SMS LA SMPP Manual

Table of Content

1 Introduction 3
1.1 Summary 3
1.2 Scope 3
1.3 Target Readership 3
1.4 Abbreviations 3
1.5 Terminology 4
1.6 Additional Documents 4
SMS LA SMPP Manual Release Management 4
1.7 Customer Contact 6

2 Technical Concept 7
2.1 General Overview 7
2.2 Window Size Applications 8
2.3 CP Whitelisting 8
2.4 Message Inspection 8
2.5 Message Buffer 8
2.6 Throttling 8
2.7 SMS Failover 9
2.8 SMS with correct source address format 10
2.9 UCS2 support for deliver_sm 10

3 SMPP Interface 11
3.1 SMPP Sessions 11
3.1.1 TCP/IP 11
SRAS 11
Bound TX, RX and TRX 11
3.2 SMPP Operations 12
Session Management Operations 12
Message Submission Operations 13
Message Delivery Operations 13
Anciliary Submission Operations 14
3.3 SMPP PDU Definitions 14
3.3.1 Parameter Type Definitions 14
3.3.2 Bind Operations 15
3.3.3 Enquire Link Operation 19
3.3.4 Generic NACK Operation 20
3.3.5 Message Submission Operations 22
3.3.6 Message Delivery Operations 25
3.3.7 Message Cancel Operations 27
3.3.8 Message Query Operations 29
3.3.9 Message Replace Operations 30
3.3.10 Command Status Error Codes 32
3.3.11 SMPP Examples 33

1 Introduction

This document describes the SMPP implementation used in the SMS LA service operated by Swisscom.

All in this document mentioned SMPP functionalities and parameters will work with Swisscom. For all other functionalities is no guaranty of proper functionalities.

1.1 Summary

SMPP, stands for Short Message Peer-to-Peer, is a protocol used by the telecommunications industry for exchanging SMS messages between Short Message Service Center (SMSC) and/or External Short Messaging Entities (ESME) at Swisscom also known as Large Account (LA).
The SMPP protocol is sent via TCP/IP, which allows fast deliver of SMS messages. As a backup, but slow, it is also possible to send SMS via SRAS over ISDN.
The most commonly used versions of SMPP are v3.3, the most widely supported standard, and v3.4, which adds transceiver support (single connections that can send and receive messages). Data exchange may be synchronous, where each peer must wait for a response for each PDU being sent, and asynchronous, where multiple requests can be issued in one go and acknowledged in a skew order by the other peer.

Recommendation:
For a more performant throughput, we strongly recommend to use asynchronous mode!

- SMPP version 5.0: "Short Message Peer to Peer Protocol Specification", version 5.0, 19-Feb-2003, SMS Forum [1]
- SMPP version 3.4: "Short Message Peer to Peer Protocol Specification", v3.4, 12-Oct-1999, Issue 1.2, SMPP Developers Forum [2]

All those specifications are available from *http://www.smsforum.net*

1.2 Scope

The product “SMS Large Account” offers an interface to third parties to the GSM network of Swisscom.

This document describes the SMS service based on SMPP protocol with the Swisscom specific settings. This technical description provides the basis to build a SMS Large Account service by third parties.

1.3 Target Readership

- Third party product managers
- Third party technical staff
- Swisscom product managers
- Swisscom technical staff

1.4 Abbreviations

API	Application Programming Interface
Client ID	encrypted MSISDN
CP	Content Provider
CUC	Customer Care
EMS	Enhanced Message Service
ESME	External Short Message Entity (also known as Large Account)
GSM	Global System for Mobile communications
IMEI	International Mobile Equipment Identity

ISO	International Standards Organization
LA	Large Account
MB	Message Bureau - This is typically an operator message bureau.
MC	Message Centre - A generic term used to describe various types of SMS
MO	Mobile Originating
MSISDN	Mobile Station ISDN number
MT	Mobile Terminating
PDU	Protocol Data Unit
SIS	Subscriber Information Server
SME	Short Message Entity
SMPP	Short Message Peer to Peer Protocol
SMS	Short Message Service
SMSC	Short Message Service Center
SRAS	SMS Request Access Service
TP	Third Party
UDHI	User Data Header Indicator

1.5 Terminology

Third Party	Swisscom Third Party (or Content Provider) Business customer, connecting to the mobile network and offering a service to the mobile end user
Short ID	"abbreviated number of a SMS service" known to the mobile customer, also called ESME or Short Code (e.g. 222 for SBB timetable). Short ID is only within the mobile network of Swisscom valid.

1.6 Additional Documents

[1] SMPP V5.0: "Short Message Peer to Peer Protocol Specification" version 5.0 19-Feb-2003, SMS Forum http://docs.nimta.com/smppv50.pdf (see also http://en.wikipedia.org/wiki/Short_Message_Peer-to-Peer)

[2] SMPP V3.4: "Short Message Peer to Peer Protocol Specification", v3.4, 12-Oct-1999, Issue 1.2, SMPP Developers Forum http://docs.nimta.com/SMPP_v3_4_Issue1_2.pdf (see also http://en.wikipedia.org/wiki/Short_Message_Peer-to-Peer)

[3] GSM 03.40: ETSI standard GSM 03.40 http://www.etsi.org/deliver/etsi_gts/03/0340/05.03.00_60/gsmts_0340v050300p.pdf

SMS LA SMPP Manual Release Management

The table summarizes the major differences in this document due to a new document release.

Version	Chapter	Description
1.0	all	According Swisscom SMPP project
1.1	3.1.1	IP addresses added
1.2	1.8	New Hotline number
1.3	2.7	SMS with correct source address format
1.4	3.1.11	SMPP examples
1.5	all	Small adaptions
1.6	all	Updated
1.7	2.5 and 3.3.10	Throttling and error handling
1.8	2.8	UCS2 messages
1.9	1.7	Unified email support addresses, removed Partner Integration

2.0	1.1, 2.5 + 2.6	Added asynchronous mode as recommendation in chapter 1.1 and Chapter 2.5 (Message Buffer) and corrections to 2.6

1.7 Customer Contact

For **administrative questions** please contact Swisscom Provider Support:

- Phone: +41 (0) 800 848 900
- Email: Provider.Support@swisscom.com
- Address: Swisscom (Schweiz) AG
  Enterprise Solutions
  Mobile Business Solutions
  Design & Delivery
  Provider Support
  PO - 3050 Bern

For **technical support** please contact Swisscom Provider Support:

- Email: Provider.Support@swisscom.com
- Internet: www.swisscom.ch/iservclient

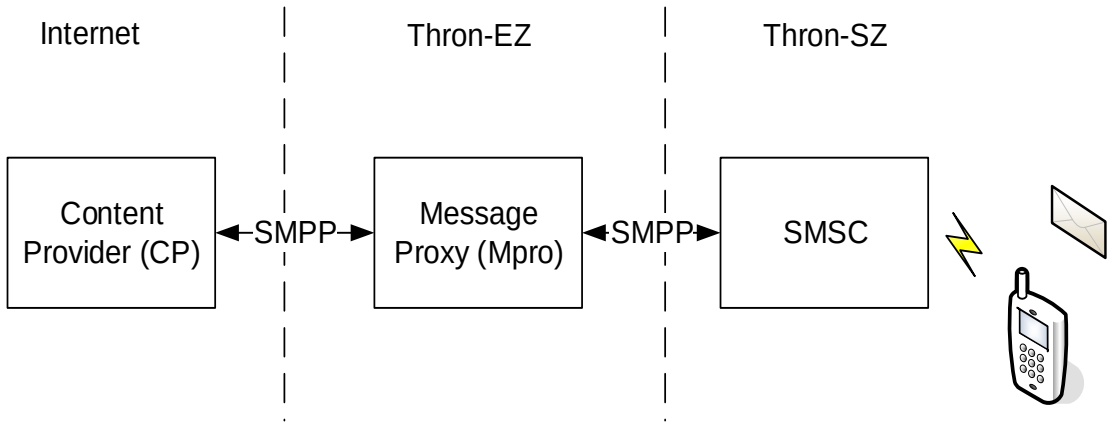
Swisscom Helpline:

- Phone: +41 (0) 800 848 900

2 Technical Concept

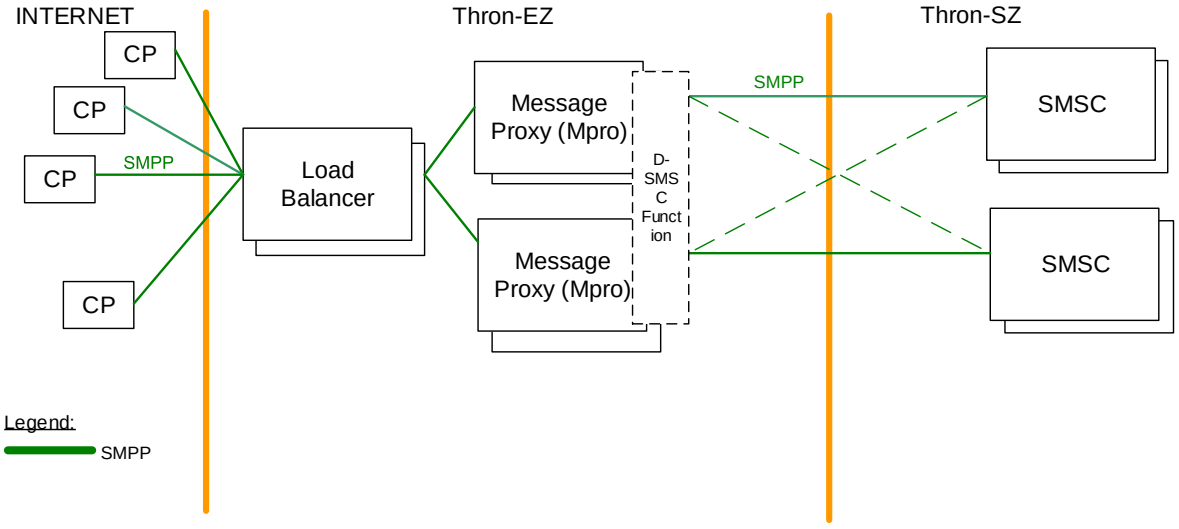
2.1 General Overview

Swisscom Mobile offers Third parties (Large Account customers) connections to its GSM mobile network via the Messaging Proxy for sending receiving SMS. The Messaging Proxy acts as gateway for incoming and outgoing SMS.
Large Account customers (Content Provider or Third Parties) have the possibility to communicate with the SMSC either with SMPP or REST protocol. It depends on the LA settings.



Picture 1: Overview new SMPP connections

MT SMS: Content Providers in Internet send SMPP messages to the Messaging Proxy. The Messaging Proxy forwards the messages to the SMSC in the secure zone.
MO-SMS: The SMSC forwards Mobile Originated messages from end-users to the Messaging Proxy. The Messaging Proxy forwards the messages to the Content Provider in Internet.



Picture 2: Detailed View of MP connections

2.2 Window Size Applications

SMPP is an asynchronous protocol. This means LA and SMSC can send several requests at a time to the other party. Each SMPP request has an identifier called a sequence number that is used uniquely to identify the PDU in the context of its originating entity and the current SMPP session. The PDU sequence numbers make the request/response matching possible, regardless of when an asynchronous response arrives.
Therefore, windowing can be applied furthermore.
The SMSC supports a sliding window size in both directions separately. The windowing size can be set between 1 and 10 (default is 1).

2.3 CP Whitelisting

This additional check ensures that a given CP is not able to invoke a service belonging to another CP. It's applied within the MP Proxy workflows which checks if the originating IP address and port is part of the list of the configured IPs for the given CP.

2.4 Message Inspection

All SMPP messages are checked against IP address and short code within system_id field in Bind Syntax.
The SMPP messages (requests) have an associated response message (acknowledgement) to indicate whether the message was received or returns an error code in case of failure.
If one of the communicating parties receives a SMPP message it has to decode it and determine what sort of SMPP message it is. If the SMPP header is correct, then the message body will be decoded. In case of a failure an error code will be inserted in the particular response message.

2.5 Message Buffer

To ensure quality of service and persistent synchronisation of cluster the SMSC stores each message prior to transmission (submit/delivery) in a store/queue/buffer. The capacity of this buffer is limited. Once the buffer is full no more message (MO/MT) is accepted and error code ESME_RMSGQFUL is forwarded to the CP as response to the command triggering the overflow. Full buffer may happen on sending of BULK batch of messages when notifications or messages are not fetched while sending by the CP.
By reacting to error messages within your application, a full buffer can be avoided.

More about error codes and error handling you can find on chapter 3.3.10.

2.6 Throttling

Each ESME is provisioned with a dedicated throughput. If this throughput level is reached, SMSC will throttle the SMS traffic.
In this case SMSC sends a negative acknowledgment in a response message (ESME_RTHROTTLED) to the content provider application when the configured throughput is exceeded by the sending application. So, the application knows that it has exceeded the agreed throughput and that these messages have not been accepted by the SMSC and are not delivered to recipients.

For the CP developers of the SMPP application, it is essential that they themselves control the throughput of their message sender in order not to receive such NACK from the SMSC. Also, they must implement a logic in their application to handle such error messages.

It's important to know that the throughput is calculated over all established sessions as a sum of all active message operations (submit_sm, deliver_sm, query_sm …) per ESME. Exceeding your throughput may not only restrict the CP's ability to send messages, but also the CP's ability to receive messages and notifications.

More about error codes and error handling you can find on chapter 3.3.10.

2.7 SMS Failover

For SMPP connections a CP will be connected to a single load-balancer IP address. The load balancer will distribute SMPP connections in a round-robin fashion between the redundant Message Proxy (Mpro) instances. Each MP maintains exactly one outbound connection (to either primary or backup DSMSC) for each inbound connection from the CP for SMPP.
When an inbound/outbound connection drops or if a connection was idle for too long, the MP can instantly close its matching outbound/inbound connection and force the CP to re-establish the dropped SMPP session. To re-establish the SMPP sessions the following session management operations are available: ***unbind, bind, enquire_link***.
This is needed to enforce retry with potential failover in cases where no proper connection closure was possible, e.g. upon power outage; otherwise the communicating parties will forever wait.

The SMSC can deliver messages to CP over one or more links. It depends if the LA is registered as a single address or multiple address large account. For the latter situation on CP side, with multiple addresses the SMSC uses the other links as failover possibilities.

Recommendation:
We strongly recommend setting the connection timeout on CP side (application) ≥ 5 seconds. (Reason is: if there is a SMSC failover it can go up to 5 seconds until CP will get an answer. Otherwise in this situation CP will not be able to send any SMS)

2.8 SMS with correct source address format

While sending SMS, it is possible to define the source address in an alphanumeric or numeric format.

If the numeric format is chosen, here are the important parameters, which have to be set in the right way:

- **source_addr_ton = 1** (International)
- **source_addr_npi = 1** (ISDN)
- **source_addr = 41791112233 (example)**, it has to be a valid phone number in international format with no 00 or + in front of sender's number. The + sign will be added by receiver's mobile. The receiver will see sender's number in this format: +41791112233 (example).
  We recommend to use our feature "Global Reply" that provides a virtual MSISDN (+4179807xxxx); all MO messages are routed back to your SMS Large Account. For more information, please contact provider.support@swisscom.com

If the alphanumeric format is chosen, here are the important parameters:

- **source_addr_ton = 5** (Alphanumeric number)
- **source_addr_npi = 0** (Unknown)
- **source_addr = <any text>**. The receiver will see sender's number as a text and can not send any answer to this alphanumeric address.

2.9 UCS2 support for deliver_sm

As some mobile devices encode their MO SMS content into UCS2 formatted messages, the ESME application should be able to receive and handle 7Bit ascii and as well UCS2 encoded content! Otherwise you probably will not be able to fulfil your service.

3 SMPP Interface

3.1 SMPP Sessions

SMPP sessions can be established either via TCP/IP, IPSS or SRAS over ISDN.

To transfer data via SMPP the following sequence for a session set up is used:

Sequence	Session state
Setting up a TCP/ IP network connection between LA and the SMSC	Open
Setting up the SMPP session, initiated by LA (Login with “bind”)	Bound
Transferring data	Bound
Closing the SMPP session	Unbound
Closing the network connection	Closed

SMPP is basically a set of operations, each one taking the form of a request and response Protocol Data Unit (PDU).

3.1.1 TCP/IP

Establishing a Session first requires the ESME to connect to the MC. This is achieved using a TCP/IP connection. Throughput is max. 10 SMS/sec for one session (it depends on the contract type). It is possible to have up to 5 sessions.

Third Party can connect to the following address:
→ **messagingproxy.swisscom.ch (currently IP 217.192.8.32)** at port **4200**

A session will be closed by MC when there is inactivity for a certain time (Swisscom setting: 5 min.).
For holding open a session ESME can send a “keep alive” to MC → ***enquire_link***

SRAS

SRAS connection is established via an ISDN connection.
Throughput is max. 3 SMS/sec (it depends on the contract type)

Modem Protocol which is supported by Swisscom::

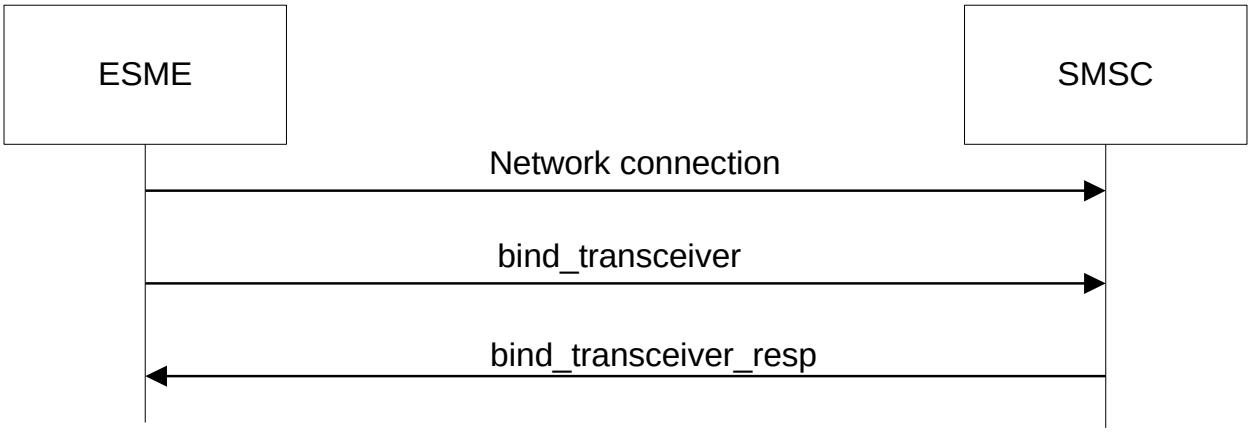
- X.75
- V.120
- V.110 asynchronous (LLC needs to be set correct)

A session will be closed by MC when there is inactivity for a certain time (Swisscom setting: 5 min.).
For holding open a session ESME can send a “keep alive” to MC → ***enquire_link***

Bound TX, RX and TRX

The session set up is for Transmitter, Receiver and transceiver the same procedure. Here an example with a transceiver session establishment.

A connected ESME has requested to bind as a Transceiver (by issuing a bind_transceiver PDU) and has received a bind_transceiver_resp PDU from the MC authorising its Bind request. An ESME bound as a Transceiver is authorised to use all operations supported by a Transmitter ESME and a Receiver ESME. A transceiver session is effectively the combination of a Transmitter and a Receiver session. Thus an ESME bound as a transceiver may send short messages to a MC for onward delivery to a Mobile Station or to another ESME and may also receive short messages from a MC, which may be originated by a mobile station, by another ESME or by the MC itself (for example a MC delivery receipt). Refer to section 2.4 for a full list of applicable operations in Bound_TRX state.



Picture 3: Bound_TRX State

3.2 SMPP Operations

Session Management Operations

SMPP PDU Name	Description
bind_transmitter	Authentication PDU used by a transmitter ESME to bind to the Message Centre. The PDU contains identification information and an access password for the ESME.
bind_transmitter_resp	Message Centre response to a bind_transmitter PDU. This PDU indicates the success or failure of the ESME's attempt to bind as a transmitter.
bind_receiver	Authentication PDU used by a receiver ESME to bind to the Message Centre. The PDU contains identification information, an a cess password for the ESME and may also contain routing info mation specifying the range of addresses serviced by the ESME.
bind_receiver_resp	Message Centre response to a bind_receiver PDU. This PDU indicates the success or failure of the ESME's attempt to bind as a receiver.
bind_transceiver	Authentication PDU used by a transceiver ESME to bind to the Message Centre. The PDU contains identification information, an access password for the ESME and may also contain routing information specifying the range of addresses serviced by the ESME.
bind_transceiver_resp	Message Centre response to a bind_transceiver PDU. This PDU indicates the success or failure of the ESME's attempt to bind as a transceiver.
unbind	This PDU can be sent by the ESME or MC as a means of initiating the termination of a SMPP session.

unbind_resp	This PDU can be sent by the ESME or MC as a means of acknowledging the receipt of an unbind request. After sending this PDU the MC typically closes the network connection.
enquire_link (only direction from ESME to SMSC)	This PDU can be sent by the ESME or MC to test the network connection. The receiving peer is expected to acknowledge the PDU as a means of verifying the test. This is also used for the "keep alive" functionality to keep the session open.
enquire_link_resp	This PDU is used to acknowledge an enquire_link request sent by an ESME or MC.
generic_nack	This PDU can be sent by an ESME or MC as a means of indicating the receipt of an invalid PDU. The receipt of a generic_nack usually indicates that the remote peer either cannot identify the PDU or has deemed it an invalid PDU due to its size or content.

Message Submission Operations

SMPP PDU Name	Description
submit_sm	A transmitter or transceiver ESME, wishing to submit a short message, can use this PDU to specify the sender, receiver and text of the short message. Other attributes include message priority, data coding scheme, validity period etc.
submit_sm_resp	The MC response to a submit_sm PDU, indicating the success or failure of the request. Also included is a MC message_id that can be used in subsequent operations to query, cancel or replace the contents of an undelivered message.
data_sm	data_sm is a streamlined version of the submit_sm operation, designed for packet-based applications that do not demand extended functionality normally available in the submit_sm operation. ESMEs implementing WAP over a SMS bearer typically use this operation.
data_sm_resp	The MC response to a data_sm PDU, indicating the success or failure of the request. Also included is a MC message_id that can be used in subsequent operations to query, cancel or replace the contents of an undelivered message.

Note: submit_multi is not supported

Message Delivery Operations

SMPP PDU Name	Description
deliver_sm	Deliver_sm is the symmetric opposite to submit_sm and is used by a MC to deliver a message to a receiver or transceiver ESME.
deliver_sm_resp	This PDU indicates the ESMEs acceptance or rejection of the delivered message. The error returned by the ESME can cause the message to be retried at a later date or rejected there and then.
data_sm	Data_sm can also be used for message delivery from Message Centre to the ESME. ESMEs implementing WAP over SMS typically use this operation.
data_sm_resp	The ESME response to a data_sm PDU, indicating the success or failure of the MC-initiated delivery request.

Anciliary Submission Operations

SMPP PDU Name	Description
cancel_sm	This PDU is used to cancel a previously submitted message. The PDU contains the source address of the original message and the message_id returned in the original submit_sm_resp, submit_multi_resp or data_sm_resp PDU. This PDU may also omit the message_id and instead contain a source address, destination address and optional service_type field as a means of cancelling a range of messages sent from one address to another.
cancel_sm_resp	The MC returns this PDU to indicate the success or failure of a cancel_sm PDU.
query_sm	This PDU is used to query the state of a previously submitted message. The PDU contains the source address of the original message and the message_id returned in the original submit_sm_resp, submit_multi_resp or data_sm_resp PDU.
query_sm_resp	The MC returns a query_sm_resp PDU as a means of indicating the result of a message query attempt. The PDU will indicate the success or failure of the attempt and for successful attempts will also include the current state of the message.
replace_sm	The replace_sm PDU is used by an ESME to pass a message_id of a previously submitted message along with several other fields used to update the text, validity period and other attributes of the message.
replace_sm_resp	The replace_sm_resp PDU indicates the success or failure of a replace_sm PDU

3.3 SMPP PDU Definitions

3.3.1 Parameter Type Definitions

All SMPP PDUs comprise of organised sets of parameters. These parameters can have any of the following formats:

Note: Values depicted with a 0x prefix are in Hexadecimal format, meaning that each digit represents 4 binary bits. Thus, a 2-digit hex number is represented by 1 octet of data.

Parameter Type	Description
Integer	An unsigned integer value, which can be 1, 2 or 4 octets in size. The octets are always encoded in Most Significant Byte (MSB) first order, otherwise known as Big Endian Encoding. Example: A 1-octet Integer with a value 5, would be encoded in a single octet with the value 0x05. A 2-octet integer with the decimal value of 41746 would be encoded as 2 octets with the value 0xA312
C-Octet String	A C-Octet String is a sequence of ASCII characters terminated with a NULL octet (0x00). Example: The string “Hello” would be encoded in 6 octets (5 characters of “Hello” and NULL octet) as follows: 0x48656C6C6F00

Octet String	An Octet String is a sequence of octets not necessarily terminated with a NULL octet. Such fields using Octet String encoding, typically represent fields that can be used to encode raw binary data.
Tagged Length Value (TLV)	A Tagged Length Value Field is a special composite field that comprises of three parts: • A 2-octet Integer (Tag). The tag identifies the parameter. • A 2-octet Integer (Length) The length field indicates the length of the value field in octets. Note that this length does not include the length of the tag and length fields. • An Octet String (Value) The value field contains the actual data for the TLV field. The Tag identifies the parameter. The Length indicates the size of the Value field in octets. An example of a TLV is the *dest_bearer_type*. Its Tag is 0x0007 and has a value size of 1 octet. The value 0x04 indicates USSD as a bearer type. In its encoded form, this TLV would appear as follows: 0x0007000104

For additional information, please see SMPP specification in chapter 3 [1]

3.3.2 Bind Operations

The purpose of the SMPP bind operation is to register an instance of an ESME with the MC system and request a SMPP session over this network connection for the submission or delivery of messages. Thus, the Bind operation may be viewed as a form of MC login request to authenticate the ESME entity wishing to establish a connection.
As described previously, an ESME may bind to the MC as a Transmitter (called ESME Transmitter), a Receiver (called ESME Receiver), or a Transceiver (called ESME Transceiver). There are three SMPP bind PDUs to support the various modes of operation, namely *bind_transmitter, bind_transceiver* and *bind_receiver*. The command_id field setting specifies which PDU is being used.
An ESME may bind as both a SMPP Transmitter and Receiver using separate *bind_transmitter* and *bind_receiver* operations (having first established two separate network connections). Alternatively an ESME can also bind as a Transceiver having first established a single network connection.

bind_transmitter Syntax

The format of the SMPP *bind_transmitter* PDU is definded in the following table:

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *bind_transmitter* PDU.
command_id	4	Integer	0x00000002
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a unique sequence number. The associated *bind_transmitter_resp* PDU will echo the same sequence number.
system_id	Variable max. 16	C-Octet String	Identifies the ESME system requesting to bind as a transmitter with the MC.

			In this field must contain the Short ID (short code)
password	Variable max. 9	C-Octet String	The password may be used by the MC to authenticate the ESME requesting to bind (max. 8 ASCII characters)
system_type	Variable max. 13	C-Octet String	Identifies the type of ESME system requesting to bind as a transmitter with the MC.
interface_version	1	Integer	Indicates the version of the SMPP protocol supported by the ESME.
addr_ton	1	Integer	Indicates Type of Number of the ESME address. If not known set to NULL.
addr_npi	1	Integer	Numbering Plan Indicator for ESME address. If not known set to NULL.

bind_transmitter_resp Syntax

The format of the SMPP *bind_transmitter_resp* PDU is definded in the following table:

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *bind_transmitter_resp* PDU.
command_id	4	Integer	0x80000002
command_status	4	Integer	Indicates status (success or error code) of original *bind_transmitter* request.
sequence_number	4	Integer	Set to sequence number of original *bind_transmitter* request.
system_id	Variable max. 16	C-Octet String	MC identifier.Identifies the MC to the ESME. **In this field must contain the Short ID (short code)**
Optional TLVs:			
sc_interface_version		TLV	SMPP version supported by MC

bind_receiver Syntax

The format of the SMPP *bind_receiver* PDU is defined in the following table.

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *bind_receiver* PDU.

command_id	4	Integer	0x00000001
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a unique sequence number. The associated *bind_receiver_resp* PDU will echo the same sequence number.
system_id	Variable max. 16	C-Octet String	Identifies the ESME system requesting to bind as a transmitter with the MC. **In this field must contain the Short ID (short code)**
password	Variable max. 9	C-Octet String	The password may be used by the MC to authenticate the ESME requesting to bind (max. 8 ASCII characters)
system_type	Variable max. 13	C-Octet String	Identifies the type of ESME system requesting to bind as a transmitter with the MC.
interface_version	1	Integer	Indicates the version of the SMPP protocol supported by the ESME.
addr_ton	1	Integer	Indicates Type of Number of the ESME address. If not known set to NULL.
addr_npi	1	Integer	Numbering Plan Indicator for ESME address. If not known set to NULL.

bind_receiver_resp Syntax

The format of the SMPP *bind_receiver_resp* PDU is definded in the following table:

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *bind_receiver_resp* PDU.
command_id	4	Integer	0x80000001
command_status	4	Integer	Indicates status (success or error code) of original *bind_receiver* request.
sequence_number	4	Integer	Set to sequence number of original *bind_receiver* request.
system_id	Variable max. 16	C-Octet String	MC identifier.Identifies the MC to the ESME. **In this field must contain the Short ID (short code)**
Optional TLVs:			
sc_interface_version		TLV	SMPP version supported by MC

bind_transceiver Syntax

The format of the SMPP *bind_transceiver* PDU is defined in the following table.

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *bind_transceiver* PDU.
command_id	4	Integer	0x00000009
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a unique sequence number. The associated *bind_transceiver_resp* PDU will echo the same sequence number.
system_id	Variable max. 16	C-Octet String	Identifies the ESME system requesting to bind as a transmitter with the MC. **In this field must contain the Short ID (short code)**
password	Variable max. 9	C-Octet String	The password may be used by the MC to authenticate the ESME requesting to bind (max. 8 ASCII characters)
system_type	Variable max. 13	C-Octet String	Identifies the type of ESME system requesting to bind as a transmitter with the MC.
interface_version	1	Integer	Indicates the version of the SMPP protocol supported by the ESME.
addr_ton	1	Integer	Indicates Type of Number of the ESME address. If not known set to NULL.
addr_npi	1	Integer	Numbering Plan Indicator for ESME address. If not known set to NULL.

bind_transceiver_resp Syntax

The format of the SMPP *bind_transceiver_resp* PDU is definded in the following table:

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *bind_transceiver_resp* PDU.
command_id	4	Integer	0x80000009
command_status	4	Integer	Indicates status (success or error code) of original *bind_transceiver* request.
sequence_number	4	Integer	Set to sequence number of original *bind_transceiver* request.
system_id	Variable max. 16	C-Octet String	MC identifier.Identifies the MC to the ESME. **In this field must contain the Short ID (short code)**

Optional TLVs:			
sc_interface_version		TLV	SMPP version supported by MC

unbind Syntax

The purpose of the SMPP *unbind* operation is to deregister an instance of an ESME from the MC and inform the MC that the ESME no longer wishes to use this network connection for the submission or delivery of messages.
Thus, the *unbind* operation may be viewed as a form of MC logoff request to close the current SMPP session. The format of the SMPP *unbind* PDU is defined in the following table:

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *unbind* PDU.
command_id	4	Integer	0x00000006
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a unique sequence number. The associated *unbind_resp* PDU will echo the same sequence number

unbind_resp Syntax

The SMPP *unbind_resp* PDU is used to reply to an *unbind* request. It comprises the SMPP message header only.
The format of the SMPP *unbind_resp* PDU is defined in the following table:

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *unbind_resp* PDU.
command_id	4	Integer	0x80000006
command_status	4	Integer	Indicates outcome of original *unbind* request.
sequence_number	4	Integer	Set to sequence number of original *unbind* request.

3.3.3 Enquire Link Operation

This PDU can be originated by the ESME towards MC and is used to provide a “keep alive” or confidence check of the communication path between an ESME and a MC. On receipt of this request the receiving party should respond with an *enquire_link_resp*, thus verifying that the application level connection between the MC and the ESME is functioning.

Enquire_link Syntax

The *enquire_link* PDU is used for “keep alive”

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *enquire_link* PDU.
command_id	4	Integer	0x00000015
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a unique sequence number. The associated *enquire_link_resp* PDU should echo the same sequence number

enquire_link_resp Syntax

The *enquire_link_resp* PDU is used to reply to an *enquire_link* request.

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *unbind_resp* PDU.
command_id	4	Integer	0x80000015
command_status	4	Integer	Indicates outcome of original *unbind* request.
sequence_number	4	Integer	Set to the same sequence number of original *enquire_link* PDU

3.3.4 Generic NACK Operation

The *generic_nack* PDU is used to acknowledge the submission of an unrecognized or corrupt PDU.

generic_nack Syntax

Following is the format of the SMPP *generic_nack* PDU. It comprises the SMPP message header only.

SMPP PDU Name	Size octets	Type	Description

command_length	4	Integer	Defines the overall length of the *generic_nack* PDU.
command_id	4	Integer	0x80000000
command_status	4	Integer	Error code corresponding to reason for sending the *generic_nack*.
sequence_number	4	Integer	Set to sequence number of original PDU or to NULL if the original PDU cannot be decoded.

3.3.5 Message Submission Operations

Message submission operations provide an ESME with the ability to submit messages for onward delivery to mobile stations.

submit_sm Syntax

This operation is used by an ESME to submit a short message to the MC for onward transmission to a specified short message entity (SME):

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Defines the overall length of the *submit_sm* PDU.
command_id	4	Integer	0x00000004
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a unique sequence number. The associated *submit_sm_resp* PDU will echo the same sequence number.
service_type	Variable max. 6	C-Octet String	The *service_type* parameter can be used to indicate the SMS Application service associated with the message. Specifying the *service_type* allows the ESME to avail of enhanced messaging services such as “replace by *service_type*” or to control the teleservice used on the air interface. Set to NULL for default MC settings
source_addr_ton	1	Integer	Type of Number for source address. If not known, set to NULL (Unknown).
source_addr_npi	1	Integer	Numbering Plan Indicator for source address. If not known, set to NULL (Unknown).
source_addr	Variable max. 21	C-Octet String	Address of SME which originated this message. If not known, set to NULL (Unknown).
dest_addr_ton	1	Integer	Type of Number for destination
dest_addr_npi	1	Integer	Numbering Plan Indicator for destination
destination_addr	Variable max. 21	C-Octet String	Destination address of this short message For mobile terminated messages, this is the directory number of the recipient MS
esm_class	1	Integer	Indicates Message Mode and Message Type
protocol_id	1	Integer	Protocol Identifier. Network specific field.
priority_flag	1	Integer	Not supported by Swisscom
schedule_delivery_time	1 or 17	C-Octet String	The short message is to be scheduled by the MC for delivery. Set to NULL for immediate message delivery

validity_period	1 or 17	C-Octet String	The validity period of this message. Set to NULL to request the MC default validity period **Note:** this is superseded by the *qos_time_to_live* TLV if specified.
registered_delivery	1	Integer	Indicator to signify if a MC delivery receipt, manual ACK, delivery ACK or an intermediate notification is required.
replace_if_present_flag	1	Integer	Flag indicating if the submitted message should replace an existing message.
data_coding	1	Integer	Defines the encoding scheme of the short message user data.
sm_default_msg_id	1	Integer	Not supported by Swisscom
sm_length	1	Integer	Length in octets of the short_message user data.
short_message	Variable 0-255	Octet String	Up to 255 octets of short message user data. The exact physical limit for short_message size may vary according to the underlying network **Note1:** In SMPP message data length for concatenated/segmented mesages is as follows: data_coding = 8 (Binary) Total message length 140 Text message length per segment 132 data_coding = 0 (7Bit) Total message lenght 160 Text message length per segment 152 **Note2:** this field is superceded by the *message_payload* TLV if specified.
Message Submission TLVs	Variable	TLV	

submit_sm_resp Syntax

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Set to overall length of PDU.
command_id	4	Integer	0x80000004
command_status	4	Integer	Indicates outcome of *submit_sm* request.
sequence_number	4	Integer	Set to sequence number of original *submit_sm* PDU.
message_id	Variable max. 65	C-Octet String	This field contains the MC message ID of the submitted message. It may be used at a later stage to

			query the status of a message, cancel or replace the message.
Message Submission Response TLVs	Variable	TLV	

data_sm Syntax

The *data_sm* operation is similar to the *submit_sm* in that it provides a means to submit a mobile-terminated message. However, *data_sm* is intended for packet-based applications such as WAP in that it features a reduced PDU body containing fields relevant to WAP or packet-based applications.

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Set to overall length of PDU..
command_id	4	Integer	0x00000103
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a Unique sequence number. The associated *data_sm_resp* PDU will echo this sequence number.
service_type	Variable max. 6	C-Octet String	The *service_type* parameter can be used to indicate the SMS Application service associated with the message. Specifying the *service_type* allows the ESME to avail of enhanced messaging services such as “replace by *service_type*” or control the teleservice used on the air interface. Set to NULL for default MC settings
source_addr_ton	1	Integer	Type of Number for source address. If not known, set to NULL (Unknown).
source_addr_npi	1	Integer	Numbering Plan Indicator for source address. If not known, set to NULL (Unknown).
source_addr	Variable max. 65	C-Octet String	Address of SME which originated this message. If not known, set to NULL (Unknown).
dest_addr_ton	1	Integer	Type of Number for destination
dest_addr_npi	1	Integer	Numbering Plan Indicator for destination
destination_addr	Variable max. 65	C-Octet String	Destination address of this short message For mobile terminated messages, this is the directory number of the recipient MS
esm_class	1	Integer	Indicates Message Mode and Message Type
registered_delivery	1	Integer	Indicator to signify if a MC delivery receipt or an SME acknowledgement is required.
data_coding	1	Integer	Defines the encoding scheme of the short message user data.

Message Submission TLVs	Variable	TLV	

data_sm_resp Syntax

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Set to overall length of PDU.
command_id	4	Integer	0x80000103
command_status	4	Integer	Indicates outcome of *data_sm* request.
sequence_number	4	Integer	Set to sequence number of original *data_sm* PDU.
message_id	Variable max. 65	C-Octet String	This field contains the MC message ID of the submitted message. It may be used at a later stage to query the status of a message, cancel or replace the message.
Message Submission Response TLVs	Variable	TLV	

3.3.6 Message Delivery Operations

Message delivery operations provide the means of delivering short messages from a MC to an ESME. These messages typically originate from mobile stations.

deliver_sm Syntax

The *deliver_sm* is issued by the MC to send a message to an ESME. Using this command, the MC may route a short message to the ESME for delivery.

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Set to overall length of PDU..
command_id	4	Integer	0x00000005
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a unique sequence number. The associated *deliver_sm_resp* PDU will echo the same sequence number.
service_type	Variable max. 6	C-Octet String	The service_type parameter can be used to indicate the SMS Application service associated with the message. Specifying the service_type allows

			the ESME to avail of enhanced messaging services such as “replace by service_type” or control the teleservice used on the air interface. Set to NULL if not known by MC
source_addr_ton	1	Integer	Type of Number for source address.
source_addr_npi	1	Integer	Numbering Plan Indicator for source address.
source_addr	Variable max. 21	C-Octet String	Address of SME which originated this message.
dest_addr_ton	1	Integer	Type of Number for destination
dest_addr_npi	1	Integer	Numbering Plan Indicator for destination
destination_addr	Variable max. 21	C-Octet String	Destination address of this short message For mobile terminated messages, this is the directory number of the recipient MS
esm_class	1	Integer	Indicates Message Mode and Message Type
protocol_id	1	Integer	Protocol Identifier. Network specific field.
schedule_delivery_time	1 or 17	C-Octet String	The short message is to be scheduled by the receiving MC or ESME for delivery. This field is only applicable if the short message is being forwarded to another MC. In this case it is the time at which the receiving MC should schedule the short message. Set to NULL if not scheduled.
validity_period	1 or 17	C-Octet String	The validity period of this message. This field is only applicable if this short message is being forwarded to another MC. In this case it specifies how long the receiving MC should retain the SM and continue trying to deliver it. Set to NULL if the current validity period is unavailable
registered_delivery	1	Integer	Indicator to signify if a MC delivery receipt or an SME acknowledgement is required.
replace_if_present_flag	1	Integer	Flag indicating if the submitted message should replace an existing message.
data_coding	1	Integer	Defines the encoding scheme of the short message user data.
sm_length	1	Integer	Length in octets of the short_message user data.
short_message	Variable 0-255	Octet String	Up to 255 octets of short message user data. The exact physical limit for short_message size may vary according to the underlying network **Note:** this field is superceded by the *message_payload* TLV if specified. Applications which need to send messages longer than 255 octets should use the *message_payload* TLV. In this case the *sm_length* field should be set to zero
Message Delivery TLVs	Variable	TLV	

deliver_sm_resp Syntax

SMPP PDU Name	Size oc-tets	Type	Description
command_length	4	Integer	Set to overall length of PDU.
command_id	4	Integer	0x80000005
command_status	4	Integer	Indicates outcome of *deliver_sm* request.
sequence_number	4	Integer	Set to sequence number of original *deliver_sm* PDU.
message_id	Variable max. 65	C-Octet String	This field is unused and should be set to NULL
Message Delivery Response TLVs	Variable	TLV	

3.3.7 Message Cancel Operations

Ancillary submission operations provide additional management of messages submitted by ESMEs. This includes cancellation, querying and replacement of messages.
This command is issued by the ESME to cancel one or more previously submitted short messages that are pending delivery. The command may specify a particular message to cancel, or all messages matching a pa ticular source, destination and service_type.
If the message_id is set to the ID of a previously submitted message, then provided the source address supplied by the ESME matches that of the stored message, that message will be cancelled.
If the message_id is NULL, all outstanding undelivered messages with matching source and destination addresses (and service_type if specified) are cancelled.
Where the original submit_sm, data_sm or submit_multi ‘source address’ is defaulted to NULL, then the source address in the cancel_sm command should also be NULL.

cancel_sm Syntax

SMPP PDU Name	Size oc-tets	Type	Description
command_length	4	Integer	Set to overall length of PDU..
command_id	4	Integer	0x00000008
command_status	4	Integer	0x00000000

sequence_number	4	Integer	Set to a unique sequence number. The associated *cancel_sm_resp* PDU will echo the same sequence number.
service_type	Variable max. 6	C-Octet String	Set to indicate SMS Application service, if cancellation of a group of application service messages is desired. Otherwise set to NULL.
message_id	Variable max. 65	C-Octet String	Message ID of the message to be cancelled. This must be the MC assigned Message ID of the original message. Set to NULL if cancelling a group of messages.
source_addr_ton	1	Integer	Type of Number of message originator. This is used for verification purposes, and must match that supplied in the original message submission request PDU. If not known, set to NULL.
source_addr_npi	1	Integer	Numbering Plan Identity of message originator. This is used for verification purposes, and must match that supplied in the original message submission request PDU. If not known, set to NULL.
source_addr	Variable max. 21	C-Octet String	Source address of message(s) to be cancelled. This is used for verification purposes, and must match that supplied in the original message submission request PDU(s). If not known, set to NULL.
dest_addr_ton	1	Integer	Type of number of destination SME address of the message(s) to be cancelled. This is used for verification purposes, and must match that supplied in the original message submission request PDU (e.g. *submit_sm*). May be set to NULL when the *message_id* is provided.
dest_addr_npi	1	Integer	Numbering Plan Indicator of destination SME address of the message(s) to be cancelled. This is used for verification purposes, and must match that supplied in the original message submission request PDU. May be set to NULL when the *message_id* is provided.
destination_addr	Variable max. 21	C-Octet String	Destination address of message(s) to be cancelled. This is used for verification purposes, and must match that supplied in the original message submission request PDU. May be set to NULL when the *message_id* is provided.

cancel_sm_resp Syntax

The *cancel_sm_resp* PDU is used to reply to a *cancel_sm* request. It comprises the SMPP message header only.

SMPP PDU Name	Size octets	Type	Description

command_length	4	Integer	Set to overall length of PDU.
command_id	4	Integer	0x80000008
command_status	4	Integer	Indicates outcome of *cancel_sm* request.
sequence_number	4	Integer	Set to sequence number of original *cancel_sm* PDU.

3.3.8 Message Query Operations

This command is issued by the ESME to query the status of a previously submitted short message.
The matching mechanism is based on the MC assigned message_id and source address. Where the original submit_sm, data_sm or submit_multi 'source address' was defaulted to NULL, then the source address in the query_sm command should also be set to NULL.

query_sm Syntax

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Set to overall length of PDU..
command_id	4	Integer	0x00000003
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a unique sequence number. The associated *query_sm_resp* PDU will echo the same sequence number.
message_id	Variable max. 65	C-Octet String	Message ID of the message whose state is to be queried. This must be the MC assigned Message ID allocated to the original short message when submitted to the MC by the *submit_sm*, *data_sm* or *submit_multi* command, and returned in the response PDU by the MC.
source_addr_ton	1	Integer	Type of Number of message originator. This is used for verification purposes, and must match that supplied in the original request PDU (e.g. *submit_sm*). If not known, set to NULL.
source_addr_npi	1	Integer	Numbering Plan Identity of message originator. This is used for verification purposes, and must match that supplied in the original message submission request PDU. If not known, set to NULL.

source_addr	Variable max. 21	C-Octet String	Address of message originator. This is used for verification purposes, and must match that supplied in the original request PDU (e.g. *submit_sm*). If not known, set to NULL.

query_sm_resp Syntax

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Set to overall length of PDU.
command_id	4	Integer	0x80000003
command_status	4	Integer	Indicates outcome of *query_sm* request.
sequence_number	4	Integer	Set to sequence number of original *query_sm* PDU.
message_id	Variable max. 65	C-Octet String	MC Message ID of the message whose state is being queried.
final_date	1 or 17	C-Octet String	Date and time when the queried message reached a final state. For messages, which have not yet reached a final state, this field will contain a single NULL octet.
message_state	1	Integer	Specifies the status of the queried short message.
error_code	1	Integer	Where appropriate this holds a network error code defining the reason for failure of message delivery. The range of values returned depends on the underlying telecommunications network.

3.3.9 Message Replace Operations

This command is issued by the ESME to replace a previously submitted short message that is pending delivery. The matching mechanism is based on the message_id and source address of the original message.
Where the original submit_sm 'source address' was defaulted to NULL, then the source address in the replace_sm command should also be NULL.

replace_sm Syntax

SMPP PDU Name	Size octets	Type	Description
command_length	4	Integer	Set to overall length of PDU..

command_id	4	Integer	0x00000007
command_status	4	Integer	0x00000000
sequence_number	4	Integer	Set to a unique sequence number. The associated *query_sm_resp* PDU will echo the same sequence number.
message_id	Variable max. 65	C-Octet String	Message ID of the message to be replaced. This must be the MC assigned Message ID allocated to the original short message when submitted to the MC by the *submit_sm*, *data_sm* or *submit_multi* command, and returned in the response PDU by the MC.
source_addr_ton	1	Integer	Type of Number of message originator. This is used for verification purposes, and must match that supplied in the original request PDU (e.g. *submit_sm*). If not known, set to NULL.
source_addr_npi	1	Integer	Numbering Plan Indicator for source address of original message. If not known, set to NULL (Unknown).
source_addr	Variable max. 21	C-Octet String	Address of SME, which originated this message. If not known, set to NULL (Unknown).
schedule_delivery_time	1 or 17	C-Octet String	New scheduled delivery time for the short message. Set to NULL to preserve the original scheduled delivery time
validity_period	1 or 17	C-Octet String	New expiry time for the short message. Set to NULL to preserve the original validity period setting.
registered_delivery	1	Integer	Indicator to signify if a MC delivery receipt, user/manual or delivery ACK or intermediate notification is required.
sm_default_msg_id	1	Integer	Indicates the short message to send from a list of pre- defined ('canned') short messages stored on the MC. If not using a MC canned message, set to NULL.
sm_length	1	Integer	Length in octets of the short_message user data.
short_message	Variable 0-255	Octet String	Up to 255 octets of short message user data. The exact physical limit for short_message size may vary according to the underlying network **Note:** this field is superceded by the *message_payload* TLV if specified. Applications which need to send messages longer than 255 octets should use the *message_payload* TLV. In this case the *sm_length* field should be set to zero
Message Replacement Request TLVs	Variable	TLV	

replace_sm_resp Syntax

SMPP PDU Name	Size oc-tets	Type	Description
command_length	4	Integer	Set to overall length of PDU.
command_id	4	Integer	0x80000007
command_status	4	Integer	Indicates outcome of *replace_sm* request.
sequence_number	4	Integer	Set to sequence number of original *replace_sm* PDU.
Message_payload	Variable	TLV	Contains the extended short message user data. Up to 64K octets can be transmitted. Note: The short message data should be inserted in either the *short_message* or *message_payload* fields. Both fields should not be used simultaneously. The *sm_length* field should be set to zero if using the *message_payload* parameter.

3.3.10 Command Status Error Codes

The command_status field of a SMPP message response indicates the success or failure of a SMPP request. It is relevant only in the SMPP response message and should be set to NULL in SMPP request messages.
The SMPP Error status codes are returned by the MC in the command_status field of the SMPP message header and in the error_status_code field of a submit_multi_resp message.
For detailed SMPP status and error codes information see SMPP specification in chapter 4.7.6 [1].

Attention:
For the CP developers of the SMPP application, it is essential that they recognize the response messages of the submitted messages. Also after receiving an error code from the SMSC, their application must have a logic implemented to properly handle the error.
For this, we could define several classes of error codes:

- **persistent errors** (not_to_be_retried)
  These errors will simply occur again, if retried without changing anything. Most errors are of this type.
  Examples:
  - ESME_RINVMSGLEN (Message Length is invalid)
  - ESME_RINVCMDLEN (Command Length is invalid)
  - ESME_RINVCMDID (Invalid Command ID)
  - ESME_RINVBNDSTS (Incorrect BIND Status for given command)
  - ESME_RALYBND (ESME Already in Bound State)
  - ESME_RINVSRCADR (Invalid Source Address)
  - ESME_RINVDSTADR (Invalid Destination Address)
  - ESME_RINVPASWD (Invalid Password)
  - ESME_RCANCELFAIL (Cancel SM Failed)
  - ESME_RINVSRCTON (Invalid Source address TON)
  - ESME_RINVSRCNPI (Invalid Source address NPI)
  - ESME_RINVTLVSTREAM (Error in the optional part of the PDU Body)

- **non-persistent errors** (may_be_retried_after a pause)
  The receiving system (SMSC) is in an overload state (in general or for the current connection) or parts of the connected SMSC are temporarily unavailable.
  The traffic must be throttled by waiting a while before a retry attempt.
  Examples:
  - ESME_RINVMSGID (Bind Failed, could also be a persistent error)
  - ESME_RMSGQFUL (Message Queue Full)
  - ESME_RTHROTTLED (Throttling error, ESME has exceeded allowed message limits)
  - ESME_RSERTYPUNAVAIL (Specified service_type is unavailable)
  - ESME_RUNKNOWNERR (Unknown Error)
  - ESME_RSYSERR (System Error)

  If a command retry is considered (without disconnecting and reconnecting), it should always be delayed and limited to a fixed number of retries (do not retry indefinitely)

3.3.11 SMPP Examples

The use of SMPP examples is limited because it is based on hex code. But here some examples:
Login with SMPP:
00 00 00 23 00 00 00 09 00 00 00 00 00 00 00 01 31 32 33 34 00 74 65 73 74 31 32 33 34 00 00 34 00 00 00

MO to SMPP (Short-ID):
00 00 00 45 00 00 00 05 00 00 00 00 00 00 00 0A 00 02 01 37 39 31 32 33 34 35 36 37 00 04 09 31 32 33 34 00 00 00 00 00 00 00 00 00 0D 54 65 73 74 20 53 77 69 73 73 63 6F 6D 00 0E 00 01 01 00 06 00 01 01

MO to SMPP (Global Reply with long ID):
00 00 00 4A 00 00 00 05 00 00 00 00 00 00 00 0D 00 02 01 37 39 31 32 33 34 35 36 37 00 02 01 37 39 38 30 37 37 39 35 39 00 00 00 00 00 00 00 00 00 00 0D 54 65 73 74 20 53 77 69 73 73 63 6F 6D 00 0E 00 01 01 00 06 00 01 01