

– SMPP – PROTOCOLO E APLICAÇÕES

Referência		
Data	revisão	Documento
12/10/1999	1.2	Short message peer to peer protocol specification v3.4

Revisão		
Data	revisão	Documento
04/12/2002	1.0	SMPP – Protocolo e Aplicações

Registro		
Data	Nº de Registro	Cartório
04/12/2002	785545	CARTÓRIO DO 2º OFÍCIO DO REGISTRO DE TÍTULOS E DOCUMENTOS - Cartório Cecivaldo Bentes

Observações
A cópia e utilização total ou parcial deste documento estão autorizadas, desde que a fonte seja citada integralmente, constando nome completo do autor e data de registro. O descumprimento desta observação está sujeito às penalidades previstas por lei.

1 - INTRODUÇÃO	4
1.1 - OVERVIEW SMPP	4
2 - OVERVIEW DO PROTOCOLO SMPP.....	6
2.1 - DEFINIÇÕES DO PROTOCOLO SMPP	6
2.2 - DESCRIÇÃO DE UMA SESSÃO SMPP	7
2.2.1. O OUTBIND.....	8
2.3 - O PDU SMPP	8
2.4 - A CAMADA DE CONEXÕES DE REDE SMPP.....	10
2.5 - MENSAGENS ENVIADAS DA ESME PARA O SMSC.....	10
2.5.1. Respostas de mensagens SMPP do SMSC para a ESME.....	10
2.5.2. Exemplo de uma sessão típica SMPP – ESME Transmitter.....	11
2.6 - MENSAGENS ENVIADAS PELO SMSC PARA A ESME	12
2.6.1. Respostas de mensagens SMPP da ESME para o SMSC.....	12
2.6.2. Exemplo de uma sessão típica SMPP – ESME Receiver.....	12
2.7 - ENVIO DE MENSAGENS BIDIRECIONAL ENTRE UMA ESME E UM SMSC.....	13
2.7.1. Exemplo de uma sessão típica SMPP – ESME Transceiver.....	13
2.7.2. Gerenciamento de erros no SMPP.....	14
2.8 - TEMPORIZADORES SMPP	14
2.9 - TIPOS DE MENSAGENS.....	14
2.9.1. SMSC Delivery Receipt.....	14
2.9.2. Notificação Intermediária.....	15
2.9.3. SME Deliver Acknowledgement.....	15
3 - TIPOS DE PDU'S SMPP E DEFINIÇÕES DE FORMATO	16
3.1 - DEFINIÇÕES DOS TIPOS DE DADOS DO PDU SMPP.....	16
3.1.1. Notação do tamanho dos campos dos parâmetros SMPP.....	16
3.2 - OVERVIEW DO FORMATO DO PDU SMPP	17
3.2.1. Layout do PDU SMPP	18
3.2.2. Tamanho do PDU SMPP	19
3.2.3. Parâmetros Opcionais.....	19
3.2.4. Formato dos parâmetros opcionais	20
3.2.5. Premissas para a compatibilidade do SMPP com versões mais avançadas.....	20
3.2.6. Premissas para a compatibilidade do SMPP com versões mais atrasadas	21
4 - DEFINIÇÃO DO PDU SMPP	22
4.1 - OPERAÇÃO DE BIND	22
4.1.1. ESME Transmitter.....	22
4.1.1.1. Sintaxe do BIND_TRANSMITTER	22
4.1.1.2. Sintaxe do BIND_TRANSMITTER_RESP	23
4.1.2. ESME Receiver.....	24
4.1.2.1. Sintaxe do BIND_RECEIVER.....	24
4.1.2.2. Sintaxe do BIND_RECEIVER_RESP	25
4.1.3. ESME Transceiver	25
4.1.3.1. Sintaxe do BIND_TRANSCEIVER.....	26
4.1.3.2. Sintaxe do BIND_TRANSCEIVER_RESP	27
4.1.4. Operação de OUTBIND.....	27
4.1.4.1. Sintaxe do OUTBIND.....	27
4.2 - OPERAÇÃO DE UNBIND	28
4.2.1. Sintaxe do UNBIND	28
4.2.2. Sintaxe do UNBIND_RESP.....	28
4.3 - GENERIC_NACK PDU	28
4.3.1. Sintaxe do PDU GENERIC_NACK.....	29
4.4 - OPERAÇÃO DE SUBMIT_SM	29
4.4.1. Sintaxe do SUBMIT_SM.....	30
4.4.1.1. Source and Destination Addressing.....	31
4.4.1.2. Operação de reposição de mensagens com SUBMIT_SM.....	32

4.4.1.3.	Sintaxe do SUBMIT_SM_RESP.....	32
4.5 -	OPERAÇÃO DE DELIVER_SM	33
4.5.1.	Sintaxe do DELIVER_SM.....	33
4.5.2.	Sintaxe do DELIVER_SM_RESP	35
4.6 -	OUTRAS OPERAÇÕES DO SMPP	35
4.6.1.	DATA_SM	35
4.6.2.	QUERY_SM.....	35
4.6.3.	CANCEL_SM	36
4.6.4.	REPLACE_SM	36
4.6.5.	ENQUIRE_LINK.....	36
4.6.6.	ALERT_NOTIFICATION.....	36
5 -	DEFINIÇÃO DOS PARÂMETROS SMPP.....	37
5.1 -	PARÂMETROS DO CABEÇALHO SMPP (COMMAND HEADER)	37
5.1.1.	Command_length	37
5.1.2.	Command_id	37
5.1.2.1.	Conjunto de comandos SMPP.....	37
5.1.3.	Command_status.....	39
5.1.4.	Sequence_number	41
5.2 -	PARÂMETROS MANDATÓRIOS DO SMPP	41
5.2.1.	System_id	41
5.2.2.	Password.....	41
5.2.3.	System_type.....	41
5.2.4.	Interface_version.....	41
5.2.5.	addr_ton, source_addr_ton, dest_addr_ton, esme_addr_ton.....	42
5.2.6.	addr_npi, source_addr_npi, dest_addr_npi, esme_addr_npi.....	42
5.2.7.	Address_range.....	43
5.2.8.	Source_addr.....	43
5.2.9.	Destination_addr.....	43
5.2.10.	Esme_addr	43
5.2.11.	Service_type	44
5.2.12.	Esm_class.....	44
5.2.13.	Protocol_id.....	46
5.2.14.	Priority_flag.....	46
5.2.15.	schedule_delivery_time	46
5.2.16.	Validity_period.....	47
5.2.17.	Registery_delivery.....	47
5.2.18.	replace_if_present_flag.....	48
5.2.19.	data_coding.....	48
5.2.20.	sm_default_msg_id.....	49
5.2.21.	Sm_length.....	49
5.2.22.	Short_message.....	49
5.2.23.	Message_id.....	49
5.2.24.	Number_of_dests.....	49
5.2.25.	Dest_flag.....	49
5.2.26.	No_unsuccess.....	49
5.2.27.	Dl_name	49
5.2.28.	Message_state	49
6 -	DEFINIÇÕES GERAIS.....	50
6.1 -	DEFINIÇÕES DE HORA	50
6.1.1.	Formato de tempo absoluto.....	50
6.1.2.	Formato de tempo relativo.....	50
6.2 -	DEFINIÇÕES DE TEMPORIZADORES	51
7 -	TÓPICO ESPECIAL – O RECIBO DE ENTREGA DO SMPP.....	52

1 - Introdução

1.1 - Overview SMPP

O protocolo SMPP (Short Message Peer to Peer) é um protocolo aberto, desenvolvido para proporcionar uma interface para a comunicação de dados flexível, para a transferência de short messages entre um Short Message Center (SMSC), GSM USSD (Unstructured Supplementary Services Data) ou outro tipo qualquer de message center, e uma aplicação SMS, como por exemplo, uma plataforma de Voice Mail, servidor de E-mail, Servidor Proxy WAP ou outra gateway de mensagens qualquer.

Nota: O protocolo SMPP utiliza o termo SMSC (Short Message Service Center) quando se refere à entidade servidora da conexão SMPP. No caso da entidade cliente da conexão SMPP, o nome adotado pelo protocolo é ESME (External Short Message Entity)

A versão V3.4 do protocolo SMPP contempla, dentre outras, as seguintes tecnologias:

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

O protocolo SMPP permite:

- Transmitir mensagens de uma ESME para um único ou múltiplos destinos via SMSC;
- Que uma ESME possa receber mensagens de um terminal móvel via o SMSC;
- Enviar mensagens com confirmação de recebimento;
- Cancelar ou repor mensagens;
- Consultar o status de entrega de uma determinada mensagem;
- Agendar a entrega de mensagens, selecionando a data e a hora de entrega;
- Selecionar o modo de transmissão da mensagem, i.e. datagrama ou store and forward
- Definir prioridade de entrega para as mensagens;
- Definir o tipo de codificação dos dados da mensagem;
- Definir um período de validade para a mensagem;
- Associar um tipo de serviço para cada mensagem.

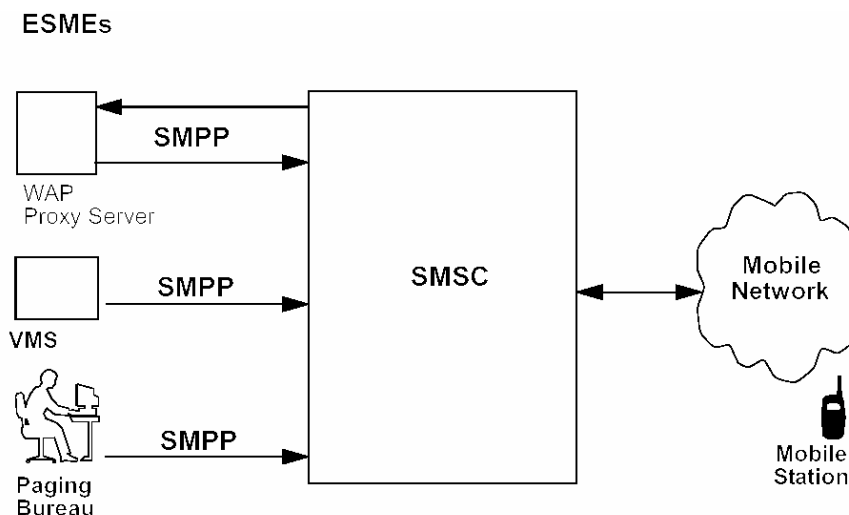


fig 1. Exemplos de aplicações que utilizam SMPP

O protocolo SMPP está sendo utilizado recentemente para permitir a troca de SMS entre operadoras. Para este intuito foram desenvolvidas gateways para trabalhar com este protocolo, convertendo-o de uma tecnologia para outra (e.g. de TDMA para GSM). Este tipo de conversão se faz bastante necessário no cenário brasileiro dada a diversidade de tecnologias adotadas pelas operadoras. Têm-se como exemplo destas novas aplicações os contratos de interconexão de SMS entre as operadoras TIM e Telemig Celular, TIM e Teleshia Celular, TIM e Global Telecom, TIM e BCP, Telemig Celular e TCO, Telesp Celular e BCP, Telesp Celular e TESS, dentre outras.

2 - Overview do protocolo SMPP

2.1 - Definições do protocolo SMPP

O protocolo SMPP pode trafegar sobre a pilha TCP/IP ou X.25. Em linhas gerais, a filosofia do protocolo SMPP consiste em abrir sessões específicas, permanentes, semi-permanentes ou dinâmicas, entre cada entidade. Por estas sessões são enviados os pacotes ou PDU (Protocol Data Unit), contendo as informações daquela operação SMPP específica. Fazendo uma analogia, a seção seria como uma rodovia por onde trafegam os caminhões, neste caso PDU's com as suas respectivas cargas, ou operações. Da mesma forma que uma rodovia, as sessões SMPP podem ser unidirecionais ou bidirecionais.

O SMPP utiliza um sistema de troca de mensagens e confirmações de recebimento para garantir a confiabilidade das transações. O protocolo SMPP define:

- Um conjunto de operações para a troca de mensagens entre a ESME e o SMSC;
- Os dados que uma entidade pode trocar com a outra, durante uma operação SMPP.

Todas as operações de envio de mensagens SMPP devem ser seguidas de uma mensagem de resposta. A única exceção à esta regra é no caso da mensagem de **ALERT_NOTIFICATION**, que não requer resposta.

Os três grupos distintos de transações de mensagens SMPP são os seguintes:

- i) mensagens enviadas a partir da ESME para o SMSC;
- ii) mensagens enviadas a partir do SMSC para a ESME;
- iii) Mensagens trocadas entre a ESME e o SMSC simultaneamente;

A figura 2 mostra os três tipos de seção possíveis dentro do protocolo SMPP. Note o sentido de envio das mensagens.

- (1) Conexão Transmitter
- (2) Conexão Receiver
- (3) Conexão Transceiver

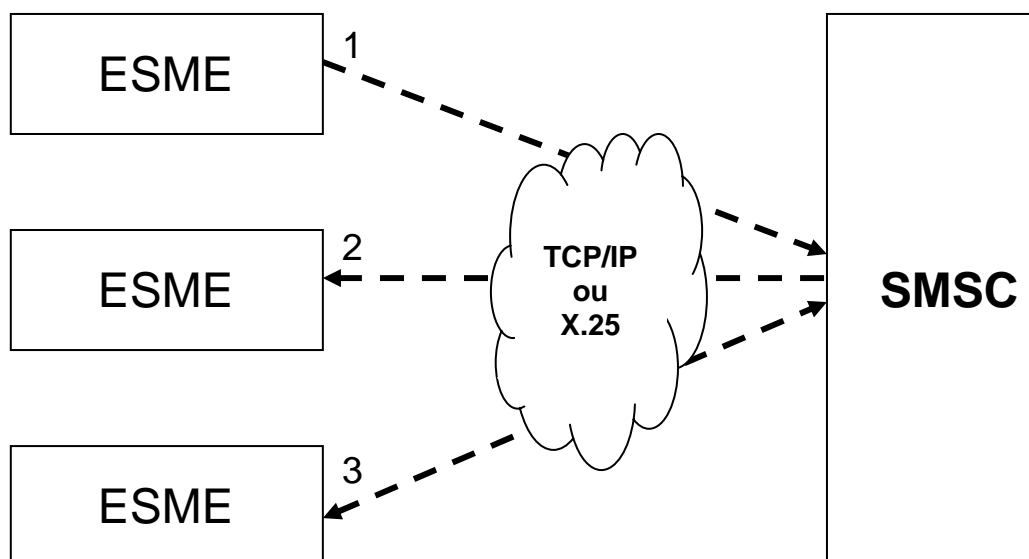


fig 2. Tipos de conexão SMPP

2.2 - Descrição de uma sessão SMPP

Uma sessão SMPP entre uma ESME e um SMSC é iniciada pelo estabelecimento de um link TCP/IP ou X.25 entre estas duas entidades. Em seguida a ESME, que segundo a nossa definição é a entidade cliente da sessão SMPP, faz a solicitação da conexão, chamada BIND. Se esta entidade deseja enviar mensagens, deve fazer um “BIND TRANSMITTER”, se deseja receber mensagens, deve fazer um “BIND RECEIVER” e se deseja tanto enviar quanto receber, deve fazer ou um “BIND TRANSMITTER” e um “BIND RECEIVER” ou simplesmente um “BIND TRANSCEIVER”. Os passos para esta conexão estão descritos abaixo:

- **OPEN** (Conectado via TCP/IP ou X.25, aguardando o BIND) A ESME já possui um caminho lógico via TCP/IP ou X.25, entretanto ainda não solicitou a abertura da conexão SMPP;
- **BOUND_TX** A ESME solicitou a abertura da conexão SMPP transmitter, enviando um PDU (*Protocol Data Unit*) *bind_transmitter*. Ao receber este PDU, o SMSC devolve um *bind_transmitter_resp*, informando se aceitou ou não a conexão. Caso a conexão seja estabelecida, a ESME já estará apta a enviar mensagens para o SMSC;
- **BOUND_RX** A ESME solicitou a abertura da conexão SMPP receiver, enviando um PDU *bind_receiver*. Ao receber este PDU, o SMSC devolve um *bind_receiver_resp*, informando se aceitou ou não a conexão. Caso a conexão seja estabelecida, a ESME já estará apta a receber mensagens do SMSC;
- **CLOSED** A ESME solicitou a desconexão ao SMSC.

2.2.1. O OUTBIND

Uma outra operação do SMPP é o **OUTBIND**, que é a mensagem SMPP enviada pelo SMSC para a ESME, solicitando que a ESME envie um BIND_RECEIVER. Observe que a regra da ESME solicitar a conexão não foi quebrada, uma vez que o SMSC não estabeleceu a conexão e sim a ESME. O diagrama abaixo ilustra esta situação:

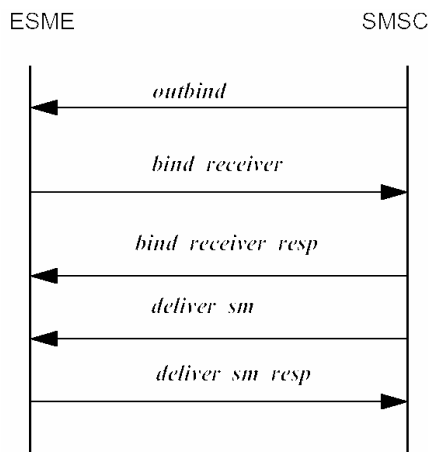


fig 3. A operação de OUTBIND

2.3 - O PDU SMPP

A tabela abaixo mostra em quais situações são aceitos cada tipo de PDU SMPP.

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	Yes	Yes
	BOUND_RX	Yes	Yes
	BOUND_TRX	Yes	Yes
<i>unbind_resp</i>	BOUND_TX	Yes	Yes
	BOUND_RX	Yes	Yes
	BOUND_TRX	Yes	Yes
<i>submit_sm</i>	BOUND_TX	Yes	No
	BOUND_TRX	Yes	No
<i>submit_sm_resp</i>	BOUND_TX	No	Yes
	BOUND_TRX	No	Yes

SMPP PDU Name	Required SMPP Session State	Issued by ESME	Issued by SMSC
<i>submit_sm_multi</i>	BOUND_TX	Yes	No
	BOUND_TRX	Yes	No
<i>submit_sm_multi_resp</i>	BOUND_TX	No	Yes
	BOUND_TRX	No	Yes
<i>Data_sm</i>	BOUND_TX	Yes	Yes
	BOUND_RX	Yes	Yes
	BOUND_TRX	Yes	Yes
<i>Data_sm_resp</i>	BOUND_TX	Yes	Yes
	BOUND_RX	Yes	Yes
	BOUND_TRX	Yes	Yes
<i>deliver_sm</i>	BOUND_RX	No	Yes
	BOUND_TRX	No	Yes
<i>deliver_sm_resp</i>	BOUND_RX	Yes	No
	BOUND_TRX	Yes	No
<i>query_sm</i>	BOUND_TX	Yes	No
	BOUND_TRX	Yes	No
<i>query_sm_resp</i>	BOUND_TX	No	Yes
	BOUND_TRX	No	Yes
<i>cancel_sm</i>	BOUND_TX	Yes	No
	BOUND_TRX	Yes	No
<i>cancel_sm_resp</i>	BOUND_TX	No	Yes
	BOUND_TRX	No	Yes
<i>replace_sm</i>	BOUND_TX	Yes	No
<i>replace_sm_resp</i>	BOUND_TX	No	Yes
<i>enquire_link</i>	BOUND_TX	Yes	Yes
	BOUND_RX	Yes	Yes
	BOUND_TRX	Yes	Yes
<i>enquire_link_resp</i>	BOUND_TX	Yes	Yes
	BOUND_RX	Yes	Yes
	BOUND_TRX	Yes	Yes
<i>alert_notification</i>	BOUND_RX	No	Yes
	BOUND_TRX	No	Yes
<i>generic_nack</i>	BOUND_TX	Yes	Yes
	BOUND_RX	Yes	Yes
	BOUND_TRX	Yes	Yes

tabela 1 - Sumário dos tipos de PDU SMPP

2.4 - A camada de conexões de rede SMPP

A camada inferior ao nível SMPP é a camada de transporte, que pode ser TCP/IP ou X.25. No caso das interconexões realizadas com o ISR até a presente data, utilizamos o protocolo TCP/IP. Por ser um protocolo do nível de aplicação, o SMPP não se preocupa com o transporte da mensagem, deixando esta tarefa a cargo da camada inferior. Esta camada deverá prover a confiabilidade necessária para a troca de mensagens. Se necessário, é possível solicitar a segmentação do pacote SMPP à entidade de nível inferior (TCP/IP ou X.25) que irá enviá-la. Ao recebê-la, a entidade de nível inferior do outro lado deverá reorganizar o pacote e entregá-lo para o nível superior.

2.5 - Mensagens enviadas da ESME para o SMSC

Uma ESME que envia short messages para um SMSC deve estar conectada a este SMSC como ESME Transmitter ou ESME Transceiver (vide tabela 1). Exemplos de PDU's que podem ser enviados para a entrega de mensagens ou dados são os PDU's *submit_sm* e *data_sm*. Os PDUS *query_sm*, *cancel_sm* e *replace_sm* são utilizados para controlar as mensagens enviadas para o SMSC, permitindo que a ESME possa questionar qual é o status de uma determinada mensagem, cancelar uma mensagem ou substituir uma mensagem, respectivamente. Deve-se observar que sempre que uma mensagem destas enviada, o SMSC deve responder com a mensagem de resposta correspondente, contendo o status da operação. A única exceção é a operação *alert_notification*. Estas mensagens serão detalhadas mais adiante.

2.5.1. Respostas de mensagens SMPP do SMSC para a ESME

Conforme dito anteriormente, para cada envio de operação SMPP, a parte recebedora do PDU deve sinalizar o recebimento do pacote com uma mensagem de resposta, indicando o status da operação. Desta forma, as mensagens de resposta, enviadas pelo SMSC, em resposta aos pacotes enviados pela ESME são as mensagens *submit_sm_resp*, *data_sm_resp*, *query_sm_resp*, *cancel_sm_resp* e *replace_sm_resp*. Estas mensagens serão detalhadas mais adiante.

2.5.2. Exemplo de uma sessão típica SMPP – ESME Transmitter

A figura abaixo mostra como é a seqüência de mensagem/resposta em uma sessão SMPP – ESME Transmitter

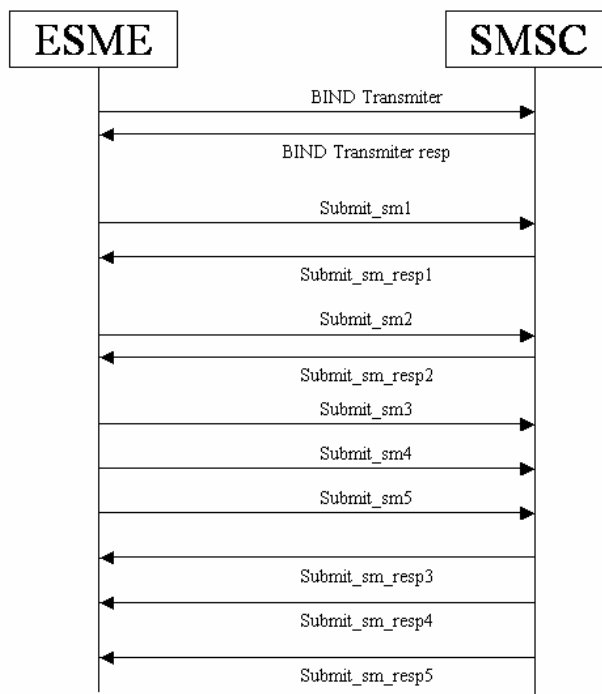


fig 4. Exemplo típico de uma seção SMPP – ESME Transmitter

A troca de mensagens e respostas entre a ESME e o SMSC pode ocorrer de forma síncrona (mensagens 1 e 2) ou assíncrona (mensagens 3, 4 e 5). A primeira é chamada *Stop and Wait* e a segunda é chamada *Sliding Window, Janela Deslizante*, ou simplesmente *Janelamento*.

Uma série de mensagens enviadas de forma assíncrona deve ser seguida de uma série de respostas em seqüência, respeitando a ordem de recebimento das mensagens, entretanto o funcionamento desta forma não é mandatório na especificação do protocolo. A entidade recebedora deve ser capaz de gerenciar estas mensagens. A quantidade de mensagens enviadas é chamada de *janela* ou *window*. No caso da figura 4, a janela utilizada foi de três mensagens.

O protocolo não estabelece um tamanho máximo para a janela a ser utilizada, entretanto, recomenda-se que não sejam utilizadas janelas maiores do que 10(dez) mensagens.

A ESME deve retornar as respostas na mesma ordem em que recebeu as mensagens. O único PDU de resposta relevante que a ESME passa para o SMSC é o ***enquire_link_response***.

2.6 - Mensagens enviadas pelo SMSC para a ESME

Uma ESME que recebe short messages para um SMSC deve estar conectada a este SMSC como ESME Receiver ou ESME Transceiver (vide tabela 1). Exemplos de PDU's que podem ser enviados para a entrega de mensagens ou dados são os PDU's ***deliver_sm*** e ***data_sm***. Deve-se observar que sempre que uma mensagem destas enviada, o SMSC deve responder com a mensagem de resposta correspondente, contendo o status da operação. Estas mensagens serão detalhadas mais adiante.

2.6.1. Respostas de mensagens SMPP da ESME para o SMSC

Conforme dito anteriormente, para cada envio de operação SMPP, a parte recebedora do PDU deve sinalizar o recebimento do pacote com uma mensagem de resposta, indicando o status da operação. Desta forma, as mensagens de resposta, enviadas pela ESME, em resposta aos pacotes enviados pelo SMSC são as mensagens ***deliver_sm_resp*** e ***data_sm_resp***. Estas mensagens serão detalhadas mais adiante.

2.6.2. Exemplo de uma sessão típica SMPP – ESME Receiver

A figura a seguir mostra como é a seqüência de mensagem/resposta em uma sessão SMPP – ESME Receiver

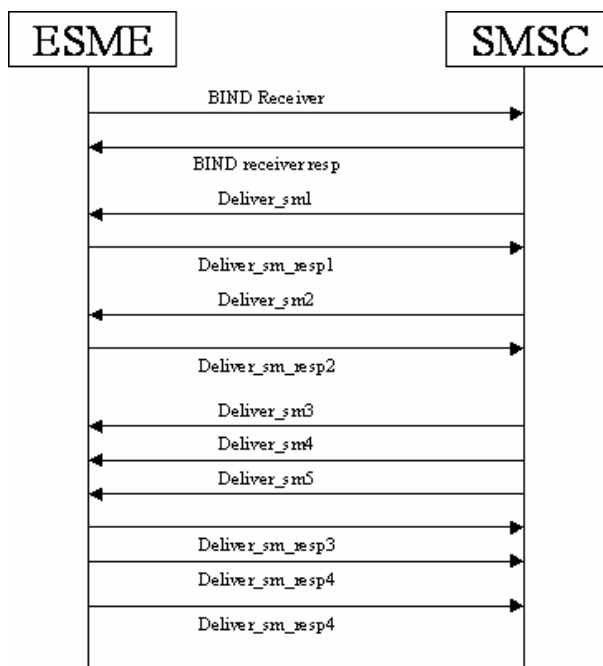


fig 5. Exemplo típico de uma seção SMPP – ESME Receiver

Da mesma forma como mostrado anteriormente, no caso da ESME Transmitter, a ESME receiver também pode trabalhar com o controle de fluxo utilizando o *Stop and Wait* ou *Windowing*.

2.7.2. Gerenciamento de erros no SMPP

Todas as operações SMPP são seguidas do respectivo PDU de resposta, exceto a operação **alert_notification**. A função do PDU de resposta é informar à entidade remetente que o PDU foi recebido, assim como qual foi o resultado da operação solicitada. Caso a operação solicitada seja concluída com sucesso, o campo *command_status* do PDU de resposta virá preenchido com um **OK**. Caso seja encontrada alguma não conformidade no campo de dados do pacote, i.e. a configuração errônea de um parâmetro, este campo virá preenchido com o código de erro correspondente. Caso seja encontrado algum erro no cabeçalho do PDU, a entidade de destino irá responder com um “GENERIC_NACK” PDU, significando que não foi possível a identificação do tipo de PDU recebido.

2.8 - Temporizadores SMPP

Para garantir a eficiência nas transações SMPP, é recomendável que cada sessão SMPP seja gerenciada utilizando temporizadores configuráveis, tanto na ESME, quanto no SMSC, de acordo com as seguintes considerações:

- Um temporizador inicial para garantir que quando uma ESME inicia uma sessão SMPP, isto ocorra dentro de um determinado período, após a abertura da conexão de rede entre as plataformas;
- Um temporizador para permitir que tanto a ESME quanto o SMSC verifiquem o status da conexão via o comando **enquire_link**;
- Uma inatividade da conexão SMPP deve especificar um período máximo para a espera por novas mensagens, após o qual a conexão poderá ser finalizada;
- Um temporizador SMPP que irá especificar qual é o maior tempo permitido entre a chegada de uma mensagem e o envio de uma resposta.

Trataremos dos temporizadores SMPP mais adiante.

2.9 - Tipos de mensagens

Complementando os tipos “normais” de short messages, mensagens especiais podem ser transferidas entre ESME e SMSC em uma operação **submit_sm**, **deliver_sm** ou **data_sm**. O tipo destas mensagens é definido no parâmetro *esm_class* destas operações SMPP. Os tipos suportados são os seguintes:

2.9.1. SMSC Delivery Receipt

Esta mensagem é usada para transportar um recibo de entrega da mensagem por uma SMSC. A SMSC, detectando que o destino final da mensagem foi alcançado, deve gerar um recibo de entrega para a entidade originadora da mensagem. O recibo de entrega é codificado como um pacote de dados do usuário, na operação **deliver_sm** ou **data_sm**. Os campos abaixo são relevantes nas operações **deliver_sm** ou **data_sm**, quanto utilizadas para transportar recibos de entrega de mensagens:

- *source address* (i.e. *source_addr_ton*, *source_addr_npi*, *source_addr*)

O endereço fonte é obtido a partir do endereço de destino na mensagem original;

- *destination address* (i.e. *dest_addr_ton*, *dest_addr_npi*, *destination_addr*)

O endereço fonte é obtido a partir do endereço fonte na mensagem original;

- *esm_class*
- *message_state*
- *network_error_code*
- *receipted_message_id*

2.9.2. Notificação Intermediária

Uma notificação intermediária é uma forma especial de mensagem que o SMSC utiliza para enviar para a ESME mensagens que deverão ser entregues em um destinatário específico. Isto proporciona um status intermediário da entrega da mensagem. As aplicações típicas são:

- Proporcionar uma notificação de “capacidade de memória excedida” para uma plataforma Voice Mail;
- Informar a primeira tentativa de entrega de uma mensagem que, por não ter sido entregue, foi armazenada para tentativas futuras.

2.9.3. SME Deliver Acknowledgement

Apesar do nome, a mensagem com o tipo SME Deliver Acknowledgement não é uma confirmação de que a mensagem foi entregue e sim uma confirmação de que a mensagem foi lida pelo destinatário. Esta facilidade é encontrada principalmente nos aparelhos Nokia, onde pode-se definir no menu de configuração de mensagem se é necessária uma confirmação de leitura para a mesma. Este tipo de mensagem **não** é suportado em todos os tipos de rede.

3 - Tipos de PDU's SMPP e definições de formato

3.1 - Definições dos tipos de dados do PDU SMPP

Os seguintes tipos de dados são utilizados para definir parâmetros SMPP:

Integer Um valor indeterminado que define o número de octetos. Os octetos serão sempre transmitidos com o bit mais significativo primeiro (Big Endian).

C-Octet String NULO Uma série de caracteres ASCII, terminados com o caractere NULO

C-Octet String (decimal) Uma série de caracteres ASCII, onde cada caractere representa um dígito decimal (de 0 a 9) e termina com o caractere NULO

C-Octet String (HEXA) Uma série de caracteres ASCII, onde cada caractere representa um dígito hexadecimal (de 0 a F) e termina com o caractere NULO

Octet String Uma série de octetos, não necessariamente terminada com o caractere NULO

3.1.1. Notação do tamanho dos campos dos parâmetros SMPP

A seguinte notação será utilizada nesta documentação para os parâmetros descritos:

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.

tabela 2 - Notação do tamanho dos parâmetros

3.2 - Overview do formato do PDU SMPP

O formato geral do PDU SMPP consiste de um cabeçalho PDU, seguido de um pacote de dados, tal como a maioria dos PDU's dos sistemas de comunicação digital. Este formato está descrito abaixo:

PDU SMPP				
Cabeçalho do PDU SMPP (Mandatório)				Corpo do PDU (Opcional)
<i>Command length</i>	<i>Command Id</i>	<i>Command status</i>	<i>Sequence number</i>	<i>Campo de dados</i>
4 Octetos	Comprimento=(tamanho informado no <i>command length</i> - 4) Octetos			

tabela 3 - Formato do PDU SMPP

O cabeçalho SMPP é mandatório em qualquer PDU e deve sempre estar presente. O campo de dados do PDU é opcional e pode não estar presente na mensagem. O formato de cada PDU SMPP será mais detalhado adiante.

3.2.1. Layout do PDU SMPP

	SMPP PDU Field	Size Octets	Type	Description
header	<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	The <i>command_id</i> field identifies the particular SMPP PDU, e.g., <i>submit_sm</i> , <i>query_sm</i> , etc. A unique command identifier is allocated to each SMPP request PDU in the range: 0x00000000 to 0x000001FF A unique command identifier is also allocated to each SMPP response PDU in the range: 0x80000000 to 0x800001FF (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.
	<i>command_status</i>	4	Integer	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. The complete list of SMPP Error codes is defined in Chapter 5.
	<i>sequence_number</i>	4	Integer	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. 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. The <i>sequence_number</i> may range from: 0x00000001 to 0x7FFFFFFF.
body	<i>Mandatory Parameters</i>	var.	mixed	A list of mandatory parameters corresponding to that SMPP PDU defined in the <i>command_id</i> field. The complete list of mandatory parameters is detailed in section 4. "SMPP PDU Definition" with the description of each SMPP PDU.
	<i>Optional Parameters</i>			A list of Optional Parameters corresponding to that SMPP PDU defined in the <i>command_id</i> field and included as required. The complete list of optional parameters is detailed in section 4. "SMPP PDU Definition" with the description of each SMPP PDU.

tabela 4 - Descrição do formato do PDU SMPP

3.2.2. Tamanho do PDU SMPP

O campo *command_length* no começo do cabeçalho do PDU SMPP indica o tamanho total do PDU SMPP em número de octetos. O campo *command_length* contem um inteiro de 4 octetos, transmitido no formato Big Endian.

Para decodificar um PDU SMPP, a ESME ou o SMSC deve primeiro ler o *command_length* para determinar o tamanho deste PDU. O valor indicado neste campo é subtraído de quatro octetos, sendo que o valor resultante corresponde ao tamanho restante da mensagem.

Exemplo:

Suponhamos que uma mensagem tenha a seguinte seqüência de dados no cabeçalho:

```
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
```

Os valores estão representados em formato hexadecimal. Desta forma, decodificando a seqüência acima obtemos:

```
00 00 00 2F Command Length 0x0000002F
00 00 00 02 Command ID 0x00000002 (bind_transmitter)
00 00 00 00 Command Status 0x00000000
00 00 00 01 Sequence Number 0x00000001
```

Os dados restantes representam o campo de dados do PDU SMPP, cujo exemplo corresponde à um PDU da operação *BIND_TRANSMITTER*.

3.2.3. Parâmetros Opcionais

Parâmetros Opcionais são campos que podem estar presentes nas mensagens SMPP. Estes parâmetros fornecem mecanismos para a introdução futura de novos parâmetros, à medida que forem definidos em futuras versões do protocolo SMPP.

Os parâmetros opcionais devem sempre aparecer no final do PDU SMPP. Entretanto eles podem ser incluídos em qualquer ordem conveniente, dentro da seção de parâmetros opcionais do PDU SMPP.

Para um PDU SMPP em particular, a ESME ou o SMSC devem incluir todos ou nenhum dos parâmetros opcionais definidos como o necessário no contexto da aplicação em particular. Por exemplo, sistemas de paging podem precisar somente de incluir o campo relacionado ao "*callback number*" em uma operação de *submit_sm*.

3.2.4. Formato dos parâmetros opcionais

Todos os parâmetros opcionais devem seguir o seguinte formato TLV (Tag Length Value) ou ICV (Identificador Comprimento Valor).

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.

tabela 5 - Formato dos parâmetros opcionais

3.2.5. Premissas para a compatibilidade do SMPP com versões mais avançadas

Um procedimento para compatibilidade com versões mais avançadas permite que uma entidade funcional (i.e. SMSC ou ESME) utilize uma versão do protocolo SMPP para comunicar-se com uma outra entidade utilizando uma versão mais aprimorada do protocolo. Desta forma, novos desenvolvimentos para PDU's SMPP existentes são implementados utilizando parâmetros opcionais.

As seguintes premissas devem ser seguidas para garantir esta compatibilidade:

- Se uma entidade SMPP recebe um PDU ou comando desconhecido, esta entidade **deve** retornar um PDU **generic_nack**, indicando um *command_id* no campo *command_status* do cabeçalho.
- A entidade SMPP que recebe uma mensagem que possui Parâmetros Opcionais deve primeiramente inspecionar o campo de identificador do Parâmetro Opcional, da seguinte forma:
 - ✓ Se o identificador do Parâmetro Opcional for reconhecido e suportado pela entidade SMPP, para aquela aplicação em particular, este parâmetro deverá ser processado;
 - ✓ Se o identificador do Parâmetro Opcional for reconhecido mas não for esperado pela entidade SMPP, para aquela operação em particular, este parâmetro deverá ser ignorado;
 - ✓ Se o identificador do Parâmetro Opcional não for reconhecido ou não suportado pela entidade SMPP, para aquela operação em particular, este parâmetro deverá ser ignorado e o próximo parâmetro opcional da seqüência deverá ser processado;

- Uma entidade SMPP que recebe um parâmetro com o valor “reservado”, deve utilizar o valor padrão, se uma configuração padrão é definida, caso contrário o parâmetro deverá ignorado;
- Se um valor de um parâmetro é desconhecido ou inválido, a entidade SMPP deve retornar um erro, indicando que o valor daquele parâmetro é inválido;
- Uma entidade SMPP que detecta que um parâmetro opcional, necessário naquela operação específica, não estava presente no PDU SMPP, deve retornar uma mensagem com um erro dizendo “*Expected Optional Parameter missing*”.
- Um campo com tamanho variável pode ter seu comprimento máximo estendido em versões futuras do protocolo SMPP. Uma entidade SMPP que recebe um parâmetro cujo comprimento é maior do que o suportado por aquela entidade deve rejeitar aquele parâmetro com o código de erro indicando “*invalid parameter length*”.

3.2.6. Premissas para a compatibilidade do SMPP com versões mais atrasadas

Um procedimento para compatibilidade com versões mais atrasadas permite que uma entidade funcional (i.e. SMSC ou ESME) utilize uma versão do protocolo SMPP para comunicar-se com uma outra entidade utilizando uma versão mais antiga do protocolo.

As seguintes premissas devem ser seguidas para garantir esta compatibilidade:

- PDU's existentes não devem ser removidos do protocolo;
- O efeito de uma mensagem existente, recebida em um novo formato modificado, deve ter o mesmo efeito que seria esperado quando utilizado um protocolo anterior;
- Parâmetros Opcionais não deverão tornar-se mandatórios e vice-versa.
- Novos parâmetros mandatórios não deverão ser adicionados;
- O significado de parâmetros existentes não deverá ser mudado em novas versões do protocolo;

Como o conceito de Parâmetros Opcionais foi introduzido nesta versão do protocolo, as seguintes premissas foram definidas:

- Um SMSC que trabalha com SMPP v3.4 ou superior não deve enviar parâmetros opcionais para uma ESME que utilize versões inferiores. Uma SMSC, que suporta uma versão anterior à v3.4, deverá preencher o parâmetro *interface_version* com valor menor que 0x34;
- Um SMSC, que suporta uma versão igual ou superior à v3.4, deverá preencher o parâmetro *sc_interface_version* com o valor suportado;
- Um SMSC que trabalha com uma versão do SMPP igual ou superior à v3.4 não deve gerar message ID's maiores que 8 (oito) octetos, quando esta comunicando-se com uma ESME que utilize uma versão mais antiga do SMPP.

4 - Definição do PDU SMPP

4.1 - Operação de BIND

A operação de BIND é a operação de tentativa de abertura do canal SMPP entre uma ESME e um SMSC. Conforme dito anteriormente, uma ESME pode fazer um *bind_transmitter*, *bind_receiver* ou *bind_transceiver*. Caso a ESME tente conectar-se com um *bind_transceiver* em um SMSC não suporte esta operação, este deve retornar uma mensagem de resposta contendo o erro "Invalid Command ID" e a ESME deverá tentar conectar-se novamente, desta vez com um *bind_transmitter* ou *bind_receiver* ou um par *bind_transmitter*, *bind_receiver*. Caso a ESME tente conectar-se com um *bind_transmitter* ou *bind_receiver* em um SMSC não suporte esta operação, este deve retornar uma mensagem de resposta contendo o erro "Invalid Command ID" e a ESME deverá tentar conectar-se novamente, desta vez com um *bind_transceiver*.

4.1.1. ESME Transmitter

Uma ESME conectada como uma transmitter está autorizada a enviar mensagens a um SMSC e receber as respostas SMPP correspondentes. Uma ESME indica que não deseja receber mensagens originadas de outra SME (i.e. terminais móveis), conectando-se como transmitter. Verifique na tabela 1 quais são as operações permitidas a uma ESME conectada como transmitter.

4.1.1.1. Sintaxe do BIND_TRANSMITTER

	Field Name	Size Octets	Type	Description	Ref.
header	<i>Command_length</i>	4	Integer	Defines the overall length of the <i>bind_transmitter</i> PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_transmitter</i> request.	
	<i>Command_status</i>	4	Integer	Not used in <i>bind_transmitter</i> PDU. Must be set to NULL.	
	<i>Sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>bind_transmitter_resp</i> PDU will echo the same sequence number.	
body	<i>system_id^b</i>	var. max 16	C-Octet String	Identifies the ESME system requesting to bind as a transmitter with the SMSC.	
	<i>password^c</i>	var. 16	C-Octet String	The password may be used by the SMSC to authenticate the ESME requesting to bind.	
	<i>system_type^d</i>	var. 16	C-Octet String	Identifies the type of ESME system requesting to bind as a transmitter with the SMSC.	
	<i>interface_version</i>	1	Integer	Indicates the version of the SMPP Protocol supported by the ESME.	

	Field Name	Size Octets	Type	Description	Ref.
body	<i>addr_ton</i>	1	Integer	Indicates Type of Number of the ESME address. If not known set to NULL	
	<i>addr_npi</i>	1	Integer	Numbering Plan Indicator for ESME address. If not known set to NULL.	
	<i>address_range</i>	Var. max 41	C-Octet String	The ESME address. If not known set to NULL.	

tabela 6 - PDU do BIND_TRANSMITTER

- 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 the password 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.

4.1.1.2. Sintaxe do BIND_TRANSMITTER_RESP

	Field Name	Size Octets	Type	Description	Ref.
header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>bind_transmitter_resp</i> PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_transmitter_resp</i> request.	
	<i>command_status</i>	4	Integer	Indicates status (success or error code) of original <i>bind_transmitter</i> request.	
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>bind_transmitter</i> request.	
body	<i>system_id</i> ^b	var. max 16	C-Octet String	SMSC identifier. Identifies the SMSC to the ESME.	
OPTIONAL PARAMETERS for BIND_TRANSMITTER_RESP					
	<i>sc_interface_version</i>		TLV	SMPP version supported by SMSC	

tabela 7 - PDU do BIND_TRANSMITTER_RESP

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.

4.1.2. ESME Receiver

Uma ESME conectada como uma receiver está autorizada a receber mensagens de um SMSC e retornar as respostas SMPP correspondentes. Verifique na tabela 1 quais são as operações permitidas a uma ESME conectada como receiver.

4.1.2.1. Sintaxe do BIND_RECEIVER

	Field Name	Size Octets	Type	Description	Ref.
header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>bind_receiver</i> PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_receiver</i> request.	
	<i>command_status</i>	4	Integer	Not used in <i>bind_receiver</i> PDU. Must be set to NULL.	
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>bind_receiver_resp</i> PDU will echo the same sequence number.	
body	<i>system_id^b</i>	var. max 16	C-Octet String	Identifies the ESME system requesting to bind as a receiver with the SMSC.	
	<i>password^c</i>	var. 16	C-Octet String	The password may be used by the SMSC to authenticate the ESME requesting to bind.	
	<i>system_type^d</i>	var. 16	C-Octet String	Identifies the type of ESME system requesting to bind as a receiver with the SMSC.	
	<i>interface_version</i>	1	Integer	Indicates the version of the SMPP Y Protocol supported by the ESME.	
	<i>addr_ton</i>	1	Integer	Indicates Type of Number of the ESME address. If not known set to NULL	
	<i>addr_npi</i>	1	Integer	Numbering Plan Indicator for ESME address. If not known set to NULL.	
	<i>address_range</i>	Var. max 41	C-Octet String	The ESME address. If not known set to NULL.	

tabela 8 - PDU do BIND_RECEIVER

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 the password 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.

4.1.2.2. Sintaxe do BIND_RECEIVER_RESP

	Field Name	Size Octets	Type	Description	Ref.
header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>bind_receiver_resp</i> PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_receiver_resp</i> request.	
	<i>command_status</i>	4	Integer	Indicates status (success or error code) of original <i>bind_receiver</i> request.	
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>bind_receiver</i> request.	
body	<i>system_id^b</i>	var. max 16	C-Octet String	SMSC identifier. Identifies the SMSC to the ESME.	
OPTIONAL PARAMETERS for BIND_RECEIVER_RESP					
	Optional Parameter Name	Type	Description	Ref.	
	<i>sc_interface_version</i>	TLV	SMPP version supported by SMSC		

tabela 9 - PDU do BIND_RECEIVER_RESP

Note: The body portion of the SMPP *bind_receiver_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_receiver* request, the SMSC *system_id* is not returned.

4.1.3. ESME Transceiver

Uma ESME conectada como uma transceiver está autorizada receber/enviar mensagens de/para o SMSC e retornar as respostas SMPP correspondentes, em uma única sessão SMPP. Verifique na tabela 1 quais são as operações permitidas a uma ESME conectada como transceiver.

4.1.3.1. Sintaxe do BIND_TRANSCEIVER

	Field Name	Size Octets	Type	Description	Ref.
header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>bind_transceiver</i> PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_transceiver</i> request.	
	<i>command_status</i>	4	Integer	Not used in <i>bind_transceiver</i> PDU. Must be set to NULL.	
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated <i>bind_transceiver_resp</i> PDU will echo the same sequence number.	
body	<i>system_id</i> ^b	var. max 16	C-Octet String	Identifies the ESME system requesting to bind as a transceiver with the SMSC.	
	<i>password</i> ^c	var. 16	C-Octet String	The password may be used by the SMSC to authenticate the ESME requesting to bind.	
	<i>system_type</i> ^d	var. 16	C-Octet String	Identifies the type of ESME system requesting to bind as a transceiver with the SMSC.	
	<i>interface_version</i>	1	Integer	Indicates the version of the SMPP Y Protocol supported by the ESME.	
	<i>addr_ton</i>	1	Integer	Indicates Type of Number of the ESME address. If not known set to NULL	
	<i>addr_npi</i>	1	Integer	Numbering Plan Indicator for ESME address. If not known set to NULL.	
	<i>address_range</i>	Var. max 41	C-Octet String	The ESME address. If not known set to NULL.	

tabela 10 - PDU do BIND_TRANSCEIVER

- 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 the password 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.

4.1.3.2. Sintaxe do BIND_TRANSCEIVER_RESP

	Field Name	Size Octets	Type	Description	Ref.
header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>bind_transceiver_resp</i> PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to <i>bind_transceiver_resp</i> request.	
	<i>command_status</i>	4	Integer	Indicates status (success or error code) of original <i>bind_transceiver</i> request.	
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>bind_transceiver</i> request.	
body	<i>system_id</i> ^b	var. max 16	C-Octet String	SMSC identifier. Identifies the SMSC to the ESME.	
OPTIONAL PARAMETERS for BIND_TRANSCEIVER_RESP					
	Optional Parameter Name		Type	Description	Ref.
	<i>sc_interface_version</i>		TLV	SMPP version supported by SMSC	

tabela 11 - PDU do BIND_TRANSCEIVER_RESP

Note: The body portion of the SMPP *bind_transceiver_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_transceiver* request, the SMSC *system_id* is not returned.

4.1.4. Operação de OUTBIND

Esta operação é utilizada pelo SMSC para sinalizar à ESME para que esta origine um *bind_receiver* ao SMSC.

4.1.4.1. Sintaxe do OUTBIND

	Field Name	Size Octets	Type	Description	Ref.
Header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>outbind</i> PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to <i>outbind</i> request.	
	<i>command_status</i>	4	Integer	Not used in <i>outbind</i> PDU. Must be set to NULL.	
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number.	
Body	<i>system_id</i>	var. max 16	C-Octet String	SMSC identifier. Identifies the SMSC to the ESME	
	<i>password</i>	var. 16	C-Octet String	The password may be used by the ESME for security reasons to authenticate the SMSC originating the <i>outbind</i> .	

tabela 12 - PDU do OUTBIND

4.2 - Operação de UNBIND

O propósito da operação de **unbind** de acordo com o protocolo SMPP é cancelar o registro de uma instância ESME a partir do SMSC e informar ao SMSC que a ESME não deseja mais utilizar esta conexão de rede para o envio ou recebimento de mensagens. Desta forma, a operação de **unbind** pode ser vista como uma forma do SMSC requisitar um fechamento da sessão SMPP corrente.

4.2.1. Sintaxe do UNBIND

	Field Name	Size Octets	Type	Description	Ref.
Header	<i>command_length</i>	4	Integer	Defines the overall length of the unbind PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to unbind request.	
	<i>command_status</i>	4	Integer	Not used in unbind PDU. Must be set to NULL.	
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated unbind_resp PDU will echo the same sequence number.	

tabela 13 - PDU do UNBIND

4.2.2. Sintaxe do UNBIND_RESP

	Field Name	Size Octets	Type	Description	Ref.
Header	<i>command_length</i>	4	Integer	Defines the overall length of the unbind PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to unbind request.	
	<i>command_status</i>	4	Integer	Not used in unbind PDU. Must be set to NULL.	
	<i>sequence_number</i>	4	Integer	Set to sequence number of original unbind PDU request.	

tabela 14 - PDU do UNBIND_RESP

4.3 - GENERIC_NACK PDU

Este PDU de resposta é um reconhecimento negativo de um PDU SMPP enviado com um cabeçalho inválido. A resposta **generic_nack** é retornada nos seguintes casos:

➤ *Invalid command_length*

Se a entidade recebedora, ao decodificar o PDU SMPP, detecta um *command_length* inválido (tanto curto quando longo), ela deve assumir que o dado está corrompido. Neste caso uma resposta com o **generic_nack** é retornada ao originador da mensagem;

➤ *command_id* desconhecido

Se um *command_id* desconhecido ou inválido é recebido, uma resposta com o **generic_nack** é retornada ao originador da mensagem.

4.3.1. Sintaxe do PDU GENERIC_NACK

	Field Name	Size Octets	Type	Description	Ref.
Header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>generic_nack</i> PDU.	
	<i>command_id</i>	4	Integer	Value corresponding to <i>generic_nack</i> request.	
	<i>command_status</i>	4	Integer	Error code corresponding to reason for sending the <i>generic_nack</i> .	
	<i>sequence_number</i>	4	Integer	Set to sequence number of original PDU or to NULL if the original PDU cannot be decoded.	

tabela 15 - PDU do GENERIC_NACK

4.4 - Operação de SUBMIT_SM

A operação de *submit_sm* é utilizada pela ESME para enviar mensagens para o SMSC. A sessão utilizada para enviar estas mensagens é a sessão ESME Transmitter. Assim que o SMSC recebe o pacote *submit_sm*, e este é verificado como correto, esta entidade envia um PDU de resposta *submit_sm_resp* contendo um código que será utilizado para identificar aquela mensagem. Este código é o *message_id*, que será utilizado posteriormente para associar a mensagem *submit_sm* ao recibo de confirmação de entrega correspondente. Conforme dito anteriormente, as mensagens de resposta *submit_sm_resp* não precisam necessariamente serem precedidas do *submit_sm* correspondente, mas devem estar OBRIGATORIAMENTE em ordem, de forma que o *message_id* de uma mensagem não seja atribuído à outra. Caso ocorra algum erro no envio desta mensagem *submit_sm*, o SMSC irá retornar o código de erro da operação para o ESME e a mensagem não será aceita. Dependerá das regras de tratamento de erro de cada plataforma ESME, qual será a decisão a ser tomada com relação a este pacote *submit_sm* rejeitado. Esta decisão pode ser descartar o pacote, armazenar o pacote em buffer, esperando uma intervenção no sistema para corrigir o problema, ou uma tentativa de reenvio, de acordo com o estabelecido na máquina. A mesma decisão depende do código de erro devolvido, ou seja, não necessariamente a ESME irá descartar, armazenar ou retransmitir todos os pacotes com algum código de erro.

4.4.1. Sintaxe do SUBMIT_SM

O formato do PDU *submit_sm* é definido pela seguinte tabela:

	Field Name	Size Octets	Type	Description	Ref.
Header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>submit_sm</i> PDU.	
	<i>command_id</i>	4	Integer	<i>submit_sm</i>	
	<i>command_status</i>	4	Integer	Not used. Set to NULL.	
	<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.	
MANDATORY PARAMETERS	<i>service_type</i>	Var. Max 6	C-Octet String	<p>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</p> <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. <p>Set to NULL for default SMSC settings.</p>	
	<i>source_addr_ton</i>	1	Integer	Type of Number for source address. If not known, set to NULL (Unknown).	
	<i>source_addr_npi</i>	1	Integer	Numbering Plan Indicator for source address. If not known, set to NULL (Unknown).	
	<i>source_addr</i>	Var. Max 21	C-Octet String	Address of SME which originated this message. If not known, set to NULL (Unknown).	
	<i>dest_addr_ton</i>	1	Integer	Type of Number for destination	
	<i>dest_addr_npi</i>	1	Integer	Numbering Plan Indicator for Destination.	
	<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.	
	<i>esm_class</i>	1	Integer	Indicates Message Mode & Message Type.	
	<i>Protocol_id</i>	1	Integer	Protocol Identifier. Network specific field.	
	<i>priority_flag</i>	1	Integer	Designates the priority level of the message.	

	Field Name	Size Octets	Type	Description	Ref.
MANDATORY PARAMETERS	<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.	
	<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.	
	<i>registered_delivery</i>	1	Integer	Indicator to signify if an SMSC delivery receipt or an SME acknowledgement is required.	
	<i>replace_if_present_flag</i>	1	Integer	Flag indicating if submitted message should replace an existing message.	
	<i>Data_coding</i>	1	Integer	Defines the encoding scheme of the short message user data.	
	<i>sm_default_message_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.	
	<i>sm_length</i>	1	Integer	Length in octets of the <i>short_message</i> user data.	
	<i>short_message</i>	Var. Max 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:</p> <p>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>	

tabela 16 - PDU do SUBMIT_SM

Nota: Os parâmetros opcionais do PDU *submit_sm* poderão ser encontrados na documentação entregue em anexo (**SMPP Protocol Specification v3.4**), e não farão parte do escopo deste curso.

4.4.1.1. Source and Destination Addressing

O PDU *submit_sm* inclui formas de provisionar tanto o "endereço fonte" ou *source address*, quanto o "endereço de destino" ou *destination address*. O *endereço fonte* é composto dos campos *source_addr_ton*, *source_addr_npi* e *source_addr* e o *endereço de destino* é composto dos campos *dest_addr_ton*, *dest_addr_npi* e *destination_addr*.

Uma ESME Transmitter pode inserir valores nulos nos campos do *endereço fonte*. Neste caso, o SMSC deve inserir neste campo o valor padrão para aquela ESME em particular. Este comportamento foi projetado para interfaces que normalmente não são familiares ao conceito de *endereço fonte* para short messages, como por exemplo, sistemas de Voice Mail.

4.4.1.2. Operação de reposição de mensagens com SUBMIT_SM

O protocolo SMPP oferece uma operação dedicada à reposição de mensagens, chamada *replace_sm*, mas a operação *submit_sm* também pode ser utilizada para este fim. A função de reposição utilizando *submit_sm* pode ser ativada preenchendo o campo *replace_if_present_flag* com 1 (um). Alternativamente o administrador do SMSC pode implementar um tipo de serviço específico para a funcionalidade de reposição. Neste caso, a função de reposição de mensagem será ativada no PDU *submit_sm* através do campo *service_type*, preenchendo-o com o valor definido. Em todos os casos, o campo contendo a mensagem, na mensagem que será reposta, será substituído pelo texto contido no campo *short_message* da última mensagem enviada.

A diferença em utilizar o método de reposição por *submit_sm* e *replace_sm* é que no primeiro caso, se a mensagem não é encontrada no buffer do SMSC (i.e. se já tiver sido entregue), uma nova mensagem será enviada para o destinatário. Já no segundo caso a mensagem não será entregue e o SMSC irá retornar um erro no PDU *replace_sm_resp*.

4.4.1.3. Sintaxe do SUBMIT_SM_RESP

O PDU *submit_sm_resp* possui o seguinte formato:

	Field Name	Size Octets	Type	Description	Ref.
Header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>submit_sm_resp</i> PDU.	
	<i>command_id</i>	4	Integer	<i>submit_sm_resp</i>	
	<i>command_status</i>	4	Integer	Indicates outcome of <i>submit_sm</i> request.	
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>submit_sm</i> PDU	
Body	<i>message_id</i>	var. max 65	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.	

tabela 17 - PDU do SUBMIT_SM_RESP

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

4.5 - Operação de DELIVER_SM

A operação de **deliver_sm** é utilizada pelo SMSC para submeter mensagens para uma ESME. Utilizando este comando, o SMSC pode rotear uma mensagem para entrega.

Adicionalmente, o SMSC utiliza a operação de **deliver_sm** para transferir tipos específicos de short messages para o ESME, dentre eles o SMSC Delivery Receipt, que nada mais é do que o recibo de confirmação de entrega da mensagem, que informa à ESME que aquela mensagem descrita neste pacote foi devidamente entregue no aparelho de destino. Desta forma é possível implementar um sistema de tarifação baseado na entrega de mensagens.

4.5.1. Sintaxe do DELIVER_SM

	Field Name	Size Octets	Type	Description	Ref.
Header	<i>command_length</i>	4	Integer	Defines the overall length of the deliver_sm PDU.	
	<i>command_id</i>	4	Integer	deliver_sm	
	<i>command_status</i>	4	Integer	Unused. Set to NULL	
	<i>sequence_number</i>	4	Integer	Set to a unique sequence number. The associated deliver_sm_resp PDU should echo the same sequence number.	
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.	
	<i>source_addr_ton</i>	1	Integer	Type of Number for source address. If not known, set to NULL (Unknown).	
	<i>source_addr_npi</i>	1	Integer	Numbering Plan Indicator for source address. If not known, set to NULL (Unknown).	
	<i>source_addr</i>	Var. Max 21	C-Octet String	Address of SME which originated this message. If not known, set to NULL (Unknown).	
	<i>dest_addr_ton</i>	1	Integer	Type of Number for destination	
	<i>dest_addr_npi</i>	1	Integer	Numbering Plan Indicator for Destination.	
	<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.	
	<i>esm_class</i>	1	Integer	Indicates Message Type and enhanced network services	

	Field Name	Size Octets	Type	Description	Ref.
MANDATORY PARAMETERS	<i>Protocol_id</i>	1	Integer	Protocol Identifier. Network specific field.	
	<i>priority_flag</i>	1	Integer	Designates the priority level of the message.	
	<i>Schedule_delivery_time</i>	1 or 17	C-Octet String	This field is unused for <i>deliver_sm</i> . It must be set to NULL.	
	<i>Validity_period</i>	1 or 17	C-Octet String	This field is unused for <i>deliver_sm</i> . It must be set to NULL.	
	<i>registered_delivery</i>	1	Integer	Indicates if an ESME acknowledgement is required.	
	<i>replace_if_present_flag</i>	1	Integer	Not used in <i>deliver_sm</i> . It must be set to NULL.	
	<i>Data_coding</i>	1	Integer	Indicates the encoding scheme of the short message.	
	<i>sm_default_msg_id</i>	1	Integer	Unused for <i>deliver_sm</i> . It must be set to NULL.	
	<i>sm_length</i>	1	Integer	Length in octets of the <i>short_message</i> user data in octets.	
<i>short_message</i>	Var. Max 0 - 254	Octet String	Up to 254 octets of short message user data. 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. 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.		

tabela 18 - PDU do DELIVER_SM

Nota: Os parâmetros opcionais do PDU *deliver_sm* poderão ser encontrados na documentação entregue em anexo (**SMPP Protocol Specification v3.4**), e não farão parte do escopo deste curso.

4.5.2. Sintaxe do DELIVER_SM_RESP

	Field Name	Size Octets	Type	Description	Ref.
Header	<i>command_length</i>	4	Integer	Defines the overall length of the <i>deliver_sm_resp</i> PDU.	
	<i>command_id</i>	4	Integer	<i>deliver_sm_resp</i>	
	<i>command_status</i>	4	Integer	Indicates outcome of <i>deliver_sm</i> request.	
	<i>sequence_number</i>	4	Integer	Set to sequence number of original <i>deliver_sm</i> PDU	
Body	<i>message_id</i>	1	C-Octet String	This field is unused and is set to NULL.	

tabela 19 - PDU do DELIVER_SM_RESP

O recibo de confirmação de entrega, que utiliza o PDU *deliver_sm* será detalhado em um tópico especial mais adiante.

4.6 - Outras operações do SMPP

Existem outras operações suportadas pelo protocolo SMPP. Estas operações não serão detalhadas neste curso. Posteriormente será desenvolvido um módulo avançado, destinado ao detalhamento das operações SMPP não descritas neste curso.

4.6.1. DATA_SM

Esta operação é utilizada para transmitir dados entre a ESME e o SMSC. Pode ser utilizado por ambas as entidades. Este comando é uma alternativa aos comando *submit_sm* e *deliver_sm*. Foi introduzido como um novo comando para ser utilizado por aplicações interativas, como por exemplo, operações via WAP. A ESME pode utilizar este comando para requisitar que o SMSC transfira uma mensagem para um móvel. Desta forma, o SMSC pode utilizar este comando para transferir mensagens originadas por terminais para uma ESME. Da mesma forma que a operação *deliver_sm*, esta operação pode ser utilizada para transferir o recibo de confirmação de entrega para uma ESME.

4.6.2. QUERY_SM

Este comando é utilizado pelo ESME para questionar o SMSC sobre o status de uma determinada mensagem submetida previamente.

Este mecanismo é baseado no *message_id* associado, repassado pelo SMSC, e pelo *endereço fonte*.

4.6.3. CANCEL_SM

Este comando é utilizado pela ESME para cancelar uma mensagem previamente enviada para o SMSC, que ainda estão com a entrega pendente. Este comando pode especificar uma única mensagem, cancelando-a, ou pode também cancelar mensagens específicas de acordo com o *endereço fonte* ou *endereço de destino*.

4.6.4. REPLACE_SM

Conforme descrito anteriormente, esta operação permite que uma determinada mensagem seja substituída por outra mensagem. Novamente o mecanismo de *message_id* é utilizado para detectar a mensagem que se deseja repor.

4.6.5. ENQUIRE_LINK

Esta mensagem pode ser utilizada tanto pelo ESME quanto pelo SMSC, para proporcionar uma checagem confiável do caminho de comunicação entre a ESME e o SMSC. Uma das entidades, ao receber esta mensagem, deve respondê-la com uma mensagem ***enquire_link_resp***, indicando que está "viva", ou seja, o nível de aplicação entre as entidades está funcionando.

4.6.6. ALERT_NOTIFICATION

Este comando é enviado pelo SMSC para a ESME quando o SMSC detecta que um assinante em particular se tornou disponível para receber mensagens e um sinalizador de entrega foi enviado para aquele assinante por uma operação ***data_sm*** anterior.

Esta mensagem pode ser utilizada por exemplo como um gatilho para uma operação "Push" para dados de um Servidor WAP Proxy.

Nota: Esta operação é a única que não contém uma mensagem de resposta associada, ou seja, não existe um PDU ***alert_notification_resp***.

5 - Definição dos parâmetros SMPP

Esta seção descreve os parâmetros SMPP que podem ser utilizados em um comando SMPP.

5.1 - Parâmetros do cabeçalho SMPP (Command Header)

5.1.1. Command_length

O parâmetro *command_length* indica o tamanho total em octetos de uma mensagem SMPP. São considerados o cabeçalho SMPP (incluindo o próprio campo *command_length*), os parâmetros mandatórios e os parâmetros opcionais.

5.1.2. Command_id

O campo *command_id* indica o tipo de mensagem que o PDU SMPP representa, por exemplo, *submit_sm*, *deliver_sm* etc.

Um identificador de comando é destinado para cada primitiva de solicitação SMPP.

Um identificador de comando de resposta é destinado para cada primitiva de resposta SMPP.

Verifique na tabela a seguir a faixa de valores reservados.

5.1.2.1. Conjunto de comandos SMPP

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

Command ID	Value
<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
<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

tabela 20 - Valores para o campo *command_id* do PDU SMPP

5.1.3. Command_status

O campo *command_status* indica o sucesso ou falha de uma solicitação SMPP. Somente é relevante em mensagens de resposta SMPP. Nas mensagens de solicitação SMPP, deve ser preenchido com o valor valor nulo. O código de erro retornado no comando de resposta SMPP é informado neste campo. O conjunto completo de códigos de erro e seus valores associados estão apresentados na **tabela 21**.

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
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	Dest. flag is invalid (<i>submit_multi</i>)
Error Code	Value	Description

<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
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.
Error Code	Value	Description
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

tabela 21 - Códigos de erro do SMPP

5.1.4. Sequence_number

Um número de sequência permite que um PDU de resposta seja relacionado a um PDU de solicitação. O PDU de resposta SMPP deve preservar este campo. A faixa de valores possíveis deste campo vai de 0x00000001 a 0x7FFFFFFF.

5.2 - Parâmetros mandatórios do SMPP

5.2.1. System_id

O parâmetro *system_id* é utilizado para identificar uma ESME ou um SMSC no momento do BIND.

5.2.2. Password

O parâmetro *password* é utilizado pelo SMSC para autenticar uma ESME que está solicitando o BIND. Também é utilizado pela ESME para autenticar um SMSC, no caso da operação de *outbind*.

5.2.3. System_type

O parâmetro *system_type* é utilizado para categorizar o tipo da ESME que está solicitando o BIND ao SMSC. Por exemplo, "VMS", no caso de um Sistema de Voice Mail, ou "OTA", para um sistema de Over-The-Air activation.

A especificação do *system_type* é opcional. Alguns SMSC podem não solicitar que a ESME informe este detalhe. Neste caso a ESME pode preencher este campo com o valor nulo.

5.2.4. Interface_version

Este parâmetro é utilizado para indicar a versão do protocolo SMPP utilizado pela entidade. As seguintes versões de interface são definidas:

Interface Version	Value
Indicates that the EMSE supports	0x00-0x33

version 3.3 or earlier of the SMPP protocol.	
Indicates that the ESME is supporting SMPP version 3.4	0x34
<i>All other values reserved</i>	

tabela 22 - Valores disponíveis para o campo *interface_version*

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

Estes campos definem o tipo de número ou TON (*Type Of Number*) para ser utilizado nos parâmetros da SME. Os valores possíveis são apresentados na **tabela 23**.

TON	Value	Decimal
Unknown	00000000	0
International	00000001	1
National	00000010	2
Network Specific	00000011	3
Subscriber Number	00000100	4
Alphanumeric	00000101	5
Abbreviated	00000110	6
<i>All other values reserved</i>		

tabela 23 - Valores possíveis para TON (*Type Of Number*)

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

Estes campos definem o indicador de plano numérico, ou NPI (*Numeric Plan Indicator*) que será utilizado nos parâmetros da SME. Os valores possíveis são apresentados na **tabela 24**.

NPI	Value	Decimal
Unknown	00000000	0
ISDN (E163/E164)	00000001	1
Data (X.121)	00000011	3
Telex (F.69)	00000100	4
Land Mobile (E.212)	00000110	6
National	00001000	8
Private	00001001	9
ERMES	00001010	10
Internet (IP)	00001110	14
WAP Client Id (to be defined by WAP Forum)	00010010	18
<i>All other values reserved</i>		

tabela 24 - Valores possíveis para NPI (*Numeric Plan Indicator*)

5.2.7. Address_range

O parâmetro *address_range* é utilizado nas operações de *bind_receiver* e *bind_tranceiver* para especificar um conjunto de endereços das SME's servidas pelo cliente ESME. Um endereço único pode ser especificado no parâmetro *address_range*. A notação de expressão regular do UNIX deve ser utilizada para especificar a faixa de endereços.

5.2.8. Source_addr

Este campo especifica o endereço da SME que originou a mensagem. Uma ESME que é implementada com um único endereço SME, i.e. Voice Mail, pode preencher este campo com o valor nulo, para permitir que o SMSC preencha este campo com o valor padrão.

5.2.9. Destination_addr

Este campo especifica o endereço da SME de destino da mensagem. Para mensagens terminadas, este endereço especifica o DN do terminal.

5.2.10. Esme_addr

Especifica o endereço da ESME para onde as mensagens de *alert_notification* deverão ser enviadas.

5.2.11. Service_type

O parâmetro *service_type* pode ser usado para indicar o serviço de aplicação SMS associado à mensagem, permitindo à ESME:

- Disponibilizar serviços avançados de mensagens, tais como aqueles descritos no item 4.4.1.2;
- Controlar o teleserviço utilizado na interface aérea (i.e. IS-136/TDMA, IS-95/CDMA)

O SMSC pode associar implicitamente a função "*replace if present*" do *service_type* indicado em uma submissão de mensagem. Por exemplo, o SMSC pode sempre repor uma mensagem pendente de entrega que tenha o mesmo originador e destinatário da mensagem submetida. Desta forma, um SMSC pode garantir que um sistema de Voice Mail, utilizando o *service_type* "VMA" tem no máximo uma notificação por destinatário, invocando automaticamente a função "*replace if present*".

Os tipos genéricos de *service_type* estão apresentados abaixo:

“” (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

Outros valores podem ser definidos, bastando apenas um acordo mútuo entre os administradores da ESME e do SMSC.

5.2.12. Esm_class

O parâmetro *esm_class* é utilizado para indicar atributos especiais associados às mensagens.

Em mensagens enviadas da ESME para o SMSC, este parâmetro é codificado da seguinte forma:

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)

Bits	7 6 5 4 3 2 1 0	Meaning
<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)

Em mensagens enviadas do SMSC para a ESME, este parâmetro é codificado da seguinte forma:

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
<i>all other values reserved</i>		

5.2.13. Protocol_id

GSM

Configurado de acordo com a norma GSM 03.40 [GSM 03.40]

ANSI-136 (TDMA)

Para mensagens terminadas, este campo não é utilizado, portanto deve ser ignorado pelo SMSC.

Para mensagens originadas, o SMSC deve configurar este valor para nulo.

IS-95-(CDMA)

Para mensagens terminadas, este campo não é utilizado, portanto deve ser ignorado pelo SMSC.

Para mensagens originadas, o SMSC deve configurar este valor para nulo.

5.2.14. Priority_flag

O parâmetro *priority_flag* permite que mensagens originadas pela SME associem níveis de prioridade às mensagens.

São suportados quatro níveis de prioridade:

0= Nível 0 menor prioridade

1= Nível 1 de prioridade

2= Nível 2 de prioridade

3= Nível 3 maior prioridade

>3=Reservado

Estes valores são aplicados nas diferentes redes conforme descrito na **tabela 25**:

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>			

tabela 25 - Valores para a prioridade das mensagens SMPP

5.2.15. schedule_delivery_time

Este parâmetro especifica a hora da primeira tentativa de entrega da mensagem. Pode definir tanto uma hora absoluta quanto uma hora relativa, tomando como base a hora do SMSC. O tipo de codificação da hora será definido adiante.

5.2.16. Validity_period

O parâmetro *validity_period* indica o tempo de expiração da mensagem no SMSC, após o qual a mensagem deverá ser descartada, se não entregue ao destino. O tipo de codificação da hora será definido adiante.

5.2.17. Registry_delivery

O parâmetro *registry_delivery* é utilizado para solicitar um recibo de entrega ao SMSC, assim que a mensagem for entregue ao destino ou assim que a mensagem for apagada do buffer desta entidade. Os valores possíveis são definidos a seguir:

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 0 0 x x	No recipient SME acknowledgment requested (default)
	x x x 0 1 x x	SME Delivery Acknowledgement requested
	x x x 1 0 x x	SME Manual/User Acknowledgment requested
	x x x 1 1 x x	Both Delivery and Manual/User Acknowledgment requested Intermediate Notification (bit 5)
	x x x 0 x x x x	No Intermediate notification requested (default)
	x x x 1 x x x x	Intermediate notification requested **
	<i>all other values reserved</i>	

O valor padrão para o parâmetro *registered_delivery* é 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*

Conforme descrito anteriormente, o campo *replace_if_present_flag* é utilizado para solicitar ao SMSC a reposição de uma mensagem previamente enviada pelo ESME. O SMSC irá repor uma mensagem existente que tenha um determinado *endereço fonte*, *endereço de destino* e *service_type*. Os valores possíveis para o campo *replace_if_present_flag* são apresentados abaixo:

- 0 Não repor (padrão)
- 1 Repor
- 2 - 255 reservado

5.2.19. *data_coding*

Indica o tipo de codificação da mensagem. É representado da seguinte forma:

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

5.2.20. **sm_default_msg_id**

O parâmetro *sm_default_msg_id* especifica o índice de uma mensagem pré definida.

0	reserved
1 - 254	Allowed values
255	Reserved

5.2.21. **Sm_length**

O parâmetro *sm_length* especifica o tamanho da mensagem contida no campo *short_message* em octetos. Caso o campo *message_payload* seja utilizado para transportar a mensagem, este campo deverá ser configurado com o valor zero.

5.2.22. **Short_message**

Neste campo está contida a mensagem, no caso das operações ***submit_sm***, ***deliver_sm*** e ***data_sm***, e, no caso das mensagens de confirmação de entrega utilizando, ***deliver_sm*** e ***data_sm***, o recibo de entrega da mensagem.

5.2.23. **Message_id**

Este campo é preenchido com a única identificação que pode referenciar exatamente uma única mensagem dentre as várias mensagens submetidas. É preenchido de acordo com a implementação do SMSC. É um valor retornado pelo SMSC na mensagem de resposta das operações ***submit_sm***, ***submit_multi_sm***, e ***data_sm***, que será relacionado ao número retornado no recibo de entrega.

5.2.24. **Number_of_dests**

Indica o número de destinos das mensagens ***submit_multi_sm***.

5.2.25. **Dest_flag**

Indica se o número de destino é um assinante único ou uma lista de distribuição

5.2.26. **No_unsuccess**

Indica o número de mensagens que não tiveram sucesso na entrega, no caso das mensagens ***submit_multi_sm***.

5.2.27. **DI_name**

É o nome de referência para a lista de distribuição.

5.2.28. **Message_state**

É a resposta à solicitação ***query_sm***.

6 - Definições Gerais

6.1 - Definições de hora

Neste protocolo, todos os campos relacionados a data e hora seguirão o formato a seguir, codificados em ASCII:

"YYMMDDhhmmsstnp" onde:

"YY"	últimos dois dígitos do ano (00-99)
"MM"	mês(01-12)
"DD"	Dia(01-31)
"hh"	Hora(00-23)
"mm"	Minutos(00-59)
"ss"	Segundos(00-59)
"t"	dezenas de segundos (0-9)
"nn"	diferença de tempo em quarto de horas entre a hora local e a UTC (Universal Time Constant) (00-48)
"p" - "+"	A hora local está avançada em um quarto em relação à UTC
"-"	A hora local está atrasada em um quarto em relação à UTC
"R"	A hora local é relativa à hora corrente no SMSC

6.1.1. Formato de tempo absoluto

Este é o formato de tempo padrão utilizado pelo SMPP. Entregas agendadas e tempo de expiração são especificados no seu formato global UTC, incluindo a variação de um quarto de hora e símbolo de direção "+" ou "-".

6.1.2. Formato de tempo relativo

O tempo relativo pode ser indicado configurando o sinalizador de orientação do UTC para "R" ao invés de "+" ou "-". Desta forma será tomado como base a hora e data do SMSC.

6.2 - Definições de temporizadores

É recomendável que os seguintes temporizadores sejam implementados em sessões SMPP transmitter e receiver. Todos estes temporizadores são configuráveis.

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).

tabela 26 - Tipos de temporizadores do SMPP

7 - Tópico Especial – O Recibo de Entrega do SMPP

O SMPP implementa um recibo de entrega da mensagem pelo SMSC via os PDU's *deliver_sm* e *data_sm*, que indica o status de entrega da mensagem.

O conteúdo informativo de um recibo de entrega pode ser inserido no parâmetro *short_message* da operação de *deliver_sm*. O formato deste recibo de entrega é especificado pelo fabricante mas deve seguir o seguinte formato:

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

Os campos acima são explicados na tabela seguinte:

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.
dlvrD	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)
done date	10	C-Octet Fixed Length String	The time and date at which the short message reached its 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.

tabela 27 - Formato do recibo de entrega das mensagens SMPP

Os estados de entrega da mensagem podem ser os seguintes:

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

tabela 28 - Estados possíveis para a entrega de mensagens SMPP