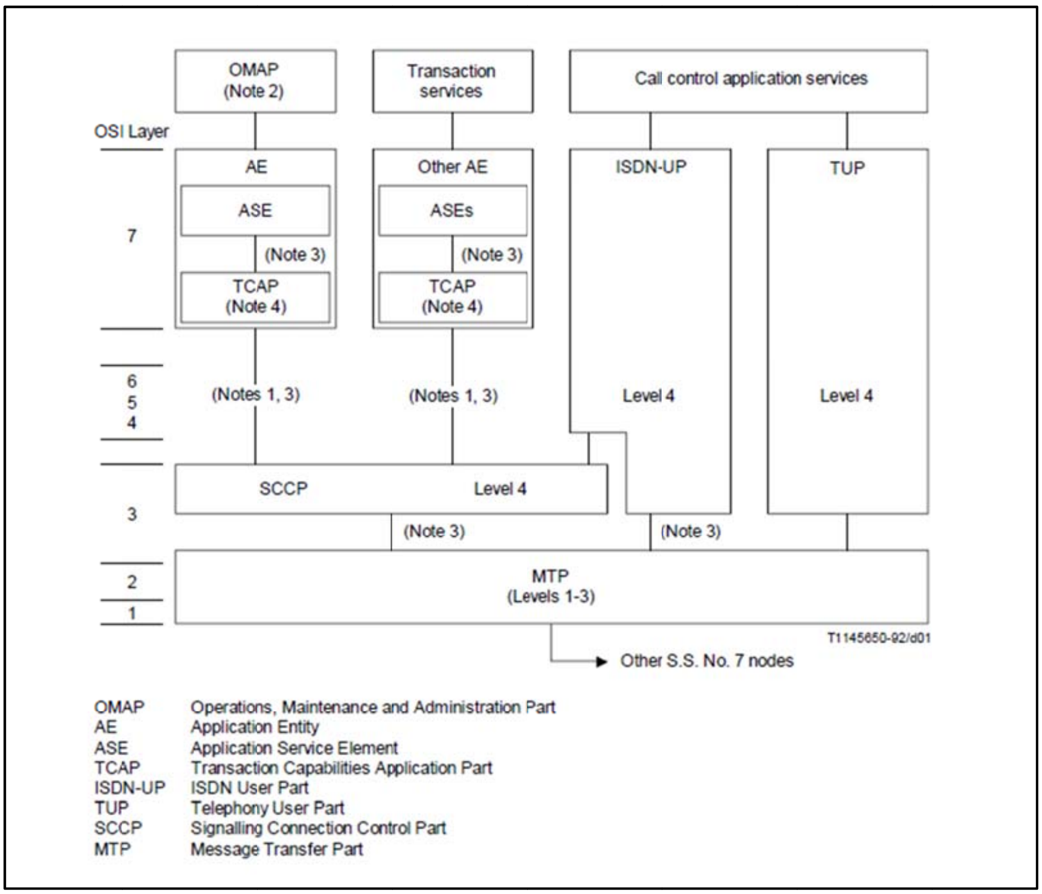


Figure 2 OSI and ITU-T protocol mapping (ITU-T Q.1400, page 5, figure 1)

2.1.3 Message Transfer Part layers 1-3 (MTP)

First two of the OSI model layers are implemented by MTP1-3 (Message Transfer Protocol) as shown in Figure 2.

MTP1 is the lowest level implementation that provides OSI models Data Link level. Specification generalizes MTP1 as a bidirectional transmission path for signaling: comprising two data channels operating together in opposite directions at the same data rate. It constitutes the lowest functional level (level 1) in the Signalling System No. 7 functional hierarchy. (ITU-T Q.702, 1.1 General, page 1)

MTP2 is the link level protocol that provides communication between two adjacent nodes which are called signaling points. Specification describes the functions and procedures for and relating to the transfer of signalling messages over one signalling data link. The signalling link functions, together with a signalling data link as bearer, provide a signalling link for reliable transfer of signalling messages between two directly connected signalling points (ITU-T Q.703, 1.1 Introduction, page 1).

MTP3 is the network level protocol that provides communication between two nodes over the one or more nodes, thus providing message routing and signalling status functionalities between two end-points. Specification assumes that MTP3 is carried over MTP1 and MTP2 layers. Every network node carrying MTP3, also known as signalling point, has a unique address called PC (Point Code). MTP3 messages have Originating and Destination Point Code which are relatively source and destination addresses of the message. (ITU-T Q.704, 1.1 General characteristics of the signalling network functions, page 1).

2.1.4 Signalling Connection Control Part (SCCP)

MTP3 and SCCP together are covering OSI model layer 3 as shown in Figure 2. As stated before, MTP3 is providing end-to-end message transferring. SCCP provides also the flow-control, segmentation, connectionless and connection oriented services and more error correction services.

Signalling Connection Control Part (SCCP) provides five network services, where only first Class 0 is concerned in this work. It is called Basic connectionless class which provides seamless data stream providing segmentation services independent of lower layer MTP3 transfer unit sizes.

SCCP provides also addressing functionalities over the MTP3 point codes. Addressing and routing scheme can consist of GT (Global Title) number which can be translated to the MTP3 Point Code depending of the Global Title translation rules. Global Title numbers are formatted according to the same way that usual telephone numbers and are therefore internationally unique. Using GT numbers, SCCP messages can be pointed to the different elements over MTP3 network. It allows upper level protocols to send messages directly to the other carrier's network nodes or directly to subscribers without need to know underlying MTP3 point code addresses. In addition to Point Code or Global Title, there is also SSN (Subsystem Number) which distinguishes different services inside the same Global Title.

SCCP functionality is fully defined in (ITU-T Q.714).

2.1.5 Transaction Capabilities Application Part (TCAP)

Transaction Capabilities Application Part (TCAP) functional specification generalizes protocol to provide functions and protocols to a large variety of applications distributed over switches and specialized centres in telecommunication networks (e.g. databases) (ITU-T Q.771, 1.1 General, page 1).

From second to third generation networks TCAP is usually carried over SCCP in circuit-switched network. It is end-to-end protocol that provides unique transactions between two end-points, thus supporting many conversations between same end-points at the same time. Transactions are distinguished by transaction id which makes it possible to have several dialogues open between same network nodes.

Each TCAP message has three parts (ITU Q.772, 1 General):

- Transaction portion has two main parameters: Message type, which defines whether the dialogue is initiated by the begin primitive, continued with continue primitive or to be release by abort or end primitive (ITU-T Q.772, 2.1 Message Type, page 1). Second parameter are Originating and Destination Transaction id pair which are identifiers that makes the dialogue unique between the two end-points (ITU-T Q.772, 2.2 Transaction IDs, page 1)
- Dialogue portion includes dialogue request or response part (ITU-T Q.772, 4. Dialogue portion, page 7) which is protocol data unit definition that describes the actual dialogue as being one of the following types: request, response or abort.
- Component portion (ITU-T Q.772, 3 Component portion, page 3) can include components filled by user data that are invoked (sent to the remote peer) and result returned (response received from the remote). User data refers to other protocols that are using TCAP as a transaction layer.

TCAP is fully defined in specifications (ITU-T Q.771 - Q.775)

2.1.6 Mobile Application Part (MAP)

Mobile Application Part (MAP) it is the major and most relevant protocol used inside mobile network. It uses TCAP and SCCP services for communicating between two nodes. There are several operations defined which are used between different network nodes in mobile network. It is a protocol used inside the carrier's own network and also interworking protocol between the different operators when subscriber is roaming in visitor network.

Examples of mobile network procedure areas that rely on MAP protocol:

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

MAP specification defines in general own service user and provider layers and primitives for opening and closing the dialogue between the remote end, but those are mapped to corresponding TCAP component types (3GPP TS 29.002, chapter 6).

Only the actual MAP operation consists of the operation arguments and response which are transferred in TCAP component's user data. The MAP operation is usually defined to be having at least some mandatory request parameters and one or more optional return parameters depending of the service. All services and protocol details are fully specified in (3GPP TS 29.002).

2.2 Internet standards

In the packet based domain, mostly known by the Internet related applications, there is IETF (Internet Engineering Task Force) organization which is group of technical people that share mission to "*to make the Internet work better*" (IETF, homepage / About the IETF).

IETF's widely known set of technical documents are RFC (Request For Comments) document sets which define all major protocols used in Internet and other IP based networks. They have been defined Internet Protocol, Hyper Text Transfer Protocol and much more protocols used every day by public users especially needed in the web surfing. Below are introduced specifications that were relevant part of this thesis work.

First protocol described is IP (Internet Protocol) in next chapter. Link layer protocols carrying IP are omitted since there is wide variety of protocols and they are not relevant for this thesis.

2.2.1 Internet Protocol (IP)

Internet Protocol is widely known and used network level protocol in Internet and intra-networks. IP provides network level connectivity between two end-points. End-points are assigned with 32-bit address which usually is denoted in dotted decimal format, for example 192.168.0.1.

Being network level protocol, IP provides routing functionalities over devices acting as a router or gateway between IP networks. IP is connectionless protocol providing only fragmentation and reassembly services depending of the underlying link-level protocol transfer capabilities. It does not guarantee any sequenced order for receiving packets nor error correction in the receiving side.

Internet Protocol has header of 14 fields, where 13 are mandatory and one is optional (RFC 791, chapter 3.1.). Header includes, for example:

- Source and destination IP address (2 x 32 bits)
- Protocol (1 octet) defines what protocol is carried inside the IP data portion

In addition to the header, IP has the data portion which capsulate actual data of the carried protocol. Internet Protocol version 4 is fully specified in (RFC 791).

2.2.2 Framework for signaling transmission (SIGTRAN)

SIGTRAN states for Framework for signaling transmission. The framework describes relationships between functional and physical entities exchanging information, such as Signalling Gateways and Media Gateway Controllers. It identifies interfaces where signaling transport may be used and the functional and performance requirements that apply from existing Switched Circuit Network signaling protocols (RFC 2719).

In brief, this framework is umbrella for specifications what are needed to provide IP based connectivity between the Mobile network specifications described in 2.1 and software running over the IP based stack. Framework specification itself does not introduce any physical protocols or data elements, only logical concepts and implementation functionalities and points to the other recommendations of the actual protocols.

Relevant to this thesis work, SIGTRAN framework introduces the Signalling Gateway (SG) and Media Gateway functions which are for example logical nodes that combine mobile network protocols and IP based protocols (RFC 2719, chapter 2.1 Gateway component functions). In this work, we will focus only to the Signalling Gateway which will combine Mobile network specifications and IP based protocols together.

SIGTRAN framework is fully defined in RFC 2719.

2.2.3 Stream Control Transmission Protocol (SCTP)

SCTP (Stream Control Transmission Protocol) is designed to transport Public Switched Telephone Network (PSTN) signaling messages over IP networks, but is capable of broader applications (RFC 4960, Abstract).

SCTP is transport layer (OSI) protocol serving in similar, but more advanced ways over IP like known Transmission Control Protocol (TCP) (RFC 793) and Universal Datagram Protocol (UDP) (RFC 768) protocols.

As mentioned earlier, TCP and UDP were not chosen to carry telephone and mobile network protocols over IP. Recommendation lists four reasons why TCP was not enough extensible to fulfill requirements for example transferring telephony signalling protocols (RFC 4960, 1.1 Motivation), for example:

- TCP provides both reliable data transfer and strict order-of-transmission delivery of data. Some applications need reliable transfer without sequence maintenance, while others would be satisfied with partial ordering of the data. In both of these cases, the head-of-line blocking offered by TCP causes unnecessary delay.

- The stream-oriented nature of TCP is often an inconvenience. Applications must add their own record marking to delineate their messages, and must make explicit use of the push facility to ensure that a complete message is transferred in a reasonable time.
- The limited scope of TCP sockets complicates the task of providing highly-available data transfer capability using multi-homed hosts.
- TCP is relatively vulnerable to denial-of-service attacks, such as SYN attacks

As stated in chapter 2.2.1 the IP protocol is connectionless protocol, therefore SCTP provides reliable way to transfer packets between two end points. SCTP is concerned to be message-oriented rather than stream based like TCP. Messages are group of bytes which are sent in different streams. There can be one or more streams in one connection. SCTP user may also decide whether messages in each stream must be ordered or not.

SCTP packet format consists of the Common Header and one or more data Chunks (RFC 4960, 3. SCTP Packet format):

- Common Header includes 16-bit source and destination port same way like TCP protocol. Port numbers distinguishes between different SCTP connections.
- Chunk Field has following fields: Type, Flags and Length that describes the data carried in Chunk Value field. Mostly used chunk types with descriptions are:
    - Initiation (INIT) and acknowledgement (INIT ACK): Used to negotiate connection and creates the SCTP association between two end-points. End-points agree with association specific parameters, for example amount of streams (RFC 4960, chapter 3.3.2).
    - Heartbeat (HEARTBEAT) and acknowledgement (HEARTBEAT ACK): Probe between end-points to keep connection state and path active (RFC 4960, chapters 3.3.5-3.3.6, 8.3)
    - Payload data (DATA), Selective Acknowledgement (SACK): Transports the SCTP user’s data in specified stream and keeps the ordering

using sequence numbers (RFC 4960, chapter 3.3.1). DATA chunks are acknowledged using SACK chunk (RFC 4960, chapter 3.3.4).

- Abort (ABORT) Shutdown (SHUTDOWN) and acknowledgement (SHUTDOWN ACK): Releases the SCTP connection and frees the resources (RFC 4960, 3.3.8)

Addressing also supports multi-homing which allows adding more than one receiver transport address for the end point. In case of error situations, different transport addresses can be taken into use for making new path for the association.

SCTP is introduced in RFC 3286 and functionally described in RFC 4960. There exists also several other RFC recommendations for SCTP extensions, but those are not relevant for this work.

2.2.4 SCCP User Adaptation layer (SUA)

SUA (Signalling connection control part User Adaptation layer) is a protocol for the transport of any SCCP -User (described in chapter 2.1.4) signalling over IP using the Stream Control Transmission Protocol (described in chapter 2.2.3).

According to the recommendation SUA is designed to be modular and symmetric, to allow it to work in diverse architectures, such as a Signalling Gateway to IP Signalling Endpoint architecture as well as a peer-to-peer IP Signalling Endpoint architecture (RFC 3868, Abstract). In this chapter we describe how the SUA is used in Signalling Gateway configuration, it could be also used directly between the IP based protocol stack hosts.

In chapter 2.1 the mobile network specifications protocol stack was introduced from the MTP1 level to upper MAP protocol. Figure 3 shows the Signalling Gateway functionality where mobile network protocol stack is shown in left side. SUAP states for SCCP User Protocol, which represents for example MAP over TCAP. In the right side, there is Internet Protocol based protocol stack described in this chapter.

Signalling Gateway (SG) functionality in the middle converts signalling between mobile network and IP based stack.

```
********   SS7   ***************   IP   ********
* SEP  *---------*             *--------*      *
*  or  *         *     SG      *        * ASP  *
* STP  *         *             *        *      *
********         ***************        ********

+------+                                +------+
| SUAP |                                | SUAP |
+------+         +------+------+        +------+
| SCCP |         | SCCP | SUA  |        | SUA  |
+------+         +------+------+        +------+
| MTP3 |         | MTP3 |      |        |      |
+------+         +------+ SCTP |        | SCTP |
| MTP2 |         | MTP2 |      |        |      |
+------+         +------+------+        +------+
|  L1  |         |  L1  |  IP  |        |  IP  |
+------+         +------+------+        +------+
    |                |      |               |
    +----------------+      +---------------+

  SUAP - SCCP/SUA User Protocol (TCAP, for example)
  STP  - SS7 Signalling Transfer Point
```

Figure 3 Signalling Gateway combines mobile and IP based protocol stacks (RFC 3868, chapter 1.3.1)

SUA specification introduces two logical entities, Application Server (AS) and Application Server Process (ASP). Application Server represents the logical unit that has service functionalities, for example Short Message Center processes. Application Server Process is the logical process inside the AS which represents the actual process where the SCTP connection is physically established. Routing key and context is the set of signalling parameters that associates given ASP to the service at specified address.

SUA message consists of common header, which has message class, type and data with length fields. For this thesis, relevant message classes and class specific messages are introduces in the Table 1. Messages are described in role between the Active Server and Signalling Gateway.

Table 1 SUA protocol message classes, types and descriptions

Class	Type(s)	Description
Application Server Process State Maintenance	Asp Up (UP) Asp Up Ack (UP ACK) Asp Down (DOWN) Asp Down Ack (DOWN ACK)	AS notifies the SG that specific AS is in the up or down state. AS is identified by the routing key and context. Messages are acknowledged by respectively ACK messages.
ASP Traffic Maintenance	Asp Active (ACTIVE), Asp Active Ack (ACTIVE ACK)	ASP notifies the SG that specific AS is in the up or down state for the traffic. Messages are acknowledged by respectively ACK messages.
Connectionless Messages	Connectionless Data Transfer (CLDT)	Connectionless upper layer protocol data (for example TCAP) is sent using this message.
Signalling Network Management	Destination Unavailable (DUNA) Destination available (DAVA)	DUNA is received by AS when SG notifies that SCCP User or lower layers have become unreachable. Respectively DAVA is sent when state changes back to reachable.

SUA protocol is fully defined in recommendation (RFC 3868).

3 DEVELOPMENT TOOLS

The tools used in this thesis were chosen to be found from the free Linux distributions commonly used. Only major requirement was that no commercial tools or libraries should be chosen.

3.1 Linux distribution

Debian Linux 6.2 was used as the development and server environment (DEBIAN). Debian Linux has many of the installation modes, but basic development tools can be installed during the operating system installation or afterwards via package management tools.

3.2 C/C++ development environment

Debian Linux development environment included widely known GNU C/C++ compiler which was used to compile and link the C/C++ code. GCC/G++ -version was 4.6.3.

3.3 ASN.1 compiler and libraries

Mobile network protocols defined in 2.1 are usually described using ASN.1 (Abstract syntax notation number 1) which is included in appendices of the related specification. ASN.1 describes the protocol messages in abstract level without information how the actual bit-level construction is done. Specification defines that ASN.1 described data in mobile network is transferred using Basic Encoding Rules (BER) which gives the concrete representation of the different data types (ASN1DEF).

Evaluation of ASN.1 compilers and run-time environment was difficult since almost all products found were commercial. One free open source based project was found and tested to be suitable for development. Lew Walkin has developed Asn1c named encoder and decoder compiler project (ASN1C) which supports Distinguished Encoding Rules (DER). The BER rules are subset of the DER rules, thus it was compatible for generating data structures for the mobile network development.

The compiler generates the header and source files based for the corresponding ASN.1 definition file, more specific, encode-decode methods are created per each data type. Compiler created source and headers are linked into application and can be then called in runtime. To create BER data, one must fill compile created structure by own user data, call the encode method which returns the user data as encoded BER byte stream. Accordingly received BER data, for example from the network, can be given to decode function which fills the compiler created data structure with decoded values.

3.4 Testing tools

CppUnit testing framework (CPPUNIT) was used to make simple unit tests per each protocol class. Unit testing is sophisticated testing method for each protocol message and corresponding message handlers. For every message creation method there shall be unit test which verifies that the given message parameters are correctly constructed in bit-level.

4 APPLICATION DEVELOPMENT

Main goal of this thesis is to prove that every layer of the protocol stack can be implemented by the free tools. This chapter describes design principles and details of each protocol layer implementation for the proof-of-concept SMSC.

4.1 Functional requirements

Fully functional and production mature SMSC would include a lot of features, covering all related standards and provide reliable delivery and message caching using robust storage. Also making the commercial product of the SMSC functionality itself should include high availability and scaling to different loads of productions. These are common subjects also to other high scale applications, so in this thesis we focus to design only basic short message delivery to prove implementation of the key features and protocol layers.

Proof of concept SMSC shall receive the short message, make decision who is the receiver and send the message forward. It does only registration to the signalling gateway and is shown as active for the mobile network elements. No management or statistic interfaces are defined nor any of message caching.

4.2 Architecture design

Application was chosen to work on the top of IP protocol domain instead of connecting directly to the circuit-switched network. Typically every host computer has Ethernet and IP protocol based physical interface. Connecting to the circuit switched would need physical card to connect MTP1 physical layer. Therefore, SIGTRAN and SUA protocols over IP were chosen to be protocol stack for the application.

Application was identified to be event driven. Message events are received from the network and application produces corresponding response message. Architecture was based on main loop which receives the messages, interprets the data and constructs the response message.

Class hierarchy was designed to be protocol specific. Protocol specific classes try to avoid dependencies of the other layers and are also easily unit tested. Programming language was naturally chosen to be C/C++ because it was important to keep possibility to make bit-level changes if needed.

4.3 SCTP protocol implementation

In Linux kernel there is SCTP implementation over IP protocol which is referred usually to LK-SCTP in Linux kernel terminology (Linux Kernel-SCTP). It is kernel built in implementation which is accessed by the normal socket API in similar manner that is used to program TCP or UDP based applications. It is hosted and documented in (LK-SCTP, project homepage).

Some major SCTP specific parameters are added to the socket API. For example, creating SCTP socket involves the IPPROTO_SCTP argument:

```
A one-to-many style interface with 1 to MANY relationship
between socket and associations where the outbound  asso-
ciation  setup  is       implicit. The syntax of a one-
to-many style socket() call is

sd = socket(PF_INET, SOCK_SEQPACKET, IPPROTO_SCTP);
```

(Linux manual, man 7 sctp)

There are some added functions in addition to normal recv() and send() functions related to the socket API. SCTP specific function names are prefixed with sctp_ and usually include structure for SCTP specific parameters for the connection to be managed. All SCTP related extensions are documented by the Linux Programmer's Manual, the man pages in Linux Debian. LK-SCTP provides also the tools for making basic SCTP connection tests. In Debian Linux it is packaged by name lksctp-tools.

SMSC application initiates the main loop which eventually waits message by calling sctp_recvmsg(). After interpreted incoming message, outgoing messages are sent using sctp_sendmsg().

4.4 SUA protocol implementation

Outgoing ASP UP and ASP ACTIVE messages were implemented for setting application into active state. Upper layer data, the TCAP described in next chapter, is sent and received by supporting CLDT message which carriers the TCAP protocol described in next chapter.

SUA messages were implemented using only defined octet sequences from the specification (described in chapter 2.2.4). Messages are constructed including first the common header which distinguishes version and message type. Actual message parameters are added using Tag-Length-Value TLV coding. A simple class was created which constructs first the header part, parameters according to the message type and eventually the user data. Constructed SUA message is then passed to the SCTP connection using protocol identifier 4 which identifies protocol to be SCCP SUA.

4.5 TCAP protocol implementation

TCAP protocol was implemented using ASN.1 descriptions from the ITU-T specifications described in chapter 2.1.5. ASN1C compiler (from chapter 3.3) was used to generate source and header files from the TCAP portions.

Implemented TCAP message was Begin primitive with Invoke component. It consists of dialogue part which is filled by corresponding version and random transaction id which is generated when message is constructed. In case of sending End primitive to the received Begin, the transaction id is copied from the received message.

Component part is defined as an Invoke type component which includes the carried user data. In this application the user data is assumed to be incoming MAP message described in next chapter.

4.6 MAP protocol implementation

Like TCAP protocol, MAP is also described in ASN.1 syntax in the corresponding specification. For this application only the incoming and outgoing short message operations was needed, thus the implemented MAP operations were *moForwardSM* and *mtForwardSM*.

Incoming message (moForwardSM) and parameters are described in Table 2 and outgoing message (mtForwardSM) in
Table 3. For both messages, common mandatory parameters (M in Request column) are SM RP DA (destination address), SM RP OA (originating address) and SM RP UI (user information).

Table 2 Incoming moForwardSM operation parameters (3GPP TS 29.002, Table 12.2/1)

Table 12.2/1: MAP-MO-FORWARD-SHORT-MESSAGE

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
SM RP DA	M	M(=)		
SM RP OA	M	M(=)		
SM RP UI	M	M(=)	C	C(=)
IMSI	C	C(=)		
User error			C	C(=)
Provider error				O

Table 3 Outgoing mtForwardSM operation parameters (3GPP TS 29.002, Table 12.9/1)

Table 12.9/1: MAP-MT-FORWARD-SHORT-MESSAGE

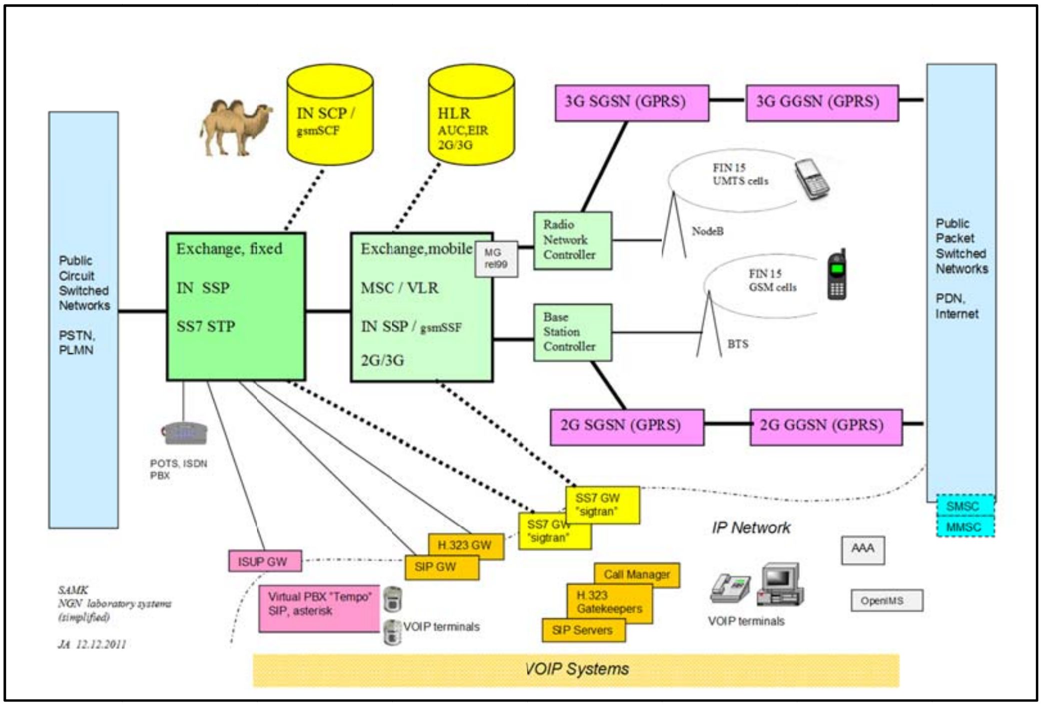
Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
SM RP DA	M	M(=)		
SM RP OA	M	M(=)		
SM RP UI	M	M(=)	C	C(=)
More Messages To Send	C	C(=)		
User error			C	C(=)
Provider error				O

Application supports decoding incoming moForwardSM message which also have IMSI parameter identifying the message sender. Based on that IMSI parameter, application chooses the corresponding MSISDN number into SM RP DA parameter for outgoing message. This IMSI and MSISDN mapping is done over the simple configuration instead of querying from the network. According to the MAP specification, short message receiver should be queried from the network using SEND-ROUTING-INFO-FOR-SM operation as described in (3GPP TS 29.002, chapter 23.3, page 717).

SMS RP UI includes the payload of the actual user typed short message. It has own SMS point-to-point protocol what is described in (GSM 03.40). Incoming moForwardSM MAP-message includes SMS-SUBMIT data type (from mobile to service center) which is converted to the SMS-DELIVER data type (from service center to mobile) and included in outgoing mtForwardSM MAP-message. SMS-SUBMIT and SMS-DELIVER data types and contents are described in (GSM 03.40, chapter 9.2.2).

5 TESTING ENVIRONMENT AND RESULTS

Application was tested using real GSM and UMTS mobile network in SAMK Next Generation Networks laboratory (NGN LAB). Laboratory equipment is shown in Picture 1.



Picture 1 SAMK NGN laboratory equipment (NGNLAB, Equipment link, referred 28.4.2013)

Application was executed in Linux server. The server was normal computer equipped with IP network interface. It was configured to establish SCTP and SUA connections to the Cisco C2651XM ITP router which is presented “SS7 GW” in yellow box in Picture 1. Cisco ITP router was acting as Signalling Gateway between the Nokia Mobile Exchange (MSC/VLR) and the application.

Cellular phone SIM cards were configured to use Short Message Service Center address which is GT number configured in Cisco ITP router to route messages to and from Linux server. The GT number was also configured to the MSC/VLR for routing traffic to application to and from Cisco ITP signalling gateway.

5.1 Message flow

Protocol stack and message flow in testing environment are shown below in Figure 4.

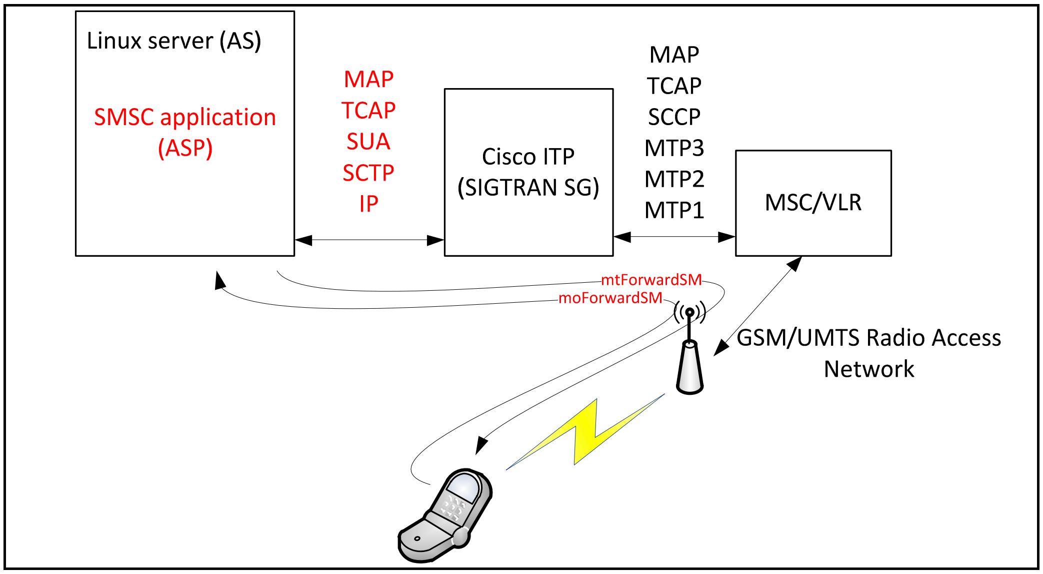


Figure 4 Message flow (implemented parts in this theses are red colored)

As stated in previous chapter, Cisco ITP was used as a Signalling Gateway. It was connected to the Nokia MSC/VLR with configuration that it will route MAP messages between the AS and mobile phone. Cisco ITP signalling gateway transfers the upper layers of the protocol stacks (TCAP and MAP) between the circuit-switched and IP based stacks.

Appendices in this thesis are the network captures taken from the Linux server's network interface. Showing the protocol layers from the IP to MAP (APPENDIX 1, 2, 3).

5.2 Transition to active state

Application has configuration that defines the destination IP address of the Cisco ITP router. After startup, it established SCTP connection to the Cisco and sends SUA messages (ASP UP and ASP ACTIVE) and waits for the corresponding acknowledges. When the responses are received, application enters the waiting mode for incoming traffic messages described in next chapter.

Example of these activation state packets is illustrated in appendices (APPENDIX 1).

5.3 Incoming message into Linux server

In the laboratory there were two mobile phones attached to the network. Both subscriptions had short-message services available by the network configuration in HLR (Home Location Register). In SIM card, there is Short Message Service Center setting defining the destination address (mobile number) where the messages from the phone are sent. That number is always the destination of SMSC, not the message receiver which is carried inside the MAP operation. When the application receives the TCAP message, that SMSC address is shown in TCAP destination address filled by the MSC/VLR element. Respectively TCAP source address is address of the MSC/VLR because that element was the initiator of this dialogue. The SMSC address is then also the same number that was configured into Cisco ITP router to route messages to and from the Linux server.

Test message was sent from the phone, which was seen as moForwardSM MAP message over the SIGTRAN in the Linux server. Protocol stack of this message is shown in appendices (APPENDIX 2).

5.4 Outgoing message from the Linux server

As described in chapter 4.6 the incoming moForwardSM MAP message includes the MSISDN number of the destination but not the IMSI which is needed for the outgoing mtForwardSM MAP message. Production ready Short Message Center would query this from the HLR, but due to thesis work limits we implemented mapping table of MSISDN and IMSI numbers used in the laboratory.

Protocol stack of outgoing message is shown in appendices (APPENDIX 3).

6 CONCLUSION

Selection of the protocols and specifications for this thesis work was very wide and due to the schedule it was clear that I shall limit the functionality to be only proof of concept of the fully functional SMSC. However, all the relevant functional phases of the SMSC were implemented like message receiving and sending. I think that proves that making fully functional and tested product would only need more effort and planning since all technical issues was observed already.

The IP based protocols were clearly described in RFCs, so it was trivial to develop byte-based messages. ASN.1 based protocols (TCAP and MAP) were more complex because ASN1C framework had only basic set of tools for decoding and encoding. I didn't found much of the examples for the TCAP and MAP sessions from the Internet so mostly development was based on the specifications. SAMK NGN laboratory traces from the real mobile network were useful and decreased the testing time.

Developing this proof of concept SMSC into commercial level would naturally need more implementation of the functionalities and wide testing. However, no technical issues were found that would prevent to make full implementation according to the specifications. The major conclusion is that all researched network protocols, functionalities and even network elements from the mobile network could be implemented fully by using freely available open source development tools and libraries. I think that use of open source models could still expand more in the mobile network industry. It might also open the markets for the smaller companies who need reference implementations for their product development.

Therefore, I also wanted to publish this proof of concept implementation under the free software license. Hoping it would help to increase development using open source especially in telecommunications industry (SIGTRANSMSC).

REFERENCES

3GPP 3rd Generation Partnership Project homepage. Referred 4.4.2013. http://www.3gpp.org

3GPP TS 23.002

3GPP Technical Standard 23.002, version 3.4.0, release 99. http://www.etsi.org/deliver/etsi_ts/123000_123099/123002/03.04.00_60/ts_123002v030400p.pdf

3GPP TS 29.002

3GPP Technical Standard 29.002, version 7.15.0, release 99. http://www.etsi.org/deliver/etsi_ts/129000_129099/129002/07.15.00_60/ts_129002v071500p.pdf

ASN1C ASN.1 decoder & encoder compiler project. Referred 4.5.2013. http://www.lionet.info/asn1c

ASN1DEF Abstract Syntax Notation One (ASN.1) Recommendations. Referred 4.5.2013. http://www.itu.int/ITU-T/studygroups/com17/languages/

CPPUNIT CppUnit, C/C++ unit testing framework, project homepage, referred 17.5.2013. http://cppunit.sourceforge.net/

DEBIAN Debian, free Linux operating system. Referred 4.5.2013. http://www.debian.org

ETSI European Telecommunications Standard Institute. Homepage referred 4.4.2013. http://www.etsi.org

GSM 03.40 GSM 03.40 Technical Specification, Technical realization of the Short Message Service (SMS) Point-to-Point (PP), 5.3.0, 1996. http://www.etsi.org/deliver/etsi_gts/03/0340/05.03.00_60/gsmts_0340v050300p.pdf

IETF Homepage referred 31.3.2013. http://www.ietf.org

ITU Q.702 ITU-T Recommendation Q.702 (11/88), MTP 1, signaling data link. http://www.itu.int/rec/T-REC-Q.702

ITU Q.703 ITU-T Recommendation Q.703 (07/96), MTP 2, signaling link. http://www.itu.int/rec/T-REC-Q.703

ITU Q.704 ITU-T Recommendation Q.704 (07/96), MTP 3, signaling network messages and functions. http://www.itu.int/rec/T-REC-Q.704

ITU Q.711 ITU-T Recommendation Q.714 (03/01), Functional description of the signalling connection control part. http://www.itu.int/rec/T-REC-Q.711

ITU Q.771-775

ITU-T Recommendation Q.771 (06/97), Functional descripton of transaction capabilities. http://www.itu.int/rec/T-REC-Q.771

ITU-T Recommendaton Q.772 (06/97), Transaction capabilities information element definitions. http://www.itu.int/rec/T-REC-Q.772

ITU-T Recommendaton Q.773 (06/97), Transaction capabilities formats and encoding (+ ASN.1). http://www.itu.int/rec/T-REC-Q.773

ITU-T Recommendaton Q.774 (06/97), Transaction capabilities procedures. http://www.itu.int/rec/T-REC-Q.774

ITU-T Recommendaton Q.775 (06/97), Guidelines for using transact tion capabilities. http://www.itu.int/rec/T-REC-Q.775

ITU.T Q.1400

ITU-T Recommendation Q.1400 (03/93), Architecture framework for the development of signalling and OA&M protocols using OSI concepts. http://www.itu.int/rec/T-REC-Q.1400

Korhonen 2003

Korhonen, Juha 2003. Introduction to 3G mobile communications – 2nd edition, Artech House, Inc.

LK-SCTP Linux kernel SCTP implementation project homepage, referred 16.5.2013. http://lksctp.sourceforge.net/

NGNLAB Satakunta University of Applied Sciencies, Next Generation Networks laboratory, referred 28.4.2013. http://www.samk.fi/ngn

RFC 768 RFC 768, Universal Datagram Protocol

http://www.ietf.org/rfc/rfc768.txt

RFC 791 RFC 791 Internet Protocol, Protocol Specification.

http://www.ietf.org/rfc/rfc791.txt

RFC 793 RFC 793, Transmission Control Protocol

http://www.ietf.org/rfc/rfc793.txt

RFC 2719 RFC 2719, Framework Architecture for Signalling Transport

http://tools.ietf.org/html/rfc2719

RFC 3286 RFC 3286, an introduction to Stream Control Transmission Protocol http://www.ietf.org/rfc/rfc3286.txt

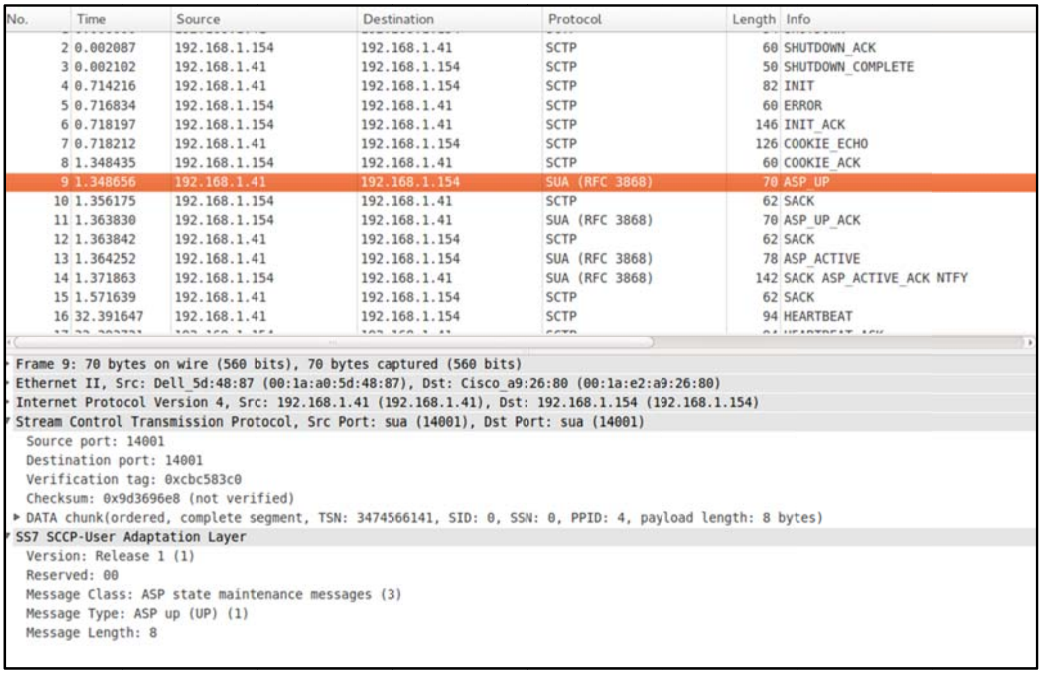
RFC 3868 RFC3868, Signalling Connection Control Part User Adaptation Layer (SUA). http://tools.ietf.org/html/rfc3868

RFC 4960 RFC 4960 Stream Control Transmission Protocol. http://tools.ietf.org/html/rfc4960

SIGTRANSMSC

SIGTRAN based proof of concept Short Message Center project by this thesis, source and project repository, referred 16.5.2013. http://sourceforge.net/projects/sigtran-smsc/

Screen capture is from the Wireshark network capturing software showing the connection establish and activation messages from Linux server (SMSC application) to Cisco ITP signalling gateway.



No.	Time	Source	Destination	Protocol	Length	Info
2	0.002087	192.168.1.154	192.168.1.41	SCTP	60	SHUTDOWN_ACK
3	0.002102	192.168.1.41	192.168.1.154	SCTP	50	SHUTDOWN_COMPLETE
4	0.714216	192.168.1.41	192.168.1.154	SCTP	82	INIT
5	0.716834	192.168.1.154	192.168.1.41	SCTP	60	ERROR
6	0.718197	192.168.1.154	192.168.1.41	SCTP	146	INIT_ACK
7	0.718212	192.168.1.41	192.168.1.154	SCTP	126	COOKIE_ECHO
8	1.348435	192.168.1.154	192.168.1.41	SCTP	60	COOKIE_ACK
9	1.348656	192.168.1.41	192.168.1.154	SUA (RFC 3868)	70	ASP_UP
10	1.356175	192.168.1.154	192.168.1.41	SCTP	62	SACK
11	1.363830	192.168.1.154	192.168.1.41	SUA (RFC 3868)	70	ASP_UP_ACK
12	1.363842	192.168.1.41	192.168.1.154	SCTP	62	SACK
13	1.364252	192.168.1.41	192.168.1.154	SUA (RFC 3868)	78	ASP_ACTIVE
14	1.371863	192.168.1.154	192.168.1.41	SUA (RFC 3868)	142	SACK ASP_ACTIVE_ACK NTFY
15	1.571639	192.168.1.41	192.168.1.154	SCTP	62	SACK
16	32.391647	192.168.1.41	192.168.1.154	SCTP	94	HEARTBEAT

Frame 9: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
Ethernet II, Src: Dell_5d:48:87 (00:1a:a0:5d:48:87), Dst: Cisco_a9:26:80 (00:1a:e2:a9:26:80)
Internet Protocol Version 4, Src: 192.168.1.41 (192.168.1.41), Dst: 192.168.1.154 (192.168.1.154)
Stream Control Transmission Protocol, Src Port: sua (14001), Dst Port: sua (14001)
Source port: 14001
Destination port: 14001
Verification tag: 0xcbc583c0
Checksum: 0x9d3696e8 (not verified)
DATA chunk(ordered, complete segment, TSN: 3474566141, SID: 0, SSN: 0, PPID: 4, payload length: 8 bytes)
SS7 SCCP-User Adaptation Layer
Version: Release 1 (1)
Reserved: 00
Message Class: ASP state maintenance messages (3)
Message Type: ASP up (UP) (1)
Message Length: 8

Picture 2 Packets needed for transition to active state.

Source IP 192.168.1.41: Linux server

Destination IP 192.168.1.154: Cisco ITP Gateway

Packets 2-3: Previous connection shutdown

Packets 4-8: SCTP 4-way handshake initiated by Linux server (packet 5 is error of unsupported features, but does not abort the handshake)

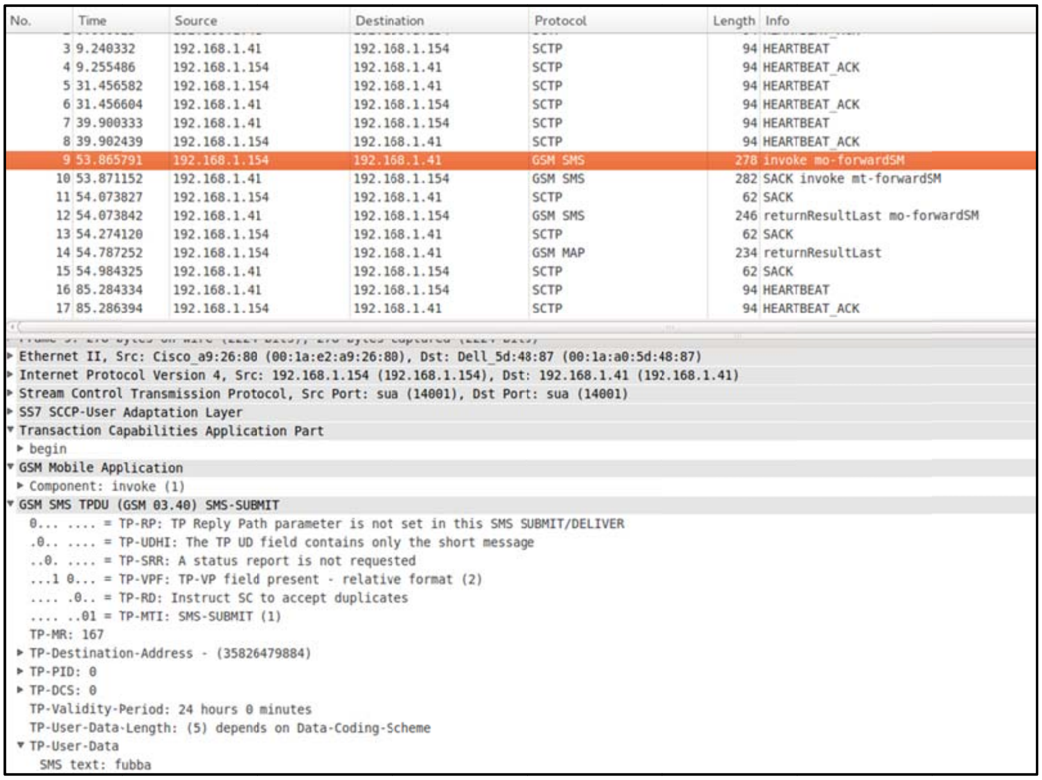
Packet 9: ASP UP sent to Cisco ITP

Packet 10-11: SCTP and SUA level acknowledgements

Packet 13: ASP ACTIVE sent to Cisco ITP and acknowledged by packet 14 ASP ACTIVE ACK

Packet(s) 16+: SCTP Heartbeats keeps the connection alive

Screen capture is from the Wireshark network capturing software showing the incoming short message from the network element (MSC/VLR) to the Linux server (SMSC application).



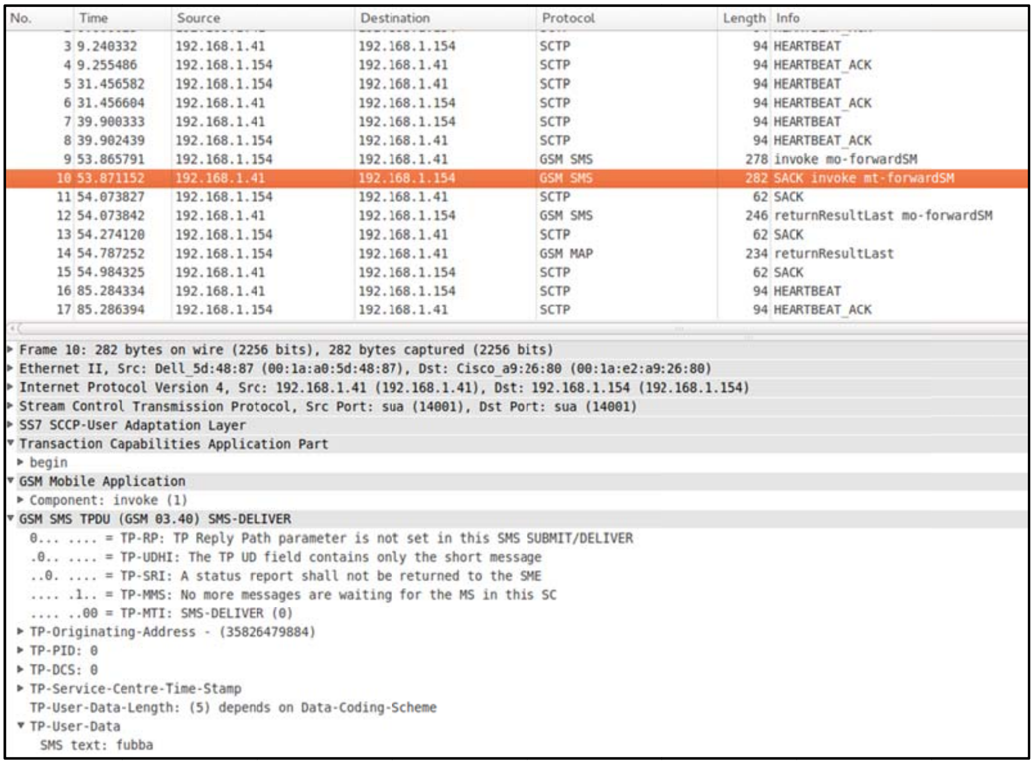
Picture 3 Incoming message to the Linux host

Packets 3-8 are heartbeat and acknowledgements of the connection between Linux server and the Cisco ITP Gateway. Heartbeats are sent when the connection is in the active state, but no other SCTP messages are transferred.

Packet 9 is short-message (text: *fubba)* sent from the mobile phone to destination address +35826479884. Short-message is carried in SMS-SUBMIT protocol data unit (GSM SMS TPDU in picture) which is invoke type TCAP component of the MAP moForwardSM message.

Packet 12 is TCAP layer acknowledgement back to the mobile network which was the sender of the MAP message.

Screen capture is from the Wireshark network capturing software showing the outgoing short-message from Linux server (SMSC application) to the network element (MSC/VLR).



Picture 4 Outgoing message from the Linux host

Packets 3-8 are heartbeat and acknowledgements of the connection between Linux server and the Cisco ITP Gateway. Heartbeats are sent when the connection is in the active state, but no other SCTP messages are transferred.

Packet 10 Outgoing short-message (text: *fubba)* is generated by the Linux server (from the received message). Short-message is carried in SMS-DELIVER protocol data unit (GSM SMS TPDU) which is invoke type of TCAP component of the MAP mtForwardSM message.

Packet 14 is TCAP layer acknowledgement of the outgoing short-message to the subscriber