__This specification__ describes the situation of the Proximus network and services. It will be subject to modifications for corrections or when the network or the services will be modified. Please take into account that modifications can appear at any moment. Therefore, the reader is requested to check regularly with the most recent list of available specifications that the document in one's possession is the latest version.

Proximus can't be held responsible for any damages due to the use of a version of this specification which is not included in the most recent list of available specifications (list always available with a request to the e-mail address mentioned in the underneath paragraph).

Whilst every care has been taken in the preparation and publication of ***this document****, errors in content, typographical or otherwise, may occur. If you have remarks concerning its accuracy, please send a mail to the following address Proximus.uni.spec@Proximus.be and your remark will be transmitted to the right Proximus department.*

The User Network Interface Specifications published via Internet are available for your information but have no official value. The only documents with an official value are printed on a specific paper.

If you want ***to get an official version of this User Network Interface Specification****, please order it by sending your request by mail to Proximus.uni.spec@Proximus.be*

SMS

INTEROPERABILITY

Table of Contents

0. Document history

Every update of this document results in a complete new version with new version number and release date.

Version	Date	Main or important changes since previous version
1.0	8 JUL 2002	- First version
2.0	10 FEB 2003	- SMPP server functionality added
2.1	15 OCT 2004	- Some small updates

1. Scope

This document describes the details of implementing an interface to Proximus's network for the purpose of exchanging SMS messages between the fixed subscribers of Proximus and the subscribers of a MOLO or OLO operator.

The concept of this document is a description of the specific compliancy/deviations to the used SMPP specifications (see References [1][2]) with regards to the implementation. This implies that these referenced documents shall be followed unless otherwise noted.

For convenience, the structure of the following paragraphs follows the organization of the SMPP specification.

2. References

Whenever a date of edition is mentioned, the document with this date should be consulted. If no date is present, the latest version of this document should be consulted.

1. SMPP Interface specification v 3.3	Short message peer to peer (SMPP) interface specification. - Aldiscon - 14/01/1996 This document can be obtained via: http://www.smsforum.net
2. SMPP Interface specification v 3.4	Short message peer to peer (SMPP) interface specification. - SMPP developer forum - 12/10/1999 Issue 1.2. Link http://www.smsforum.net

3. Connection as SMPP SERVER

This part specifies the details in case of an interconnection where the M(OLO) operator acts as SMPP server and the Proximus network acts as SMPP client.

3.1. Physical

The physical connectivity shall be provided via a (redundant) leased line (PVC) TCP/IP connection. The IP port number which is to be used for the SMPP connection will be 8100.

3.2. Logical

The logical connection shall be implemented according to the SMPP v3.3 standards (see References [1]).
The Proximus platform - referenced in this document as ESME - shall act as client and the MOLO or OLO operator SMSC platform shall act as server.

3.3. Protocol messages

The communication between the ESME and the SMSC shall be asynchronous.

All messages sent, either from ESME to SMSC, or SMSC to ESME, will generate immediate responses.
See § 5.1 for the time-out value which is to be applied.

The timing by which the messages are send is limited. This is controlled via a throttling mechanism/timer.
If within the throttling period (see § 5.1) more then 1 message is sent, they shall be discarded by the other peer.

3.4. Use of Primitives

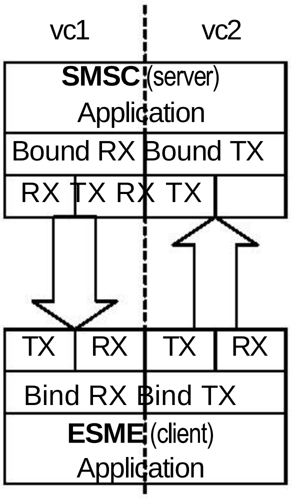
3.4.1. Initiation of Communication with SMSC

The ESME establishes communication with the SMSC, by an implementation specific mechanism. A model is displayed in the picture on the right.

Two 'virtual connections' (vc1 and vc 2) are required. One will be used for messages originating in the ESME system, and the response messages for them. (e.g. submit_sm, query_sm, cancel_sm, etc…), while the other will be used for messages originating in the SMSC and their responses (e.g. deliver_sm).

Once a 'virtual connection' has been established, each of the two processes on the ESME should send either a Bind-Transmitter request or a Bind-Receiver request. If a Bind Transmitter request is sent, the process on the SMSC that receives it will receive messages originating in the ESME system. If a Bind Receiver request is sent, the process on the SMSC that receives it will forward messages to the ESME.

Responses will invariably be returned on the same 'virtual connection' as the corresponding request messages.



Connectivity to the SMSC pool of the OLO network can occur in 2 ways:

1. The pool of SMSC's are accessed via a single centralised logical connection (1 IP address).
The ESME shall bind once to the centralised message application router.
One SEP is to be used for which the throttling timer shall apply.

2. Logical connectivity is required to each individual SMSC (multiple IP addresses)
The ESME shall bind (create a SEP) to each individual SMSC.
For each of the SEP's the same throttling timer shall apply.

The type of connectivity can vary depending on the OLO which is to be interconnected to. Proximus recommends **implementation 1** for the interconnection to the **MOLO or OLO operator SMSC infrastructure.**

3.4.2. Steady-State Communication with the SMSC

Full compliancy required.

3.4.3. Terminating Communication with the SMSC

If, at any time, either the ESME or the SMSC needs to terminate communications with the other, it should issue an "unbind" request over the appropriate 'virtual connection'.
Note: The SMPP AIM (on SMSC level) will also terminate if the ESME (or indeed the Network) releases the 'virtual connection'.

Once the communication is established (both RX and TX are bound between ESME and SMSC), it remains permanently established.
This implies that the ESME (nor the SMSC) shall never send an unbind command.

3.4.4. Error Handling and Retransmission

Full compliancy required.

In case of failure on the underlying communication network, the SMPP connection shall be closed by the ESME. The ESME shall attempt to reconnect in the same fashion as used during startup.

3.4.5. Protocol Message Types

3.4.5.1. ESME to SMSC

The interface shall support all messages indicated under this section with the exception of Query_last_msgs, Query_msg_details, Cancel_sm Replace_SM.

3.4.5.2. SMSC to ESME

The interface shall support all messages indicated under this section with the exception of Query_last_msgs_Resp, Query_msg_details_Resp, Cancel_sm_Resp, Replace_SM_Resp.

3.5. Message layouts

3.5.1. Definitions

Full compliancy required.

3.5.2. Message Header Format

Full compliancy required.

3.5.3. Message Body Formats

3.5.3.1. "BIND" operation

Full compliancy required.

The interface may act as either an ESME in it's own right or as an agent for the transport of messages to or from other ESME's.

The interface shall act as an ESME in it's own right.

3.5.3.1.1. "BIND_RECEIVER" Syntax

See table under § 5.1 for the value of the required parameters in this operation.

3.5.3.2. "UNBIND" Operation.

Full compliancy required.

Under normal operational circumstances this command shall not be sent by the ESME. Although it could be sent in case the SMPP connection must be closed for maintenance (or other) reasons.

3.5.3.3. "SUBMIT_SM" Operation.

This operation shall be used only to send messages (SMS, notification, etc…) from Proximus to other operator. Full compliancy required.

3.5.3.3.1. "SUBMIT_SM" Syntax

See table under § 5.1 for the value of the required parameters in this operation.

No specific value for the validity_period shall be provided (NULL) by the ESME. At SMSC level, the default value of 72 h shall be applied.

The "replace_if_present_flag" and" priority_flag" will not be used by the UMS. The following default values shall be applied:

- **replace_if_present_flag** = 0
- **priority_flag** = 0

This operation is also used for SMS delivery ack notification from ESME to SMSC in case of mobile to fixed SMS transmission.
In this case, the **esm_class** value shall contain **NULL** which is the only possible value for a submit_sm operation.

3.5.3.4. "SUBMIT_MULTI" Operation

This operation shall not be used.

3.5.3.5. "DELIVER_SM" Operation

This operation shall be used only to send SMS messages from other operator to Proximus. Full compliancy required.

3.5.3.5.1. "DELIVER_SM" Syntax

Full compliancy required.

No specific value for the validity_period shall be provided (NULL) by the SMSC. At ESME level, the value has to be specified at application level.

The "replace_if_present_flag" and "priority_flag" will not be used by the UMS. The following default values shall be applied:

- **replace_if_present_flag** = 0
- **priority_flag** = 0

This operation is also used for SMS final delivery receipt notification from SMSC to ESME in case of fixed to mobile SMS transmission.
In this case, the SMSC shall map the **esm_class** value to contain **0x04** to indicate that the deliver_sm message is a notification message and not a normal SMS message.

3.5.3.5.2. "DELIVER_SM_RESP" Syntax

Full compliancy required.

At the reception of a message (SMS, notifications, etc…) from a MOLO or OLO operator, the Proximus UMS platform shall verify the destination number. If this destination number is invalid (e.g. destination is black listed, etc…), an error code indicating the reason for failure (see table below) will be returned in the deliver_sm_resp message to the MOLO or OLO operator.

For normal SMS, the deliver_sm_resp shall be sent by ESME after the blacklist and other general restriction checks defined in the UMS service. Following error codes shall be used:

condition	***value***	***description***
destination number in black list	ESME_RINVDSTADR (0x0B)	Invalid destination address

3.5.3.6. QUERY

Full compliancy required.

3.5.3.7. "CANCEL_SM" Operation

This operation shall not be used.

3.5.3.8. "REPLACE_SM" Operation

This operation shall not be used.

3.5.3.9. "ENQUIRE_LINK" Operation

Full compliancy required.

If no protocol commands are sent over the link for more than a specific time (see § 5.1), the enquire_link command shall be sent by the ESME and the SMSC in order to verify the availability of the other end.

When the SMSC does not receive a enquire_link_resp within the standard response time, it shall tear down the SMPP allocated processes at SMSC level after which the ESME shall reinitiate the connection via the standard bind procedure.

When the ESME does not receive a enquire_link_resp within the standard response time, it shall unbind both the RX and TX process and reattempt to connect to the SMSC via the standard bind procedure.
If after a predefined number of unsuccessful bind attempts, the ESME fails to connect to the SMSC, then an alarm shall be generated at the UMS platform.

3.5.3.10. "PARAM_RETRIEVE" Operation

This operation shall not be used. The ESME shall not retrieve parameters from the SMSC.

4. Connection as SMPP client

This part specifies the details in case of an interconnection where the M(OLO) operator acts as SMPP client and the Proximus network acts as SMPP server.

This type of connection is to be used by sms service providers who interconnect to Proximus to offer short code SMS services. (eg. Voting via SMS,…)

4.1. Physical

The physical connection shall be established via a (redundant) TCP/IP connection from the ESME (external client) to the SMPP server. Therefore the SMPP server shall listen on port 8100 for incoming connections.

4.2. Logical

The logical connection shall be implemented according to the SMPP v3.4 standards (see References [2]).
The Proximus platform - hereafter referenced as SMPP server - shall act as server and the MOLO or OLO operator - hereafter referenced as ESME - shall act as SMPP client.

4.3. Initiation of Communication with SMPP server.

The ESME establishes communication with the SMSC by first setting up the TCP/IP connection and next issuing a bind command. Note that between these 2 actions, the timer session_init_timer (§ 5.1) must be respected.

It shall only be possible for the ESME to bind in transceiver mode. Other bind requests (eg. as transmitter or receiver shall be refused).

4.4. Terminating Communication with the SMPP server.

If, at any time, either the ESME or the SMPP server needs to terminate communications with the other, it should issue an "unbind" request.

4.5. SMPP protocol overview.

Following SMPP operations will be supported by the SMPP server:

4.5.1. ESME to SMPP server

The interface shall support all smpp operations as defined in the SMPP v3.4 standards (see References [2]). with the exception of:
Query_last_msgs, Query_msg_details, Cancel_sm Replace_SM.

The data_sm operation is supported with following restrictions:
- esm_class = 0x00
- short_message = normal text, max. 160 characters.

4.5.2. SMPP server to ESME

The interface shall support all smpp operations as defined in the SMPP v3.4 standards (see References [2]). with the exception of:
Query_last_msgs_Resp, Query_msg_details_Resp, Cancel_sm_Resp, Replace_SM_Resp.

4.5.3. Messages mode

Only the default "store and forward" mode (esm_class = 0x00) shall be supported to exchange SMS.

4.5.4. Enquire link operation.

The smpp server shall issue enquire_link commands to verify the status of the connected ESME. The SMPP server shall send these commands in case that no SMPP commands have been sent or received for a specific time out period. (see § 5.1)

4.6. SMPP parameter definition

Only the parameters used by the supported SMPP operations (§ 4.5) can be used.

4.6.1. Data coding

The supported character encoding types are:

- SMS default 7bit character set as defined in GSM 03.38 (7bit packing)
- Octet unspecified (8 bit binary) character mapping according the HP Roman 8 character set

4.7. Network implementation

4.7.1. Maximum message length.

The maximum message length will be 160 characters.

4.8. General definitions

Full compliancy required for the supported operations.

5. System configuration

5.1. SMPP Configurable Parameters

Parameter	***type of SMPP connection***		***value***	***description***
	server(*)	***client(*)***		
esme_ack_timeout_sec	X	X	30 sec.	timeout before which an acknowledgement to the corresponding command (eg. Submit_sm -> submit_sm_resp) must arrive at ESME level.
Smsc_ack_timeout_sec	X		30 sec.	timeout before which an acknowledgement to the corresponding command (eg. deliver_sm -> deliver_sm_resp) must arrive at SMSC level.
esme_buf_pool_sze	X		~500 messages	size of buffer pool for unacknowledged deliver_sm commands.
esme_equire_link_timer	X	X	90 sec.	Time out after which in case of no transmission of a protocol command, the ESME/SMSC shall send an enquire_link command to verify the status of the link.
esme_throttling_timer	X		1 sec.	min. Interval between 2 subsequent submit_sm messages.
Smpp_mux_network_conn_timer	X		30 sec.	time between sending a connection request to a receiver ESME and receiving an acknowledgement for the connection.
Smpp_mux_connection_timer	X		30 sec.	time between network connection and bind request.
session_init_timer		X	5 sec.	This timer specifies the time lapse allowed between a network connection being established and a bind_tranceiver request being sent to the SMSC. This timer should be active on the SMSC.
inactivity_timer	X	X	180 sec.	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).

(*) The applicability for a specific parameter is marked with a X.