

Short Message Peer to Peer Protocol Specification v3.4

Document Version:- 12-Oct-1999 Issue 1.2

Short Message Peer to Peer Protocol Specification v3.4

12-Oct-1999 Issue 1.2

© 1999 SMPP Developers Forum.

COPYRIGHT

All rights reserved. This document or any part thereof may not, without the prior written consent of SMPP Developers Forum, be copied, reprinted or reproduced in any material form including, but without prejudice to the foregoing and not by way of exception photocopying, transcribing, transmitting or storing in any medium or translating into any language, in any form or by any means, including but not limited to, electronic, mechanical, xerographic, optical, magnetic, digital or other methodology.

DISCLAIMER

WHILST THE GREATEST CARE HAS BEEN TAKEN TO ENSURE THE ACCURACY OF THE INFORMATION AND DATA CONTAINED HEREIN, SMPP DEVELOPERS FORUM DOES NOT WARRANT THE ACCURACY OR SUITABILITY OF SAME FOR ANY SPECIFIC USE. SMPP DEVELOPERS FORUM EXPRESSLY DISCLAIMS ALL AND ANY LIABILITY TO ANY PERSON, WHETHER A PURCHASER OR OTHERWISE, IN RESPECT OF ANY CONSEQUENCES OF ANYTHING DONE OR OMITTED TO BE DONE BY ANY SUCH PERSON IN PARTIAL OR TOTAL RELIANCE UPON THE WHOLE OR ANY PART OF THE CONTENTS OF THIS PUBLICATION OR ANY DERIVATIVE THEREOF.

THE INFORMATION CONTAINED HEREIN IS BELIEVED TO BE ACCURATE AND RELIABLE. HOWEVER, SMPP DEVELOPERS FORUM ACCEPTS NO RESPONSIBILITY FOR IT'S USE BY ANY MEANS OR IN ANY WAY WHATSOEVER. SMPP DEVELOPERS FORUM SHALL NOT BE LIABLE FOR ANY EXPENSES, COSTS OR DAMAGE THAT MAY RESULT FROM THE USE OF THE INFORMATION CONTAINED HOWSOEVER ARISING IN THIS DOCUMENT OR ANY DERIVATIVE THEREOF.

NOTE 1:THE INFORMATION CONTAINED IN THE WITHIN DOCUMENT AND ANY DERIVATIVE THEREOF IS SUBJECT TO CHANGE WITHOUT NOTICE.

NOTE 2:THE CORPORATE NAME OF SMPP DEVELOPERS FORUM IS NORTHGROVE LIMITED, COMPANY NUMBER 309113, REGISTERED OFFICE GARDNER HOUSE, WILTON PLACE, DUBLIN 2.

Errata

Erratum	Description of Correction to address Erratum	Change Request Reference
<p>In the SMPP Protocol Specification v3.4. version 30-July-1999 Issue 1.1 section 4.1.5 “Bind_Transceiver” the <i>interface_version</i> field was inadvertently not included in the <i>bind_transceiver</i> PDU.</p>	<p>The erratum was corrected in the SMPP Protocol Specification v3.4 version 12-Oct-1999 Issue 1.2 as follows:</p> <p>In section 4.1.5 “Bind_Transceiver” the <i>interface_version</i> field was added as a mandatory field to the <i>bind_transceiver</i> PDU.</p> <p>Since it is a mandatory field all implementations of the SMPP Protocol Specification v3.4 must include the <i>interface_version</i> field when using the <i>bind_transceiver</i> PDU.</p>	<p>SMPPV3.4-05Oct99-01</p>

Table of Contents

1.	Introduction	8
1.1	SMPP Overview	8
1.2	Scope	9
1.3	Glossary	10
1.4	References	11
2.	SMPP Protocol Overview	12
2.1	SMPP Protocol Definition	13
2.2	SMPP Session Description	14
2.2.1	Outbind	16
2.3	SMPP PDUs	17
2.4	SMPP Network Layer Connections	19
2.5	SMPP messages sent from ESME to SMSC	20
2.5.1	SMPP Message Response from SMSC to ESME	20
2.5.2	Typical SMPP session sequence - ESME Transmitter	21
2.6	SMPP messages sent from SMSC to ESME	23
2.6.1	SMPP Message Response from ESME to SMSC	23
2.6.2	Typical SMPP session sequence - ESME Receiver	24
2.7	Duplex message exchange between an SMSC and an ESME	26
2.7.1	Typical SMPP session sequence - ESME Transceiver	27
2.8	SMPP Error Handling	29
2.9	SMPP Timers	29
2.10	Message Modes	30
2.10.1	Store and Forward Message Mode	30
2.10.2	Datagram Message Mode	32
2.10.3	Transaction Message Mode	33
2.11	Message Types	34
3.	SMPP PDU Type and Format Definitions	36
3.1	SMPP PDU - Type Definitions	36
3.1.1	SMPP Parameter Field Size Notation	37
3.2	SMPP PDU Format - Overview	38
3.2.1	SMPP PDU Layout	39
3.2.2	SMPP PDU Length	41
3.2.3	SMPP Message length and extended message length	41
3.2.4	Optional Parameters	42
3.2.4.1	Optional Parameter Format	42
3.3	Guidelines for SMPP Forward Compatibility	43
3.4	Guidelines for SMPP Backward Compatibility	44
4.	SMPP PDU Definition	45
4.1	“BIND” Operation	45
4.1.1	“BIND_TRANSMITTER” Syntax	46
4.1.2	“BIND_TRANSMITTER_RESP” Syntax	47
4.1.3	“BIND_RECEIVER” Syntax	48
4.1.4	“BIND_RECEIVER_RESP”	50
4.1.5	“BIND_TRANSCEIVER” Syntax	51
4.1.6	“BIND_TRANSCEIVER_RESP”	53

4.1.7	“OUTBIND” Operation	54
4.1.7.1	“OUTBIND” Syntax	54
4.2	“UNBIND” Operation	55
4.2.1	“UNBIND”	56
4.2.2	“UNBIND_RESP”	56
4.3	“GENERIC_NACK” PDU	57
4.3.1	“GENERIC_NACK” Syntax	57
4.4	“SUBMIT_SM” Operation	58
4.4.1	“SUBMIT_SM” Syntax	59
4.4.1.1	Source and Destination Addressing	66
4.4.1.2	Message Replace operation in “SUBMIT_SM”	66
4.4.2	“SUBMIT_SM_RESP”	67
4.5	“SUBMIT_MULTI” Operation	68
4.5.1	“SUBMIT_MULTI” Syntax	69
4.5.1.1	Destination Address definition	75
4.5.1.2	Distribution List (DL) definition	75
4.5.2	“SUBMIT_MULTI_RESP” Syntax	76
4.5.2.1	Unsuccessful deliveries	77
4.6	“DELIVER_SM” Operation	78
4.6.1	“DELIVER_SM” Syntax	79
4.6.2	“DELIVER_SM_RESP” Syntax	85
4.7	“DATA_SM” Operation	86
4.7.1	“DATA_SM” Syntax	87
4.7.2	“DATA_SM_RESP” Syntax	93
4.8	“QUERY_SM” Operation	94
4.8.1	“QUERY_SM” Syntax	95
4.8.2	“QUERY_SM_RESP” Syntax	96
4.9	“CANCEL_SM” Operation	97
4.9.1	“CANCEL_SM” Syntax	98
4.9.2	“CANCEL_SM_RESP” Syntax	100
4.10	“REPLACE_SM” Operation	101
4.10.1	“REPLACE_SM” Syntax	102
4.10.2	“REPLACE_SM_RESP” Syntax	104
4.11	“ENQUIRE_LINK” Operation	105
4.11.1	“ENQUIRE_LINK” Syntax	106
4.11.2	“ENQUIRE_LINK_RESP” Syntax	106
4.12	“ALERT_NOTIFICATION” Operation	107
4.12.1	“ALERT_NOTIFICATION” Syntax	108
5.	SMPP Parameter Definition	109
5.1	Command Header Parameters	109
5.1.1	command_length	109
5.1.2	command_id	109
5.1.2.1	SMPP Command set	110
5.1.3	command_status	112
5.1.4	sequence_number	115
5.2	Mandatory SMPP Parameters	116
5.2.1	system_id	116
5.2.2	password	116
5.2.3	system_type	116
5.2.4	interface_version	116

5.2.5	addr_ton, source_addr_ton, dest_addr_ton, esme_addr_ton.....	117
5.2.6	addr_npi, source_addr_npi, dest_addr_npi, esme_addr_npi.....	118
5.2.7	address_range.....	118
5.2.8	source_addr.....	119
5.2.9	destination_addr.....	119
5.2.10	esme_addr.....	119
5.2.11	service_type.....	120
5.2.12	esm_class.....	121
5.2.13	protocol_id.....	123
5.2.14	priority_flag.....	123
5.2.15	schedule_delivery_time.....	124
5.2.16	validity_period.....	124
5.2.17	registered_delivery.....	124
5.2.18	replace_if_present_flag.....	125
5.2.19	data_coding.....	126
5.2.20	sm_default_msg_id.....	127
5.2.21	sm_length.....	128
5.2.22	short_message.....	128
5.2.23	message_id.....	128
5.2.24	number_of_dests.....	128
5.2.25	dest_flag.....	129
5.2.26	no_unsuccess.....	129
5.2.27	dl_name.....	129
5.2.28	message_state.....	130
5.3	SMPP Optional Parameter Description.....	131
5.3.1	Optional Parameter Tag Identifiers.....	131
5.3.2	SMPP Optional Parameter Tag definitions.....	132
5.3.2.1	dest_addr_subunit.....	134
5.3.2.2	source_addr_subunit.....	134
5.3.2.3	dest_network_type.....	135
5.3.2.4	source_network_type.....	135
5.3.2.5	dest_bearer_type.....	136
5.3.2.6	source_bearer_type.....	136
5.3.2.7	dest_telematics_id.....	137
5.3.2.8	source_telematics_id.....	137
5.3.2.9	qos_time_to_live.....	138
5.3.2.10	payload_type.....	138
5.3.2.11	additional_status_info_text.....	139
5.3.2.12	receipted_message_id.....	139
5.3.2.13	ms_msg_wait_facilities.....	140
5.3.2.14	privacy_indicator.....	141
5.3.2.15	source_subaddress.....	142
5.3.2.16	dest_subaddress.....	143
5.3.2.17	user_message_reference.....	143
5.3.2.18	user_response_code.....	144
5.3.2.19	language_indicator.....	144
5.3.2.20	source_port.....	145
5.3.2.21	destination_port.....	145
5.3.2.22	sar_msg_ref_num.....	146
5.3.2.23	sar_total_segments.....	147
5.3.2.24	sar_segment_seqnum.....	147

5.3.2.25	sc_interface_version	148
5.3.2.26	display_time.....	148
5.3.2.27	ms_validity	149
5.3.2.28	dpf_result	149
5.3.2.29	set_dpf.....	150
5.3.2.30	ms_availability_status.....	151
5.3.2.31	network_error_code.....	152
5.3.2.32	message_payload	153
5.3.2.33	delivery_failure_reason	153
5.3.2.34	more_messages_to_send.....	154
5.3.2.35	message_state	154
5.3.2.36	callback_num.....	155
5.3.2.37	callback_num_pres_ind.....	156
5.3.2.38	callback_num_atag	157
5.3.2.39	number_of_messages.....	158
5.3.2.40	sms_signal.....	158
5.3.2.41	alert_on_message_delivery.....	159
5.3.2.42	its_reply_type	159
5.3.2.43	its_session_info.....	160
5.3.2.44	ussd_service_op.....	161
6.	Network Implementation.....	162
6.1	Network Error Codes	162
6.2	Maximum Message Length.....	162
7.	General Definitions	163
7.1	Time Definitions	163
7.1.1	Time Format.....	163
7.1.1.1	Absolute Time format.....	163
7.1.1.2	Relative Time Format	164
7.2	Timer Definitions.....	165
	Appendix A	166
	Appendix B	167
	Appendix C	169

1. Introduction

1.1 SMPP Overview

The Short Message Peer to Peer (SMPP) protocol is an open, industry standard protocol designed to provide a flexible data communications interface for transfer of short message data between a Message Center, such as a Short Message Service Centre (SMSC), GSM Unstructured Supplementary Services Data (USSD) Server or other type of Message Center and a SMS application system, such as a WAP Proxy Server, EMail Gateway or other Messaging Gateway.

Note: For sake of brevity, the term SMSC will be used throughout this document to describe any SMPP ‘server’ entity to which an SMPP ‘client’, termed an External Short Message Entity (ESME), can be connected.

SMPP Release v3.4 supports Digital Cellular Network technologies including:-

- GSM
- IS-95 (CDMA)
- ANSI-136 (TDMA)
- iDEN

Using the SMPP protocol, an SMS application system called the ‘External Short Message Entity’ (ESME) may initiate an application layer connection with an SMSC over a TCP/IP or X.25 network connection and may then send short messages and receive short messages to and from the SMSC respectively. The ESME may also query, cancel or replace short messages using SMPP.

SMPP supports a full featured set of two-way messaging functions such as:-

- Transmit messages from an ESME to single or multiple destinations via the SMSC
- An ESME may receive messages via the SMSC from other SME’s (e.g. mobile stations).
- Query the status of a short message stored on the SMSC
- Cancel or replace a short message stored on the SMSC
- Send a registered short message (for which a ‘delivery receipt’ will be returned by the SMSC to the message originator)
- Schedule the message delivery date and time
- Select the message mode, i.e. datagram or store and forward
- Set the delivery priority of the short message
- Define the data coding type of the short message
- Set the short message validity period
- Associate a service type with each message e.g. voice mail notification

1.2 Scope

This document defines Version 3.4 of the SMPP protocol and specifies the command and response format to be used when implementing an SMPP v3.4 protocol interface.

It is intended for designers and implementers of an SMPP v3.4 interface between an SMSC and an External Short Message Entity (ESME), as illustrated in the following diagram.

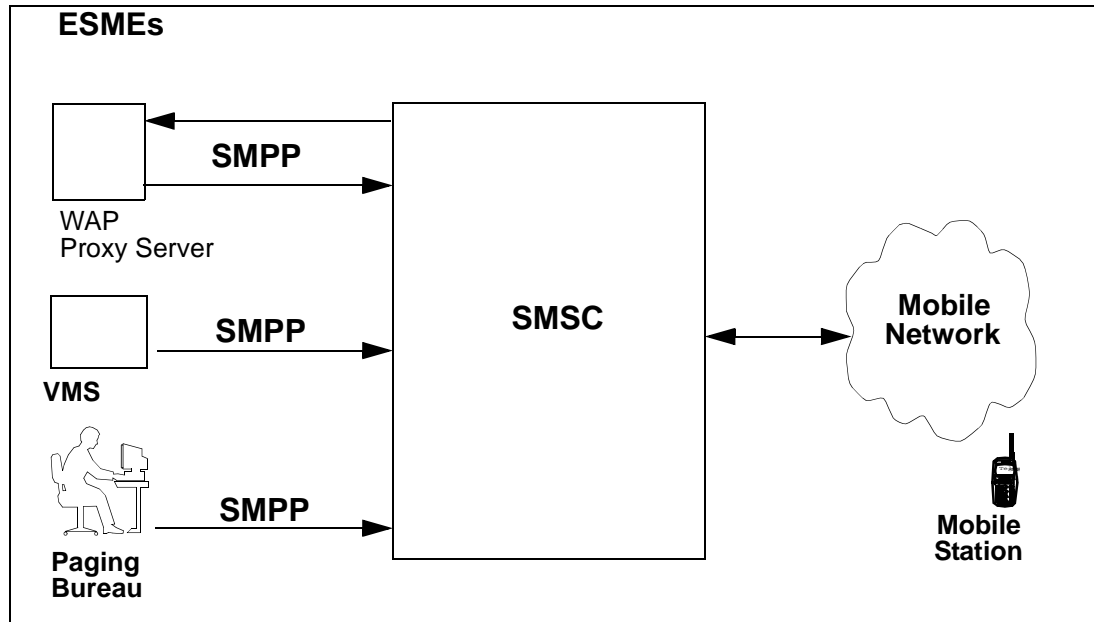


Figure 1-1: Context of SMPP in a Mobile Network

1.3 Glossary

ACK	Acknowledgement
API	Application Programming Interface
CDR	Call Detail Record
ESME	External Short Message Entity. Refer to note[1]
ETSI	European Telecommunications Standards Institute
HEADER	Leading portion of the SMPP message, common to all SMPP PDUs
MB	Message Bureau - This is typically an operator message bureau.
MSB	Most Significant Byte
MSC	Mobile Switching Centre
MS	Mobile Station
MWI	Message Waiting Indication
NACK	Negative Acknowledgement
NSAP	Network Service Access Point
PDU	Protocol Data Unit
PSSD	Process Unstructured Supplementary Services Data
PSSR	Process Unstructured Supplementary Services Request
SME	Short Message Entity
SMSC	Short Message Service Centre
SMPP	Short Message Peer to Peer Protocol
UDHI	User Data Header Indicator
URL	Uniform Resource Locator
USSN	Unstructured Supplementary Services Notification
USSR	Unstructured Supplementary Services Request
VMA	VoiceMail Alert
VPS	Voice Processing System
TIA	Telecommunications Industry Association
WAP	Wireless Application Protocol (http://www.wapforum.org)
WCMP	Wireless Control Message Protocol
WDP	Wireless Datagram Protocol

Note 1: In the context of this document ESME refers to such external sources and sinks of short messages as Voice Processing Systems, WAP Proxy Servers or Message Handling computers. It specifically excludes SMEs which are located within the Mobile Network, i.e., a mobile station (MS).

1.4 References

Ref.	Document Title	Document Number	Version Number
[GSM 03.40]	Technical Realisation of the Short Message Service Point to Point	GSM 03.40 http://www.etsi.fr	v5.7.1
[GSM 03.38]	“Digital Cellular telecommunications system (Phase 2+); Alphabets and language specific information”.	[GSM 03.38] http://www.etsi.fr	v5.5.1 Sept. ‘97
[GSM MAP 09.02]	GSM Mobile Application Part	[GSM MAP 09.02] http://www.etsi.fr	v5.11.0
[IS637]	Short Message Service for Spread Spectrum Systems	TIA/EIA/IS-637-A	Rev A
[TSAR]	Teleservice Segmentation and Reassembly (TSAR)	TIA/EIA-136-620	Rev 0
[CMT-136]	Short Message Service - Cellular Messaging Teleservice	TIA/EIA-136-710-A	Rev A
[GUTS]	General UDP Transport Service (GUTS)	TIA/EIA-136-750	Rev 0
[WAPARCH]	Wireless Application Protocol Architecture Specification	WAP Forum http://www.wapforum.org	Version 30-Apr.-1998
[WCMP]	Wireless Control Message Protocol Specification	WAP Forum http://www.wapforum.org	Version 12-June-1998
[WDP]	Wireless Datagram Protocol Specification	WAP Forum http://www.wapforum.org	Version 10-Feb.-1999
[ITUT X.213]	Open Systems Interconnection - Network Service Definition	[ITUT X.213]	11/95
[KOR ITS]	PCS operators common standards for handset-SMS functionalities	PCS standardization committee PCS-SMS-97-05-28	1.06 Rev 99-04-30

2. SMPP Protocol Overview

Short Message Peer to Peer (SMPP) protocol is an open message-transfer protocol that enables short message entities (SMEs) outside the mobile network to interface with an SMSC. Non-mobile entities that submit messages to, or receive messages from an SMSC are known as External Short Message Entities (ESMEs).

The SMPP protocol defines:

- a set of operations for the exchange of short messages between an ESME and an SMSC
- the data that an ESME application must exchange with an SMSC during SMPP operations.

Subscribers to an SMS-capable Cellular Network may receive short messages on a Mobile Station (MS) from one or more ESMEs. The means whereby these messages arrive at the ESME via an interface other than SMPP is beyond the scope of this document. However, examples of such ESME applications include:-

- Voicemail alerts originating from a VPS (Voice Processing System), indicating voice messages at a customer's mailbox.
- Numeric and alphanumeric paging services
- Information services. For example, an application that enables mobile subscribers to query currency rates or share-price information from a database or the WWW and have it displayed as a short message on the handsets.
- Calls directly dialled or diverted to a message-bureau operator, who forwards the message to the SMSC, for onward delivery to a subscriber's handset.
- A fleet management application that enables a central station to use the SMSC to determine the location of its service vehicles and notify the closest vehicle of a service request in their area.
- Telemetry applications. For example, a house-hold meter that transmits a short message to a utility company's billing system to automatically record customer usage.
- WAP Proxy Server. A WAP Proxy Server acts as the WAP gateway for wireless internet applications. A WAP Proxy Server may select an SMS or USSD bearer for sending WDP datagrams to and receiving WDP datagrams from a mobile station.

2.1 SMPP Protocol Definition

SMPP is based on the exchange of request and response protocol data units (PDUs) between the ESME and the SMSC over an underlying TCP/IP or X.25 network connection. The SMPP protocol defines:

- a set of operations and associated Protocol Data Units (PDUs) for the exchange of short messages between an ESME and an SMSC
- the data that an ESME application can exchange with an SMSC during SMPP operations.

Note* Every SMPP operation must consist of a request PDU and associated response PDU. The receiving entity must return the associated SMPP response to an SMPP PDU request.

* The only exception to this rule is

- the *alert_notification* PDU for which there is no response

The exchange of messages between an ESME and SMSC via SMPP may be categorised under three distinct groups of transactions as follows:

- messages sent from the ESME (Transmitter) to the SMSC
- messages sent from the SMSC to the ESME (Receiver)
- messages sent from the ESME (Transceiver) to the SMSC and messages sent from the SMSC to the ESME (Transceiver)

The following Figure 2-1 illustrates the above categories, which are explained in more detail in subsequent sections.

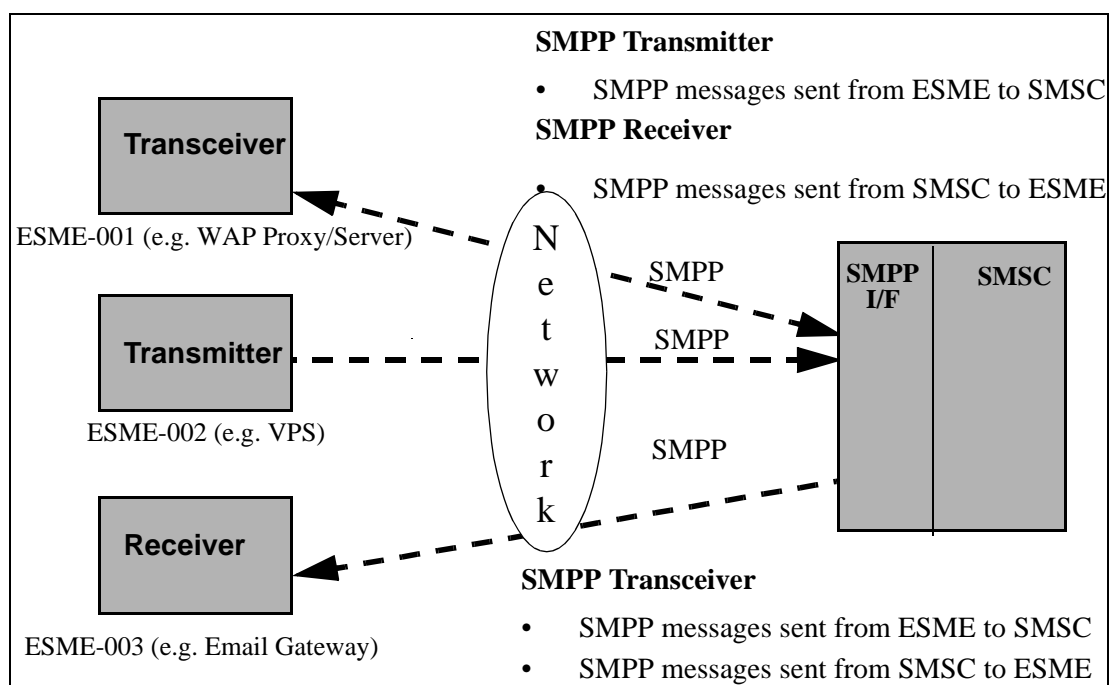


Figure 2-1: SMPP interface between SMSC and ESME

2.2 SMPP Session Description

An SMPP session between an SMSC and an ESME is initiated by the ESME first establishing a network connection with the SMSC and then issuing an SMPP Bind request to open an SMPP session. An ESME wishing to submit and receive messages is required to establish two network connections (TCP/IP or X.25) and two SMPP sessions (Transmitter and Receiver). Alternatively, in this version of the protocol an ESME may establish an SMPP Transceiver session over a single network connection.

During an SMPP session, an ESME may issue a series of requests to an SMSC and shall receive the appropriate responses to each request, from the SMSC. Likewise, the SMSC may issue SMPP requests to the ESME, which must respond accordingly.

The SMPP session may be defined in terms of the following possible states:

- **OPEN** (Connected and Bind Pending)
An ESME has established a network connection to the SMSC but has not yet issued a Bind request.

- **BOUND_TX**
A connected ESME has requested to bind as an ESME Transmitter (by issuing a *bind_transmitter* PDU) and has received a response from the SMSC authorising its bind request.

An ESME bound as a transmitter may send short messages to an SMSC for onward delivery to a Mobile Station or to another ESME. The ESME may also replace, query or cancel a previously submitted short message.

- **BOUND_RX**
A connected ESME has requested to bind as an ESME Receiver (by issuing a *bind_receiver* PDU) and has received a response from the SMSC authorising its Bind request.

An ESME bound as a receiver may receive short messages from an SMSC which may be originated by a mobile station, by another ESME or by the SMSC itself (for example an SMSC delivery receipt).

- **BOUND_TRX**
A connected ESME has requested to bind as an ESME Transceiver (by issuing a *bind_transceiver* PDU) and has received a response from the SMSC authorising its Bind request. An ESME bound as a Transceiver supports the complete set of operations supported by a Transmitter ESME and a Receiver ESME.

Thus an ESME bound as a transceiver may send short messages to an SMSC for onward delivery to a Mobile Station or to another ESME. The ESME may also receive short messages from an SMSC which may be originated by a mobile station, by another ESME or by the SMSC itself (for example an SMSC delivery receipt).

- **CLOSED** (Unbound and Disconnected)
An ESME has unbound from the SMSC and has closed the network connection. The SMSC may also unbind from the ESME.

2.2.1 Outbind

The purpose of the *outbind* operation is to allow the SMSC signal an ESME to originate a *bind_receiver* request to the SMSC. An example of where such a facility might be applicable would be where the SMSC had outstanding messages for delivery to the ESME.

An outbind SMPP session between an SMSC and an ESME can be initiated by the SMSC first establishing a network connection with the ESME.

Once a network connection has been established, the SMSC should bind to the ESME by issuing an “*outbind*” request. The ESME should respond with a “*bind_receiver*” request to which the SMSC will reply with a “*bind_receiver_resp*”.

If the ESME does not accept the outbind session (e.g. because of an illegal *system_id* or *password* etc.) the ESME should disconnect the network connection.

Once the SMPP session is established the characteristics of the session are that of a normal SMPP receiver session.

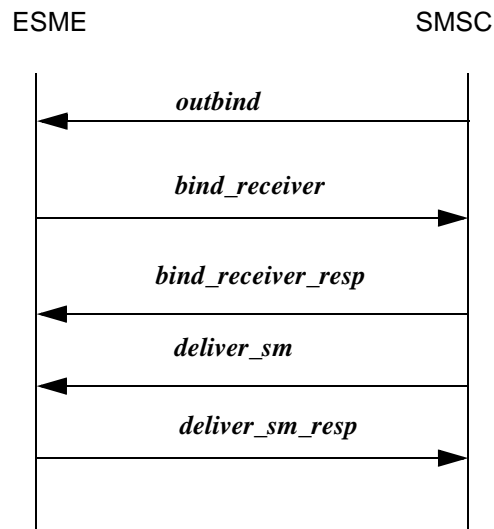


Figure 2-2: Sample Outbind Sequence

2.3 SMPP PDUs

The following table lists the SMPP PDU set and the context in which each PDU may be used:

SMPP PDU Name	Required SMPP Session State	Issued by ESME	Issued by SMSC
<i>bind_transmitter</i>	OPEN	Yes	No
<i>bind_transmitter_resp</i>	OPEN	No	Yes
<i>bind_receiver</i>	OPEN	Yes	No
<i>bind_receiver_resp</i>	OPEN	No	Yes
<i>bind_transceiver</i>	OPEN	Yes	No
<i>bind_transceiver_resp</i>	OPEN	No	Yes
<i>outbind</i>	OPEN	No	Yes
<i>unbind</i>	BOUND_TX BOUND_RX BOUND_TRX	Yes Yes Yes	Yes Yes Yes
<i>unbind_resp</i>	BOUND_TX BOUND_RX BOUND_TRX	Yes Yes Yes	Yes Yes Yes
<i>submit_sm</i>	BOUND_TX BOUND_TRX	Yes Yes	No No
<i>submit_sm_resp</i>	BOUND_TX BOUND_TRX	No No	Yes Yes
<i>submit_sm_multi</i>	BOUND_TX BOUND_TRX	Yes Yes	No No
<i>submit_sm_multi_resp</i>	BOUND_TX BOUND_TRX	No No	Yes Yes
<i>data_sm</i>	BOUND_TX BOUND_RX BOUND_TRX	Yes Yes Yes	Yes Yes Yes
<i>data_sm_resp</i>	BOUND_TX BOUND_RX BOUND_TRX	Yes Yes Yes	Yes Yes Yes
<i>deliver_sm</i>	BOUND_RX BOUND_TRX	No No	Yes Yes
<i>deliver_sm_resp</i>	BOUND_RX BOUND_TRX	Yes Yes	No No
<i>query_sm</i>	BOUND_TX BOUND_TRX	Yes Yes	No No
<i>query_sm_resp</i>	BOUND_TX BOUND_TRX	No No	Yes Yes

Table 2-1: SMPP PDU Summary List

SMPP PDU Name	Required SMPP Session State	Issued by ESME	Issued by SMSC
<i>cancel_sm</i>	BOUND_TX BOUND_TRX	Yes Yes	No No
<i>cancel_sm_resp</i>	BOUND_TX BOUND_TRX	No No	Yes Yes
<i>replace_sm</i>	BOUND_TX	Yes	No
<i>replace_sm_resp</i>	BOUND_TX	No	Yes
<i>enquire_link</i>	BOUND_TX BOUND_RX BOUND_TRX	Yes Yes Yes	Yes Yes Yes
<i>enquire_link_resp</i>	BOUND_TX BOUND_RX BOUND_TRX	Yes Yes Yes	Yes Yes Yes
<i>alert_notification</i>	BOUND_RX BOUND_TRX	No No	Yes Yes
<i>generic_nack</i>	BOUND_TX BOUND_RX BOUND_TRX	Yes Yes Yes	Yes Yes Yes

Table 2-1: SMPP PDU Summary List

2.4 SMPP Network Layer Connections

The underlying transport interface between the SMSC and ESME may be based on a TCP/IP or X.25 network connection.

SMPP is an application layer protocol and is not intended to offer transport functionality. It is therefore assumed that the underlying network connection will provide reliable data transfer from point to point including packet encoding, windowing, flow control and error handling.

Thus, at the SMPP level, the ESME and SMSC should treat the network connection as a reliable transport which manages delivery and receipt of SMPP PDUs.

The following diagram illustrates a generic SMPP interface implementation between an ESME and SMSC.

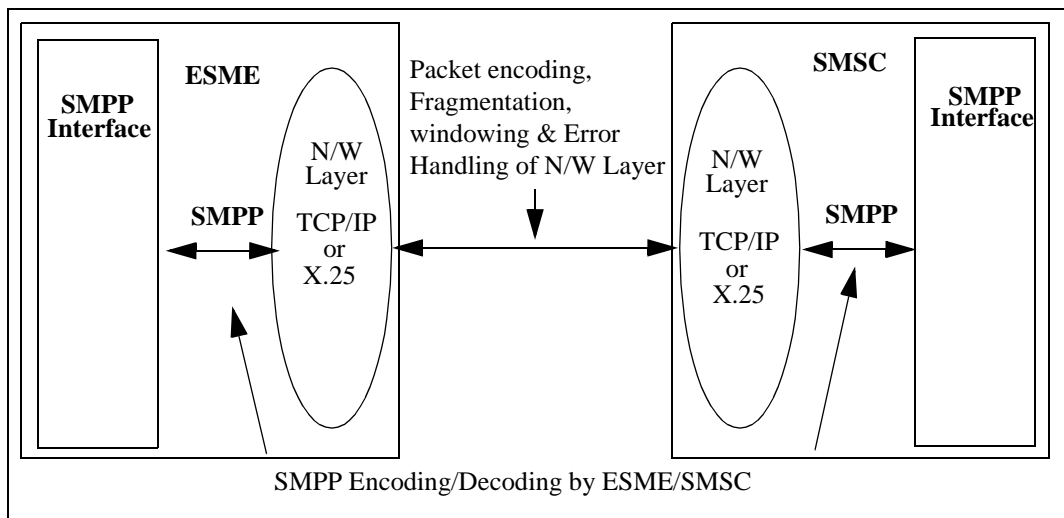


Figure 2-3: Model of SMPP SMSC-ESME Interface

If required, it is expected that the network layer at the sending entity will handle the segmentation of the SMPP PDUs for transmission as a series of fragmented packets over the network connection. Likewise, the network layer of the receiving entity, shall re-assemble a fragmented SMPP PDU before passing the entire SMPP PDU to the SMPP layer.

2.5 SMPP messages sent from ESME to SMSC

An ESME which sends short messages to an SMSC must be connected to the SMSC as an ESME Transmitter or an ESME Transceiver.

Examples of SMPP message Protocol Data Units (PDUs) which may be sent from an ESME transmitter to the SMSC include:

- *submit_sm*
- *data_sm*

In addition to submission of messages to the SMSC, an ESME may perform the following SMPP operations using the message identifier returned by the SMSC in the message acknowledgement:

- *query_sm* - Query the SMSC for the status of a previously submitted message
- *cancel_sm* - Cancel delivery of a previously submitted message
- *replace_sm* - Replace a previously submitted message

SMPP PDUs sent to the SMSC by an ESME must, when received, be acknowledged with a PDU response by the SMSC.

Refer to Table 2-1 for details on the SMPP operations which may be sent from an ESME to the SMSC.

2.5.1 SMPP Message Response from SMSC to ESME

The SMPP PDU response for a message submission to the SMSC will include a message identifier (which must be a unique handle assigned to that particular message) and a status which informs the ESME whether the submitted message is valid (i.e. accepted by the SMSC for onward delivery) or invalid. In the latter case, the SMSC will return an appropriate error status.

- *submit_sm_resp*
- *data_sm_resp*
- *query_sm_resp*
- *cancel_sm_resp*
- *replace_sm_resp*

2.5.2 Typical SMPP session sequence - ESME Transmitter

The following diagram illustrates a typical SMPP request/response sequence between an SMSC and an ESME bound as a Transmitter.

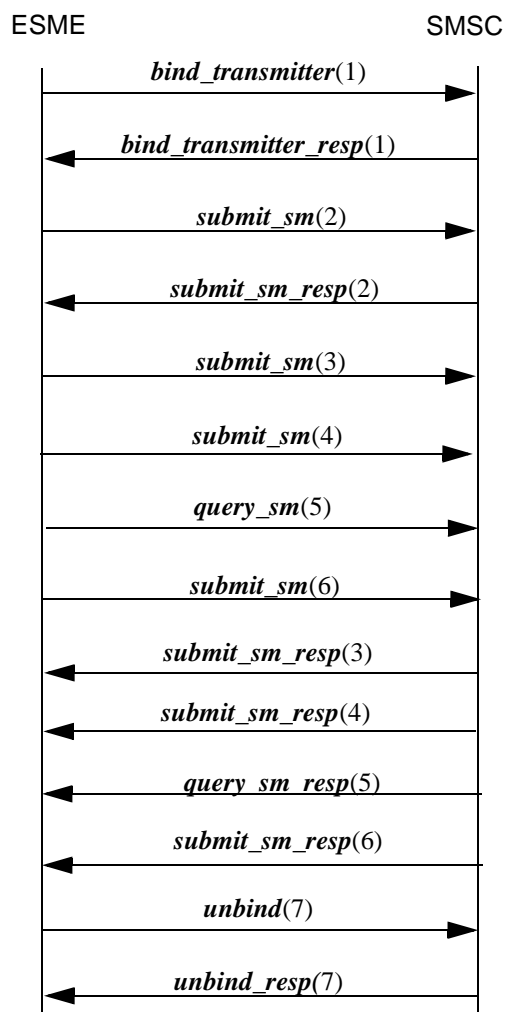


Figure 2-4: Typical SMPP request/response sequence for an ESME Transmitter

- The exchange of SMPP request and response PDUs between an ESME Transmitter and SMSC may occur synchronously or asynchronously as shown above. Thus an ESME may, if desired, send multiple requests to the SMSC, without synchronously waiting for the associated response PDUs.
- A series of successive SMPP requests issued asynchronously by an ESME (as denoted by the number in parentheses in Figure 2-4 above) must be followed shortly after by a series of associated responses from the SMSC.
- SMPP responses should be returned by the SMSC in the same order in which the original requests were received from the ESME. However this is not mandatory within SMPP and the ESME should be capable of handling responses received out of sequence.

- The ESME should return SMPP responses to the SMSC in the same order in which the original requests were received. The only relevant PDU response that an ESME Transmitter returns in a transmitter session is an *enquire_link_resp*.

Note: The maximum number of outstanding (i.e. unacknowledged) SMPP operations between an ESME and SMSC and vice versa is not specified explicitly in the SMPP Protocol Specification and will be governed by the SMPP implementation on the SMSC.

However, as a guideline it is recommended that no more than 10 (ten) SMPP messages are outstanding at any time.

2.6 SMPP messages sent from SMSC to ESME

The SMSC may deliver short messages to an ESME. In this case the ESME must be connected to the SMSC as an ESME Receiver or as an ESME Transceiver.

Typical applications in which an ESME would operate as an SMPP Receiver include:-

- an e-mail gateway accepting messages originated by mobile stations for onward delivery to internet e-mail boxes.
- The SMSC may also send a 'delivery receipt' to the ESME which contains a returned delivery status of a previously submitted short message.

Examples of SMPP message Protocol Data Units (PDUs) which may be sent from an SMSC to an ESME receiver include:

- *deliver_sm*
- *data_sm*

SMPP PDUs delivered to an ESME by the SMSC must be acknowledged with a SMPP PDU response by the ESME when received*.

* Exceptions to this rule are:

- the *alert_notification* PDU.

Refer to Table 2-1 for details on the SMPP operations which may be sent from an SMSC to an ESME.

2.6.1 SMPP Message Response from ESME to SMSC

The SMPP PDU response from an ESME Receiver must preserve the PDU transaction identifier (contained in the *sequence_number* parameter) sent by the SMSC. The response must also include the command status which informs the SMSC whether the message delivered to the ESME was valid (i.e. accepted by the ESME) or invalid. In the latter case, the ESME should return an appropriate SMPP error status.

Examples of SMPP message responses which may be sent from an ESME receiver to the SMSC include:

- *deliver_sm_resp*
- *data_sm_resp*

2.6.2 Typical SMPP session sequence - ESME Receiver

The following diagram illustrates a typical SMPP request/response sequence between an SMSC and an ESME bound as a Receiver.

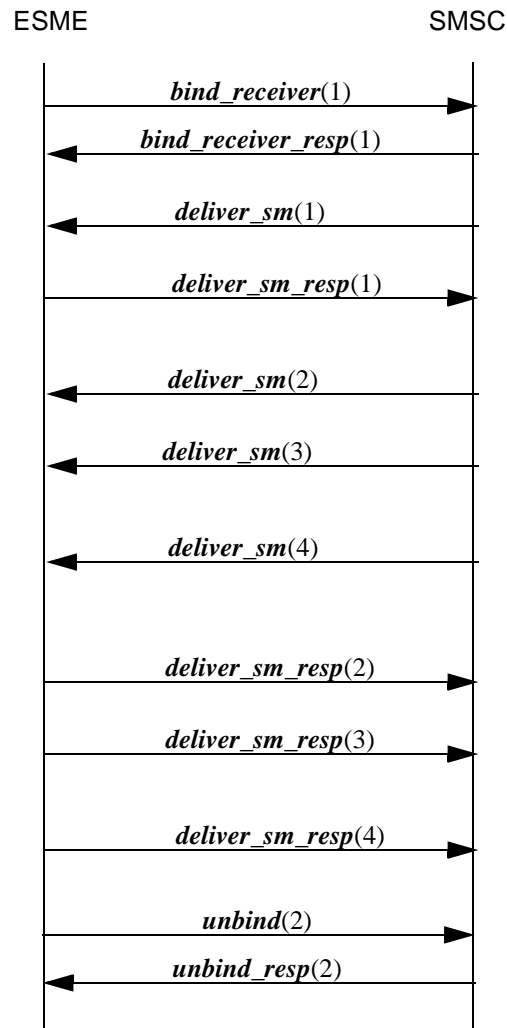


Figure 2-5: Typical SMPP request/response sequence for an ESME Receiver

- The exchange of SMPP request and response PDUs between an SMSC and ESME Receiver may be implemented synchronously or asynchronously as shown above. Thus the SMSC may send multiple *deliver_sm* requests to the ESME, without synchronously waiting for the associated response PDUs.
- A series of successive SMPP requests issued asynchronously by an SMSC (as denoted by the number in parentheses) must be followed shortly after by a series of associated responses from the ESME.
- The ESME should always return SMPP responses to the SMSC in the same order in which the original requests were received. However this is not mandatory within SMPP and the SMSC should be capable of handling responses received out of sequence.

- SMPP responses should be returned by the SMSC in the same order in which the original requests were received from the ESME. However this is not mandatory within SMPP and the ESME should be capable of handling responses received out of sequence.

Note: The maximum number of outstanding (i.e. unacknowledged) SMPP operations between an ESME and SMSC and vice versa is not specified explicitly in the SMPP Protocol Specification and will be governed by the SMPP implementation on the SMSC.

However, as a guideline it is recommended that no more than 10 (ten) SMPP messages are outstanding at any time.

2.7 Duplex message exchange between an SMSC and an ESME

The SMSC and ESME may operate a duplex messaging session, i.e. messages are exchanged in both directions. In this case the ESME must be connected to the SMSC as an ESME Transceiver.

Typical applications in which an ESME would operate as an SMPP Transceiver include:-

- Two-way message exchange between a mobile station and an ESME, e.g a WAP Proxy/Server. The mobile subscriber initiates an information request to the WAP Proxy Server and the information response is returned via the SMSC to the mobile station.

Examples of SMPP message Protocol Data Units (PDUs) which may be sent on an SMPP Transceiver session include:

- *data_sm*
- *submit_sm*
- *deliver_sm*

In addition to submission of messages to the SMSC, an ESME may perform the following SMPP operations using the message identifier returned by the SMSC in the message acknowledgement:

- *query_sm* - Query the SMSC for the status of a previously submitted message
- *cancel_sm* - Cancel delivery of a previously submitted message
- *replace_sm* - Replace a previously submitted message

SMPP PDUs delivered to an ESME by the SMSC (or vice versa) must be acknowledged with a PDU response when received*.

* Exceptions to this rule are:

- the *alert_notification* PDU.

Refer to Table 2-1 for details on the SMPP operations which may be sent on an SMPP Transceiver session.

2.7.1 Typical SMPP session sequence - ESME Transceiver

The following diagram illustrates a typical SMPP request/response sequence between an SMSC and an ESME bound as a Transceiver.

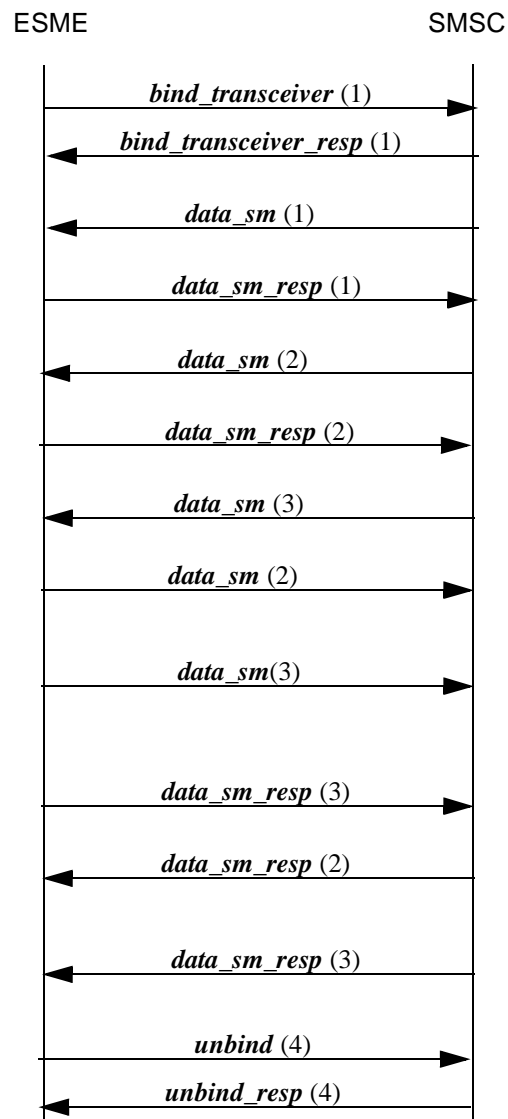


Figure 2-6: Typical SMPP request/response sequence for an ESME Transceiver

- The exchange of SMPP request and response PDUs between an SMSC and ESME Transceiver may be implemented synchronously or asynchronously as shown above. Thus the SMSC may send multiple *data_sm* requests to the ESME, without synchronously waiting for the associated response PDUs.
- A series of successive SMPP requests issued asynchronously by an SMSC (as denoted by the number in parentheses) must be followed shortly after by a series of associated responses from the ESME. The *sequence_number* parameter in the SMPP header is used to correlate the SMPP response PDU with the SMPP request PDU.

- The ESME should always return SMPP PDU responses to the SMSC in the same order in which the original requests were received. However this is not mandatory within SMPP and the SMSC should be capable of handling responses received out of sequence
- SMPP responses should be returned by the SMSC in the same order in which the original requests were received from the ESME. However this is not mandatory within SMPP and the ESME should be capable of handling responses received out of sequence.

Note: The maximum number of outstanding (i.e. unacknowledged) SMPP operations between an ESME and SMSC and vice versa is not specified explicitly in the SMPP Protocol Specification and will be governed by the SMPP implementation on the SMSC.

However, as a guideline it is recommended that no more than 10 (ten) SMPP messages are outstanding at any time.

2.8 SMPP Error Handling

All SMPP operations consist of a request PDU and associated response PDU, with the exception of the *alert_notification* PDU (for which there is no SMPP response).

In all other cases, the receiving entity must return the associated SMPP response PDU to an SMPP request PDU, indicating that the original PDU has been received at the destination. Until such a response is received by the originator, it must be assumed that the PDU has not been received at the destination.

In the event that the original SMPP request PDU is found to contain an error, the receiving entity must return a response with an appropriate error code inserted in the *command_status* field of the response PDU header (Ref. Section 3.2, “SMPP PDU Format - Overview”).

If the receiving entity finds an error in the PDU header, it should return a *generic_nak* PDU to the originator (Ref. Section 4.3, “GENERIC_NACK” PDU).

2.9 SMPP Timers

To ensure the efficient exchange of SMPP transactions, it is recommended that each SMPP session be managed using configurable timers on both the ESME and SMSC communicating SMPP entities as follows:-

- An SMPP session initiation timer to ensure that when an ESME initiates an SMPP session, that this occurs within a specified period after opening a network connection to the SMSC.
- An SMPP session timer to enable either the ESME or SMSC request the SMPP session status of the other communicating SMPP entity via the *enquire_link* command.
- An SMPP inactivity timer which should specify the maximum period after which time, if no SMPP messages are exchanged, the SMPP session may be dropped gracefully.
- An SMPP transaction timer which specifies the time lapse allowed between an SMPP request and the corresponding SMPP response.

Further details on implementation of SMPP timers are included in Section 7.2, “Timer Definitions” .

2.10 Message Modes

SMPP offers a Message Mode option which, if supported on the SMSC, allows an ESME to select the SMSC message delivery mechanism. The typical delivery mechanisms that may be offered by an SMSC are:-

- Store and Forward
- Datagram
- Transaction mode

These Message Modes are described in more detail in the following sections.

2.10.1 Store and Forward Message Mode

The conventional approach to SMS has been to store the message in a SMSC storage area (e.g. message database) before forwarding the message for delivery to the recipient SME. With this model, the message remains securely stored until all delivery attempts have been made by the SMSC. This mode of messaging is commonly referred to as “store and forward”.

SMPP supports the “store and forward” delivery mechanism via the *submit_sm* operation, which enables the ESME to send a message to the SMSC where it is stored until it is successfully delivered or until the message validity period expires. The store and forward mode is also supported via the *data_sm* operation.

The “store and forward” message mode also facilitates subsequent SMPP operations on the stored short message such as *query_sm*, *replace_sm* and *cancel_sm*. The *submit_sm* PDU also facilitates “replace-if-present” functionality which requires that the original message be stored on the SMSC.

Note: To determine the eventual outcome of the SMS delivery, the ESME must request an SMSC Delivery Receipt in the *submit_sm* or *data_sm* operation.

The following diagram shows the message flow for a store and forward message where the ESME is bound both as a Transmitter and as a Receiver. The ESME has requested an SMSC Delivery Receipt.

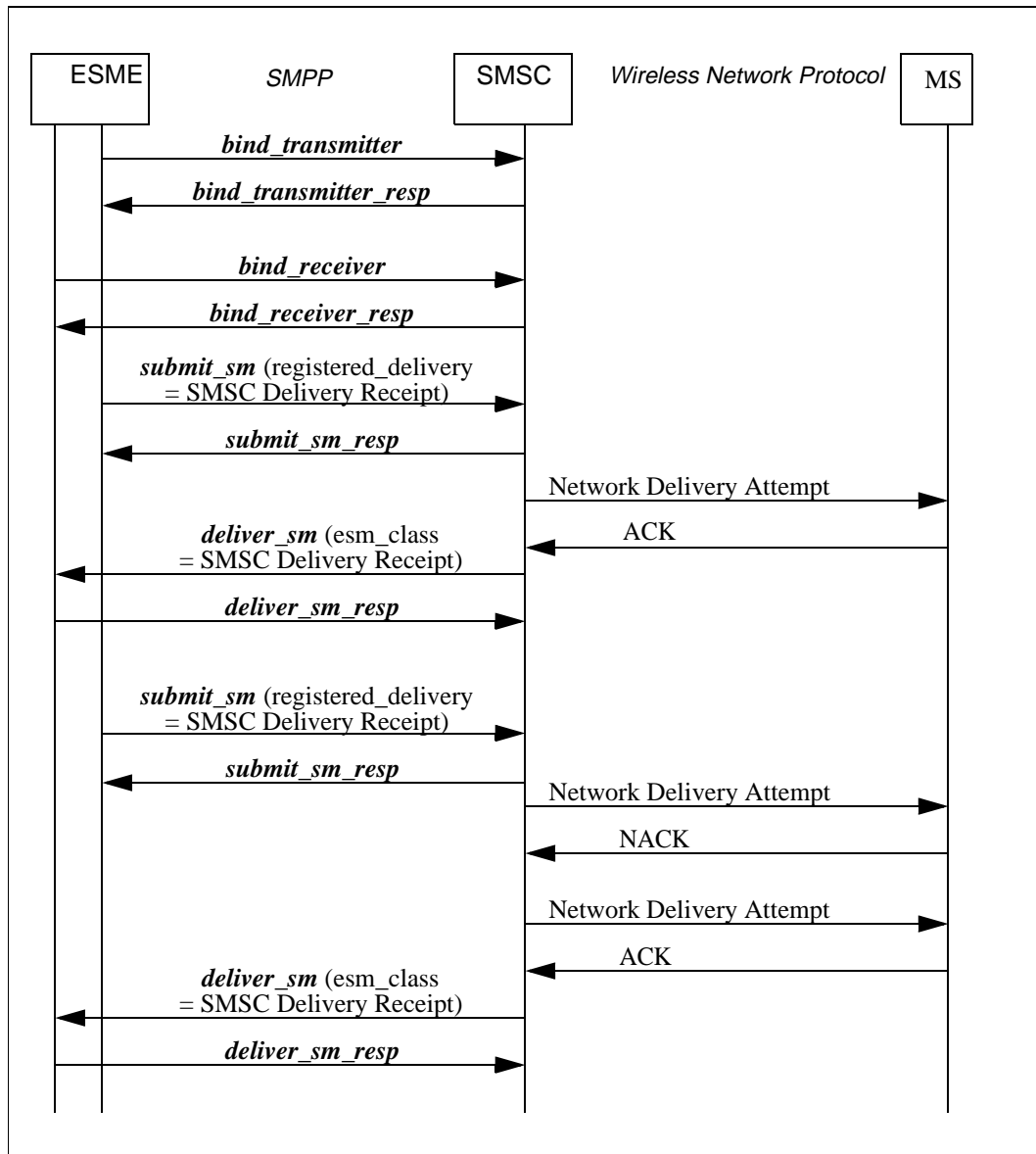


Figure 2-7: Typical SMPP sequence for a registered store and forward message

2.10.2 Datagram Message Mode

The Datagram Message Mode emulates the datagram paradigm used in other data communication protocols such as UDP datagram packet transfer and focuses on high message throughput without the associated secure storage and retry guarantees of Store and Forward Message Mode. In Datagram Message Mode the message originator (i.e. the ESME) does not receive any form of delivery acknowledgement.

In Datagram Message Mode, typical SMSC functions such as scheduled delivery, registered delivery etc. do not apply. Datagram Message Mode is designed for high throughput applications that may not require the highly secure delivery functionality offered by the Store and Forward message mode. It is ideally suited for applications where the data content is transient in nature, for example, vehicle tracking applications.

SMPP supports datagram mode via the *data_sm* operation. The *esm_class* parameter is used to select Datagram Message Mode. Refer to section 5.2.12, “esm_class” for details on the *esm_class* parameter.

The datagram mode is also supported in the *submit_sm* operation to provide easy evolution for existing SMPP applications.

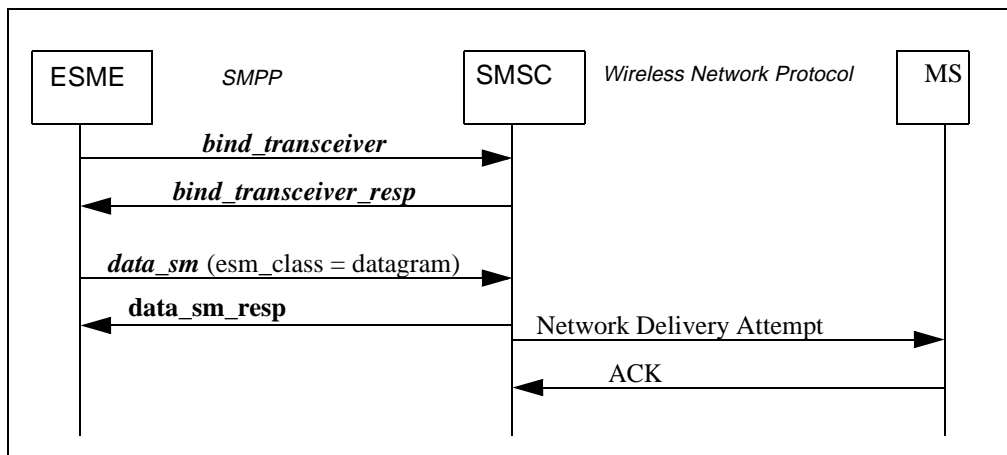


Figure 2-8: Typical SMPP sequence for message delivery in Datagram message mode

2.10.3 Transaction Message Mode

Transaction Message Mode allows the ESME message originator to receive a form of delivery acknowledgment (that indicates if the message has been successfully or unsuccessfully delivered to the destination MS) within the SMPP response PDU.

Transaction Message Mode is designed for applications that involve real-time messaging where an ESME requires a synchronous end-to-end delivery outcome, without the need for long term SMSC storage. Such applications could include for example multicast of dispatch information to vehicle fleets, etc.

SMPP supports Transaction Message Mode via the *data_sm* operation only. The *esm_class* parameter is used to select Transaction Message Mode. Refer to section 5.2.12, for details on the *esm_class* parameter.

Note: The fundamental difference between the Datagram and Transaction Message Modes is that in Transaction Message Mode, the ESME receives a *data_sm_resp* indicating the end-to-end delivery outcome. In Datagram Message Mode, the response PDU just indicates that the message has been accepted by the SMSC over the SMPP connection.

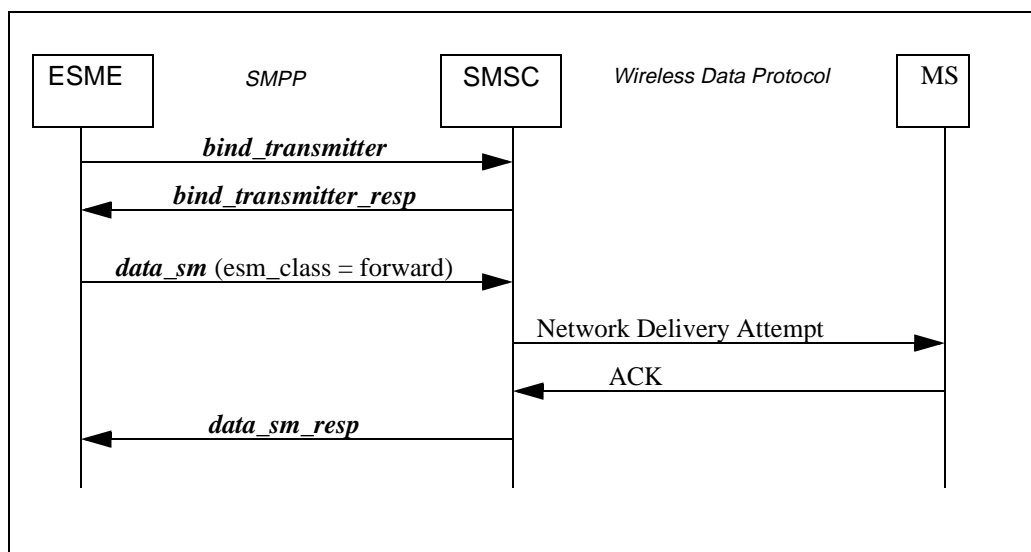


Figure 2-9: Typical SMPP sequence for message delivery in Transaction message mode

2.11 Message Types

In addition to “normal” short messages, special messages can be transferred between ESME and the SMSC in a *submit_sm*, *deliver_sm* or a *data_sm* operation. The message type is defined in the *esm_class* parameter of the above SMPP operations.

The following message types are supported in SMPP:

SMSC Delivery Receipt

This message type is used to carry an SMSC delivery receipt. The SMSC, on detecting the final state of a registered message stored in the SMSC, should generate a receipt message addressed to the originator of the message. The SMSC Delivery Receipt is carried as the user data payload in the SMPP *deliver_sm* or *data_sm* operation.

The following fields are relevant in the *deliver_sm* and *data_sm* operations when used for transmitting delivery receipts.

- source address (i.e. *source_addr_ton*, *source_addr_npi*, *source_addr*)
The source address will be taken from the destination address of the original short message which generated the delivery receipt.
- destination address (i.e. *dest_addr_ton*, *dest_addr_npi*, *destination_addr*)
The destination address will be taken from the source address of the original short message which generated the delivery receipt.
- *esm_class*
- *message_state*
- *network_error_code*
- *receipted_message_id*

Intermediate Notification

An intermediate notification is a special form of message that the SMSC may send to an ESME for a mobile terminated message delivery. It provides an intermediate status of a message delivery attempt.

Typical uses are

- to provide a “memory capacity exceeded” notification to a Voice Mail System.
- to report the outcome of the first delivery attempt that has failed but the message is still held in the SMSC for further delivery attempts.

Support for Intermediate Notification functionality is specific to the SMSC implementation and the SMSC Service Provider and is beyond the scope of this specification.

SME Delivery Acknowledgement

Despite its name, an SME Delivery Acknowledgement is not an indication that the short message has arrived at the SME, but rather an indication from the recipient SME that the user has read the short message.

For an MS-based SME, an SME Delivery Acknowledgement is sent when the MS user or MS application has read the message from the SMS storage unit (e.g. SIM card).

For a fixed SME (i.e. ESME) the circumstances in which an SME Delivery Acknowledgement may be sent are beyond the scope of this specification.

Note: The SME Delivery Acknowledgement function may not be supported on all network types.

SME Manual/User Acknowledgement

A Manual/User Acknowledgement is an application generated reply message sent in response to an application request message. For example, this message type could contain a selected menu item number from a menu list sent in an application request message.

Note: The Manual/User Acknowledgement function may not be supported on all network types.

Conversation Abort

This message type is unique to Interactive Teleservice defined by the Korean CDMA carriers organization. It is sent by a MS-based SME to indicate the unexpected termination of an interactive session. A Conversation Abort may be carried in a *deliver_sm* or *data_sm* PDU.

Note: The Conversation Abort function is not supported on all network types.

3. SMPP PDU Type and Format Definitions

3.1 SMPP PDU - Type Definitions

The following SMPP PDU data type definitions are used to define the SMPP parameters:

Integer	An unsigned value with the defined number of octets. The octets will always be transmitted MSB first (Big Endian).
C-Octet String	A series of ASCII characters terminated with the NULL character.
C-Octet String (Decimal)	A series of ASCII characters, each character representing a decimal digit (0 - 9) and terminated with the NULL character.
C-Octet String (Hex)	A series of ASCII characters, each character representing a hexadecimal digit (0 - F) and terminated with the NULL character.
Octet String	A series of octets, not necessarily NULL terminated.

- Notes:**
- (i) Reference made to NULL settings of Octet-String fields implies that the field consists of a single NULL character, i.e., an octet encoded with value 0x00 (zero).
 - (ii) Where reference is made to NULL settings of Integer fields, this implies that the field is zero filled.
 - (iii) In the case of all C-Octet String formats, the maximum field size is shown as a combination of string length and the NULL terminator, i.e., an 8-character C-Octet String is encoded in 9 octets when the NULL terminator is included.

3.1.1 SMPP Parameter Field Size Notation

The following notation style is used throughout. Note that some SMPP strings are optional and others mandatory.

Size octets	Type	Description of String type specified
4	Integer	Fixed size integer field. In this example the integer is of size 32 bits (4 octets)
Var Max 16	C-Octet String	This string is of variable length from 1-15 ASCII characters, followed by an octet containing the NULL terminator. An empty string is encoded as a single octet containing the NULL character (0x00).
Fixed 1 or 17	C-Octet String	This string has two possible lengths:- 1 octet containing the NULL character or a fixed number of characters terminated with the NULL character (in this example 16 characters plus the NULL character).
Var 0 - 254	Octet String	Variable size octet string field. In this example the size of the octet string field can vary from 0 to 254 octets.

Table 3-1: C-Octet String Notation

3.2 SMPP PDU Format - Overview

The general format of an SMPP PDU consists of a PDU header followed by a PDU body as outlined in the following table.

SMPP PDU				
PDU Header (mandatory)				PDU Body (Optional)
<i>command length</i>	<i>command id</i>	<i>command status</i>	<i>sequence number</i>	<i>PDU Body</i>
4 octets	Length = (Command Length value - 4) octets			

Table 3-2: SMPP PDU Format Overview

The SMPP Header is a mandatory part of every SMPP PDU and must always be present. The SMPP PDU Body is optional and may not be included with every SMPP PDU.

The format of each SMPP PDU is described in more detail in the following section 4. "SMPP PDU Definition".

3.2.1 SMPP PDU Layout

	SMPP PDU Field	Size octets	Type	Description
H E A D E R	<i>command_length</i>	4	Integer	The <i>command_length</i> field defines the total octet length of the SMPP PDU packet including the length field.
	<i>command_id</i>	4	Integer	<p>The <i>command_id</i> field identifies the particular SMPP PDU, e.g., <i>submit_sm</i>, <i>query_sm</i>, etc.</p> <p>A unique command identifier is allocated to each SMPP request PDU in the range: 0x00000000 to 0x000001FF</p> <p>A unique command identifier is also allocated to each SMPP response PDU in the range: 0x80000000 to 0x800001FF</p> <p>(Note that an SMPP response <i>command_id</i> is identical to the corresponding request SMPP <i>command_id</i>, but with bit 31 set). Refer to chapter 5. for details of the complete SMPP Command ID set.</p>
	<i>command_status</i>	4	Integer	<p>The <i>command_status</i> field indicates the success or failure of an SMPP request. It is relevant only in the SMPP response PDU and it must contain a NULL value in an SMPP request PDU.</p> <p>The complete list of SMPP Error codes is defined in Chapter 5.</p>
	<i>sequence_number</i>	4	Integer	<p>This field contains a sequence number which allows SMPP requests and responses to be associated for correlation purposes. The use of sequence numbers for message correlation allows SMPP PDUs to be exchanged asynchronously.</p> <p>Assignment of the <i>sequence_number</i> is the responsibility of the SMPP PDU originator. The <i>sequence_number</i> should be increased monotonically for each submitted SMPP request PDU and must be preserved in the associated SMPP response PDU.</p> <p>The <i>sequence_number</i> may range from: 0x00000001 to 0x7FFFFFFF.</p>

Table 3-3: SMPP PDU Format Description

B O D Y	<i>Mandatory Parameters</i>	var.	mixed	<p>A list of mandatory parameters corresponding to that SMPP PDU defined in the <i>command_id</i> field.</p> <p>The complete list of mandatory parameters is detailed in section 4. "SMPP PDU Definition" with the description of each SMPP PDU.</p>
	<i>Optional Parameters</i>	var.	mixed	<p>A list of Optional Parameters corresponding to that SMPP PDU defined in the <i>command_id</i> field and included as required.</p> <p>The complete list of optional parameters is detailed in section 4. "SMPP PDU Definition" with the description of each SMPP PDU.</p>

Table 3-3: SMPP PDU Format Description

Note: Some SMPP PDUs may only have a Header part only, for example, the *enquire_link* PDU.

3.2.2 SMPP PDU Length

The *command_length* field at the beginning of the SMPP PDU header, indicates the total number of octets contained in that SMPP PDU. The *command_length* field contains a 4-octet integer transmitted in Big Endian format.

To decode an SMPP PDU, the ESME or SMSC should first read the *command_length* field (4 octets) to determine the PDU length. The amount of remaining data is then determined by subtracting the length of the *command_length* field (4 octets) from this total PDU length as provided by the *command_length* field value. Thus, extracting a command length of value N, indicates that N-4 octets are remaining for the given PDU.

Example:-

The following data-stream example illustrates how the SMPP PDU header is encoded:

```
00 00 00 2F 00 00 00 02 00 00 00 00 00 00 00 01 53 4D 50 50 33 54 45 53 54 00
73 65 63 72 65 74 30 38 00 53 55 42 4D 49 54 31 00 00 01 01 00
```

Note: Values are shown in Hex format.

The header would be decoded as follows:

00 00 00 2F	Command Length	0x0000002F
00 00 00 02	Command ID	0x00000002 (<i>bind_transmitter</i>)
00 00 00 00	Command Status	0x00000000
00 00 00 01	Sequence Number	0x00000001

The remaining data represents the PDU body (which in this example relates to the *bind_transmitter* PDU).

3.2.3 SMPP Message length and extended message length

The length of the short message text (or user data) is defined in the *sm_length* field of the *submit_sm*, *submit_multi*, *deliver_sm* and *replace_sm* SMPP PDUs.

The maximum message length which can be specified in *sm_length* field (see section 5.2.21) is 254 octets. If an ESME wishes to submit a message of length greater than 254 octets, the *sm_length* field must be set to NULL and the *message_payload* optional parameter must be populated with the message length value and user data.

SMPP supports extended message lengths in the *submit_sm*, *submit_multi*, *data_sm* and *deliver_sm* PDUs.

Refer to section 3.2.4 "Optional Parameters" for detail on Optional Parameters.

Note: The actual short message length which can be transmitted to a MS may vary according to the underlying network.

3.2.4 Optional Parameters

Optional Parameters are fields, which may be present in a message. Optional Parameters provide a mechanism for the future introduction of new parameters, as and when defined in future versions of the SMPP protocol.

Optional Parameters must always appear at the end of a PDU , in the "Optional Parameters" section of the SMPP PDU. However, they may be included in any convenient order within the "Optional Parameters" section of the SMPP PDU and need not be encoded in the order presented in this document.

For a particular SMPP PDU, the ESME or SMSC may include some, all or none of the defined optional parameters as required for the particular application context. For example a paging system may only need to include the “callback number” related optional parameters in a *submit_sm* operation.

3.2.4.1 Optional Parameter Format

All optional parameters have the following general TLV (Tag, Length, Value) format. The definition of the Tag, Length and Value for each optional parameter is defined in chapter 5.

Parameter Name	Size	Type	Description
<i>Tag</i>	2	Integer	The <i>Tag</i> field is used to uniquely identify the particular optional parameter in question. The optional parameter <i>Tag</i> field is always 2 octets in length.
<i>Length</i>	2	Integer	The <i>Length</i> field indicates the length of the <i>Value</i> field in octets. Note that this length does not include the length of the <i>Tag</i> and <i>Length</i> fields. The optional parameter <i>Length</i> field is always 2 octets in length.
<i>Value</i>	variable	variable	The <i>Value</i> field contains the actual data for the optional parameter in question.

Table 3-4: Optional Parameter Format

3.3 Guidelines for SMPP Forward Compatibility

Forward Compatibility procedures allow a functional entity (i.e. SMSC or ESME) using one version of the SMPP protocol to easily communicate with an entity using a later, more enhanced version of the protocol. Hence, new enhancements to existing SMPP PDUs are implemented using optional parameters.

The following guidelines must be followed in SMPP implementations to ensure that this process is implemented successfully and consistently:

- If an SMPP entity receives an unrecognized PDU/command, it **must** return a ***generic_nack*** PDU indicating an invalid *command_id* in the *command_status* field of the header.
- The SMPP entity receiving a message which includes Optional Parameters shall first inspect the *Tag* field of the Operational Parameter, as follows:
 - If the Optional Parameter Tag is recognized and supported by the receiving SMPP entity for the particular SMPP operation, the Optional Parameter shall be processed.
 - If an Optional Parameter Tag is recognized but not expected for the particular SMPP operation, the optional parameter shall be ignored.
 - If the Optional Parameter Tag is unrecognized or unsupported by the receiving SMPP entity, that particular Optional Parameter shall be ignored and the next Optional Parameter in the sequence shall be processed.
- An SMPP entity receiving a parameter value defined as “reserved” should use the default value if a “default” setting is defined, otherwise the parameter should be ignored.
- If the Parameter value is otherwise unrecognized or invalid, the SMPP entity should return an error indicating the Parameter Value is invalid.
- An SMPP entity detecting that an Optional Parameter, which is required in the context of the operation, is not present should return a response message with an “Expected Optional Parameter missing” error.
- A Variable length field Parameter may have its maximum length definition extended in subsequent versions of the SMPP protocol. An SMPP entity receiving a variable length Parameter whose length is greater than the maximum length the entity supports for that Parameter should reject the Parameter with an error indicating “invalid parameter length”.

3.4 Guidelines for SMPP Backward Compatibility

Backward Compatibility procedures allow a functional entity using one version of the SMPP protocol to communicate with an entity using an older version of the protocol.

The following guidelines **must** be followed in SMPP implementations to ensure that this process is implemented successfully and consistently:

- Existing SMPP PDUs must not be removed from the protocol.
- The effect of receiving any existing message in a new modified format must be same as that in previous versions. Thus the addition of new parameters or parameter values is purely additive.
- Optional parameters shall not become mandatory parameters.
- Mandatory parameters shall not become optional parameters.
- Additional mandatory parameters shall not be added to an existing SMPP PDU.
- Existing mandatory parameters shall not be removed from an existing SMPP PDU.
- The meaning of any existing parameter value shall not be changed in the new version of the protocol.

As the concept of Optional Parameters was introduced in this version of the protocol, the following special guidelines are defined:

- An SMSC that implements SMPP v3.4 or a later version of this protocol must not send optional parameters to an ESME that implements an earlier SMPP version (e.g. v3.3). An SMSC shall determine the SMPP version supported by an ESME during the bind operation. An ESME supporting SMPP v3.3 or earlier will set the *interface_version* parameter in the bind operation to a value less than 0x34.
- An SMSC supporting v3.4 or later should return the SMPP version it supports in the *sc_interface_version* parameter of the bind response PDU. If a bind response does not contain the *sc_interface_version* parameter, then the ESME should assume that the SMSC does not support the use of optional parameters.
- An ESME that implements SMPP v3.4 or a later version of this protocol must not send optional parameters to an SMSC that implements an earlier version of this protocol. The ESME shall determine the SMSC version support from the SMPP bind response PDU.
- An SMSC that implements SMPP v3.4 or later must not generate message IDs greater than 8 octets when communicating with an ESME that supports SMPP v3.3 or earlier.

4. SMPP PDU Definition

4.1 “BIND” Operation

The purpose of the SMPP bind operation is to register an instance of an ESME with the SMSC system and request an SMPP session over this network connection for the submission or delivery of messages. Thus, the Bind operation may be viewed as a form of SMSC login request to authenticate the ESME entity wishing to establish a connection.

As described previously, an ESME may bind to the SMSC as either 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 an 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.

If an SMSC does not support the *bind_transmitter* and *bind_receiver* operations then it should return a response message with an “Invalid Command ID” error and the ESME should reattempt to bind using the *bind_transceiver* operation. Similarly if an SMSC does not support the *bind_transceiver* command then it should return a response message with an “Invalid Command ID” error and the ESME should reattempt to bind using the *bind_transmitter* or *bind_receiver* operations or both *bind_transmitter* and *bind_receiver* operations as appropriate.

ESME Transmitter

An ESME bound as a Transmitter is authorised to send short messages to the SMSC and to receive the corresponding SMPP responses from the SMSC.

An ESME indicates its desire not to receive (mobile) originated messages from other SME’s (e.g. mobile stations) by binding as a Transmitter.

Refer to section 2.3 for a summary list of the SMPP PDUs available to an ESME Transmitter.

ESME Receiver

An ESME bound as a Receiver is authorised to receive short messages from the SMSC and to return the corresponding SMPP message responses to the SMSC.

Refer to section 2.3 for a summary list of the SMPP PDUs available to an ESME Receiver.

ESME Transceiver

An ESME bound as a Transceiver is allowed to send messages to the SMSC and receive messages from the SMSC over a single SMPP session.

Refer to section 2.3 for a summary list of the SMPP PDUs available to an ESME Transceiver.

4.1.1 “BIND_TRANSMITTER” Syntax

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

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Defines the overall length of the <i>bind_transmitter</i> PDU.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_transmitter</i> request.	5.1.2
	<i>command_status</i>	4	Integer	Not used in <i>bind_transmitter</i> PDU. Must be set to NULL.	5.1.3
	<i>sequence_number</i> ^a	4	Integer	Set to a unique sequence number. The associated <i>bind_transmitter_resp</i> PDU will echo the same sequence number.	5.1.4
B O D Y	<i>system_id</i> ^b	Var. max 16	C- Octet String	Identifies the ESME system requesting to bind as a transmitter with the SMSC.	5.2.1
	<i>password</i> ^c	Var. max 9	C- Octet String	The password may be used by the SMSC to authenticate the ESME requesting to bind.	5.2.2
	<i>system_type</i> ^d	Var. 13	C- Octet String	Identifies the type of ESME system requesting to bind as a transmitter with the SMSC.	5.2.3
	<i>interface_version</i>	1	Integer	Indicates the version of the SMPP protocol supported by the ESME.	5.2.4
	<i>addr_ton</i>	1	Integer	Indicates Type of Number of the ESME address. If not known set to NULL	5.2.5
	<i>addr_npi</i>	1	Integer	Numbering Plan Indicator for ESME address. If not known set to NULL.	5.2.6
	<i>address_range</i>	Var. max 41	C- Octet String	The ESME address. If not known set to NULL.	5.2.7

Table 4-1: SMPP *bind_transmitter* PDU

- There is no specific requirement on how the *sequence_number* should be set. However, it is recommended that the sequence number be a monotonically increasing number.
- The recommended use of *system_id* is to identify the binding entity, e.g., “InternetGW” in the case of an Internet Gateway or ‘VMS’ for a Voice Mail System.
- The *password* is used for authentication to secure SMSC access. The ESME may set the password to NULL to gain insecure access (if allowed by SMSC administration).
- The *system_type* (optional) may be used to categorise the system, e.g., “EMAIL”, “WWW”, etc.

4.1.2 “BIND_TRANSMITTER_RESP” Syntax

The SMPP *bind_transmitter_resp* PDU is used to reply to a *bind_transmitter* request. The format of the SMPP *bind_transmitter_resp* PDU is defined in the following table.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Defines the overall length of the <i>bind_transmitter_resp</i> PDU.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_transmitter_resp</i> .	5.1.2
	<i>command_status</i>	4	Integer	Indicates status (success or error code) of original <i>bind_transmitter</i> request.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>bind_transmitter</i> request.	5.1.4
B O D Y	<i>system_id</i>	Var. max 16	C- Octet String	SMSC identifier. Identifies the SMSC to the ESME.	5.2.1
OPTIONAL PARAMETERS for BIND_TRANSMITTER_RESP					
	<i>sc_interface_version</i>		TLV	SMPP version supported by SMSC	5.3.2.25

Table 4-2: *bind_transmitter_resp* PDU

Note: The body portion of the SMPP *bind_transmitter_resp* PDU is not returned if the *command_status* field contains a non-zero value; i.e., if there is an error in the original *bind_transmitter* request, the SMSC *system_id* is not returned.

4.1.3 “BIND_RECEIVER” Syntax

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

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Defines the overall length of the PDU in octets.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_receiver</i> request.	5.1.2
	<i>command_status</i>	4	Integer	Not used in <i>bind_receiver</i> PDU. Set to NULL .	5.1.3
	<i>sequence_number^a</i>	4	Integer	Set to a unique sequence number. The associated <i>bind_receiver_resp</i> PDU will echo the same sequence number.	5.1.4
B O D Y	<i>system_id^b</i>	Var. max 16	C- Octet String	Identifies the ESME system requesting to bind as a receiver with the SMSC.	5.2.1
	<i>password^c</i>	Var. max 9	C- Octet String	The password may be used by the SMSC for security reasons to authenticate the ESME requesting to bind.	5.2.2
	<i>system_type^d</i>	Var. max 13	C- Octet String	Identifies the type of ESME system requesting to bind as a receiver with the SMSC.	5.2.3
	<i>interface_version</i>	1	Integer	Identifies the version of the SMPP protocol supported by the ESME.	5.2.4
	<i>addr_ton^e</i>	1	Integer	Type of Number (TON) for ESME address(es) served via this SMPP receiver session. Set to NULL if not known.	5.2.5
	<i>addr_npi^e</i>	1	Integer	Numbering Plan Indicator (NPI) for ESME address(es) served via this SMPP receiver session. Set to NULL if not known.	5.2.6
	<i>address_range^e</i>	Var. max 41	C- Octet String	A single ESME address or a range of ESME addresses served via this SMPP receiver session. The parameter value is represented in UNIX regular expression format (see Appendix A). Set to NULL if not known.	5.2.7

Table 4-3: SMPP *bind_receiver* PDU

- a. There is no specific requirement on how the *sequence_number* should be set. However, it is recommended that the sequence number be a monotonically increasing number.
- b. The recommended use of *system_id* is to identify the binding entity, e.g., “InternetGW” in the case of an Internet Gateway or ‘VMS’ for a Voice Mail System.
- c. The *password* is used for authentication to secure SMSC access. The ESME may set it to NULL to gain insecure access (if allowed by SMSC administration).
- d. The *system_type* (optional) may be used to categorise the system, e.g., “EMAIL”, “WWW”, etc.
- e. The *addr_ton*, *addr_npi* and *addr_range* parameters may be used by the ESME to provide an identification of the SME address(es) that the ESME serves.

4.1.4 “BIND_RECEIVER_RESP”

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

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Defines the overall length of the PDU.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_receiver_resp</i> .	5.1.2
	<i>command_status</i>	4	Integer	Indicates status (success or error code) of original <i>bind_receiver</i> request.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>bind_receiver</i> request.	5.1.4
B O D Y	<i>system_id</i>	Var. max 16	C- Octet String	SMSC identifier. Identifies the SMSC to the ESME.	5.2.1
OPTIONAL PARAMETERS for BIND_RECEIVER_RESP					
	Optional Parameter Name	Type		Description	Ref.
	<i>sc_interface_version</i>	TLV		SMPP version supported by SMSC	5.3.2.25

Table 4-4: *bind_receiver_resp* PDU

Note: The *bind_receiver_resp* PDU Body is not returned if the *command_status* field contains a non-zero value, i.e., if there is an error in the original *bind_receiver* request, the SMSC *system_id* is not returned.

4.1.5 “BIND_TRANSCEIVER” Syntax

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

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Defines the overall length of the PDU.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_transceiver</i> request.	5.1.2
	<i>command_status</i>	4	Integer	Not used in <i>bind_transceiver</i> PDU. Set to NULL.	5.1.3
	<i>sequence_number^a</i>	4	Integer	Set to a unique sequence number. The associated <i>bind_transceiver_resp</i> PDU will echo the same sequence number.	5.1.4
B O D Y	<i>system_id^b</i>	Var. max 16	C- Octet String	Identifies the ESME system requesting to bind as a transceiver with the SMSC.	5.2.1
	<i>password^c</i>	Var. max 9	C- Octet String	The password may be used by the SMSC to authenticate the ESME requesting to bind.	5.2.2
	<i>system_type^d</i>	Var. max 13	C- Octet String	Identifies the type of ESME system requesting to bind as a transceiver with the SMSC.	5.2.3
	<i>interface_version</i>	1	Integer	Identifies the version of the SMPP protocol supported by the ESME.	5.2.4
	<i>addr_ton^e</i>	1	Integer	Type of Number (TON) for ESME address(es) served via this SMPP transceiver session. Set to NULL (Unknown) if not known.	5.2.5
	<i>addr_npi^e</i>	1	Integer	Numbering Plan Indicator (NPI) for ESME address(es) served via this SMPP transceiver session. Set to NULL (Unknown) if not known.	5.2.6
	<i>address_range^e</i>	Var. max 41	C- Octet String	A single ESME address or a range of ESME addresses served via this SMPP transceiver session. This field may be used by the SMSC for authentication, verification or routing purposes. Set to NULL if not known.	5.2.7

Table 4-5: SMPP *bind_transceiver* PDU

- a. There is no specific requirement on how the *sequence_number* should be set. However, it is recommended that the sequence number be a monotonically increasing number.
- b. The recommended use of *system_id* is to identify the binding entity, e.g., “InternetGW” in the case of an Internet Gateway or ‘VMS’ for a Voice Mail System.
- c. The *password* is used for authentication to secure SMSC access. The ESME may set it to NULL to gain insecure access (if allowed by SMSC administration).
- d. The *system_type* (optional) may be used to categorise the system, e.g., “EMAIL”, “WWW”, etc.
- e. The use of the parameters *addr_ton*, *addr_npi* and *addr_range* is SMSC implementation specific. By specifying these fields in the *bind_transceiver* operation, the ESME is providing the SMSC with the SME address(es) that it serves.

4.1.6 “BIND_TRANSCEIVER_RESP”

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

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Defines the overall length of the PDU.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_transceiver_resp</i> .	5.1.2
	<i>command_status</i>	4	Integer	Indicates status (success or error code) of original <i>bind_transceiver</i> request.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>bind_transceiver</i> request.	5.1.4
B O D Y	<i>system_id</i>	Var. max 16	C- Octet String	SMSC identifier. Identifies the SMSC to the ESME.	5.2.1
OPTIONAL PARAMETERS for BIND_RECEIVER_RESP					
	Optional Parameter Name	Type		Description	Ref.
	<i>sc_interface_version</i>	TLV		SMPP version supported by SMSC	5.3.2.25

Table 4-6: *bind_transceiver_resp* PDU

4.1.7 “OUTBIND” Operation.

This operation is used by the SMSC to signal an ESME to originate a *bind_receiver* request to the SMSC.

4.1.7.1 “OUTBIND” Syntax

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

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Defines the overall length of the PDU.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>outbind</i> .	5.1.2
	<i>command_status</i>	4	Integer	Not used in <i>outbind</i> PDU. Set to NULL .	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number.	5.1.4
B O D Y	<i>system_id</i>	Var. max 16	C- Octet String	SMSC identifier. Identifies the SMSC to the ESME.	5.2.1
	<i>password^a</i>	Var. max 9	C- Octet String	The password may be used by the ESME for security reasons to authenticate the SMSC originating the <i>outbind</i> .	5.2.2

a. The *password* is used for authentication to secure ESME access. The SMSC may set it to NULL to gain insecure access (if allowed by ESME administration).

4.2 “UNBIND” Operation

The purpose of the SMPP *unbind* operation is to deregister an instance of an ESME from the SMSC and inform the SMSC 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 SMSC logoff request to close the current SMPP session.

4.2.1 “UNBIND”

The format of the SMPP *unbind* PDU is defined in the following table. The *command_id* field must include the Command ID value corresponding to the *unbind* operation.

H E A D E R	Field Name	Size octets	Type	Description	Ref.
	<i>command_length</i>	4	Integer	Defines the overall length of the PDU.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>unbind</i> request.	5.1.2
	<i>command_status</i>	4	Integer	Not used. Set to NULL .	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>unbind_resp</i> PDU will echo the same sequence number.	5.1.4

Table 4-7: SMPP *unbind* PDU format

4.2.2 “UNBIND_RESP”

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. The *command_id* field must include the Command ID value corresponding to the *unbind_resp* operation.

H E A D E R	Field Name	Size octets	Type	Description	Ref.
	<i>command_length</i>	4	Integer	Defines the overall length of the PDU.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>unbind_resp</i> PDU.	5.1.2
	<i>command_status</i>	4	Integer	Indicates outcome of original <i>unbind</i> request.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>unbind</i> request.	5.1.4

Table 4-8: SMPP *unbind_resp* PDU format

4.3 “GENERIC_NACK” PDU

This is a generic negative acknowledgement to an SMPP PDU submitted with an invalid message header. A *generic_nack* response is returned in the following cases:

- Invalid *command_length*
If the receiving SMPP entity, on decoding an SMPP PDU, detects an invalid *command_length* (either too short or too long), it should assume that the data is corrupt. In such cases a *generic_nack* PDU must be returned to the message originator.
- Unknown *command_id*
If an unknown or invalid *command_id* is received, a *generic_nack* PDU must also be returned to the originator.

4.3.1 “GENERIC_NACK” Syntax

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

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Defines the overall length of the PDU.	5.1.1
	<i>command_id</i>	4	Integer	Value corresponding to <i>generic_nack</i> PDU.	5.1.2
	<i>command_status</i>	4	Integer	Error code corresponding to reason for sending the <i>generic_nack</i> .	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of original PDU or to NULL if the original PDU cannot be decoded.	5.1.4

Table 4-9: SMPP *generic_nack* PDU format

4.4 “SUBMIT_SM” Operation

This operation is used by an ESME to submit a short message to the SMSC for onward transmission to a specified short message entity (SME). The *submit_sm* PDU does **not** support the transaction message mode.

4.4.1 “SUBMIT_SM” Syntax

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

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>submit_sm</i>	5.1.2
	<i>command_status</i>	4	Integer	Not used. Set to NULL.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a Unique sequence number. The associated <i>submit_sm_resp</i> PDU will echo this sequence number.	5.1.4
M A N D A T O R Y P A R A M E T E R S	<i>service_type</i>	Var. max 6	C- Octet String	The <i>service_type</i> parameter can be used to indicate the SMS Application service associated with the message. Specifying the <i>service_type</i> allows the ESME to <ul style="list-style-type: none"> • avail of enhanced messaging services such as “replace by service” type • to control the teleservice used on the air interface. Set to NULL for default SMSC settings.	5.2.11
	<i>source_addr_ton</i>	1	Integer	Type of Number for source address. If not known, set to NULL (Unknown).	5.2.5
	<i>source_addr_npi</i>	1	Integer	Numbering Plan Indicator for source address. If not known, set to NULL (Unknown).	5.2.6
	<i>source_addr</i>	Var. max 21	C- Octet String	Address of SME which originated this message. If not known, set to NULL (Unknown).	5.2.8

Table 4-10: *submit_sm* PDU

	Field Name	Size octets	Type	Description	Ref.
M A N D A T O R Y P A R A M E T E R S	<i>dest_addr_ton</i>	1	Integer	Type of Number for destination.	5.2.5
	<i>dest_addr_npi</i>	1	Integer	Numbering Plan Indicator for destination.	5.2.6
	<i>destination_addr</i>	Var. max 21	C- Octet String	Destination address of this short message. For mobile terminated messages, this is the directory number of the recipient MS.	5.2.9
	<i>esm_class</i>	1	Integer	Indicates Message Mode & Message Type.	5.2.12
	<i>protocol_id</i>	1	Integer	Protocol Identifier. Network specific field.	5.2.13
	<i>priority_flag</i>	1	Integer	Designates the priority level of the message.	5.2.14
	<i>schedule_delivery_time</i>	1 or 17	C- Octet String	The short message is to be scheduled by the SMSC for delivery. Set to NULL for immediate message delivery.	5.2.15
	<i>validity_period</i>	1 or 17	C- Octet String	The validity period of this message. Set to NULL to request the SMSC default validity period.	5.2.16
	<i>registered_delivery</i>	1	Integer	Indicator to signify if an SMSC delivery receipt or an SME acknowledgement is required.	5.2.17
	<i>replace_if_present_flag</i>	1	Integer	Flag indicating if submitted message should replace an existing message.	5.2.18
<i>data_coding</i>	1	Integer	Defines the encoding scheme of the short message user data.	5.2.19	

Table 4-10: *submit_sm* PDU

	Field Name	Size octets	Type	Description	Ref.
M A N D A T O R Y P A R A M E T E R S	<i>sm_default_msg_id</i>	1	Integer	Indicates the short message to send from a list of pre-defined ('canned') short messages stored on the SMSC. If not using an SMSC canned message, set to NULL.	5.2.20
	<i>sm_length</i>	1	Integer	Length in octets of the <i>short_message</i> user data.	5.2.21
	<i>short_message</i>	Var. 0-254	Octet String	<p>Up to 254 octets of short message user data. The exact physical limit for <i>short_message</i> size may vary according to the underlying network.</p> <p>Applications which need to send messages longer than 254 octets should use the <i>message_payload</i> parameter. In this case the <i>sm_length</i> field should be set to zero.</p> <p>Note: The short message data should be inserted in either the <i>short_message</i> or <i>message_payload</i> fields. Both fields must not be used simultaneously.</p>	5.2.22

Table 4-10: *submit_sm* PDU

OPTIONAL PARAMETERS for SUBMIT_SM				
	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>user_message_reference</i>	TLV	ESME assigned message reference number.	5.3.2.17
	<i>source_port</i>	TLV	Indicates the application port number associated with the source address of the message. This parameter should be present for WAP applications.	5.3.2.20
	<i>source_addr_subunit</i>	TLV	The subcomponent in the destination device which created the user data.	5.3.2.2
	<i>destination_port</i>	TLV	Indicates the application port number associated with the destination address of the message. This parameter should be present for WAP applications.	5.3.2.21
	<i>dest_addr_subunit</i>	TLV	The subcomponent in the destination device for which the user data is intended.	5.3.2.1
	<i>sar_msg_ref_num</i>	TLV	The reference number for a particular concatenated short message.	5.3.2.22
	<i>sar_total_segments</i>	TLV	Indicates the total number of short messages within the concatenated short message.	5.3.2.23
	<i>sar_segment_seqnum</i>	TLV	Indicates the sequence number of a particular short message fragment within the concatenated short message.	5.3.2.24
	<i>more_messages_to_send</i>	TLV	Indicates that there are more messages to follow for the destination SME.	5.3.2.34
	<i>payload_type</i>	<u>TLV</u>	defines the type of payload (e.g. WDP, WCMP, etc.).	<u>5.3.2.10</u>

Table 4-10: *submit_sm* PDU

	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>message_payload</i>	TLV	<p>Contains the extended short message user data. Up to 64K octets can be transmitted.</p> <p>Note: The short message data should be inserted in either the <i>short_message</i> or <i>message_payload</i> fields. Both fields should not be used simultaneously.</p> <p>The <i>sm_length</i> field should be set to zero if using the <i>message_payload</i> parameter.</p>	5.3.2.32
	<i>privacy_indicator</i>	TLV	Indicates the level of privacy associated with the message.	5.3.2.14
	<i>callback_num</i>	TLV	A callback number associated with the short message. This parameter can be included a number of times for multiple callback addresses.	5.3.2.36
	<i>callback_num_pres_ind</i>	TLV	<p>Defines the callback number presentation and screening. If this parameter is present and there are multiple instances of the <i>callback_num</i> parameter then this parameter must occur an equal number of instances and the order of occurrence determines the particular <i>callback_num_pres_ind</i> which corresponds to a particular <i>callback_num</i>.</p>	5.3.2.37

Table 4-10: *submit_sm* PDU

	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>callback_num_atag</i>	TLV	Associates a displayable alphanumeric tag with the callback number. If this parameter is present and there are multiple instances of the <i>callback_num</i> parameter then this parameter must occur an equal number of instances and the order of occurrence determines the particular <i>callback_num_atag</i> which corresponds to a particular <i>callback_num</i> .	5.3.2.38
	<i>source_subaddress</i>	TLV	The subaddress of the message originator.	5.3.2.15
	<i>dest_subaddress</i>	TLV	The subaddress of the message destination.	5.3.2.16
	<i>user_response_code</i>	TLV	A user response code. The actual response codes are implementation specific.	5.3.2.18
	<i>display_time</i>	TLV	Provides the receiving MS with a display time associated with the message.	5.3.2.26
	<i>sms_signal</i>	TLV	Indicates the alerting mechanism when the message is received by an MS.	5.3.2.40
	<i>ms_validity</i>	TLV	Indicates validity information for this message to the recipient MS.	5.3.2.27
	<i>ms_msg_wait_facilities</i>	TLV	This parameter controls the indication and specifies the message type (of the message associated with the MWI) at the mobile station.	5.3.2.13
	<i>number_of_messages</i>	TLV	Indicates the number of messages stored in a mail box	5.3.2.39
	<i>alert_on_msg_delivery</i>	TLV	Request an MS alert signal be invoked on message delivery.	5.3.2.41
<i>language_indicator</i>	TLV	Indicates the language of an alphanumeric text message.	5.3.2.19	

Table 4-10: *submit_sm* PDU

	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>its_reply_type</i>	TLV	The MS user's reply method to an SMS delivery message received from the network is indicated and controlled by this parameter.	5.3.2.42
	<i>its_session_info</i>	TLV	Session control information for Interactive Teleservice.	5.3.2.43
	<i>ussd_service_op</i>	TLV	This parameter is used to identify the required USSD Service type when interfacing to a USSD system.	5.3.2.44

Table 4-10: *submit_sm* PDU

4.4.1.1 Source and Destination Addressing

The *submit_sm* PDU includes provision for both ‘source address’ and ‘destination address’. The ‘source address’ is comprised of the *source_addr_ton*, *source_addr_npi* and *source_addr* fields and ‘destination address’ is comprised of the *dest_addr_ton*, *dest_addr_npi* and *destination_addr* fields.

An ESME Transmitter may enter NULL values in the ‘source address’ fields. In this event, the SMSC may then substitute a default address for that particular ESME. This feature is designed for interfaces that are not normally familiar with the notion of a source address for a short message, e.g., paging systems, voice mail system.

4.4.1.2 Message Replace operation in “SUBMIT_SM”

Though SMPP offers a dedicated *replace_sm* operation, the *submit_sm* operation also facilitates replacement of a short message which has been previously submitted but has not yet been delivered to the designated destination.

The replace function can be activated in the *submit_sm* PDU by setting the *replace_if_present_flag* to 1 (one).

Alternatively, an SMSC administrator may define a specific *service_type* to provide ‘replace-if-present’ functionality. In this case, the replace function can be activated in the *submit_sm* PDU by setting the *service_type* field to the defined value.

For both methods of replacing a message using the *submit_sm* operation, the data contained in the short message found in the SMSC, whose source and destination addresses and *service_type* match those addresses specified in the latest *submit_sm* operation, will be replaced with the text contained in the *short_message* field of that latest *submit_sm* operation.

Note:

If the *submit_sm* PDU is used to replace an as yet undelivered message in the SMSC, and a matching message is not found in the SMSC, a new short message will be submitted to the SMSC.

If a matching message is not found when using the *replace_sm* operation, the *replace_sm* operation will not result in a new message being submitted to the SMSC - an SMPP error will be returned to the ESME in the *replace_sm_resp* PDU.

4.4.2 “SUBMIT_SM_RESP”

This is the response to the *submit_sm* PDU and has the following format:

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>submit_sm_resp</i>	5.1.2
	<i>command_status</i>	4	Integer	Indicates outcome of <i>submit_sm</i> request.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>submit_sm</i> PDU.	5.1.4
B O D Y	<i>message_id</i>	Var. max 9 33 <u>65</u>	C- Octet String	This field contains the SMSC 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.	5.2.23

Table 4-11: *submit_sm_resp* PDU

Note: The *submit_sm_resp* PDU Body is not returned if the *command_status* field contains a non-zero value.

4.5 “SUBMIT_MULTI” Operation

The *submit_multi* operation may be used to submit an SMPP message for delivery to multiple recipients or to one or more Distribution Lists. The *submit_multi* PDU does **not** support the transaction message mode.

4.5.1 “SUBMIT_MULTI” Syntax

Following is the format of the SMPP *submit_multi* PDU. The *command_id* field contains the command identifier code for *submit_multi*.

	Field Name	Size octets	Type	Description	Ref.
HEADERS	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>submit_multi</i>	5.1.2
	<i>command_status</i>	4	Integer	Not used. Set to NULL .	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>submit_multi_resp</i> PDU will echo the same sequence number.	5.1.4
MANDATORY PARAMETERS	<i>service_type</i>	Var max 6	C- Octet String	The <i>service_type</i> parameter can be used to indicate the SMS Application service associated with the message. Specifying the <i>service_type</i> allows the ESME to <ul style="list-style-type: none"> • avail of enhanced messaging services such as replace by service type • to control the teleservice used on the air interface. Set to NULL for default SMSC settings.	5.2.11
	<i>source_addr_ton</i>	1	Integer	Type of Number for source address. If not known, set to NULL (Unknown).	5.2.5
	<i>source_addr_npi</i>	1	Integer	Numbering Plan Indicator for source. If not known, set to NULL (Unknown).	5.2.6
	<i>source_addr</i>	Var. max 21	C- Octet String	Address of SME which originated this message. If not known, set to NULL (Unknown).	5.2.8

Table 4-12: *submit_multi* PDU

	Field Name	Size octets	Type	Description	Ref.
M A N D A T O R Y P A R A M E T E R S	<i>number_of_dests</i>	1	Integer	Number of destination addresses - indicates the number of <i>dest_address</i> structures that are to follow. A maximum of 254 destination addresses are allowed. Note: Set to 1 when submitting to one SME Address OR when submitting to one Distribution List.	5.2.24
	<i>dest_address(es)</i>	Var. n[2-24] See Ref.	See Ref.	Contains one or more (<i>number_of_dests</i>) SME addresses or/and Distribution List names.	Table 4-13
	<i>esm_class</i>	1	Integer	Indicates Message Mode & Message Type.	5.2.12
	<i>protocol_id</i>	1	Integer	Protocol Identifier. Network specific field.	5.2.13
	<i>priority_flag</i>	1	Integer	Designates the priority level of the message.	5.2.14
	<i>schedule_delivery_time</i>	1 or 17	C- Octet String	The short message is to be scheduled by the SMSC for delivery. Set to NULL for immediate message delivery.	5.2.15
	<i>validity_period</i>	1 or 17	C- Octet String	The validity period of this message. Set to NULL to request the SMSC default validity period.	5.2.16
	<i>registered_delivery</i>	1	Integer	Indicator to signify if an SMSC delivery receipt or an SME acknowledgement is required.	5.2.17
<i>replace_if_present_flag</i>	1	Integer	Reserved. Must be set to NULL.	5.2.18	
<i>data_coding</i>	1	Integer	Indicates the encoding scheme of the short message.	5.2.19	

Table 4-12: *submit_multi* PDU (Continued)

	Field Name	Size octets	Type	Description	Ref.
M A N D A T O R Y P A R A M E T E R S	<i>sm_default_msg_id</i>	1	Integer	Indicates the short message to send from a list of predefined (“canned”) short messages stored on the SMSC. If not using an SMSC predefined message, set to NULL.	5.2.20
	<i>sm_length</i>	1	Integer	Length in octets of the <i>short_message</i> user data.	5.2.21
	<i>short_message</i>	Var. 0-254	Octet String	Up to 254 octets of short message user data. The exact physical limit for <i>short_message</i> size may vary according to the underlying network. Applications which need to send messages longer than 254 octets should use the <i>message_payload</i> parameter. In this case the <i>sm_length</i> parameter should be set to zero. Note: The short message data should be inserted in either the <i>short_message</i> or <i>message_payload</i> parameters. Both parameters must not be used simultaneously.	5.2.22

Table 4-12: *submit_multi* PDU (Continued)

OPTIONAL PARAMETERS for SUBMIT_MULTI				
	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>user_message_reference</i>	TLV	ESME assigned message reference number.	5.3.2.17
	<i>source_port</i>	TLV	Indicates the application port number associated with the source address of the message. This parameter should be present for WAP applications.	5.3.2.20
	<i>source_addr_subunit</i>	TLV	The subcomponent in the destination device which created the user data.	5.3.2.2
	<i>destination_port</i>	TLV	Indicates the application port number associated with the destination address of the message. This parameter should be present for WAP applications	5.3.2.21
	<i>dest_addr_subunit</i>	TLV	The subcomponent in the destination device for which the user data is intended.	5.3.2.1
	<i>sar_msg_ref_num</i>	TLV	The reference number for a particular concatenated short message.	5.3.2.22
	<i>sar_total_segments</i>	TLV	Indicates the total number of short messages within the concatenated short message.	5.3.2.23
	<i>sar_segment_seqnum</i>	TLV	Indicates the sequence number of a particular short message fragment within the concatenated short message.	5.3.2.24
	<i>payload_type</i>	<u>TLV</u>	Defines the type of payload (e.g. WDP, WCMP, etc.)	<u>5.3.2.10</u>

Table 4-12: *submit_multi* PDU (Continued)

	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>message_payload</i>	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 <i>short_message</i> or <i>message_payload</i> fields. Both fields should not be used simultaneously. The <i>sm_length</i> field should be set to zero if using the <i>message_payload</i> parameter.	5.3.2.32
	<i>privacy_indicator</i>	TLV	Indicates the level of privacy associated with the message.	5.3.2.14
	<i>callback_num</i>	TLV	A callback number associated with the short message. This parameter can be included a number of times for multiple callback addresses.	5.3.2.36
	<i>callback_num_pres_ind</i>	TLV	Identifies the presentation and screening associated with the callback number. If this parameter is present and there are multiple instances of the <i>callback_num</i> parameter then this parameter must occur an equal number of instances and the order of occurrence determines the particular <i>callback_num_pres_ind</i> which corresponds to a particular <i>callback_num</i> .	5.3.2.37

Table 4-12: *submit_multi* PDU (Continued)

	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>callback_num_atag</i>	TLV	Associates a displayable alphanumeric tag with the callback number. If this parameter is present and there are multiple instances of the <i>callback_num</i> parameter then this parameter must occur an equal number of instances and the order of occurrence determines the particular <i>callback_num_atag</i> which corresponds to a particular <i>callback_num</i> .	5.3.2.38
	<i>source_subaddress</i>	TLV	The subaddress of the message originator.	5.3.2.15
	<i>dest_subaddress</i>	TLV	The subaddress of the message destination.	5.3.2.16
	<i>display_time</i>	TLV	Provides the receiving MS based SME with a display time associated with the message.	5.3.2.26
	<i>sms_signal</i>	TLV	Indicates the alerting mechanism when the message is received by an MS.	5.3.2.40
	<i>ms_validity</i>	TLV	Indicates validity information for this message to the recipient MS.	5.3.2.27
	<i>ms_msg_wait_facilities</i>	TLV	This parameter controls the indication and specifies the message type (of the message associated with the MWD) at the mobile station.	5.3.2.13
	<i>alert_on_msg_delivery</i>	TLV	Requests an MS alert signal be invoked on message delivery.	5.3.2.41
	<i>language_indicator</i>	TLV	Indicates the language of an alphanumeric text message.	5.3.2.19

Table 4-12: *submit_multi* PDU (Continued)

4.5.1.1 Destination Address definition

	Field Name	Size octets	Type	Description	Ref.
	<i>dest_flag</i>	1	Integer	Flag which will identify whether destination address is a Distribution List name or SME address.	5.2.25
	<i>SME Address</i> <i>or</i> <i>Distribution List Name</i>	See Ref.	See Ref.	Depending on <i>dest_flag</i> this could be an SME Address or a Distribution List Name.	Table 4-14& Table 4-15

Table 4-13: dest_address

	Field Name	Size octets	Type	Description	Ref.
	<i>dest_addr_ton</i>	1	Integer	Type of Number for destination SME.	5.2.5
	<i>dest_addr_npi</i>	1	Integer	Numbering Plan Indicator for destination SME.	5.2.6
	<i>destination_addr</i>	Var. max. 21	C- Octet String	Destination Address for this short message.	5.2.9

Table 4-14: SME_dest_address

4.5.1.2 Distribution List (DL) definition

	Field Name	Size octets	Type	Description	Ref.
	<i>dl_name</i>	Var. max. 21	C- Octet String	Name of Distribution List.	5.2.27

Table 4-15: DL Name

4.5.2 “SUBMIT_MULTI_RESP” Syntax

The following is the format of the SMPP *submit_multi_resp* PDU. The *command_id* field contains the command identifier code for *submit_multi_resp*.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>submit_multi_resp</i>	5.1.2
	<i>command_status</i>	4	Integer	Outcome of <i>submit_multi</i> request.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>submit_multi</i> PDU.	5.1.4
B O D Y	<i>message_id</i>	Var. max 65	C- Octet String	The SMSC message ID of the submitted message.	5.2.23
	<i>no_unsuccess</i>	1	Integer	The number of messages to destination SME addresses that were unsuccessfully submitted to the SMSC.	5.2.26
	<i>unsuccess_sme(s)</i>	Var. n[7-27] See Ref.	See Ref.	Contains one or more (<i>no_unsuccess</i>) SME address(es) or/and Distribution List names to which submission was unsuccessful.	Table 4-17

Table 4-16: *submit_multi_resp* PDU

4.5.2.1 Unsuccessful deliveries

Field	Size octets	Type	Description	Ref.
<i>dest_addr_ton</i>	1	Integer	Type of number for destination SME.	5.2.5
<i>dest_addr_npi</i>	1	Integer	Numbering Plan Indicator for destination SME	5.2.6
<i>destination_addr</i>	Var. max. 21	C-Octet String	Destination Address of destination SME	5.2.9
<i>error_status_code</i>	4	Integer	Indicates the success or failure of the <i>submit_multi</i> request to this SME address.	5.1.3

Table 4-17: Unsuccess_smes

4.6 “DELIVER_SM” Operation

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

In addition the SMSC uses the *deliver_sm* operation to transfer the following types of short messages to the ESME:-

- SMSC Delivery Receipt. A delivery receipt relating to a message which had been previously submitted with the *submit_sm* operation and the ESME had requested a delivery receipt via the *registered_delivery* parameter. The delivery receipt data relating to the original short message will be included in the *short_message* field of the *deliver_sm*. (Reference Appendix B for an example Delivery Receipt format.)
- SME Delivery Acknowledgement. The user data of the SME delivery acknowledgement is included in the *short_message* field of the *deliver_sm*
- SME Manual/User Acknowledgement. The user data of the SME delivery acknowledgement is included in the *short_message* field of the *deliver_sm*
- Intermediate Notification.

4.6.1 “DELIVER_SM” Syntax

The *deliver_sm* PDU has the same format as the *submit_sm* PDU. For this reason, some fields are unused.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>deliver_sm</i>	5.1.2
	<i>command_status</i>	4	Integer	Unused. Set to NULL.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>deliver_sm_resp</i> PDU should echo the same sequence number.	5.1.4
M A N D A T O R Y P A R A M E T E R S	<i>service_type</i>	Var max 6	C- Octet String	The <i>service_type</i> parameter can be used to indicate the SMS Application service associated with the message.	5.2.11
	<i>source_addr_ton</i>	1	Integer	Type of Number for source address. If not known, set to NULL (Unknown).	5.2.5
	<i>source_addr_npi</i>	1	Integer	Numbering Plan Indicator for source. If not known, set to NULL (Unknown).	5.2.6
	<i>source_addr</i>	Var. max 21	C- Octet String	Address of SME which originated this message. If not known, set to NULL (Unknown).	5.2.8
	<i>dest_addr_ton</i>	1	Integer	Type of number of destination SME.	5.2.5
	<i>dest_addr_npi</i>	1	Integer	Numbering Plan Indicator of destination SME.	5.2.6
	<i>destination_addr</i>	Var. max 21	C- Octet String	Destination address of destination SME.	5.2.9

Table 4-18: *deliver_sm* PDU

	Field Name	Size octets	Type	Description	Ref.
M A N D A T O R Y P A R A M E T E R S	<i>esm_class</i>	1	Integer	Indicates Message Type and enhanced network services.	5.2.12
	<i>protocol_id</i>	1	Integer	Protocol Identifier. Network Specific Field.	5.2.13
	<i>priority_flag</i>	1	Integer	Designates the priority level of the message.	5.2.14
	<i>schedule_delivery_time</i>	1	C-Octet String	This field is unused for <i>deliver_sm</i> . It must be set to NULL.	5.2.15
	<i>validity_period</i>	1	C-Octet String	This field is unused for <i>deliver_sm</i> . It must be set to NULL.	5.2.16
	<i>registered_delivery</i>	1	Integer	Indicates if an ESME acknowledgement is required.	5.2.17
	<i>replace_if_present_flag</i>	1	Integer	Not used in <i>deliver_sm</i> . It must be set to NULL.	5.2.18
	<i>data_coding</i>	1	Integer	Indicates the encoding scheme of the short message.	5.2.19
	<i>sm_default_msg_id</i>	1	Integer	Unused in <i>deliver_sm</i> . It must be set to NULL.	5.2.20
<i>sm_length</i>	1	Integer	Length of short message user data in octets.	5.2.21	

Table 4-18: *deliver_sm* PDU

	Field Name	Size octets	Type	Description	Ref.
M A N D A T O R Y P A R A M E T E R S	<i>short_message</i>	Var. 0-254	Octet String	<p>Up to 254 octets of short message user data.</p> <p>When sending messages longer than 254 octets the <i>message_payload</i> parameter should be used and the <i>sm_length</i> parameter should be set to zero.</p> <p>Note: The message data should be inserted in either the <i>short_message</i> or the <i>message_payload</i> parameters. Both parameters must not be used simultaneously.</p>	5.2.22

Table 4-18: *deliver_sm* PDU

OPTIONAL PARAMETERS for DELIVER_SM				
	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>user_message_reference</i>	TLV	A reference assigned by the originating SME to the message. In the case that the <i>deliver_sm</i> is carrying an SMSC delivery receipt, an SME delivery acknowledgement or an SME user acknowledgement (as indicated in the <i>esm_class</i> field), the <i>user_message_reference</i> parameter is set to the message reference of the original message.	5.3.2.17
	<i>source_port</i>	TLV	Indicates the application port number associated with the source address of the message. The parameter should be present for WAP applications.	5.3.2.20
	<i>destination_port</i>	TLV	Indicates the application port number associated with the destination address of the message. The parameter should be present for WAP applications.	5.3.2.21
	<i>sar_msg_ref_num</i>	TLV	The reference number for a particular concatenated short message.	5.3.2.22
	<i>sar_total_segments</i>	TLV	Indicates the total number of short messages within the concatenated short message.	5.3.2.23
	<i>sar_segment_seqnum</i>	TLV	Indicates the sequence number of a particular short message fragment within the concatenated short message.	5.3.2.24
	<i>user_response_code</i>	TLV	A user response code. The actual response codes are SMS application specific.	5.3.2.18

Table 4-18: *deliver_sm* PDU

	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>privacy_indicator</i>	TLV	Indicates a level of privacy associated with the message.	5.3.2.14
	<i>payload_type</i>	TLV	Defines the type of payload (e.g. WDP, WCMP, etc.)	5.3.2.10
	<i>message_payload</i>	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 <i>short_message</i> or <i>message_payload</i> fields. Both fields should not be used simultaneously. The <i>sm_length</i> field should be set to zero if using the <i>message_payload</i> parameter.	5.3.2.32
	<i>callback_num</i>	TLV	A callback number associated with the short message. This parameter can be included a number of times for multiple call back addresses.	5.3.2.36
	<i>source_subaddress</i>	TLV	The subaddress of the message originator.	5.3.2.15
	<i>dest_subaddress</i>	TLV	The subaddress of the message destination.	5.3.2.16
	<i>language_indicator</i>	TLV	Indicates the language of an alphanumeric text message.	5.3.2.19
	<i>its_session_info</i>	TLV	Session control information for Interactive Teleservice.	5.3.2.43
	<i>network_error_code</i>	TLV	Network Error Code. May be present for Intermediate Notifications and SMSC Delivery Receipts	5.3.2.31
<i>message_state</i>	TLV	Message State. Should be present for SMSC Delivery Receipts and Intermediate Notifications.	5.3.2.35	

Table 4-18: *deliver_sm* PDU

	Optional Parameter Name	Type	Description	Ref.
	<i>receipted_message_id</i>	TLV	SMSC message ID of receipted message Should be present for SMSC Delivery Receipts and Intermediate Notifications.	5.3.2.12

Table 4-18: *deliver_sm* PDU

4.6.2 “DELIVER_SM_RESP” Syntax

The following is the format of the SMPP *deliver_sm_resp* PDU.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>deliver_sm_resp</i>	5.1.2
	<i>command_status</i>	4	Integer	Indicates outcome of <i>deliver_sm</i> request.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of <i>deliver_sm</i> PDU.	5.1.4
B O D Y	<i>message_id</i>	1	C- Octet String	This field is unused and is set to NULL.	5.2.23

Table 4-19: *deliver_sm_resp* PDU

4.7 “DATA_SM” Operation

This command is used to transfer data between the SMSC and the ESME. It may be used by both the ESME and SMSC.

This command is an alternative to the *submit_sm* and *deliver_sm* commands. It is introduced as a new command to be used by interactive applications such as those provided via a WAP framework.

The ESME may use this command to request the SMSC to transfer a message to an MS. The SMSC may also use this command to transfer an MS originated message to an ESME.

In addition, the *data_sm* operation can be used to transfer the following types of special messages to the ESME:-

- SMSC Delivery Receipt.
- SME Delivery Acknowledgement. The user data of the SME delivery acknowledgement is included in the *short_message* field of the *data_sm*
- SME Manual/User Acknowledgement. The user data of the SME delivery acknowledgement is included in the *short_message* field of the *data_sm*
- Intermediate Notification.

4.7.1 “DATA_SM” Syntax

The following is the format of the SMPP *data_sm* PDU.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>data_sm</i>	5.1.2
	<i>command_status</i>	4	Integer	Not used. Set to NULL.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>data_sm_resp</i> PDU will echo the same sequence number. This parameter is used to facilitate transaction windowing.	5.1.4
M A N D A T O R Y P A R A M E T E R S	<i>service_type</i>	Var. max 6	C- Octet String	The <i>service_type</i> parameter can be used to indicate the SMS Application service associated with the message. Specifying the <i>service_type</i> allows the ESME/SMSC to <ul style="list-style-type: none"> to indicate the teleservice used on the air interface. 	5.2.11
	<i>source_addr_ton</i>	1	Integer	Type of Number for source address. If not known, set to “Unknown” (0x00).	5.2.5
	<i>source_addr_npi</i>	1	Integer	Numbering Plan Indicator for source address. If not known, set to “Unknown” (0x00).	5.2.6
	<i>source_addr</i>	Var. max 65	C- Octet String	Address of SME which originated this message.	5.2.8
	<i>dest_addr_ton</i>	1	Integer	Type of Number for destination.	5.2.5
	<i>dest_addr_npi</i>	1	Integer	Numbering Plan Indicator for destination.	5.2.6

Table 4-20: *data_sm* PDU

	Field Name	Size octets	Type	Description	Ref.
M A N D A T O R Y P A R S	<i>destination_addr</i>	Var. max 65	C- Octet String	Destination address of this short message. For mobile terminated messages, this is the directory number of the recipient MS.	5.2.9
	<i>esm_class</i>	1	Integer	Indicates Message Mode and Message Type.	5.2.12
	<i>registered_delivery</i>	1	Integer	Indicator for requesting a SMSC delivery receipt or an SME acknowledgement	5.2.17
	<i>data_coding</i>	1	Integer	Indicates the encoding scheme of the payload data	5.2.19

Table 4-20: *data_sm* PDU

OPTIONAL PARAMETERS for DATA_SM				
	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>source_port</i>	TLV	Indicates the application port number associated with the source address of the message. This parameter should be present for WAP applications.	5.3.2.20
	<i>source_addr_subunit</i>	TLV	The subcomponent in the destination device which created the user data.	5.3.2.2
	<i>source_network_type</i>	TLV	The correct network associated with the originating device.	5.3.2.4
	<i>source_bearer_type</i>	TLV	The correct bearer type for the delivering the user data to the destination.	5.3.2.6
	<i>source_telematics_id</i>	TLV	The telematics identifier associated with the source.	5.3.2.8
	<i>destination_port</i>	TLV	Indicates the application port number associated with the destination address of the message. This parameter should be present for WAP applications.	5.3.2.21
	<i>dest_addr_subunit</i>	TLV	The subcomponent in the destination device for which the user data is intended.	5.3.2.1
	<i>dest_network_type</i>	TLV	The correct network for the destination device.	5.3.2.3
	<i>dest_bearer_type</i>	TLV	The correct bearer type for the delivering the user data to the destination.	5.3.2.5
	<i>dest_telematics_id</i>	TLV	The telematics identifier associated with the destination.	5.3.2.7

Table 4-20: *data_sm* PDU

	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>sar_msg_ref_num</i>	TLV	The reference number for a particular concatenated short message.	5.3.2.22
	<i>sar_total_segments</i>	TLV	Indicates the total number of short messages within the concatenated short message.	5.3.2.23
	<i>sar_segment_seqnum</i>	TLV	Indicates the sequence number of a particular short message fragment within the concatenated short message.	5.3.2.24
	<i>more_messages_to_send</i>	TLV	Indicates that there are more messages to follow for the destination SME.	5.3.2.34
	<i>qos_time_to_live</i>	TLV	Time to live as a relative time in seconds from submission.	5.3.2.9
	<i>payload_type</i>	TLV	Defines the type of payload (e.g. WDP, WCMP, etc.).	5.3.2.10
	<i>message_payload</i>	TLV	Contains the message user data. Up to 64K octets can be transmitted.	5.3.2.32
	<i>set_dpf</i>	TLV	Indicator for setting Delivery Pending Flag on delivery failure.	5.3.2.29
	<i>receipted_message_id</i>	TLV	SMSC message ID of message being receipted. Should be present for SMSC Delivery Receipts and Intermediate Notifications.	5.3.2.12
	<i>message_state</i>	TLV	Message State. Should be present for SMSC Delivery Receipts and Intermediate Notifications.	5.3.2.35
<i>network_error_code</i>	TLV	Network error code. May be present for SMSC Delivery Receipts and Intermediate Notifications.	5.3.2.31	
<i>user_message_reference</i>	TLV	ESME assigned message reference number.	5.3.2.17	

Table 4-20: *data_sm* PDU

	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>privacy_indicator</i>	TLV	Indicates a level of privacy associated with the message.	5.3.2.14
	<i>callback_num</i>	TLV	A callback number associated with the short message. This parameter can be included a number of times for multiple call back addresses.	5.3.2.36
	<i>callback_num_pres_ind</i>	TLV	This parameter identifies the presentation and screening associated with the callback number. If this parameter is present and there are multiple instances of the <i>callback_num</i> parameter then this parameter must occur an equal number of instances and the order of occurrence determines the particular <i>callback_num_pres_ind</i> which corresponds to a particular <i>callback_num</i> .	5.3.2.37
	<i>callback_num_atag</i>	TLV	This parameter associates a displayable alphanumeric tag with the callback number. If this parameter is present and there are multiple instances of the <i>callback_num</i> parameter then this parameter must occur an equal number of instances and the order of occurrence determines the particular <i>callback_num_atag</i> which corresponds to a particular <i>callback_num</i> .	5.3.2.38
	<i>source_subaddress</i>	TLV	The subaddress of the message originator.	5.3.2.15
	<i>dest_subaddress</i>	TLV	The subaddress of the message destination.	5.3.2.16
	<i>user_response_code</i>	TLV	A user response code. The actual response codes are implementation specific.	5.3.2.18

Table 4-20: *data_sm* PDU

	Optional Parameter Name	Type	Description	Ref.
O P T I O N A L P A R A M E T E R S	<i>display_time</i>	TLV	Provides the receiving MS based SME with a display time associated with the message.	5.3.2.26
	<i>sms_signal</i>	TLV	Indicates the alerting mechanism when the message is received by an MS.	5.3.2.40
	<i>ms_validity</i>	TLV	Indicates validity information for this message to the recipient MS.	5.3.2.27
	<i>ms_msg_wait_facilities</i>	TLV	This parameter controls the indication and specifies the message type (of the message associated with the MWI) at the mobile station.	5.3.2.13
	<i>number_of_messages</i>	TLV	Indicates the number of messages stored in a mail box (e.g. voice mail box).	5.3.2.39
	<i>alert_on_msg_delivery</i>	TLV	Requests an MS alert signal be invoked on message delivery.	5.3.2.41
	<i>language_indicator</i>	TLV	Indicates the language of an alphanumeric text message.	5.3.2.19
	<i>its_reply_type</i>	TLV	The MS user's reply method to an SMS delivery message received from the network is indicated and controlled by this parameter.	5.3.2.42
	<i>its_session_info</i>	TLV	Session control information for Interactive Teleservice.	5.3.2.43

Table 4-20: *data_sm* PDU

4.7.2 “DATA_SM_RESP” Syntax

The following is the format of the SMPP *data_sm_resp* PDU.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU	5.1.1
	<i>command_id</i>	4	Integer	<i>data_sm_resp</i>	5.1.2
	<i>command_status</i>	4	Integer	Indicates outcome of <i>data_sm</i> request.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>data_sm</i> PDU.	5.1.4
B O D Y	<i>message_id</i>	Var. max 65	C- Octet String	This field contains the SMSC assigned message ID of the short message.	5.2.23
OPTIONAL PARAMETERS for DATA_SM_RESP					
	Optional Parameter Name		Type	Description	Ref.
	<i>delivery_failure_reason^a</i>		TLV	Include to indicate reason for delivery failure.	5.3.2.33
	<i>network_error_code^a</i>		TLV	Error code specific to a wireless network.	5.3.2.31
	<i>additional_status_info_text</i>		TLV	ASCII text giving a description of the meaning of the response	5.3.2.11
	<i>dpf_result^a</i>		TLV	Indicates whether the Delivery Pending Flag was set.	5.3.2.28

Table 4-21: *data_sm_resp* PDU

note

- a. These parameters are only relevant for transaction message mode.

4.8 “QUERY_SM” Operation

This command is issued by the ESME to query the status of a previously submitted short message.

The matching mechanism is based on the SMSC 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.

4.8.1 “QUERY_SM” Syntax

Following is the format of the SMPP *query_sm* PDU.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU	5.1.1
	<i>command_id</i>	4	Integer	<i>query_sm</i>	5.1.2
	<i>command_status</i>	4	Integer	Not used. Set to NULL.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>query_sm_resp</i> PDU should echo the same sequence number	5.1.4
M A N D A T O R Y P A R A M E T E R S	<i>message_id</i>	Var. Max 65	C-Octet String	Message ID of the message whose state is to be queried. This must be the SMSC assigned Message ID allocated to the original short message when submitted to the SMSC by the <i>submit_sm</i> , <i>data_sm</i> or <i>submit_multi</i> command, and returned in the response PDU by the SMSC.	5.2.23
	<i>source_addr_ton</i>	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. <i>submit_sm</i>). If not known, set to NULL.	5.2.5
	<i>source_addr_npi</i>	1	Integer	Numbering Plan Identity of message originator. This is used for verification purposes, and must match that supplied in the original request PDU (e.g. <i>submit_sm</i>). If not known, set to NULL.	5.2.6
	<i>source_addr</i>	Var. 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. <i>submit_sm</i>). If not known, set to NULL.	5.2.8

Table 4-22: *query_sm* PDU

4.8.2 “QUERY_SM_RESP” Syntax

Following is the format of the SMPP *query_sm_resp* PDU.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>query_sm_resp</i>	5.1.2
	<i>command_status</i>	4	Integer	Indicates outcome of <i>query_sm</i> request	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>query_sm</i> PDU.	5.1.4
M A N D A T O R Y P A R A M E T E R S	<i>message_id</i>	Var. max 65	C- Octet String	SMSC Message ID of the message whose state is being queried.	5.2.23
	<i>final_date</i>	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.	7.1.1
	<i>message_state</i>	1	Integer	Specifies the status of the queried short message.	5.2.28
	<i>error_code</i>	1	Integer	Where appropriate this holds a network error code defining the reason for failure of message delivery.	6.1

Table 4-23: *query_sm_resp* PDU

4.9 “CANCEL_SM” Operation

This command is issued by the ESME to cancel one or more previously submitted short messages that are still pending delivery. The command may specify a particular message to cancel, or all messages for a particular source, destination and *service_type* are to be cancelled.

- 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 given in the PDU are cancelled. If provided, *service_type* is included in this matching.

Where the original *submit_sm*, *data_sm* or *submit_multi* ‘source address’ was defaulted to NULL, then the source address in the *cancel_sm* command should also be NULL.

4.9.1 “CANCEL_SM” Syntax

Following is the format of the SMPP *cancel_sm* PDU.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>cancel_sm</i>	5.1.2
	<i>command_status</i>	4	Integer	Not used. Set to NULL.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>cancel_sm_resp</i> PDU should echo the same sequence number.	5.1.4
M A N D A T O R Y P A R A M E T E R S	<i>service_type</i>	Var. 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.	5.2.11
	<i>message_id</i>	Var. max 65	C- Octet String	Message ID of the message to be cancelled. This must be the SMSC assigned Message ID of the original message. Set to NULL if cancelling a group of messages.	5.2.23
	<i>source_addr_ton</i>	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.	5.2.5
	<i>source_addr_npi</i>	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.	5.2.6
	<i>source_addr</i>	Var. 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).	5.2.8

Table 4-24: *cancel_sm* PDU

	Field Name	Size octets	Type	Description	Ref.
M A N D A T O R Y P A R A M E T E R S	<i>dest_addr_ton</i>	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. <i>submit_sm</i>). May be set to NULL when the <i>message_id</i> is provided.	5.2.5
	<i>dest_addr_npi</i>	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 <i>message_id</i> is provided.	5.2.6
	<i>destination_addr</i>	Var. 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 <i>message_id</i> is provided.	5.2.9

Table 4-24: *cancel_sm* PDU (Continued)

4.9.2 “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.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>cancel_sm_resp</i>	5.1.2
	<i>command_status</i>	4	Integer	Indicates outcome of <i>cancel_sm</i> request.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to sequence number of <i>cancel_sm</i> PDU.	5.1.4

Table 4-25: *cancel_sm_resp* PDU

4.10 “REPLACE_SM” Operation

This command is issued by the ESME to replace a previously submitted short message that is still 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

4.10.1 “REPLACE_SM” Syntax

Following is the format of the SMPP *replace_sm* PDU. The *command_id* field contains the command identifier code for *replace_sm*.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>replace_sm</i>	5.1.2
	<i>command_status</i>	4	Integer	Not used. Set to NULL	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>replace_sm_resp</i> PDU should echo the same sequence number.	5.1.4
M A N D A T O R Y P A R A M E T E R S	<i>message_id</i>	Var. max 65	C- Octet String	SMSC message ID of the message to be replaced. This must be the message ID allocated to the original short message when submitted to the SMSC by the <i>submit_sm</i> command, and returned in the <i>submit_sm_resp</i> message by the SMSC.	5.2.23
	<i>source_addr_ton</i>	1	Integer	Type of Number of message originator. This is used for verification purposes, and must match that supplied in the corresponding <i>submit_sm</i> request. If not known, set to NULL.	5.2.5
	<i>source_addr_npi</i>	1	Integer	Numbering Plan Identity of message originator. This is used for verification purposes, and must match that supplied in the corresponding <i>submit_sm</i> request. If not known set to NULL.	5.2.6
	<i>source_addr</i>	Var. 1-21	C- Octet String	Originating address of the short message to be replaced. This is used for verification purposes, and must match that supplied in the corresponding <i>submit_sm</i> request.	5.2.8

Table 4-26: *replace_sm* PDU

	Field Name	Size octets	Type	Description	Ref.
M A N D A T O R Y P A R A M E T E R S	<i>schedule_delivery_time</i>	1 or 17	C-Octet String	New scheduled delivery time for the short message. Set to NULL, if updating of the original scheduled delivery time is not desired.	5.2.15
	<i>validity_period</i>	1 or 17	C-Octet String	New expiration time for the short message. Set to NULL, if updating of the original expiration time is not required.	5.2.16
	<i>registered_delivery</i>	1	Integer	New registered delivery parameter setting.	5.2.17
	<i>sm_default_msg_id</i>	1	Integer	New pre-defined (canned) message identifier.	5.2.20
	<i>sm_length</i>	1	Integer	Length of new short message in octets.	5.2.21
	<i>short_message</i>	Var. 0-254	Octet String	New short message to replace existing message.	5.2.22

Table 4-26: *replace_sm* PDU (Continued)

4.10.2 “REPLACE_SM_RESP” Syntax

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

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>replace_sm_resp</i>	5.1.2
	<i>command_status</i>	4	Integer	Indicates outcome of <i>replace_sm</i> request	5.1.3
	<i>sequence_number</i>	4	Integer	Expected to be the same sequence number of original <i>replace_sm</i> PDU.	5.1.4

Table 4-27: *replace_sm_resp* PDU

4.11 “ENQUIRE_LINK” Operation

This message can be sent by either the ESME or SMSC and is used to provide a confidence-check of the communication path between an ESME and an SMSC. On receipt of this request the receiving party should respond with an *enquire_link_resp*, thus verifying that the application level connection between the SMSC and the ESME is functioning. The ESME may also respond by sending any valid SMPP primitive.

4.11.1 “ENQUIRE_LINK” Syntax

The *enquire_link* PDU comprises the SMPP message header only.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU	5.1.1
	<i>command_id</i>	4	Integer	<i>enquire_link</i>	5.1.2
	<i>command_status</i>	4	Integer	Not used. Set to NULL	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>enquire_link_resp</i> PDU should echo the same sequence number	5.1.4

Table 4-28: *enquire_link* PDU

4.11.2 “ENQUIRE_LINK_RESP” Syntax

The *enquire_link_resp* PDU is used to reply to an *enquire_link* request. It comprises the SMPP message header only.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU	5.1.1
	<i>command_id</i>	4	Integer	<i>enquire_link_resp</i>	5.1.2
	<i>command_status</i>	4	Integer	Set to ESME_ROK (Success)	5.1.3
	<i>sequence_number</i>	4	Integer	Set to the same sequence number of original <i>enquire_link</i> PDU	5.1.4

Table 4-29: *enquire_link_resp* PDU

4.12 “ALERT_NOTIFICATION” Operation

This message is sent by the SMSC to the ESME, when the SMSC has detected that a particular mobile subscriber has become available and a delivery pending flag had been set for that subscriber from a previous *data_sm* operation.

It may be used for example to trigger a data content ‘Push’ to the subscriber from a WAP Proxy Server.

Note: There is no alert_notification_resp PDU.

4.12.1 “ALERT_NOTIFICATION” Syntax

Following is the format of the SMPP *alert_notification* PDU.

	Field Name	Size octets	Type	Description	Ref.
H E A D E R	<i>command_length</i>	4	Integer	Set to overall length of PDU.	5.1.1
	<i>command_id</i>	4	Integer	<i>alert_notification</i>	5.1.2
	<i>command_status</i>	4	Integer	Not used. Set to NULL.	5.1.3
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number.	5.1.4
M A N D A T O R Y P A R A M E T E R S	<i>source_addr_ton</i>	1	Integer	Type of number for the MS which has become available. If not known, set to NULL.	5.2.5
	<i>source_addr_npi</i>	1	Integer	Numbering Plan Indicator for the MS which has become available. If not known, set to NULL.	5.2.6
	<i>source_addr</i>	Var. max 65	C- Octet String	Address of MS which has become available.	5.2.8
	<i>esme_addr_ton</i>	1	Integer	Type of number for destination address which requested an alert on a particular MS becoming available. If not known, set to NULL.	5.2.5
	<i>esme_addr_npi</i>	1	Integer	Numbering Plan Indicator for ESME which requested an alert on a particular MS becoming available. If not known, set to NULL.	5.2.6
	<i>esme_addr</i>	Var. max 65	C- Octet String	Address of ESME which requested an alert on a particular MS becoming available.	5.2.10
OPTIONAL PARAMETERS for ALERT_NOTIFICATION					
	Optional Parameter Name		Type	Description	Ref.
	<i>ms_availability_status</i>		TLV	The status of the mobile station.	5.3.2.30

Table 4-30: alert_notification PDU

5. SMPP Parameter Definition

This section describes the parameters which can be specified in an SMPP command.

5.1 Command Header Parameters

5.1.1 `command_length`

The *command_length* parameter indicates the length in octets of the SMPP message. The SMPP message header (including the *command_length* field itself), the mandatory parameters and the optional parameters are all considered.

5.1.2 `command_id`

The *command_id* field identifies the type of message the SMPP PDU represents, for example, *submit_sm*, *query_sm* etc.

A command identifier is allocated to each SMPP request primitive. For reserved range value settings refer to Table 5-1:.

A response command identifier is allocated to each response primitive. For reserved range value settings refer to Table 5-1: (In general a response command identifier is identical to the corresponding request command identifier, but with bit 31 set).

5.1.2.1 SMPP Command set

The complete set of SMPP Command IDs and their associated values are defined in the following table.

Command ID	Value
<i>generic_nack</i>	0x80000000
<i>bind_receiver</i>	0x00000001
<i>bind_receiver_resp</i>	0x80000001
<i>bind_transmitter</i>	0x00000002
<i>bind_transmitter_resp</i>	0x80000002
<i>query_sm</i>	0x00000003
<i>query_sm_resp</i>	0x80000003
<i>submit_sm</i>	0x00000004
<i>submit_sm_resp</i>	0x80000004
<i>deliver_sm</i>	0x00000005
<i>deliver_sm_resp</i>	0x80000005
<i>unbind</i>	0x00000006
<i>unbind_resp</i>	0x80000006
<i>replace_sm</i>	0x00000007
<i>replace_sm_resp</i>	0x80000007
<i>cancel_sm</i>	0x00000008
<i>cancel_sm_resp</i>	0x80000008
<i>bind_transceiver</i>	0x00000009
<i>bind_transceiver_resp</i>	0x80000009
<i>Reserved</i>	0x0000000A 0x8000000A
<i>outbind</i>	0x0000000B
<i>Reserved</i>	0x0000000C - 0x00000014 0x8000000B - 0x80000014
<i>enquire_link</i>	0x00000015
<i>enquire_link_resp</i>	0x80000015
<i>Reserved</i>	0x00000016 - 0x00000020 0x80000016 - 0x80000020

Table 5-1: SMPP Command ID Values

Command ID	Value
<i>submit_multi</i>	0x00000021
<i>submit_multi_resp</i>	0x80000021
<i>Reserved</i>	0x00000022 - 0x000000FF 0x80000022 - 0x800000FF
<i>Reserved</i>	0x00000100
<i>Reserved</i>	0x80000100
<i>Reserved</i>	0x00000101 0x80000101
<i>alert_notification</i>	0x00000102
<i>Reserved</i>	0x80000102
<i>data_sm</i>	0x00000103
<i>data_sm_resp</i>	0x80000103
<i>Reserved for SMPP extension</i>	0x00000104 - 0x0000FFFF 0x80000104 - 0x8000FFFF
<i>Reserved</i>	0x00010000 - 0x000101FF 0x80010000 - 0x800101FF
<i>Reserved for SMSC Vendor</i>	0x00010200 - 0x000102FF 0x80010200 - 0x800102FF
<i>Reserved</i>	0x00010300 - 0xFFFFFFFF

Table 5-1: SMPP Command ID Values (Continued)

5.1.3 command_status

The *command_status* field of an SMPP message response indicates the success or failure of an 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 SMSC in the *command_status* field of the SMPP message header and in the *error_status_code* field of a *submit_multi_resp* message.

The complete set of SMPP Error Codes and their associated values are defined in the following table.

Error Code	Value	Description
ESME_ROK	0x00000000	No Error
ESME_RINVMSGLEN	0x00000001	Message Length is invalid
ESME_RINVCMDLEN	0x00000002	Command Length is invalid
ESME_RINVCMDID	0x00000003	Invalid Command ID
ESME_RINVBNDSTS	0x00000004	Incorrect BIND Status for given command
ESME_RALYBND	0x00000005	ESME Already in Bound State
ESME_RINVPRTFLG	0x00000006	Invalid Priority Flag
ESME_RINVREGDLVFLG	0x00000007	Invalid Registered Delivery Flag
ESME_RSYSERR	0x00000008	System Error
<i>Reserved</i>	0x00000009	Reserved
ESME_RINVSRCADR	0x0000000A	Invalid Source Address
ESME_RINVDSTADR	0x0000000B	Invalid Dest Addr
ESME_RINVMSGID	0x0000000C	Message ID is invalid
ESME_RBINDFAIL	0x0000000D	Bind Failed
ESME_RINVPASWD	0x0000000E	Invalid Password
ESME_RINVSYSID	0x0000000F	Invalid System ID
<i>Reserved</i>	0x00000010	Reserved
ESME_RCANCELFAIL	0x00000011	Cancel SM Failed
<i>Reserved</i>	0x00000012	Reserved
ESME_RREPLACEFAIL	0x00000013	Replace SM Failed

Table 5-2: SMPP Error Codes

Error Code	Value	Description
ESME_RMSGQFUL	0x00000014	Message Queue Full
ESME_RINVSERTYP	0x00000015	Invalid Service Type
<i>Reserved</i>	0x00000016- 0x00000032	Reserved
ESME_RINVNUMDESTS	0x00000033	Invalid number of destinations
ESME_RINVDLNAME	0x00000034	Invalid Distribution List name
<i>Reserved</i>	0x00000035- 0x0000003F	Reserved
ESME_RINVDESTFLAG	0x00000040	Destination flag is invalid (<i>submit_multi</i>)
<i>Reserved</i>	0x00000041	Reserved
ESME_RINVSUBREP	0x00000042	Invalid 'submit with replace' request (i.e. <i>submit_sm</i> with <i>replace_if_present_flag</i> set)
ESME_RINVESMCLASS	0x00000043	Invalid <i>esm_class</i> field data
ESME_RCNTSUBDL	0x00000044	Cannot Submit to Distribution List
ESME_RSUBMITFAIL	0x00000045	<i>submit_sm</i> or <i>submit_multi</i> failed
<i>Reserved</i>	0x00000046- 0x00000047	Reserved
ESME_RINVSRCNPI	0x00000048	Invalid Source address TON
ESME_RINVSRCNPI	0x00000049	Invalid Source address NPI
ESME_RINVDSTTON	0x00000050	Invalid Destination address TON
ESME_RINVDSTNPI	0x00000051	Invalid Destination address NPI
<i>Reserved</i>	0x00000052	Reserved
ESME_RINVSYSTYP	0x00000053	Invalid <i>system_type</i> field
ESME_RINVREPFLAG	0x00000054	Invalid <i>replace_if_present</i> flag
ESME_RINVNUMMSGS	0x00000055	Invalid number of messages
<i>Reserved</i>	0x00000056- 0x00000057	Reserved
ESME_RTHROTTLED	0x00000058	Throttling error (ESME has exceeded allowed message limits)
<i>Reserved</i>	0x00000059- 0x00000060	Reserved

Table 5-2: SMPP Error Codes

Error Code	Value	Description
ESME_RINVSCHED	0x00000061	Invalid Scheduled Delivery Time
ESME_RINVEXPIRY	0x00000062	Invalid message validity period (Expiry time)
ESME_RINVDFTMSGID	0x00000063	Predefined Message Invalid or Not Found
ESME_RX_T_APPN	0x00000064	ESME Receiver Temporary App Error Code
ESME_RX_P_APPN	0x00000065	ESME Receiver Permanent App Error Code
ESME_RX_R_APPN	0x00000066	ESME Receiver Reject Message Error Code
ESME_RQUERYFAIL	0x00000067	<i>query_sm</i> request failed
<i>Reserved</i>	0x00000068 - 0x000000BF	Reserved
ESME_RINVOPTPARSTREAM	0x000000C0	Error in the optional part of the PDU Body.
ESME_ROPTPARNOTALLWD	0x000000C1	Optional Parameter not allowed
ESME_RINVPARLEN	0x000000C2	Invalid Parameter Length.
ESME_RMISSINGOPTPARAM	0x000000C3	Expected Optional Parameter missing
ESME_RINVOPTPARAMVAL	0x000000C4	Invalid Optional Parameter Value
<i>Reserved</i>	0x000000C5 - 0x000000FD	Reserved
ESME_RDELIVERYFAILURE	0x000000FE	Delivery Failure (used for <i>data_sm_resp</i>)
ESME_RUNKNOWNERR	0x000000FF	Unknown Error
<i>Reserved for SMPP extension</i>	0x00000100- 0x000003FF	Reserved for SMPP extension
<i>Reserved for SMSC vendor specific errors</i>	0x00000400- 0x000004FF	Reserved for SMSC vendor specific errors
<i>Reserved</i>	0x00000500- 0xFFFFFFFF	Reserved

Table 5-2: SMPP Error Codes

5.1.4 **sequence_number**

A sequence number allows a response PDU to be correlated with a request PDU.

The associated SMPP response PDU must preserve this field.

The allowed *sequence_number* range is from 0x00000001 to 0x7FFFFFFF.

5.2 Mandatory SMPP Parameters

5.2.1 system_id

The *system_id* parameter is used to identify an ESME or an SMSC at bind time. An ESME *system_id* identifies the ESME or ESME agent to the SMSC. The SMSC *system_id* provides an identification of the SMSC to the ESME.

5.2.2 password

The *password* parameter is used by the SMSC to authenticate the identity of the binding ESME. The Service Provider may require ESME's to provide a password when binding to the SMSC. This password is normally issued by the SMSC system administrator.

The *password* parameter may also be used by the ESME to authenticate the identity of the binding SMSC (e.g. in the case of the *outbind* operation).

5.2.3 system_type

The *system_type* parameter is used to categorize the type of ESME that is binding to the SMSC. Examples include "VMS" (voice mail system) and "OTA" (over-the-air activation system).

Specification of the *system_type* is optional - some SMSC's may not require ESME's to provide this detail. In this case, the ESME can set the *system_type* to NULL.

5.2.4 interface_version

This parameter is used to indicate the version of the SMPP protocol. The following interface version values are defined:

Interface Version	Value
Indicates that the EMSE supports version 3.3 or earlier of the SMPP protocol.	0x00-0x33
Indicates that the ESME is supporting SMPP version 3.4	0x34
<i>All other values reserved</i>	

5.2.5 **addr_ton, source_addr_ton, dest_addr_ton, esme_addr_ton**

These fields define the Type of Number (TON) to be used in the SME address parameters. The following TON values are defined:

TON	Value
Unknown	00000000
International	00000001
National	00000010
Network Specific	00000011
Subscriber Number	00000100
Alphanumeric	00000101
Abbreviated	00000110
<i>All other values reserved</i>	

Table 5-3: TON values

5.2.6 **addr_npi, source_addr_npi, dest_addr_npi, esme_addr_npi**

These fields define the Numeric Plan Indicator (NPI) to be used in the SME address parameters. The following NPI values are defined:

NPI	Value
Unknown	00000000
ISDN (E163/E164)	00000001
Data (X.121)	00000011
Telex (F.69)	00000100
Land Mobile (E.212)	00000110
National	00001000
Private	00001001
ERMES	00001010
Internet (IP)	00001110
WAP Client Id (to be defined by WAP Forum)	<u>00010010</u>
<i>All other values reserved</i>	

Table 5-4: NPI values

5.2.7 **address_range**

The *address_range* parameter is used in the *bind_receiver* and *bind_transceiver* command to specify a set of SME addresses serviced by the ESME client. A single SME address may also be specified in the *address_range* parameter. UNIX Regular Expression notation should be used to specify a range of addresses (Refer to Appendix A.)

Messages addressed to any destination in this range shall be routed to the ESME.

Notes

For IP addresses, it is only possible to specify a single IP address. A range of IP addresses is not allowed. IP version 6.0 is not currently supported in this version of the protocol.

5.2.8 source_addr

Specifies the address of SME which originated this message. An ESME which is implemented as a single SME address, may set this field to NULL to allow the SMSC to default the source address of the submitted message.

Notes

An IP address is specified in “aaa.bbb.ccc.ddd” notation. IP version 6.0 is not supported in V3.4 of the SMPP protocol.

5.2.9 destination_addr

Specifies the destination SME address. For mobile terminated messages, this is the directory number of the recipient MS.

Notes

An IP address is specified in “aaa.bbb.ccc.ddd” notation. IP version 6.0 is not supported in V3.4 of the SMPP protocol.

5.2.10 esme_addr

Specifies the address of an ESME address to which an *alert_notification* should be routed.

Notes

An IP address is specified in “aaa.bbb.ccc.ddd” notation. IP version 6.0 is not supported in V3.4 of the SMPP protocol.

5.2.11 **service_type**

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 message 'replace_if_present' by service type (generic).
- Control the teleservice used on the air interface (e.g. ANSI-136/TDMA, IS-95/CDMA).

SMSC's may implicitly associate a "replace if present" function from the indicated *service_type* in a message submission operation, i.e., the SMSC will always replace an existing message pending delivery, that has the same originating and destination address as the submitted message. For example, an SMSC can ensure that a Voice Mail System using a *service_type* of "VMA" has at most one outstanding notification per destination MS by automatically invoking the "replace if present" function.

The following generic *service_types* are defined:

"(NULL)"	Default
"CMT"	Cellular Messaging
"CPT"	Cellular Paging
"VMN"	Voice Mail Notification
"VMA"	Voice Mail Alerting
"WAP"	Wireless Application Protocol
"USSD"	Unstructured Supplementary Services Data

All other values are carrier specific and are defined by mutual agreement between the SMSC Service Provider and the ESME application.

5.2.12 esm_class

The *esm_class* parameter is used to indicate special message attributes associated with the short message.

The *esm_class* parameter is encoded as follows in the *submit_sm*, *submit_multi* and *data_sm* (ESME -> SMSC) PDUs:

Bits	7 6 5 4 3 2 1 0	Meaning
		<u>Messaging Mode (bits 1-0)</u>
	x x x x x 0 0	Default SMSC Mode (e.g. Store and Forward)
	x x x x x 0 1	Datagram mode
	x x x x x 1 0	Forward (i.e. Transaction) mode
	x x x x x 1 1	Store and Forward mode (use to select Store and Forward mode if Default SMSC Mode is non Store and Forward)
		<u>Message Type (bits 5-2)</u>
	x x 0 0 0 0 x x	Default message Type (i.e. normal message)
	x x 0 0 1 0 x x	Short Message contains ESME Delivery Acknowledgement
	x x 0 1 0 0 x x	Short Message contains ESME Manual/User Acknowledgement
		<u>GSM Network Specific Features (bits 7-6)</u>
	0 0 x x x x x x	No specific features selected
	0 1 x x x x x x	UDHI Indicator (only relevant for MT short messages)
	1 0 x x x x x x	Set Reply Path (only relevant for GSM network)
	1 1 x x x x x x	Set UDHI and Reply Path (only relevant for GSM network)

The *esm_class* parameter is encoded as follows in a *deliver_sm* and *data_sm* (SMSC -> ESME) PDUs:

Bits	7 6 5 4 3 2 1 0	Meaning
		<u>Message Mode (bits 1-0)</u>
	x x x x x x x x	not applicable - ignore bits 0 and 1
		<u>Message Type (bits 5-2)</u>
	x x 0 0 0 0 x x	Default message Type (i.e. normal message)
	x x 0 0 0 1 x x	Short Message contains SMSC Delivery Receipt
	x x 0 0 1 0 x x	Short Message contains SME Delivery Acknowledgement
	x x 0 0 1 1 x x	reserved
	x x 0 1 0 0 x x	Short Message contains SME Manual/User Acknowledgment
	x x 0 1 0 1 x x	reserved
	x x 0 1 1 0 x x	Short Message contains Conversation Abort (Korean CDMA)
	x x 0 1 1 1 x x	reserved
	x x 1 0 0 0 x x	Short Message contains Intermediate Delivery Notification
		<i>all other values reserved</i>

	<u>GSM Network Specific Features (bits 7-6)</u>
0 0 x x x x x x	No specific features selected
0 1 x x x x x x	UDHI Indicator set
1 0 x x x x x x	Reply Path
1 1 x x x x x x	UDHI and Reply Path

all other values reserved

The default setting of the *esm_class* parameter is 0x00.

Notes

- If an ESME encodes GSM User Data Header information in the short message user data, it must set the UDHI flag in the *esm_class* field.
- If the SMSC delivers a short message that contains GSM User Data Header information encoded in the *short_message* or *message_payload* parameter, it must set the UDHI flag in the *esm_class* field.
- For GSM networks, the concatenation related optional parameters (*sar_msg_ref_num*, *sar_total_segments*, *sar_segment_seqnum*) or port addressing related optional parameters (*source_port*, *destination_port*) cannot be used in conjunction with encoded User Data Header in the *short_message* (user data) field. This means that the above listed optional parameters cannot be used if the User Data Header Indicator flag is set.

5.2.13 protocol_id

GSM

Set according to GSM 03.40 [GSM 03.40]

ANSI-136 (TDMA)

For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For ANSI-136 mobile originated messages, the SMSC should set this value to NULL.

IS-95 (CDMA)

For mobile terminated messages, this field is not used and is therefore ignored by the SMSC. For IS-95 mobile originated messages, the SMSC should set this value to NULL.

5.2.14 priority_flag

The *priority_flag* parameter allows the originating SME to assign a priority level to the short message.

Four Priority Levels are supported:

- 0 = Level 0 (lowest) priority
- 1 = Level 1 priority
- 2 = Level 2 priority
- 3 = Level 3 (highest) priority
- >3=Reserved

These are applied in different networks as follows:-

Priority Level	GSM ^a	ANSI-136	IS-95
0	non-priority	Bulk	Normal
1	priority	Normal	Interactive
2	priority	Urgent	Urgent
3	priority	Very Urgent	Emergency
<i>All other values reserved</i>			

Table 5-5: SMPP Message Priority values

- a. For GSM mobile terminated, messages with priority greater than Level 0 are treated as priority when making a delivery attempt (i.e. a delivery attempt is made even when MWD is set in the HLR).

5.2.15 `schedule_delivery_time`

This parameter specifies the scheduled time at which the message delivery should be first attempted.

It defines either the absolute date and time or relative time from the current SMSC time at which delivery of this message will be attempted by the SMSC.

It can be specified in either absolute time format or relative time format. The encoding of a time format is specified in Section 7.1.1.

5.2.16 `validity_period`

The *validity_period* parameter indicates the SMSC expiration time, after which the message should be discarded if not delivered to the destination. It can be defined in absolute time format or relative time format. The encoding of absolute and relative time format is specified in Section 7.1.1.

5.2.17 `registered_delivery`

The *registered_delivery* parameter is used to request an SMSC delivery receipt and/or SME originated acknowledgements. The following values are defined:

Bits	7 6 5 4 3 2 1 0	Meaning
		<u>SMSC Delivery Receipt (bits 1 and 0)</u>
	x x x x x 0 0	No SMSC Delivery Receipt requested (default)
	x x x x x 0 1	SMSC Delivery Receipt requested where final delivery outcome is delivery success or failure
	x x x x x 1 0	SMSC Delivery Receipt requested where the final delivery outcome is delivery failure
	x x x x x 1 1	reserved
		<u>SME originated Acknowledgement (bits 3 and 2)</u>
	x x x x 0 0 x x	No recipient SME acknowledgment requested (default)
	x x x x 0 1 x x	SME Delivery Acknowledgement requested
	x x x x 1 0 x x	SME Manual/User Acknowledgment requested
	x x x x 1 1 x x	Both Delivery and Manual/User Acknowledgment requested
		<u>Intermediate Notification (bit 5)</u>
	x x x 0 x x x x	No Intermediate notification requested (default)
	x x x 1 x x x x	Intermediate notification requested **

all other values reserved

The default setting of the *registered_delivery* parameter is 0x00.

Note: * A delivery receipt is returned only when the message has reached a non-delivered final state such as cancelled or undeliverable, etc.

** Support for Intermediate Notification Functionality is specific to the SMSC implementation and is beyond the scope of the SMPP Protocol Specification.

5.2.18 `replace_if_present_flag`

The *replace_if_present_flag* parameter is used to request the SMSC to replace a previously submitted message, that is still pending delivery. The SMSC will replace an existing message provided that the source address, destination address and *service_type* match the same fields in the new message.

0	Don't replace (default)
1	Replace
2 - 255	reserved

ESME applications that use this SMSC messaging function should use the same *service_type* and set the *replace_if_present_flag* parameter consistently to "1" for all messages, including the first message. This ensures that the SMSC has at most one message pending per destination SME for a particular application (e.g. voice mail notification).

5.2.19 data_coding

Bits	7 6 5 4 3 2 1 0	Meaning	Notes
	0 0 0 0 0 0 0 0	SMSC Default Alphabet	
	0 0 0 0 0 0 0 1	IA5 (CCITT T.50)/ASCII (ANSI X3.4)	b
	0 0 0 0 0 0 1 0	Octet unspecified (8-bit binary)	b
	0 0 0 0 0 0 1 1	Latin 1 (ISO-8859-1)	b
	0 0 0 0 0 1 0 0	Octet unspecified (8-bit binary)	a
	0 0 0 0 0 1 0 1	JIS (X 0208-1990)	b
	0 0 0 0 0 1 1 0	Cyrillic (ISO-8859-5)	b
	0 0 0 0 0 1 1 1	Latin/Hebrew (ISO-8859-8)	b
	0 0 0 0 1 0 0 0	UCS2 (ISO/IEC-10646)	a
	0 0 0 0 1 0 0 1	Pictogram Encoding	b
	0 0 0 0 1 0 1 0	ISO-2022-JP (Music Codes)	b
	0 0 0 0 1 0 1 1	reserved	
	0 0 0 0 1 1 0 0	reserved	
	0 0 0 0 1 1 0 1	Extended Kanji JIS(X 0212-1990)	b
	0 0 0 0 1 1 1 0	KS C 5601	b
	0 0 0 0 1 1 1 1	reserved	
	:		
	1 0 1 1 1 1 1 1	reserved	
	1 1 0 0 x x x x	GSM MWI control - see [GSM 03.38]	d
	1 1 0 1 x x x x	GSM MWI control - see [GSM 03.38]	d
	1 1 1 0 x x x x	reserved	
	1 1 1 1 x x x x	GSM message class control - see [GSM 03.38]	e

Notes:

- These coding schemes are common to GSM, TDMA and CDMA. The SMPP protocol allows ESME applications to use the same DCS value (i.e. the GSM 03.38 value) for all three technologies.
- In cases where a Data Coding Scheme is defined for TDMA and/ or CDMA but not defined for GSM, SMPP uses GSM 03.38 reserved values.
- There is no default setting for the data_coding parameter.
- The data_coding parameter will evolve to specify Character code settings only. Thus the recommended way to specify GSM MWI control is by specifying the relevant settings in the optional parameters *_ms_msg_wait_facilities* and *ms_validity*.
- The data_coding parameter will evolve to specify Character code settings only. Thus the recommended way to specify GSM message class control is by specifying the relevant setting in the optional parameter *dest_addr_subunit*.

5.2.20 `sm_default_msg_id`

The *sm_default_msg_id* parameter specifies the SMSC index of a pre-defined ('canned') message.

0	reserved
1 - 254	Allowed values
255	Reserved

5.2.21 sm_length

The *sm_length* parameter specifies the length of the *short_message* parameter in octets. The *sm_length* should be set to 0 in the *submit_sm*, *submit_multi*, and *deliver_sm* PDUs if the *message_payload* parameter is being used to send user data larger than 254 octets.

0	no user data in short message field
1-254	allowed
255	not allowed

5.2.22 short_message

The *short_message* parameter contains the user data. A maximum of 254 octets can be sent. ESME's should use the optional *message_payload* parameter in *submit_sm*, *submit_multi*, and *deliver_sm* to send larger user data sizes.

5.2.23 message_id

The unique message identifier reference assigned by the SMSC to each submitted short message. It is an opaque value and is set according to SMSC implementation. It is returned by the SMSC in the *submit_sm_resp*, *submit_multi_resp*, *deliver_sm_resp* and *data_sm_resp* PDUs and may be used by the ESME in subsequent SMPP operations relating to the short message, e.g. the ESME can use the *query_sm* operation to query a previously submitted message using the SMSC *message_id* as the message handle.

5.2.24 number_of_dests

The *number_of_dests* parameter indicates the number of *dest_address* structures that are to follow in the *submit_multi* operation.

A maximum of 254 destination address structures are allowed.

5.2.25 dest_flag

Flag which will identify whether destination address is a Distribution List (DL) name or SME address.

- 1 - SME Address
- 2 - Distribution List Name

5.2.26 no_unsuccess

The number of unsuccessful SME destinations to which delivery was attempted for a *submit_multi* operation.

5.2.27 dl_name

The reference name for a distribution list provisioned on the SMSC. Distribution list names are defined by mutual agreement between the SMSC and the ESME.

5.2.28 message_state

The following is a list of allowable states for a short message. The *message_state* value is returned by the SMSC to the ESME as part of the *query_sm_resp* PDU.

Message State	Value	Description
ENROUTE	1	The message is in enroute state.
DELIVERED	2	Message is delivered to destination
EXPIRED	3	Message validity period has expired.
DELETED	4	Message has been deleted.
UNDELIVERABLE	5	Message is undeliverable
ACCEPTED	6	Message is in accepted state (i.e. has been manually read on behalf of the subscriber by customer service)
UNKNOWN	7	Message is in invalid state
REJECTED	8	Message is in a rejected state

Table 5-6: Message States

5.3 SMPP Optional Parameter Description

5.3.1 Optional Parameter Tag Identifiers

Optional Parameters are fields, which may be optionally included in an SMPP message. Optional Parameters must always appear at the end of a message, in the "Optional Parameters" section of the SMPP PDU. However, they may be included in any convenient order within the "Optional Parameters" section of the SMPP PDU and need not be encoded in the order presented in this document.

For a particular SMPP PDU, the ESME or SMSC may include some, all or none of the defined optional parameters as required for the particular application context. For example a paging system may in an SMPP *submit_sm* operation, include only the "callback number" related optional parameters.

All SMPP optional parameters have a 16 bit Parameter Tag Identifier. The SMPP protocol defines the following Parameter Tag blocks:

0x0000	Reserved
0x0001 - 0x00FF	SMPP defined optional parameters
0x0100 - 0x01FF	Reserved
0x0200 - 0x05FF	SMPP defined optional parameters
0x0600 - 0x10FF	Reserved for SMPP Protocol Extension
0x1100 - 0x11FF	Reserved
0x1200 - 0x13FF	SMPP defined optional parameters
0x1400 - 0x3FFF	Reserved for SMSC Vendor specific optional parameters
0x4000 - 0xFFFF	Reserved

5.3.2 SMPP Optional Parameter Tag definitions

The SMPP supported Optional Parameters and their associated Tag Values are listed in Table 5-7 below. The optional parameters are described individually in the following sections.

Generic optional parameters may be applicable to all wireless network technologies i.e., GSM/iDEN, TDMA and CDMA.

Tag	Value	Wireless Network Technology
<i>dest_addr_subunit</i>	0x0005	GSM
<i>dest_network_type</i>	0x0006	Generic
<i>dest_bearer_type</i>	0x0007	Generic
<i>dest_telematics_id</i>	0x0008	GSM
<i>source_addr_subunit</i>	0x000D	GSM
<i>source_network_type</i>	0x000E	Generic
<i>source_bearer_type</i>	0x000F	Generic
<i>source_telematics_id</i>	0x0010	GSM
<i>qos_time_to_live</i>	0x0017	Generic
<i>payload_type</i>	0x0019	Generic
<i>additional_status_info_text</i>	0x001D	Generic
<i>receipted_message_id</i>	0x001E	Generic
<i>ms_msg_wait_facilities</i>	0x0030	GSM
<i>privacy_indicator</i>	0x0201	CDMA, TDMA
<i>source_subaddress</i>	0x0202	CDMA, TDMA
<i>dest_subaddress</i>	0x0203	CDMA, TDMA
<i>user_message_reference</i>	0x0204	Generic
<i>user_response_code</i>	0x0205	CDMA, TDMA
<i>source_port</i>	0x020A	Generic
<i>destination_port</i>	0x020B	Generic
<i>sar_msg_ref_num</i>	0x020C	Generic
<i>language_indicator</i>	0x020D	CDMA, TDMA
<i>sar_total_segments</i>	0x020E	Generic
<i>sar_segment_seqnum</i>	0x020F	Generic
<i>SC_interface_version</i>	0x0210	Generic

Table 5-7: Optional Parameter Tag values

Tag	Value	Wireless Network Technology
<i>callback_num_pres_ind</i>	0x0302	TDMA
<i>callback_num_atag</i>	0x0303	TDMA
<i>number_of_messages</i>	0x0304	CDMA
<i>callback_num</i>	0x0381	CDMA, TDMA, GSM, iDEN
<i>dpf_result</i>	0x0420	Generic
<i>set_dpf</i>	0x0421	Generic
<i>ms_availability_status</i>	0x0422	Generic
<i>network_error_code</i>	0x0423	Generic
<i>message_payload</i>	0x0424	Generic
<i>delivery_failure_reason</i>	0x0425	Generic
<i>more_messages_to_send</i>	0x0426	GSM
<i>message_state</i>	0x0427	Generic
<i>ussd_service_op</i>	0x0501	GSM (USSD)
<i>display_time</i>	0x1201	CDMA, TDMA
<i>sms_signal</i>	0x1203	TDMA
<i>ms_validity</i>	0x1204	CDMA, TDMA
<i>alert_on_message_delivery</i>	0x130C	CDMA
<i>its_reply_type</i>	0x1380	CDMA
<i>its_session_info</i>	0x1383	CDMA

Table 5-7: Optional Parameter Tag values (Continued)

5.3.2.1 dest_addr_subunit

The *dest_addr_subunit* parameter is used to route messages when received by a mobile station, for example to a smart card in the mobile station or to an external device connected to the mobile station.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>dest_addr_subunit</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	0x00 = Unknown (default) 0x01 = MS Display 0x02 = Mobile Equipment 0x03 = Smart Card 1 (expected to be SIM if a SIM exists in the MS) 0x04 = External Unit 1 5 to 255 = reserved

5.3.2.2 source_addr_subunit

The *source_addr_subunit* parameter is used to indicate where a message originated in the mobile station, for example a smart card in the mobile station or an external device connected to the mobile station.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>source_addr_subunit</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	see 5.3.2.1

5.3.2.3 dest_network_type

The *dest_network_type* parameter is used to indicate a network type associated with the destination address of a message. In the case that the receiving system (e.g. SMSC) does not support the indicated network type, it may treat this a failure and return a response PDU reporting a failure.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>dest_network_type</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	0x00 = Unknown 0x01 = GSM 0x02 = ANSI-136/TDMA 0x03 = IS-95/CDMA 0x04 = PDC 0x05 = PHS 0x06 = iDEN 0x07 = AMPS 0x08 = Paging Network 9 to 255 = reserved

5.3.2.4 source_network_type

The *source_network_type* parameter is used to indicate the network type associated with the device that originated the message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>source_network_type</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	see 5.3.2.3

5.3.2.5 dest_bearer_type

The *dest_bearer_type* parameter is used to request the desired bearer for delivery of the message to the destination address. In the case that the receiving system (e.g. SMSC) does not support the indicated bearer type, it may treat this a failure and return a response PDU reporting a failure.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>dest_bearer_type</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	0x00 = Unknown 0x01 = SMS 0x02 = Circuit Switched Data (CSD) 0x03 = Packet Data 0x04 = USSD 0x05 = CDPD 0x06 = DataTAC 0x07 = FLEX/ReFLEX 0x08 = Cell Broadcast (cellcast) 9 to 255 = reserved

5.3.2.6 source_bearer_type

The *source_bearer_type* parameter indicates the wireless bearer over which the message originated.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>source_bearer_type</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	see 5.3.2.5

5.3.2.7 dest_telematics_id

This parameter defines the telematic interworking to be used by the delivering system for the destination address. This is only useful when a specific *dest_bearer_type* parameter has also been specified as the value is bearer dependent. In the case that the receiving system (e.g. SMSC) does not support the indicated telematic interworking, it may treat this a failure and return a response PDU reporting a failure.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>dest_telematics_id</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	2	Integer	<i>to be defined</i>

5.3.2.8 source_telematics_id

The *source_telematics_id* parameter indicates the type of telematics interface over which the message originated.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>source_telematics_id</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	see 5.3.2.7

5.3.2.9 qos_time_to_live

This parameter defines the number of seconds which the sender requests the SMSC to keep the message if undelivered before it is deemed expired and not worth delivering. If the parameter is not present, the SMSC may apply a default value.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>qos_time_to_live</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	4	Integer	number of seconds for message to be retained by the receiving system.

5.3.2.10 payload_type

The *payload_type* parameter defines the higher layer PDU type contained in the message payload.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>payload_type</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	0 Default. In the case of a WAP application, the default higher layer message type is a WDP message. See [WDP] for details. 1 WCMP message. Wireless Control Message Protocol formatted data. See [WCMP] for details. <i>values 2 to 255 are reserved</i>

5.3.2.11 additional_status_info_text

The *additional_status_info_text* parameter gives an ASCII textual description of the meaning of a response PDU. It is to be used by an implementation to allow easy diagnosis of problems.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>additional_status_info_text</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1 - 256	C Octet String	Free format text to allow implementations to supply the most useful information for problem diagnosis. Maximum length is 256 octets.

5.3.2.12 receipted_message_id

The *receipted_message_id* parameter indicates the ID of the message being receipted in an SMSC Delivery Receipt. This is the opaque SMSC message identifier that was returned in the *message_id* parameter of the SMPP response PDU that acknowledged the submission of the original message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>receipted_message_id</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1 - 65	C Octet String	SMSC handle of the message being receipted.

5.3.2.13 **ms_msg_wait_facilities**

The *ms_msg_wait_facilities* parameter allows an indication to be provided to an MS that there are messages waiting for the subscriber on systems on the PLMN. The indication can be an icon on the MS screen or other MMI indication.

The *ms_msg_wait_facilities* can also specify the type of message associated with the message waiting indication.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>ms_msg_wait_facilities</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Bit mask	<p style="text-align: center;">Bits 7.....0 I00000TT</p> <p>This parameter controls the indication and specifies the message type (of the message associated with the MWI) at the mobile station.</p> <p>The Indicator is encoded in bit 7 as follows: 0 = Set Indication Inactive 1 = Set Indication Active</p> <p>The Type of Message associated with the MWI is encoded in bits 0 and 1 as follows:</p> <p>00 = Voicemail Message Waiting 01 = Fax Message Waiting 10 = Electronic Mail Message Waiting 11 = Other Message Waiting</p>

5.3.2.14 **privacy_indicator**

The *privacy_indicator* indicates the privacy level of the message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>privacy_indicator</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	0 = Privacy Level 0 (Not Restricted) (default) 1 = Privacy Level 1 (Restricted) 2 = Privacy Level 2 (Confidential) 3 = Privacy Level 3 (Secret) <i>values 4 to 255 are reserved</i>

Table 5-8: Privacy Indicator values

5.3.2.15 source_subaddress

The *source_subaddress* parameter specifies a subaddress associated with the originator of the message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	Var 2 - 23	Octet String	<p>The first octet of the data field is a Type of Subaddress tag and indicates the type of subaddressing information included, and implies the type and length of subaddressing information which can accompany this tag value in the data field.</p> <p>Valid Tag values are:</p> <p>00000001 - Reserved 00000010 - Reserved 10000000 - NSAP (Even) [ITUT X.213] 10001000 - NSAP (Odd) [ITUT X.213] 10100000 - User Specified All other values Reserved</p> <p>The remaining octets contain the subaddress.</p> <p>A NSAP address shall be encoded using the preferred binary encoding specified in [ITUT X.213]. In this case the subaddress field contains the Authority and Format Identifier.</p> <p>A User Specified subaddress is encoded according to user specification, subject to a maximum of 22 octets.</p>

5.3.2.16 **dest_subaddress**

The *dest_subaddress* parameter specifies a subaddress associated with the destination of the message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>dest_subaddress</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	Var 2 - 23	Octet String	See 5.3.2.15 for parameter encoding.

Note: The *dest_subaddress* parameter is not supported in the SMPP *submit_multi* PDU.

5.3.2.17 **user_message_reference**

A reference assigned by the originating SME to the short message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>user_message_reference</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	2	Integer	All values allowed.

5.3.2.18 user_response_code

A response code set by the user in a User Acknowledgement/Reply message. The response codes are application specific.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>user_response_code</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	0 to 255 (IS-95 CDMA) 0 to 15 (CMT-136 TDMA)

5.3.2.19 language_indicator

The *language_indicator* parameter is used to indicate the language of the short message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>language_indicator</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	0 = unspecified (default) 1 = english 2 = french 3 = spanish 4 = german 5 = Portuguese refer to [CMT-136] for other values

5.3.2.20 source_port

The *source_port* parameter is used to indicate the application port number associated with the source address of the message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>source_port</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	2	Integer	All values allowed.

5.3.2.21 destination_port

The *destination_port* parameter is used to indicate the application port number associated with the destination address of the message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>destination_port</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	2	Integer	All values allowed.

5.3.2.22 sar_msg_ref_num

The *sar_msg_ref_num* parameter is used to indicate the reference number for a particular concatenated short message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>sar_msg_ref_num</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	2	Integer	This parameter shall contain a originator generated reference number so that a segmented short message may be reassembled into a single original message. This allows the parallel transmission of several segmented messages. This reference number shall remain constant for every segment which makes up a particular concatenated short message. When present, the PDU must also contain the <i>sar_total_segments</i> and <i>sar_segment_seqnum</i> parameters. Otherwise this parameter shall be ignored.

5.3.2.23 sar_total_segments

The *sar_total_segments* parameter is used to indicate the total number of short messages within the concatenated short message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>sar_total_segments</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	This parameter shall contain a value in the range 1 to 255 indicating the total number of fragments within the concatenated short message. The value shall start at 1 and remain constant for every short message which makes up the concatenated short message. When present, the PDU must also contain the <i>sar_msg_ref_num</i> and <i>sar_segment_seqnum</i> parameters. Otherwise this parameter shall be ignored.

5.3.2.24 sar_segment_seqnum

The *sar_segment_seqnum* parameter is used to indicate the sequence number of a particular short message within the concatenated short message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>sar_segment_seqnum</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	This octet shall contain a value in the range 1 to 255 indicating the sequence number of a particular message within the concatenated short message. The value shall start at 1 and increment by one for every message sent within the concatenated short message. When present, the PDU must also contain the <i>sar_total_segments</i> and <i>sar_msg_ref_num</i> parameters. Otherwise this parameter shall be ignored.

5.3.2.25 **sc_interface_version**

The *sc_interface_version* parameter is used to indicate the SMPP version supported by the SMSC. It is returned in the bind response PDUs.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>sc_interface_version</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	values as per 5.2.4. (interface_version)

5.3.2.26 **display_time**

The *display_time* parameter is used to associate a display time of the short message on the MS.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>display_time</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	0 = Temporary 1 = Default (default) 2 = Invoke <i>values 3 to 255 are reserved</i>

5.3.2.27 ms_validity

The *ms_validity* parameter is used to provide an MS with validity information associated with the received short message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>ms_validity</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	0 = Store Indefinitely (default) 1 = Power Down 2 = SID based registration area 3 = Display Only <i>values 4 to 255 are reserved</i>

5.3.2.28 dpf_result

The *dpf_result* parameter is used in the *data_sm_resp* PDU to indicate if delivery pending flag (DPF) was set for a delivery failure of the short message..

If the *dpf_result* parameter is not included in the *data_sm_resp* PDU, the ESME should assume that DPF is not set.

Currently this parameter is only applicable for the Transaction message mode.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>dpf_result</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	0 = DPF not set 1 = DPF set <i>values 2 to 255 are reserved</i>

5.3.2.29 set_dpf

An ESME may use the *set_dpf* parameter to request the setting of a delivery pending flag (DPF) for certain delivery failure scenarios, such as

- MS is unavailable for message delivery (as indicated by the HLR)

The SMSC should respond to such a request with an *alert_notification* PDU when it detects that the destination MS has become available.

The delivery failure scenarios under which DPF is set is SMSC implementation and network implementation specific. If a delivery pending flag is set by the SMSC or network (e.g. HLR), then the SMSC should indicate this to the ESME in the *data_sm_resp* message via the *dpf_result* parameter.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>set_dpf</i>
<i>Length</i>	2	Integer	length of value part in octets
<i>Value</i>	1	Integer	0 = Setting of DPF for delivery failure to MS not requested 1 = Setting of DPF for delivery failure requested (default) <i>values 2 to 255 are reserved</i>

5.3.2.30 **ms_availability_status**

The *ms_availability_status* parameter is used in the *alert_notification* operation to indicate the availability state of the MS to the ESME.

If the SMSC does not include the parameter in the *alert_notification* operation, the ESME should assume that the MS is in an “available” state.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>ms_availability_status</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	0 = Available (Default) 1 = Denied (e.g. suspended, no SMS capability, etc.) 2 = Unavailable <i>values 3 to 255 are reserved</i>

5.3.2.31 network_error_code

The *network_error_code* parameter is used to indicate the actual network error code for a delivery failure. The network error code is technology specific.

Field	Size octets	Type	Description									
<i>Parameter Tag</i>	2	Integer	<i>network_error_code</i>									
<i>Length</i>	2	Integer	Length of value part in octets									
<i>Value</i>	3	Octet String	<table border="1"> <thead> <tr> <th>Sub-field</th> <th>Size</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Network Type</td> <td>1</td> <td>Integer</td> </tr> <tr> <td>Error Code</td> <td>2</td> <td>Integer</td> </tr> </tbody> </table> <p>The first octet indicates the network type. The following values are defined: 1 = ANSI-136 2 = IS-95 3 = GSM 4 = Reserved <i>All other values reserved.</i></p> <p>The remaining two octets specify the actual network error code appropriate to the network type.</p>	Sub-field	Size	Type	Network Type	1	Integer	Error Code	2	Integer
Sub-field	Size	Type										
Network Type	1	Integer										
Error Code	2	Integer										

5.3.2.32 message_payload

The *message_payload* parameter contains the user data.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>message_payload</i>
<i>Length</i>	2	Integer	Set to length of user data
<i>Value</i>	Variable	Octet String	Short message user data. The maximum size is SMSC and network implementation specific.

5.3.2.33 delivery_failure_reason

The *delivery_failure_reason* parameter is used in the *data_sm_resp* operation to indicate the outcome of the message delivery attempt (only applicable for transaction message mode). If a delivery failure due to a network error is indicated, the ESME may check the *network_error_code* parameter (if present) for the actual network error code.

The *delivery_failure_reason* parameter is not included if the delivery attempt was successful.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>delivery_failure_reason</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1	Integer	0 = Destination unavailable 1 = Destination Address Invalid (e.g. suspended, no SMS capability, etc.) 2 = Permanent network error 3 = Temporary network error <i>values 4 to are 255 reserved</i>

5.3.2.34 **more_messages_to_send**

The *more_messages_to_send* parameter is used by the ESME in the *submit_sm* and *data_sm* operations to indicate to the SMSC that there are further messages for the same destination SME. The SMSC may use this setting for network resource optimization.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>more_messages_to_send</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1		0 = No more messages to follow 1 = More messages to follow (default) <i>values 2 to 255 are reserved</i>

5.3.2.35 **message_state**

The *message_state* optional parameter is used by the SMSC in the *deliver_sm* and *data_sm* PDUs to indicate to the ESME the final message state for an SMSC Delivery Receipt.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>message_state</i>
<i>Length</i>	2	Integer	Length of value part in octets
<i>Value</i>	1		Values as per section 5.2.28

5.3.2.36 callback_num

The *callback_num* parameter associates a call back number with the message. In TDMA networks, it is possible to send and receive multiple callback numbers to/from TDMA mobile stations.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	Var 4 - 19	Octet String	<p>Bits 7.....0 0000000D (octet 1) 00000TTT (octet 2) 0000NNNN (octet 3) XXXXXXXX (octet 4) : : XXXXXXXX (octet N)</p> <p>The originating SME can set a Call Back Number for the receiving Mobile Station. The first octet contains the Digit Mode Indicator.</p> <p>Bit D=0 indicates that the Call Back Number is sent to the mobile as DTMF digits encoded in TBCD. Bit D=1 indicates that the Call Back Number is sent to the mobile encoded as ASCII digits.</p> <p>The 2nd octet contains the Type of Number (TON). Encoded as in section 5.2.5.</p> <p>The third octet contains the Numbering Plan Indicator (NPI). Encoded as specified in section 5.2.6</p> <p>The remaining octets contain the Call Back Number digits encoded as ASCII characters</p>

5.3.2.37 **callback_num_pres_ind**

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>callback_num_pres_ind</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Bit mask	<p style="text-align: center;">Bits 7.....0 0000ppss</p> <p>This parameter controls the presentation indication and screening of the CallbackNumber at the mobile station. If present, the <i>callback_num</i> parameter must also be present.</p> <p>The Presentation Indicator is encoded in bits 2 and 3 as follows:</p> <ul style="list-style-type: none"> 00 = Presentation Allowed 01 = Presentation Restricted 10 = Number Not Available 11 = Reserved <p>The Screening Indicator is encoded in bits 0 and 1 as follows:</p> <ul style="list-style-type: none"> 00 = User provided, not screened 01 = User provided, verified and passed 10 = User provided, verified and failed 11 = Network Provided.

5.3.2.38 callback_num_atag

The *callback_num_atag* parameter associates an alphanumeric display with the call back number.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>callback_num_atag</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	Var max 65	Octet string	<p>Alphanumeric display tag for call back number</p> <p>Bits 7.....0 EEEEEEEE (octet 1) XXXXXXXX (octet 2) : : XXXXXXXX (octet N)</p> <p>The first octet contains the encoding scheme of the Alpha Tag display characters. This field contains the same values as for Data Coding Scheme (see section 5.2.19). The following octets contain the display characters: There is one octet per display character for 7-bit and 8-bit encoding schemes. There are two octets per display character for 16-bit encoding schemes.</p>

5.3.2.39 **number_of_messages**

The *number_of_messages* parameter is used to indicate the number of messages stored in a mailbox.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>number_of_messages</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	0 to 99 = allowed values. <i>values 100 to 255 are reserved</i>

5.3.2.40 **sms_signal**

The *sms_signal* parameter is used to provide a TDMA MS with alert tone information associated with the received short message.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>sms_signal</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	2	Integer	Encoded as per [CMT-136]

5.3.2.41 alert_on_message_delivery

The *alert_on_message_delivery* parameter is set to instruct a MS to alert the user (in a MS implementation specific manner) when the short message arrives at the MS.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>alert_on_message_delivery</i>
<i>Length</i>	2	Integer	Length of Value part in octets (= 0)
<i>Value</i>	0		There is no Value part associated with this parameter.

5.3.2.42 its_reply_type

The *its_reply_type* parameter is a required parameter for the CDMA Interactive Teleservice as defined by the Korean PCS carriers [KORITS]. It indicates and controls the MS user's reply method to an SMS delivery message received from the ESME.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>its_reply_type</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Integer	0 = Digit 1 = Number 2 = Telephone No. 3 = Password 4 = Character Line 5 = Menu 6 = Date 7 = Time 8 = Continue <i>values 9 to 255 are reserved</i>

5.3.2.43 **its_session_info**

The *its_session_info* parameter is a required parameter for the CDMA Interactive Teleservice as defined by the Korean PCS carriers [KORITS]. It contains control information for the interactive session between an MS and an ESME.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>its_session_info</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	2	Octet String	<p>Bits 7.....0 SSSS SSSS (octet 1) NNNN NNNE (octet 2)</p> <p>Octet 1 contains the session number (0 - 255) encoded in binary. The session number remains constant for each session.</p> <p>The sequence number of the dialogue unit (as assigned by the ESME) within the session is encoded in bits 7..1 of octet 2.</p> <p>The End of Session Indicator indicates the message is the end of the conversation session and is encoded in bit 0 of octet 2 as follows: 0 = End of Session Indicator inactive. 1 = End of Session Indicator active.</p>

5.3.2.44 ussd_service_op

The *ussd_service_op* parameter is required to define the USSD service operation when SMPP is being used as an interface to a (GSM) USSD system.

Field	Size octets	Type	Description
<i>Parameter Tag</i>	2	Integer	<i>ussd_service_op</i>
<i>Length</i>	2	Integer	Length of Value part in octets
<i>Value</i>	1	Octet String	0 = PSSD indication 1 = PSSR indication 2 = USSR request 3 = USSN request 4 to 15 = reserved 16 = PSSD response 17 = PSSR response 18 = USSR confirm 19 = USSN confirm 20 to 31 = reserved 32 to 255 = reserved for vendor specific USSD operations

6. Network Implementation

6.1 Network Error Codes

The SMPP PDU, *query_sm_resp* contains an “error_code” field. The range of values this field may have, depends entirely on the underlying telecommunications network.

6.2 Maximum Message Length

Each network variation is limited to some fixed maximum length. This may be further affected by data coding scheme.

The SMSC, depending on configuration, may reject or truncate messages that exceed the network allowed maximum.

7. General Definitions

7.1 Time Definitions

7.1.1 Time Format

In this interface, all time/date related fields will be in ASCII with the following format:

“YYMMDDhhmmsstnnp” where

‘YY’	last two digits of the year (00-99)
‘MM’	month (01-12)
‘DD’	day (01-31)
‘hh’	hour (00-23)
‘mm’	minute (00-59)
‘ss’	second (00-59)
‘t’	tenths of second (0-9)
‘nn’	Time difference in quarter hours between local time (as expressed in the first 13 octets) and UTC (Universal Time Constant) time (00-48).
‘p’ - “+”	Local time is in quarter hours advanced in relation to UTC time.
“-”	Local time is in quarter hours retarded in relation to UTC time.
“R”	Local time is relative to the current SMSC time.

Note: Where responses are reported by the SMSC the local time of the SMSC will be given, and the format will be “YYMMDDhhmmss”, with the same definitions as above.

7.1.1.1 Absolute Time format

This is the default format used by SMPP. Scheduled delivery times and expiry times are specified in their global UTC format, including the quarter hour offset and direction symbol ‘+’ or ‘-’.

7.1.1.2 Relative Time Format

Relative Time can be indicated by setting the UTC orientation flag to 'R' instead of '+' or '-'. In this form, the SMSC interprets the time format as the number of years, months, days, hours, minutes and seconds from the current SMSC time. Values for tenths of seconds 't' and UTC offset 'nn' are ignored and should be set to '0' and '00' respectively.

For example, the following time format '020610233429000R':

- would be interpreted as a relative period of 2 years, 6 months, 10 days, 23 hours, 34 minutes and 29 seconds from the current SMSC time.

Note: An SMSC operator may choose to impose a limit on relative time offsets, thus either rejecting a message that exceeds such a limit or reducing the offset to the maximum relative time allowed.

For example:- a typical validity period might be 7 days and a typical scheduled delivery times might be 14 days from the submission time.

7.2 Timer Definitions

It is recommended that the following timers be implemented for SMPP transmitter and receiver sessions. All timers should be configurable.

Note: Definition of the various timer values is outside the scope of this SMPP Protocol Specification.

Timer	Action on expiration	Description
<i>session_init_timer</i>	The network connection should be terminated.	This timer specifies the time lapse allowed between a network connection being established and a <i>bind_transmitter</i> or <i>bind_receiver</i> request being sent to the SMSC. This timer should be active on the SMSC.
<i>enquire_link_timer</i>	An <i>enquire_link</i> request should be initiated.	This timer specifies the time lapse allowed between operations after which an SMPP entity should interrogate whether it's peer still has an active session. This timer may be active on either communicating SMPP entity (i.e. SMSC or ESME).
<i>inactivity_timer</i>	The SMPP session should be dropped.	This timer specifies the maximum time lapse allowed between transactions, after which period of inactivity, an SMPP entity may assume that the session is no longer active. This timer may be active on either communicating SMPP entity (i.e. SMSC or ESME).
<i>response_timer</i>	The entity which originated the SMPP Request may assume that Request has not been processed and should take the appropriate action for the particular SMPP operation in question.	This timer specifies the time lapse allowed between an SMPP request and the corresponding SMPP response. This timer may be active on either communicating SMPP entity (i.e. SMSC or ESME).

Appendix A

UNIX Regular Expressions

Full explanations of UNIX regular expressions can be found in section 5 of the standard on-line UNIX manuals (**man 5 regexp**). Furthermore, many UNIX books explain regular expressions and the various syntax used. This section gives useful and applicable examples of regular expressions in the context of the SMPP usage of same.

SMPP uses a regular expression in the *bind_receiver* PDU. The ESME uses this to provide routing criteria to the SMSC, namely, TON, NPI and routing_expr. The TON & NPI values are fixed values where the routing_expr itself is the regular expression.

- **^1234**
The '^' char is used to represent "beginning with", therefore ^1234 is interpreted as MSISDNs beginning with 1234. This allows an ESME specify a specific set of numbers based on a a given prefix common to all.
- **5678\$**
The '\$' char is used to represent "ending with", thus 5678\$ will match any MSISDN ending with 5678.
- **^123456\$**
A combination of '^' and '\$' at the beginning and end of a regular expression, is used to specify an absolute address, i.e the above expression will match MSISDNs beginning with and ending with 123456. The only value ever matched to this will in fact be '123456' itself.
- **[13579]\$**
values within [] denote a character class. The above expression will match MSISDNs ending with any of 1, 3, 5, 7 or 9. So this expression will match MSISDNs ending in an odd digit. If a '^' character is placed inside the '[', then the match is based on any character not in the specified class; e.g [^13579]\$ will match MSISDNs not ending with any of the specified digits.

Appendix B

Delivery Receipt Format

SMPP provides for return of an SMSC delivery receipt via the *deliver_sm* or *data_sm* PDU, which indicates the delivery status of the message.

The informational content of an SMSC Delivery Receipt may be inserted into the *short_message* parameter of the *deliver_sm* operation. The format for this Delivery Receipt message is SMSC vendor specific but following is a typical example of Delivery Receipt report.

“id:IIIIIIII sub:SSS dlvr:DDD submit date:YYMMDDhhmm done date:YYMMDDhhmm stat:DDDDDDD err:E Text:”

The fields of the above delivery receipt example are explained in the following table:

Field	Size (octets)	Type	Description
id	10	C-Octet String (Decimal)	The message ID allocated to the message by the SMSC when originally submitted.
sub	3	C-Octet String (Decimal)	Number of short messages originally submitted. This is only relevant when the original message was submitted to a distribution list. The value is padded with leading zeros if necessary.
dlvr	3	C-Octet Fixed Length String (Decimal)	Number of short messages delivered. This is only relevant where the original message was submitted to a distribution list. The value is padded with leading zeros if necessary.
submit date	10	C-Octet Fixed Length String	The time and date at which the short message was submitted. In the case of a message which has been replaced, this is the date that the original message was replaced. The format is as follows: YYMMDDhhmm where: YY = last two digits of the year (00-99) MM = month (01-12) DD = day (01-31) hh = hour (00-23) mm = minute (00-59)

Table B-1: Delivery Receipt Short Message Text Format

Field	Size (octets)	Type	Description
done date	10	C-Octet Fixed Length String	The time and date at which the short message reached it's final state. The format is the same as for the submit date.
stat	7	C-Octet Fixed Length String	The final status of the message. For settings for this field see Table B-2.
err	3	C-Octet Fixed Length String	Where appropriate this may hold a Network specific error code or an SMSC error code for the attempted delivery of the message. These errors are Network or SMSC specific and are not included here.
text	20	Octet String	The first 20 characters of the short message.

Table B-1: Delivery Receipt Short Message Text Format

Example Delivery Receipt message states:

Message State	Final Message States	Description
DELIVERED	DELIVRD	Message is delivered to destination
EXPIRED	EXPIRED	Message validity period has expired.
DELETED	DELETED	Message has been deleted.
UNDELIVERABLE	UNDELIV	Message is undeliverable
ACCEPTED	ACCEPTD	Message is in accepted state (i.e. has been manually read on behalf of the subscriber by customer service)
UNKNOWN	UNKNOWN	Message is in invalid state
REJECTED	REJECTD	Message is in a rejected state

Table B-2: Delivery Receipt Short Message Text Format

Appendix C

SMPP and Year 2000 Conformance

SMPP adopts the definition of Year 2000 conformity, as specified by the British Standards Institute. Further details on the British Standards Institute definition of Year 2000 conformity are available at:-

<http://www.bsi.org.uk/disc/year2000/2000.html>

Year 2000 Rollover Date Guideline as applied to SMPP

SMPP provides a two digit year field. Therefore, each SMPP entity **must** define a Year 2000 rollover date for 2-digit dates. As the Year 2000 rollover date will be defined for a computer platform and all its interfaces as a whole, a generic rollover date is not explicitly defined for the SMPP protocol.

SMPP developers should make the rollover date configurable within their implementations, to ensure compatibility with various SMSC platforms.

In the interest of maximising compatibility between SMPP products and platforms, it is strongly recommended that the following default guideline be adopted when implementing an SMPP interface:-

- ***The century rollover date shall be 'xx38'***

Thus, dates ending in ranges:-

- **38 to 99** shall be interpreted as meaning years **1938 to 1999** respectively
- **00 to 37** shall be interpreted as meaning years **2000 to 2037** respectively.