



SMPP Server v.2.1

COMMERCIAL OFFER

www.netstyle.com.ua

info@netstyle.com.ua

+380 44 303-9338

Contents

Terms and abbreviations.....	2
Commercial offer.....	2
Proposed timeline.....	2
Financial offer.....	2
Technical solution.....	3
Solution overview.....	3
SMPP compliance.....	3
Extra functionality requirements.....	4
Environment compatibility requirements.....	4
Performance requirements.....	4
Authentication SQL API.....	4
SQL API for message queues access.....	4

Terms and abbreviations

- **ESME** – external short message entity
- **SM** – short message
- **MO** – mobile originated
- **MT** – mobile terminated
- **DLR** – delivery report
- **SMS Platform** – core SMS platform implemented by SMS Trade
- **SMPP Server** – SMPP v3.3/v3.4 compliant server implemented by Net Style

Commercial offer

Financial offer

- SMPP server functionality: \$5,000

Total project cost will be \$5,000 without taxes.

Technical solution

Solution overview

SMPP server is a standalone application providing the following functionality:

- listening on TCP port for incoming SMPP connections from ESME;
- authenticate incoming connections from ESME by system-ID and password;
- receive MT SM from ESME and put messages in MT queue;
- poll MO queue for MO SM and DLR and send to proper ESME.

Startup parameters are defined in configuration file may be changed with average editor.

All the system events are logged using syslog.

Detailed statistics is stored in the CDR SQL table and may be used for reporting.

SMPP compliance

- SMPP server supports 3.3 and 3.4 versions of protocol.
- SMPP server supports receiver, transmitter and transceiver modes.
- SMPP server supports the following data coding:
 - default 7 bit GSM alphabet as defined in GSM 03.38
 - 8 bit binary messages
 - UCS-2BE unicode messages
- SMPP server supports message_class in MO/MT messages.
- SMPP server supports UDH in MO/MT messages.
- SMPP server supports rate limitation for each ESME.
- SMPP server supports throttling in case of total allowed bandwidth is less than sum of rate limits for active ESME connections.
- SMPP server supports delivery reports forwarding to ESME.
- SMPP server supports access restriction for only allowed IPv4 addresses list.
- SMPP server supports limit of simultaneous connections from ESME.
- SMPP server supports enquire_link handling.

Extra functionality requirements

- SMPP server provides API to access realtime server statistics
- SMPP server stores information of all messages to CDR (files or DBMS)
- SMPP server provides API for manual connections closing
- SMPP server provides storing all SMPP traffic in files for debug purposes

Environment compatibility requirements

- SMPP server must be compatible with Linux system (i586 or x86_64 architecture).
- SMPP server must be compatible with MySQL server (version 5.0 or more).

Performance requirements

- SMPP server correctly works on Linux based system with 1GB RAM
- SMPP server handle up to 1000 simultaneous SMPP sessions.
- SMPP server process up to 1000 SMS per second on Xeon 3.0 GHz, 2GB RAM, SAS HDD.

Authentication SQL API

For authentication and authorization SMS Platform must provide SQL API containing the following data.

- **esme_id** (integer) – unique record identifier of ESME descriptor
- **system_id** (string) – unique System-ID of the client's ESME

- **password** (string) – password of the client's ESME
- **bandwidth** (integer) – allowed maximum bandwidth for ESME (MT SM/sec)
- **allowed_ip** (string) – list of allowed IP addresses (comma-separated)
- **allowed_src** (string) – list of allowed source addresses for the ESME
- **max_connections** (integer) – number of simultaneous connections from the ESME
- **active** (boolean) – flag indicating that ESME is allowed for operations.

SQL API for message queues access

For passing messages between SMPP server and SMS Platform must be used SQL table of the following structure.

- **id** (integer) – unique message identifier in
- **msg_type** (string) – message type (MO, MT or DLR)
- **esme_id** (integer) – ESME identifier (reference to the table containing ESME descriptors)
- **received** (timestamp) – when the message was received by SMPP server
- **processed** (timestamp) – when the message was processed by SMPP server
- **src_addr** (string) – source address
- **dst_addr** (string) – destination address
- **body** (string) – message body as hexadecimal byte string
- **coding** (integer) – coding (0 – 7bit, 1 – binary, 2 – UCS-2BE)
- **udh** (string) – UDH as hexadecimal byte string
- **mwi** (integer) – message waiting indication (the same as in Kannel), NULL means empty
- **mclass** (integer) – message_class (the same as in Kannel), NULL means empty
- **message_id** (string) – externally used message ID (UUID)
- **validity** (integer) – validity_period in minutes
- **deferred** (integer) – deferred delivery timeout in minutes
- **registered_delivery** (integer) – 1 if DLR expected, 0 - otherwise
- **service_type** (string) – service_type as defined in SMPP specification.
- **extra** – optional parameters as JSON encoded string (e.g. SMPP optional TLV)

To keep queue small and fast messages must be deleted from queue table after processing.