Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

EL PROTOCOLO DE MENSAJES CORTOS PUNTO A PUNTO (SMPP) SOBRE TCP/IP UTILIZADO EN APLICACIONES DE MENSAJERÍA CORTA

Jaime Miguel García González
Asesorado por la Inga. Ingrid Salomé Rodríguez de Loukota

Guatemala, mayo de 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

EL PROTOCOLO DE MENSAJES CORTOS PUNTO A PUNTO (SMPP) SOBRE TCP/IP UTILIZADO EN APLICACIONES DE MENSAJERÍA CORTA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JAIME MIGUEL GARCÍA GONZÁLEZ
ASESORADO POR LA INGA. INGRID SALOMÉ RODRÍGUEZ DE LOUKOTA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, MAYO DE 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Walter Rafael Véliz Muñoz
VOCAL V	Br. Sergio Alejandro Donis Soto
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Guillermo Antonio Puente Romero
EXAMINADOR	Ing. Marvin Mariano Hernández Fernández
EXAMINADOR	Ing. José Antonio De León Escobar
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

EL PROTOCOLO DE MENSAJES CORTOS PUNTO A PUNTO (SMPP) SOBRE TCP/IP UTILIZADO EN APLICACIONES DE MENSAJERÍA CORTA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Electrónica, con fecha 23 de noviembre de 2010.

Jaime Miguel García González

Guatemala 30 de mayo 2013

Ingeniero
Carlos Eduardo Guzmán Salazar
Coordinador del Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Estimado Ingeniero Guzmán.

Me permito dar aprobación al trabajo de graduación titulado: **EL PROTOCOLO DE MENSAJES CORTOS PUNTO A PUNTO (SMPP) SOBRE TCP/IP UTILIZADO EN APLICACIONES DE MENSAJERÍA CORTA**, del señor **Jaime Miguel García González**, por considerar que cumple con los requisitos establecidos.

Por tanto, el autor de este trabajo de graduación y, yo, como su asesora, nos hacemos responsables por el contenido y conclusiones del mismo.

Sin otro particular, me es grato saludarle.

Atentamente,

Inga. Ingrid Rodríguez de Loukota
Colegiada 5,356
Asesora

Ingrid Rodríguez de Loukota
Ingeniera en Electrónica
colegiado 5356

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

Ref. EIME 10. 2014
Guatemala, 3 de febrero 2014.

Señor Director
Ing. Guillermo Antonio Puente Romero
Escuela de Ingenierìa Mecànica Elèctrica
Facultad de Ingenierìa, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado: EL PROTOCOLO DE MENSAJES CORTOS PUNTO A PUNTO (SMPP) SOBRE TCP/IP UTILIZADO EN APLICACIONES DE MENSAJERÍA CORTA, del estudiante Jaime Miguel García González, que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

Ing. Carlos Eduardo Guzmán Salazar
Coordinador Área Electrónica

sro

RSIDAD DE SAN CARLOS
DE GUATEMALA

ULTAD DE INGENIERIA

REF. EIME 10. 2014.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; JAIME MIGUEL GARCÍA GONZÁLEZ titulado: EL PROTOCOLO DE MENSAJES CORTOS PUNTO A PUNTO (SMPP) SOBRE TCP/IP UTILIZADO EN APLICACIONES DE MENSAJERÍA CORTA, procede a la autorización del mismo.

Ing. Guillermo Antonio Puente Romero

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

GUATEMALA, 14 DE MARZO 2,014.

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

DTG. 233.2014

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **EL PROTOCOLO DE MENSAJES CORTOS PUNTO A PUNTO (SMPP) SOBRE TCP/IP UTILIZADO EN APLICACIONES DE MENSAJERÍA CORTA,** presentado por el estudiante universitario **Jaime Miguel García González,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olympo Paiz Recinos
Decano

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANO
FACULTAD DE INGENIERIA

Guatemala, 20 de mayo de 2014

/gdech

ACTO QUE DEDICO A:

Dios	Por haberme brindado la oportunidad de vivir, cumplir y desarrollarme en todo aspecto de mi vida. Su presencia en todo momento fue importante para realizarme como persona.
Mis padres	Jaime Amílcar García Rivas y Maximina González Juárez cuyo apoyo incondicional me ha brindado la confianza y la oportunidad de salir adelante en cualquier reto en mi vida.
Mis hermanas	Por estar presentes en cada una de las etapas de mi vida brindándome su apoyo en todo momento.
Mis amigos	Por su constante motivación y compañerismo, cada uno ha sido importante en cada etapa de mi vida.

AGRADECIMIENTOS A:

Mis padres	Por ser un ejemplo de vida, por sus consejos y por compartir siempre en familia.
Mis abuelos	Por ser el ejemplo que sigo constantemente, compartiendo conmigo experiencias para poder realizarme como humano y profesional.
Mi asesora	Inga. Ingrid Salomé Rodríguez de Loukota. Por la dedicación, orientación y asesoría académica que me brindó durante mis estudios y la elaboración del presente trabajo de graduación.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
*	Asterisco
%	Porcentaje

GLOSARIO

1G Telefonía móvil de primera generación. Sistema de comunicación móvil, capaz de transmitir únicamente voz análoga.

2G Telefonía móvil de segunda generación. Sistema de comunicación móvil, capaz de transmitir voz en forma digital, llenando necesidades de cifrado y servicios agregados.

3G Telefonía móvil de tercera generación. Sistema de comunicación móvil, con capacidades agregadas como acceso a internet desde el celular, video llamada, correo electrónico.

Byte Un conjunto de *bits* (unos y ceros) de una longitud específica representando un valor en el esquema de codificación de una computadora.

BHSM Busy Hour Short Message, indica la cantidad de mensajes cortos enviados por hora. Acrónimo utilizado para poder realizar una proyección del tráfico SMS en una ESME o SMSC. Unidad de medida de la cantidad de mensajes en hora pico.

Call forward	Característica de algunas redes de telefonía que permiten a una llamada entrante se redireccionada a un tercero.
Called party	Se define de este modo a la persona o dispositivo que recibe una llamada dentro de una red de telefonía.
CCITT	Comité Consultivo Internacional de Telegrafía y Telefonía Comité creado en 1956 (luego renombrado a ITU-T) encargado de la estandarización en telecomunicaciones. Comité Consultivo Internacional de Telegrafía y Telefonía.
CCS	Common Channel Signaling. Red de señalización independiente de los circuitos de voz creada durante los años 60, durante el inicio de la digitalización de las redes de telefonía.
CDMA	Code Division Multiple Access. Método de acceso a canal de comunicación mediante la división de código.
Corriente DC	Es el flujo continuo de electrones a través de un conductor entre dos puntos de distinto potencial.
CRC	Cyclic Redundancy Code. Es un código de detección de error comúnmente usado en redes digitales y

	dispositivos de almacenamiento de información para detectar errores en la información original.
Codificación	Proceso por el cual la información se codifica en símbolos digitales en el transmisor y se decodifican en el receptor para maximizar la precisión de la información presentada al usuario.
CPU	Central Process Unit. El componente de una computadora que ejecuta la lógica computacional y las funciones de decisión.
Datagrama	Mensaje de longitud máxima fija, enviado sin verificación alguna que asegure su exactitud, envío o secuencia correcta con respecto a los mensajes relacionados, que lleva la dirección completa de destino utilizada para el enrutamiento.
Drivers	Un *driver (*software*)* proporciona instrucciones para cambiar el formato o la interpretación de los comandos de software para la transferencia desde y hacia dispositivos periféricos y la unidad de procesamiento central (CPU).
.	
ESME	External Short Message Entity o entidad Externa de Mensajes cortos. Se refiere a una aplicación de contenido capaz de recibir o enviar mensajes cortos de texto.

Emulador	Dispositivo o programa de ordenador que puede actuar como si se trata de otro dispositivo o programa.
Encriptación	El cifrado es el uso de un algoritmo para ocultar el significado de una pieza de información por lo que no puede ser leído y entendido.
End-to-end	Conexión entre el sistema origen y el sistema destino.
Ethernet	Red de área local (LAN), norma oficialmente conocida como IEEE 802.3 y que opera a través del cable par trenzado y por cable coaxial a velocidades a partir de 10 *Mbps*.
Extranet	Red entre organizaciones asociadas.
Frame	Término genérico específico para una serie de protocolos de comunicaciones de datos. Un *frame* de datos es una unidad lógica de datos, que normalmente es un fragmento de un conjunto mucho mayor de datos.
Frame relay	Estándar de acceso definido por el UIT-T en la Recomendación I.122. Servicios *Frame Relay* son aquellos proporcionados por las compañías de telecomunicaciones, empleando una forma de conmutación de paquetes similar a una versión

	simplificada de redes X.25. Los paquetes son en forma de *frames*, que son de longitud variable.
Full duplex	Proceso de funcionamiento de un circuito de modo que los extremos pueden transmitir y recibir simultáneamente.
Gateway	Puerta de enlace. Un *gateway* es un dispositivo repetidor electrónico que intercepta y dirige las señales eléctricas de una red a otra. En redes de datos, usualmente ejecutan procesos de conversión de código y protocolos. De acuerdo al modelo OSI, un *gateway* es un dispositivo que provee servicio a las 7 capas del modelo.
GPS	Constelación de 24 satélites en órbita que permiten determinar con precisión la posición en cualquier lugar de la Tierra con una precisión de un metro.
Handshake	La serie de señales entre un ordenador y otros dispositivos periféricos que establecen los parámetros necesarios para transferir datos.
HDLC	High level Data Link Control. Protocolo de capa de enlace (capa 2 del modelo de referencia OSI) estándar de punto a punto y comunicaciones punto a multipunto.

Header	La porción de un mensaje que contiene la información que guía el mensaje al destino correcto. Esta información puede contener las direcciones del remitente y del receptor, el nivel de prioridad, las instrucciones de ruta y los pulsos de sincronización.
Hops	Viaje individual que los paquetes recorren muchas veces, de *router* a *router*, en su camino hacia sus destinos.
Host	Dispositivo inteligente conectado a una red.
HSSI	High Speed Serial Interface. Interfaz de comunicación serial de datos optimizados para altas velocidades de hasta 52 *Mbps*. Utilizada para conectar un switch ATM a un enlace T-3.
Hub	En redes de área local, un *hub* es el núcleo de una topología de red en estrella.
Hub and spoke	Topología de red semejante a una estrella en la que existe un nodo central por el cual fluye todo el tráfico de los nodos conectados a él.
IEEE 802.3	Protocolo de red de área local comúnmente conocido como Ethernet. Ethernet tiene una velocidad de 10 *Mbps* o 100 *Mbps*.

IEEE 802.5 Un estándar de especificación de capa física de una LAN con un método de acceso de paso de *token* en una topología de anillo.

IETF Internet Engineering Task Force. Formado en 1986 cuando la internet estaba evolucionando de un experimento del Departamento de Defensa en una red académica. El IETF es uno de los dos órganos técnicos de trabajo de la Junta de Actividades de Internet.

IGRP Interior Gateway Routing Protocol. Un protocolo de enrutamiento de vector de distancia desarrollado por Cisco Systems para su uso en grandes redes heterogéneas. Aprende las mejores rutas a través de una LAN o Internet.

Interface Punto de demarcación física entre dos dispositivos en los que las señales eléctricas, conectores y el *handshake* se definen. Los procedimientos, códigos y protocolos que permiten a dos entidades que interactúan en un intercambio significativo de información.

Intranet Una red basada en protocolos TCP / IP (un internet) que pertenece a una organización y es accesible solo por miembros de la organización, los empleados u otras personas con autorización.

IP tunnel	Túnel IP. Técnica para portar cualquier protocolo por medio de un paquete TCP/IP.
IP v4	Internet Protocol Version 4. La versión actual del protocolo de internet.
IP v6	Internet Protocol Version 6. El nuevo protocolo de internet diseñado para sustituir y mejorar el presente Protocolo que se llama IPv4.
IPX	Internet Packet Exchange. Protocolo de comunicaciones Novell NetWare que se utiliza para mover datos entre el servidor y / o programas de la estación de trabajo que se ejecutan en diferentes nodos de la red.
ISDN	Integrated Services Digital Network. Es un conjunto de normas internacionales establecidas por la ITU-T (International Telecommunications Union-Telecommunications Services Sector), para un circuito de conmutación de red digital que permite el acceso a cualquier tipo de servicio (por ejemplo: voz, datos y vídeo).
IS-IS	Intermediate System to Intermediate System. El protocolo IS-IS utiliza un algoritmo de estado de enlace para proporcionar servicios de enrutamiento de TCP/IP y OSI. Se determina el mejor camino para paquetes TCP / IP y OSI a través de la red.

	Mantiene a los *routers* informados del estado de la red y los sistemas disponibles.
ITU	International Telecommunication Union. Organización con sede en Ginebra, Suiza. Es el organismo más importante de telecomunicaciones que fija las normas en el mundo. Está formado actualmente por tres grandes sectores que se establecieron en 1992: Sector de Radiocomunicaciones (ITU-R), el Sector de Desarrollo de las Telecomunicaciones (ITU-D), y el Sector de Normalización de las Telecomunicaciones (ITU-T).
ITU-T Y.101	Norma de la ITU-T donde se formaliza los términos y definiciones de Infraestructura de Información Global.
LAN	Local Area Network. Red de comunicaciones que conecta las computadoras personales, estaciones de trabajo, impresoras, servidores de archivos y otros dispositivos dentro de un edificio o un campus.
Link	El Foro ATM define un enlace (*link*) como una entidad que define una relación topológica (incluyendo la capacidad de transporte disponible) entre dos nodos de diferentes subredes.
Meta señalización	El canal de metaseñal es un canal de gestión que se utiliza para establecer señalización de circuitos

	virtuales entre la red y cada uno de los dispositivos individuales en una interfaz multipunto.
Modelo OSI	Open Systems Interconnection. El modelo de referencia OSI es el único marco internacionalmente aceptado de estándares para la comunicación entre los diferentes sistemas de varios vendedores.
MS	Mobile Station.
MSC	Mobile Service Switching Centre.
MT	Mobile Terminated.
MWI	Message Waiting Indication.
Multiplexación	Intercalar o transmitir simultáneamente dos o más mensajes en un solo canal.
NIC	Network Interface Card. Una tarjeta de red funciona con el software de red y sistema operativo del ordenador para transmitir y recibir mensajes en la red.
NNI	Network Node Interface. Interfaz entre dos segmentos de la red pública de los equipos (a diferencia de la UNI, que representa la interfaz de usuario de red).

NPI	Numbering Plan Indicator.
OSPF	Open Shortest Path First. Utiliza el estado de enlace y protocolos de puerta interior para crear un mapa de la red en cada *router* para luego utilizar el algoritmo Dijkstra del camino más corto y determinar la ruta óptima entre dispositivos de red.
Overhead	En las comunicaciones, es toda la información como el control, ruteo y comprobación de errores que se suman a la transmisión de datos de usuario. Incluye información que lleva el estado de la red o las instrucciones de funcionamiento, la información de enrutamiento de red, así como retransmisiones de mensajes de datos por el usuario que se han recibido con error.
Paquetes	Término genérico para un conjunto de datos, por lo general en forma binaria, organizados de una manera específica para su transmisión.
Paridad	Un proceso para detectar si los *bits* han sido alterados durante su transmisión.
Payload	Carga de información. La parte que representa la información útil para el usuario.
PCM	Pulse Code Modulation. Método de codificación de una señal analógica en una señal digital.

Peer to peer	Comunicación entre dos entidades que operan dentro de la misma capa de protocolo de un sistema.
Protocolo	Es un conjunto de normas que regulan el formato de los mensajes que se intercambian entre máquinas y personas.
Protocolo internet	Parte de la familia de protocolos TCP/IP que describen el software que controla la dirección de Internet de los nodos, las rutas de los mensajes salientes y reconoce los mensajes entrantes. Se utiliza en puertas de enlace para conectar las redes a nivel de capa de red y superiores.
QoS	Quality of Service. Calidad de servicio es una medida del servicio que las telecomunicaciones (voz, datos y/o vídeo) prestan a un abonado.
Ráfaga	En la comunicación de datos, una secuencia de señales, ruido, o interfaz tomada como una unidad de acuerdo con algún criterio o medida específica.
Receptor	Cualquier componente de un dispositivo de telecomunicaciones que decodifica una señal codificada en su forma deseada.
Retardo de propagación	El tiempo que tarda una señal en viajar desde el emisor hasta el receptor a través de un circuito.

	El retardo de propagación es un factor de la velocidad finita en la que las señales electromagnéticas pueden viajar a través de un medio de transmisión
Router	Un dispositivo que conecta dos segmentos de LAN, que utilizan arquitecturas similares o diferentes en la capa de red OSI, la capa 3. El *router* determina la ruta más eficaz para transmitir datos por medio del Internet. Los paquetes que contienen una dirección de red diferente a la dirección del PC de origen se remiten a una red contigua.
Señal portadora	Una forma de onda continua (normalmente eléctrica), cuyas propiedades son susceptibles de ser modulada con una segunda señal portadora de información.
Símbolo	Estado eléctrico reconocible el cual está asociado con un elemento de señal que es una señal eléctrica dentro de un período de tiempo definido.
Síncrono	Sistema que utiliza una tasa de reloj para la transmisión de datos.
SFE	Store and Forward Engine.
SM	Short Message.
SMDS	Switched Multimegabit Data Service. Una conexión de alta velocidad de datos de servicios de

	transmisión destinada a la aplicación en una red de área metropolitana (MAN).
SMPP	Short Message Peer to Peer.
SMSC	Short Message Service Center.
SMS	Short Message Service.
Socket	Una abstracción del sistema operativo que proporciona la capacidad a los programas de aplicación para acceder de forma automática los protocolos de comunicación. Desarrollado como parte de los primeros trabajos sobre TCP/IP.
SS7	Signaling System #7. Conjunto de protocolos de señalización telefónica, utilizados en la mayoría de redes de telefonía.
Stack	Lista de instrucciones internas que se ejecutan en un sistema.
Switch	Conmutador. Un dispositivo mecánico, eléctrico o electrónico que abre o cierra los circuitos, completa o interrumpe una trayectoria eléctrica, o selecciona vías o circuitos. Los *switches* trabajan en las capas 1 (Física) y 2 (enlace de datos) del modelo de referencia OSI, haciendo énfasis en la capa 2. Un *switch* verifica los datos entrantes (datos o voz) para

	determinar la dirección de destino. Con base en esa dirección, una conexión se establece a través de la matriz de conmutación entre los puertos físicos de entrada y salida del dispositivo.
Tasa de errores *bit*	Porcentaje de *bits* recibidos con error en comparación con el número total de bits recibidos.
TCP	Transmission Control Protocol. TCP es un protocolo de capa de transporte, orientado a la conexión. Proporciona una entrega confiable, ordenada, y no duplicada de *bytes* enviados a un usuario remoto o local.
TDM	Time Division Multiplex. Una técnica de multiplexación para la transmisión de una serie de señales separadas de datos, voz y/o video simultáneamente a través de un medio de comunicación intercalando una pieza de cada señal una después de la otra.
TON	Type of Number.
TRX	Transceiver Channel.
TX	Transmitter Channel.

UDP	User Datagram Protocol. Provee el intercambio de datagramas sin acuses de recibo ni entrega garantizada.
Uplink	En los satélites, es el enlace de la estación de tierra hasta el satélite.
VMS	Voice Messaging System.
WWW	World Wide Web. Universo de información accesible disponible en muchos equipos extendidos por el mundo y unidos a la red de computadoras gigantescas que se llama internet. La web tiene un cuerpo de software, un conjunto de protocolos y un conjunto de convenciones definidas para llegar a la información en la web.

RESUMEN

El presente trabajo de graduación tiene como propósito reunir en un único documento la información necesaria para dar a conocer el protocolo SMPP utilizado para establecer comunicación y realizar el envio de mensajes de texto entre entes tales como un centro de mensajería y proveedores de contenido.

Actualmente en la rama de telecomunicaciones es necesario integrar nuevos servicios de valor agregado para brindarlos a usuarios de telefonía, los cuales puedan satisfacer necesidades de intercambiar información por medio de mensajes de texto, brindando a un usuario final el envío de información necesaria para estar actualizado sobre un tema en particular. Al integrar estos servicios de mensajería en una red de telefonía celular, haciéndola capaz por medio de un protocolo estándar de comunicación como el SMPP, en el cual un usuario de una red de telefonía pueda intercambiar mensajes de texto entre redes de distintos operadores locales como internacionales, así como suscribirse a servicios con proveedores de contenido locales e internacionales para recibir informacion que le brinden un valor agregado.

En el capítulo 1 se presenta la explicación de funcionalidad de una red TCP/IP, brindando detalles sobre el significado del modelo OSI; modelo que brinda una explicación general sobre la comunicación de dos máquinas en diferentes niveles; desde el físico hasta llegar al nivel de aplicación. Utilizando el modelo TCP/IP para explicar la forma de comunicación entre dos máquinas, partiendo desde los conceptos básicos hasta el uso de *sockets* para poder realizar el envío de paquetes, con una trama que posee una estructura en

particular para intercambiar información en una conexión basada que permite validar el envío de paquetes para lograr transmitir exitosamente cada paquete.

En el capítulo 2 se expone el significado de una sesión SMPP, donde básicamente se explica el tipo de sesiones existentes en el protocolo, así como comprender el tipo de operación de envío o recepción de mensajes que es posible establecer. Se explican las versiones de SMPP existentes o creadas en la actualidad, así como el listado de código de errores establecidos en el estándar para contar con un código de errores general, en cualquier parte del mundo e indispensable para que cualquier operador de telefonía pueda hablar el mismo lenguaje.

En el capítulo 3 se muestra el concepto del PDU, término importantísimo para conocer la información necesaria incluida en el paquete para poder transmitir un mensaje. Este capítulo explica cómo se establece la comunicación y se mantiene una correspondencia a nivel de secuencia, entre los dos equipos que necesitan estar conectados e intercambiando información de forma constante.

En el capítulo 4 se presenta el protocolo SMPP aplicado a centros de mensajería o aplicaciones ESME tambien llamadas proveedores de contenido. Este capítulo presenta una explicación del funcionamiento de estos dos entes desde su arquitectura, así como su funcionamiento. En el mismo capítulo se presenta una explicación de una aplicación utilizada para realizar pruebas de funcionalidad del protocolo SMPP, y poder realizar *troubleshooting*, en caso de falla o utilizada, también, para certificar la correcta funcionalidad de una conexión entre dos operadores o proveedores de contenido.

OBJETIVOS

General

El protocolo de mensajes cortos punto a punto (SMPP) sobre TCP/IP tiene como objetivo explicar cómo se realiza la comunicación entre dos entes de mensajería; para conseguir intercambiar mensajes cortos entre dos usuarios o con una aplicación de contenido de la misma red u otra red operadora externa.

Específicos

1. Presentar los fundamentos de redes TCP/IP necesarios para definir los modelos básicos y la tecnología que utiliza de base este protocolo para interactuar y ser aplicativa.

2. Presentar los fundamentos del protocolo SMPP para explicar cada una de sus formas de funcionamiento y de operación del protocolo.

3. Presentar los fundamentos de los parámetros y formatos del protocolo SMPP, para validar las ventajas de utilizar modos síncronos o asíncronos en el intercambio de información, dependiendo la sesión que se desea establecer entre la aplicación y el centro de mensajería.

4. Presentar los fundamentos del protocolo SMPP para poder crear un documento que brinde la información indispensable, para poder desarrollar aplicaciones que utilicen el protocolo de mensajes cortos punto a punto (SMPP), tales como SMSC o proveedores de contenido.

INTRODUCCIÓN

El desarrollo y aplicación del protocolo de mensajes cortos punto a punto (SMPP) sobre TCP/IP, es el medio que se ha estudiado e implementado en la industria de las comunicaciones con el fin de intercambiar información sobre usuarios de telefonía móvil, brindando así medios de comunicación para transferir información valiosa o de importancia para un usuario.

Con el paso del tiempo se han desarrollado tecnologías capaces de hacer que en niveles de plataformas y/o aplicaciones se implementen servicios que puedan darles ventajas al consumidor, haciendo más fácil el adecuarse a los cambios en la tecnología para obtener información o realizar pagos por otros medios como un celular, utilizando un mensaje de texto (SMS) para hacerlo.

La importancia de conocer el desarrollo de este protocolo de mensajes cortos punto a punto (SMPP) es comprender su estructura y funcionamiento para poder aplicarlos en el desarrollo de aplicaciones que brinden bienestar y facilidad al usuario final que desee utilizar esta tecnología.

1. REDES TCP/IP

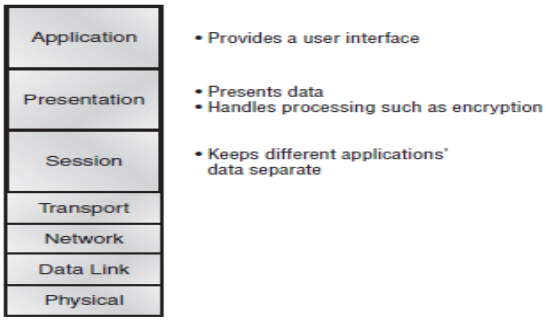
1.1. Modelo OSI

Una de las grandes funciones de las especificaciones OSI es asistir en la transferencia de datos entre elementos de distinta tecnología, tal es el ejemplo de hacer posible transferir datos entre equipos Unix y una PC o computadora de marca X.

OSI no es un modelo físico, es una guía de pasos que desarrolladores de aplicaciones pueden utilizar para crear e implementar aplicaciones que puedan funcionar sobre una red. Este también provee *framework* para crear estándar en redes, dispositivos y esquemas de interconexión de red.

El modelo OSI tiene siete diferentes capas, divididas en 2 grupos. Las primeras 3 capas superiores definen cómo las aplicaciones puedan comunicarse con cada una de las otras capas y con otros usuarios. Las últimas 4 capas inferiores definen como los datos son transmitidos de un punto a otro. La figura 1 muestra las tres capas superiores y sus funciones y la figura 2 muestra las cuatro capas inferiores y sus funciones.

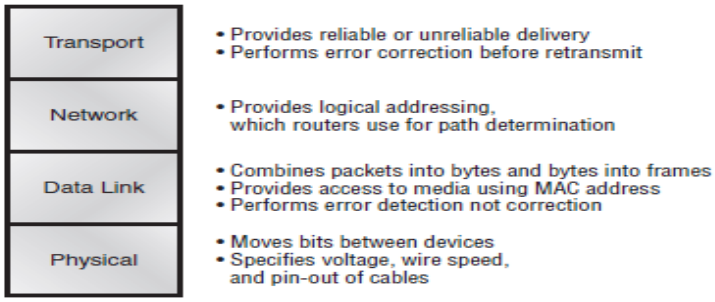
Figura 1. **Capas Superiores y sus funciones del modelo OSI**



Fuente: Cisco Certified Network Associate. *Study guide*. p. 13.

Cuando se estudia la figura 1 se comprende que las interfaces de usuarios con la computadora en la capa de aplicación y también las capas superiores, son responsables para la comunicación de aplicaciones entre los equipos. Es importante recordar que las capas superiores no conocen nada acerca de las direcciones de red o de redes.

Figura 2. **Capas inferiores y sus funciones del modelo OSI**



Fuente: Cisco Certified Network Associate. *Study guide*. p. 14.

En la figura 2, se puede apreciar que estas son las 4 capas inferiores que definen como los datos son transferidos a través de los cables físicos o a través de *switches* y *routers.* Estas capas inferiores también determinan cómo reconstruir los fragmentos de datos de una transmisión remota a una aplicación remota destinataria.

Los siguientes dispositivos de red operan en las siete capas del modelo OSI:

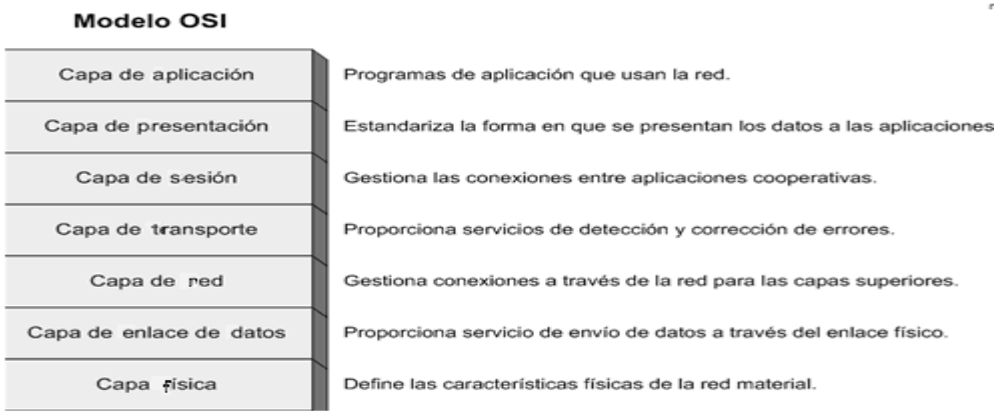
- Estaciones de administración de redes
- Servidores de aplicación y web
- Servidores Gateways
- Equipos de red

La referencia del modelo OSI cuenta con siete capas:

- Capa de aplicación (capa 7)
- Capa de presentación (capa 6)
- Capa de sesión (capa 5)
- Capa de transporte (capa 4)
- Capa de red (capa 3)
- Capa de enlace de datos (capa 2)
- Capa física (capa 1)

En la figura 3 se muestra un resumen de las funciones de cada una de las capas del modelo OSI.

Figura 3. **Capas del modelo OSI**



Fuente: elaboración propia.

Este es un esquema que brinda las bases para la creación de muchos protocolos siendo el caso del protocolo SMPP, el modelo específico de protocolo que debe ser utilizado en cada capa, este es usado como referencia ya que el mismo explica la enseñanza de comunicación de redes.

El modelo está dividido en 7 capas que descritas a continuación:

NIVEL 1 Capa física**:** esta capa, tiene a su cargo las conexiones físicas de una computadora hacia una red, correspondiendo para esta capa el medio físico así como la forma en la que se transmite la información.

Las funciones fundamentales son:

- Definir el medio físico por los que las señales eléctricas se transportarán. Estos medios pueden ser: cable de pares trenzados, coaxial, guías de onda, aire, fibra óptica.

- Definir las características materiales (componentes y conectores mecánicos), así como las eléctricas (niveles de tensión) que se utilizan en la transmisión de los datos por los medios físicos.
- Transmitir la información a través del medio, manejar las señales eléctricas del medio de transmisión, garantizando la conexión física.

NIVEL 2 Capa de enlace de datos: el nivel de enlace de datos (en inglés *data link level*) es responsable de la transferencia fiable de información a través de un circuito de transmisión de datos. Recibe peticiones de la capa de red y utiliza los servicios de la capa física. La capa de enlace de datos es responsable de la transferencia fiable de información a través de un circuito eléctrico de transmisión de datos. La transmisión de datos lo realiza mediante tramas, que son las unidades de información con sentido lógico para el intercambio de datos en la capa de enlace. También hay que tener en cuenta que en el modelo TCP/IP corresponde a la segunda capa.

Sus principales funciones son:

- Iniciación, terminación e identificación
- Segmentación y bloqueo
- Sincronización de octeto y carácter
- Delimitación de trama y transparencia
- Control de errores
- Control de flujo
- Recuperación de fallos
- Gestión y coordinación de la comunicación

Para lograr este objetivo tiene que montar bloques de información (llamados tramas en esta capa), dotarles de una dirección de capa de enlace (dirección MAC), gestionar la detección o corrección de errores, y ocuparse del

control de flujo entre equipos (para evitar que un equipo más rápido desborde a uno más lento).

En la práctica, la subcapa de acceso al medio suele formar parte de la propia tarjeta de comunicaciones, mientras que la subcapa de enlace lógico estaría en el programa adaptador de la tarjeta (*driver* en inglés).

NIVEL 3 Capa de red: se encarga de identificar el enrutamiento existente entre una o más redes. Las unidades de información se denominan paquetes, y se pueden clasificar en protocolos enrutables y protocolos de enrutamiento.

- Enrutables: viajan con los paquetes (IP, IPX, APPLETALK).
- Enrutamiento: permiten seleccionar las rutas (RIP, IGRP, EIGRP, OSPF, BGP).

El objetivo de la capa de red es hacer que los datos lleguen desde el origen al destino, aun cuando ambos no estén conectados directamente. Los dispositivos que facilitan tal tarea se denominan encaminadores, aunque es más frecuente encontrarlo con el nombre en inglés *routers*. Los *routers* trabajan en esta capa, aunque pueden actuar como *switch* de nivel 2 en determinados casos, dependiendo de la función que se le asigne. Los *firewalls* actúan sobre esta capa principalmente, para descartar direcciones de máquinas. En este nivel se realiza el direccionamiento lógico y la determinación de la ruta de los datos hasta su receptor final.

NIVEL 4 Capa de transporte: capa encargada de efectuar el transporte de los datos (que se encuentran dentro del paquete) de la máquina origen a la de destino, independizándolo del tipo de red física que se esté utilizando. La PDU de la capa 4 se llama segmento o datagrama, dependiendo de si corresponde a

TCP o UDP. Sus protocolos son TCP y UDP; el primero orientado a conexión y el otro sin conexión. Trabajan, por lo tanto, con puertos lógicos y junto con la capa red dan forma a los conocidos como Sockets IP: Puerto (191.16.200.54:80).

NIVEL 5 Capa de sesión: esta capa es la que se encarga de mantener y controlar el enlace establecido, entre dos computadores que están transmitiendo datos de cualquier índole. Por lo tanto, el servicio provisto por esta capa, es la capacidad de asegurar que dada una sesión establecida entre dos máquinas, la misma se pueda efectuar para las operaciones definidas de principio a fin, reanudándolas en caso de interrupción. En muchos casos, los servicios de la capa de sesión son parcial o totalmente prescindibles.

NIVEL 6 Capa de presentación: el objetivo es encargarse de la representación de la información, de manera que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres los datos lleguen de manera reconocible.

Esta capa es la primera en trabajar más el contenido de la comunicación que el cómo se establece la misma. En ella se tratan aspectos tales como la semántica y la sintaxis de los datos transmitidos, ya que distintas computadoras pueden tener diferentes formas de manejarlas.

También permite cifrar los datos y comprimirlos. Por lo tanto, podría decirse que esta capa actúa como un traductor.

NIVEL 7 Capa de aplicación: ofrece a las aplicaciones la posibilidad de acceder a los servicios de las demás capas, y define los protocolos que utilizan

las aplicaciones para intercambiar datos, como correo electrónico (Post Office Protocol y SMTP), gestores de bases de datos y servidor de ficheros (FTP), por UDP pueden viajar (DNS y Routing Information Protocol). Hay tantos protocolos como aplicaciones distintas, y puesto que continuamente se desarrollan nuevas aplicaciones el número de protocolos crece sin parar.

Cabe aclarar que el usuario normalmente no interactúa directamente con el nivel de aplicación. Suele interactuar con programas que a su vez interactúan con el nivel de aplicación pero ocultando la complejidad subyacente.

Entre todos estos métodos de señalización, el más importante de ellos es CCS, y se discutirá del mismo más adelante. En la siguiente sección se introducirá SS7 y se discutirá sobre su funcionalidad y arquitectura.

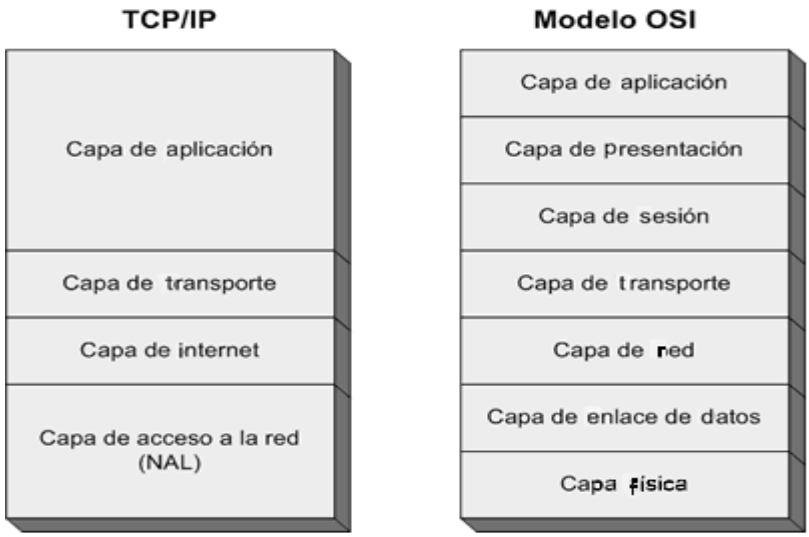
1.2. Modelo TCP/IP

El modelo de arquitectura de estos protocolos es más simple que el modelo OSI, como resultado de la agrupación de diversas capas en una sola o bien por no usar alguna de las capas propuestas en dicho modelo de referencia.

Así, por ejemplo, la capa de presentación desaparece pues las funciones a definir en ellas se incluyen en las propias aplicaciones. Lo mismo sucede con la capa de sesión, cuyas funciones son incorporadas a la capa de transporte en los protocolos TCP/IP. Finalmente la capa de enlace de datos no suele usarse en dicho paquete de protocolos.

De esta forma se queda con una modelo en cuatro capas, tal y como se ve en la figura 4:

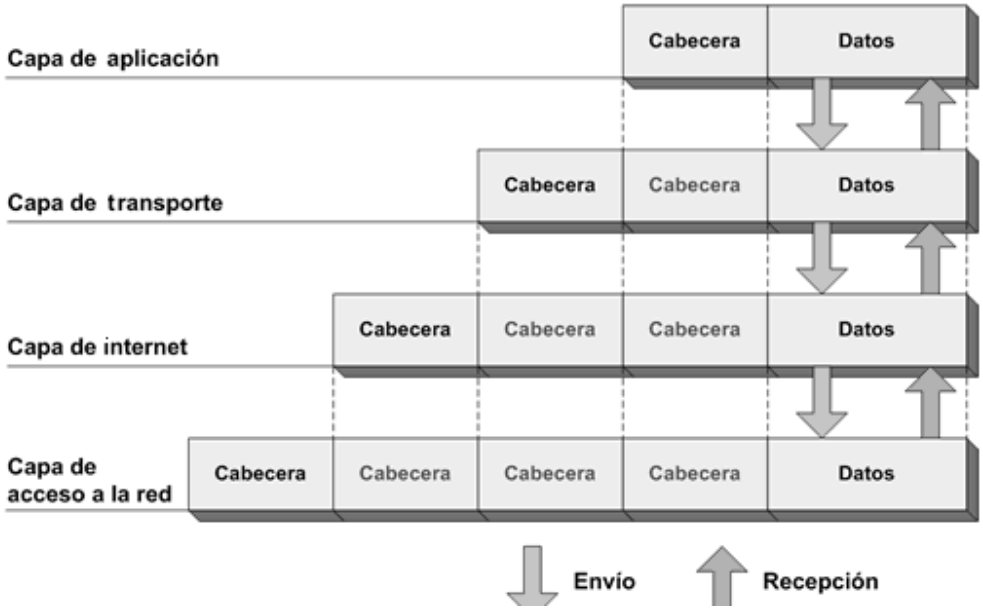
Figura 4. **Modelo TCP/IP**



Fuente: elaboración propia.

Al igual que en el modelo OSI, los datos descienden por la pila de protocolos en el sistema emisor y la escalan en el extremo receptor. Cada capa de la pila añade a los datos a enviar a la capa inferior, información de control para que el envío sea correcto. Esta información de control se denomina cabecera, pues se coloca precediendo a los datos. A la adición de esta información en cada capa se le denomina encapsulación. Cuando los datos se reciben tiene lugar el proceso inverso, es decir, según los datos ascienden por la pila, se van eliminando las cabeceras correspondientes.

Figura 5. **Modelo TCP/IP explicado con estructura de un PDU**



Fuente: elaboración propia.

Cada capa de la pila tiene su propia forma de entender los datos y, normalmente, una denominación específica que podemos ver en la tabla 1. Sin embargo, todos son datos a transmitir y los términos solo indican la interpretación que cada capa hace de los datos.

Tabla I. **Explicación de modelo TCP/IP estructurado por capas internas del mensaje**

	TCP	**UDP**
Capa de aplicación	Flujo	Mensaje
Capa de transporte	Segmento	Paquete
Capa de internet	Datagrama	Datagrama
Capa de acceso a la red	Trama	Trama

Fuente: elaboración propia.

Capa de acceso a la red

Los protocolos de esta capa proporcionan al sistema los medios para enviar los datos a otros dispositivos conectados a la red, es en esta capa donde se define como usar la red para enviar un datagrama. Es la única capa de la pila cuyos protocolos deben conocer los detalles de la red física. Este conocimiento es necesario pues son estos protocolos los que han de dar un formato correcto a los datos a transmitir, de acuerdo con las restricciones que imponga, físicamente, la red.

Las principales funciones de los protocolos definidos en esta capa son:

- Encapsulación de los datagramas dentro de los marcos a transmitir por la red.
- Traducción de las direcciones IP a las direcciones físicas de la red.

1.3. Ethernet/IP

Capa de internet

El protocolo más importante de esta capa y piedra base de toda la internet es el IP, proporciona los servicios básicos de transmisión de paquetes sobre los cuales se construyen todas las redes TCP/IP, las funciones de este protocolo incluyen:

- Definir del datagrama, que es la unidad básica de transmisión en internet.
- Definir el esquema de direccionamiento de internet.
- Mover los datos entre la capa de acceso a red y la capa de transporte.
- Encauzar los datagramas hacia sistemas remotos (*routing*)
- Realizar la fragmentación y reensamblaje de los datagramas.

El protocolo IP es un protocolo sin conexión, es decir, no intercambia información de control para establecer una conexión antes de enviar los datos. En caso de que dicha conexión fuese necesaria, el IP delegará tal labor en protocolos de otras capas.

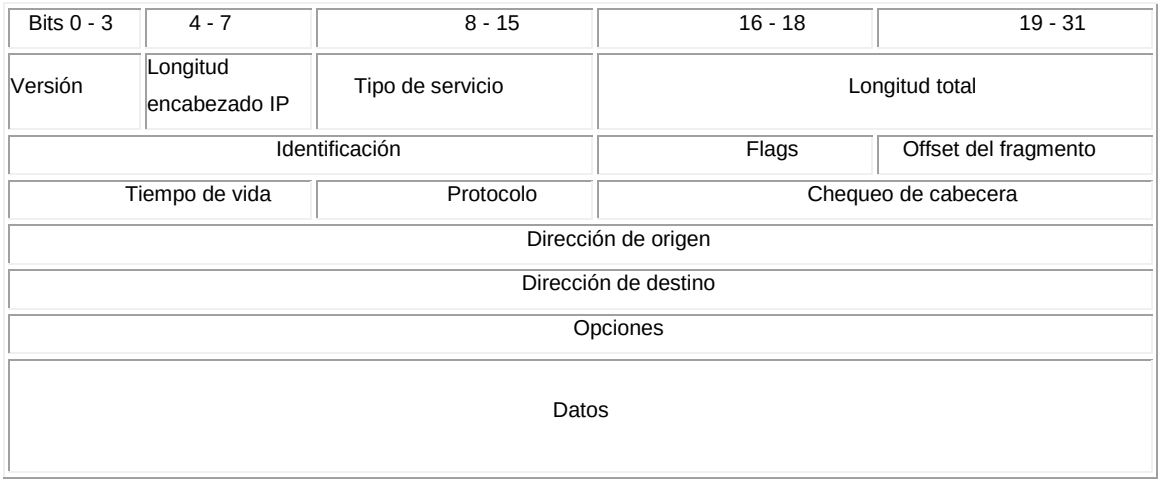
Este protocolo tampoco realiza detección de errores o recuperación de datos ante los mismos.

Los protocolos TCP/IP fueron diseñados para el intercambio de datos en ARPANET, que era una red de intercambio de paquetes. Un paquete es un bloque de datos que lleva consigo la información necesaria para enviarlo. Para aclarar esto se podría comparar un paquete con una tarjeta postal, en la que no solo se escribe un mensaje sino que además se añade los datos pertinentes para que llegue a su destinatario, nombre, dirección, etc.

Una red de intercambio de paquetes usa esta información para cambiar los paquetes de una red a otra moviéndolos hacia su destino final. Cada paquete navega por la red independientemente de cualquier otro paquete.

El datagrama es el formato del paquete que define el IP, y consta de dos partes: la cabecera y los datos.

Figura 6. **Estructura de un paquete IPv4**



Fuente: elaboración propia.

A la hora de enviar un datagrama, el IP comprueba la dirección de destino. Aquí surgen dos posibilidades:

- Que el destino sea una máquina de la red local. En este caso se envía el datagrama directamente a dicha máquina y listo.

- Que el destino sea una máquina perteneciente a otra red física. En este caso el IP encauzará el datagrama a través de *gateways* hacia su destino. El término inglés de este encauzamiento, normalmente más usado que el español, es *routing.*

Con la segunda posibilidad surge un problema más. Puesto que el datagrama va a atravesar distintas redes físicas, puede darse el caso de que su tamaño no sea adecuado para la transmisión a través de estas redes, pues cada tipo de red define un tamaño máximo para los paquetes que pueden circular por ella.

En este caso, cuando llegue al *gateway*, el IP fragmentará el datagrama en piezas más pequeñas, y a efectos de facilitar su ensamblaje posterior en la cabecera de cada pieza resultante se especificará a que datagrama pertenece y que posición tiene la pieza dentro del datagrama. Para el ensamblaje de las piezas se comprueban estos campos de la cabecera y otro más en el que se indica si hay más fragmentos que ensamblar o no.

Una vez que el datagrama llega a la máquina de destino, y en concreto a la capa de Internet, el IP habrá de enviarlo al protocolo correspondiente de la capa de transporte. Los protocolos de dicha capa tienen asignados unos números que los identifican y que quedan registrados en la cabecera del datagrama.

Otro protocolo definido en la capa de internet es el ICMP, protocolo de control de mensajes en internet. Dicho protocolo usa el sistema de envío de mensajes del IP para enviar sus propios mensajes.

Los mensajes enviados por este protocolo realizan las siguientes funciones:

- Control de flujo: cuando los datagramas llegan demasiado rápido a una máquina, de forma que esta no tiene tiempo para procesarlos, el ICMP

de dicha máquina enviará al emisor de los datagramas un mensaje para que detenga el envío temporalmente.

- Detección de destinos inalcanzables: cuando no se puede alcanzar la dirección de destino de un datagrama, la máquina que detecta el problema envía a la dirección de origen de ese datagrama un mensaje notificando dicha situación.

- Redirección de rutas: cuando a un *gateway* le llega un datagrama a enviar a una máquina y existe otro *gateway* que resulta ser una opción mejor para enviar dicho datagrama, el primer *gateway* envía el emisor un mensaje comunicándole dicha situación para que el envío se haga a través del segundo *gateway*.

- Chequeo de sistemas remotos: una máquina que necesite saber si otra máquina de otra red está conectada y operacional le enviará un mensaje, llamado echo, que la otra máquina devolverá si está conectada y operacional. El comando ping de Unix utiliza este protocolo.

Capa de transporte

Los dos protocolos más importantes de esta capa son el TCP y el UDP. El primero se encarga de los servicios de envío de datos con detección y corrección de errores. El UDP proporciona servicios de envío de datagramas sin conexión.

El protocolo UDP proporciona a los programas de aplicación acceso directo al envío de datagramas, parecido al servicio que proporciona el IP. Este

permite a las aplicaciones intercambiar mensajes con un mínimo de supervisión por parte del protocolo.

Este protocolo se usa principalmente en:

- Envío de pequeñas cantidades de datos, pues sería más costoso supervisar el establecimiento de conexiones y asegurar un envío fidedigno que retransmitir el conjunto de datos completo.

- Aplicaciones que se ajustan al modelo pregunta-respuesta. La respuesta se puede usar como una confirmación a la pregunta. Si no se recibe respuesta en un cierto periodo de tiempo, la aplicación, simplemente, vuelve a enviar la pregunta.

- Aplicaciones que tienen su propio sistema de verificar que el envío de datos ha sido fidedigno y no requieren este servicio de los protocolos de la capa de transporte.

Las aplicaciones que requieren de la capa de transporte un servicio de transmisión de datos fidedigno, usan el protocolo TCP. Este protocolo verifica que los datos se envíen a través de la red adecuadamente y en la secuencia apropiada. Las características de este protocolo son:

- Fiabilidad
- Orientado a la conexión y al flujo de datos

Para lograr la fiabilidad, el TCP se basa en un mecanismo de confirmación positiva con retransmisión (PAR, del inglés, Positive Acknowledgement with Retransmision).

Básicamente, consiste en que el emisor envíe los datos una y otra vez, hasta que reciba una confirmación de la llegada de los datos en perfecto estado.

Cada segmento de datos contiene un campo de chequeo que el sistema receptor usa para verificar la integridad de los datos. Para cada segmento recibido correctamente se envía una confirmación. Los segmentos dañados se eliminan. Tras un cierto periodo de tiempo, el emisor volverá a enviar todos aquellos segmentos para los que no ha recibido confirmación.

El protocolo TCP es un protocolo orientado a la conexión. Este protocolo establece una conexión entre las dos máquinas que se comunican, se intercambia información de control antes y después de la transmisión de los datos.

El TCP ve los datos que envía como un flujo continuo de bytes, no como paquetes independientes. Debido a esto, es necesario enviarlos en la secuencia adecuada. El TCP se cuida de mantener esta secuencia mediante los campos de número de secuencia y número de confirmación de la cabecera de segmento.

En el paso de información de control que realiza el TCP antes de establecer la conexión, se intercambian tres paquetes. Dicho intercambio se denomina apretón a tres vías.

En el primer segmento, el emisor comunica al receptor el número inicial de su secuencia. Esto se realiza poniendo este número en el campo número de secuencia de la cabecera del segmento, y activando el *bit* de sincronización de números de secuencia.

Cuando este segmento llega al receptor este contesta enviando:

- Su propio número inicial de secuencia, en el campo de número de secuencia y activando el *bit* de sincronización.

- La confirmación de recepción, indicando en el campo de confirmación el número inicial de secuencia del emisor y activando el *bit* de confirmación.

Cuando el segundo segmento llega al emisor, este confirma la recepción del mismo enviando un tercer segmento con el número de inicio de secuencia del receptor en el campo de número de confirmación y el *bit* de confirmación activado.

En este momento, el emisor tiene plena conciencia de que la máquina receptora está operacional y lista para recibir sus datos, así pues se inicia el envío de los mismos.

Según se van recibiendo datos, el receptor irá indicando al emisor la correcta recepción de los mismos. Esto se realiza periódicamente, enviando al emisor un segmento con el *bit* de confirmación activado y el número de secuencia del último byte recibido correctamente. De esta forma se evitará el tener que enviar una confirmación con cada byte recibido.

En el campo de ventana de la cabecera de este mismo segmento se indica el número de bytes que el receptor es capaz de aceptar. Este número indica al emisor que puede continuar enviando segmentos siempre y cuando la longitud en bytes de estos sea inferior al tamaño de la ventana. Un tamaño de ventana cero, indicará al emisor que detenga el envío de segmentos hasta recibir un segmento con tamaño de ventana mayor que cero.

Cuando el emisor termina de enviar los datos se establece otro apretón a tres vías que difiere del que ha tenido lugar como inicio de la conexión únicamente, en que en vez de llevar activado el *bit* de sincronización, los segmentos llevarán activado el *bit* de fin de transmisión de datos.

El TCP es también responsable de enviar los datos recibidos a la aplicación correcta. La aplicación a la que se destina los datos está identificada por un número de 16 *bits* llamado número de puerto. El número de puerto, tanto del origen como del destino, se especifica en la cabecera de cada segmento.

Capa de aplicación

En esta capa se incluyen los procesos que usan los protocolos de la capa de transporte. Hay muchos protocolos de aplicación, la mayor parte proporcionan servicios de usuario y constantemente se añaden nuevos servicios. Algunos de los protocolos más conocidos de esta capa son:

- Telnet es un protocolo que permite la conexión remota de terminales
- FTP es utilizado para efectuar transferencias interactivas de ficheros
- SMTP es el protocolo que nos permite enviar correo a través de la red

Estos tres protocolos hacen uso de los servicios orientados a la conexión del TCP.

Algunos protocolos que, en cambio, usan los servicios del UDP son:

- DNS: protocolo que traduce en direcciones IP los nombres asignados a los dispositivos de la red.

- NFS: protocolo que permite la compartición de ficheros por distintas máquinas de una red.
- RIP: utilizado por los dispositivos de la red para intercambiar información relativa a las rutas a seguir por los paquetes.

1.4. *Sockets*

Un *socket* es una abstracción con la que se denomina a uno de los extremos de una comunicación, es una generalización del mecanismo de acceso al sistema capaz de manejar la comunicación entre procesos que se comunican de manera uniforme sobre una sola máquina o en un ambiente de red.

Un *socket* es un mecanismo de comunicación, normalmente es identificado por un entero que puede ser llamado *socket* descriptor. El mecanismo de *socket* fue introducido por primera vez en 1983 en el sistema Unix BSD 4.2 en conjunto con los protocolos TCP/IP que aparecieron por primera vez a finales de 1981 en el Sistema Unix BSD 4.1.

Formalmente un *socket* es definido por un grupo de cuatro números, a saber:

- El número de identificación o dirección del *host* remoto
- El número puerto del *host* remoto
- El número de identificación o dirección del *host* local
- El número puerto del *host* local

Para los programadores de aplicaciones el mecanismo de *sockets* es accedido mediante un número de funciones o primitivas. Las primitivas involucradas con los *sockets* están implementadas como un conjunto de

llamadas al sistema (*systems calls),* que proveen el acceso a los servicios de transporte por parte de los programas de usuario, y se convierten así en la interfaz entre las aplicaciones de red y las capas más bajas de la red. A partir de aquí, se analizarán las llamadas más comunes, trabajando bajo una arquitectura cliente-servidor y un servicio de comunicaciones orientado a conexiones.

El primer paso para trabajar con *sockets* es decidir qué aplicación va a utilizarlos y qué tipo de servicios se le exigirá al nivel de transporte. Con las respuestas a mano, antes de que un *socket* pueda referenciarse debe creárselo mediante la primitiva adecuada. La llamada al sistema para llevar a cabo tal operación tiene la forma:

- Intsocket *(intsock_family, intsock_type, intprotocol):* el argumento *sock_type* selecciona un modo de transporte, entiéndase un servicio orientado a conexiones o uno sin conexión. El argumento *sock_family* identifica unívocamente a la familia de direcciones que se utilizará para referenciar el *socket*. El último argumento específica el protocolo de comunicaciones dentro de la familia seleccionada a utilizar sobre el *socket*. Colocando en cero este último parámetro se deja que el sistema decida el protocolo más adecuado. Si la operación fue correcta, la llamada *socket()* retorna un entero denominado *socket descriptor* (análogo a un *file descriptor*) que puede utilizarse para referenciar al *socket* en cualquier otra operación que se efectúe sobre él. En caso de falla, devuelve -1.

Después de su creación, un *socket* debe asociarse a una dirección local que permita su utilización por parte de los procesos interesados. Esta operación se efectúa mediante la llamada *bind()*, cuyo formato es el siguiente:

- Intbind *(intsock_*descr*, structsockaddr *addr, intaddrlen):* el argumento *sock_descr* es el *socket descriptor* utilizado para referenciar al *socket* con que se trabaja. El argumento *addres* un puntero a la estructura que contiene la dirección que quiere asociarse al *socket* y *addrlen* es el tamaño de la estructura en bytes. *bind()* retorna 0 si la llamada fue exitosa y -1 en caso de error (por ejemplo, si el *socket* apuntando ya está siendo usado por otro proceso). Los servidores necesariamente deben especificar una dirección para cada uno de sus *sockets* mientras que los clientes no necesitan obligatoriamente asociarse a una dirección específica pudiendo dejar tales cuestiones al sistema.

En interacciones basadas en un modelo cliente-servidor, el programa que actúa como un servidor se encuentra en estado pasivo esperando que lleguen pedidos provenientes de los clientes. El primer paso consisten en marcar un *socket* indicándole su deseo de establecer contacto con clientes remotos. Para ello se recurre a la llamada.

- Intlisten (i*ntsock_descr, intqueue_length*): el argumento *sock_descr* identifica al *socket* sobre el cual se está efectuando la operación, es decir, por cuál *socket* se escuchará, y el argumento *queue_length* define el número máximo de conexiones pendientes que pueden permitirse; los pedidos posteriores serán descartados. La llamada *listen* retorna 0 ante el éxito de la operación y -1 ante una falla. *listen()* tiene sólo sentido bajo un servicio orientado a conexiones.

Después que se ha creado un *socket* y se le ha asociado una dirección local, el mismo se encuentra en estado desconectado, es decir, no tiene relación alguna con una dirección remota. Para lograr un servicio orientado a

conexiones, es decir, poder transferir información bajo un concepto de *streams* confiable, el programa cliente debe emitir una llamada con la siguiente sintaxis:

- Intconnect *(intsock_descr, structsockaddr *peer_addr, intaddrlen):* aquí, *sock_descr* es un *socket descriptor*, *peer_addr*es un puntero a una estructura de direcciones que contiene la dirección del *socket* de destino, con el que se deberá conectar, y addrlen es la longitud de la dirección en bytes. La llamada retorna 0 en caso de éxito y -1 en caso de falla. Después que un proceso servidor ha ejecutado las llamadas *socket()*, *bind()*, y *listen()* para crear un *socket*, asociarle una dirección local y definir la cola para almacenar los pedidos entrantes, puede aceptar cada solicitud ejecutando la llamada.

- Intaccept *(intsock_descr, structsockaddr *peer, intaddrlen):* el argumento *sock_addr* señala el *socket* sobre el que se estaban esperando los pedidos, *peer_addr*es un puntero a la estructura que guardará la dirección del cliente y *addrlen*es la longitud de la dirección del cliente. Los dos últimos parámetros son escritos por el sistema.

La llamada *accept()* se bloquea hasta que haya un pedido de conexión proveniente de un cliente en la cola de espera. Cuando llega un pedido, es decir, cuando un cliente emite una llamada *connect()*, se crea un nuevo *socket* que será el utilizado para intercambiar datos, y retorna el nuevo *socket descriptor* y la identificación del cliente que hizo la llamada. Habitualmente cada pedido aceptado se maneja en forma concurrente haciendo que, después de salir de la *accept()*, el servidor haga un *fork* generando un proceso que trabaje sobre el *socket* recientemente creado mientras que el proceso original cierra su copia y llama nuevamente a la función *accept()* para continuar “escuchando” en el *socket* original a la espera de otros pedidos.

Una vez que se han creado los *sockets* y se ha establecido una conexión lógica que los vincule, puede procederse a la transferencia de datos entre ellos y, por ende, entre los procesos de usuario que los utilizan.

Las conocidas llamadas *read()* y *write()* pueden utilizarse para el intercambio de datos con la diferencia que el primer parámetro será un *socket descriptor* en vez de un *file descriptor*. Sin embargo, la semántica es diferente cuando estas llamadas se aplican sobre *sockets*. Mientras que en el caso de trabajar con archivos el *write()* se limita a transferir datos al caché (la escritura en disco se hará más tarde), al trabajar con *sockets* el *write()* se bloquea hasta que los datos puedan transferirse al buffer del *socket*. Dos funciones similares, denominadas *send()* y *recv()*, emplean un cuarto argumento que permite algunas operaciones especiales. En resumen, las funciones más importantes para el intercambio de datos son:

*int read(intsock_descr, structmsghdr *msg, int length);*
*int write(intsock_descr, structmsghdr *msg, int length);*
*int send(intsock_descr, structmsghdr *msg, int length, int flags);*
*int recv(intsock_descr, structmsghdr *msg, int length, int flags);*

En todas, el argumento *msg* es un puntero a la estructura de datos a enviar o dónde colocar los datos recibidos y *length* indica el tamaño de tal estructura. El parámetro *flags* tiene una codificación especial y puede utilizarse para modificar la operación de los protocolos subyacentes. Las llamadas devuelven -1 ante una falla y el número de bytes escritos o leídos en caso de éxito. Debe tenerse en cuenta también que *read()* y *recv()* retornan tan pronto como tienen algún resultado y no necesariamente aguardan el arribo de todos los datos que se esperan.

No debe olvidarse que se trata de *streams*, por lo tanto, no necesariamente se conservan los límites de los mensajes. Desafortunadamente, una serie de llamadas *write()* en un extremo no necesariamente conducen a una serie equivalente de llamadas *read()* en el otro extremo.

Para terminar las operaciones sobre un *socket* en particular puede recurrirse a la llamada:

- Intshutdown *(intsock_descr, intmode):* el argumento *mode* determina cuán abrupto será el final de la conexión y toma los valores 0, 1 o 2. Con el valor 0, no se pueden escribir más datos; con 1, se envía cualquier dato pendiente y se suspende la transmisión; con 2, no se puede enviar ni recibir más información.

Por último, al igual que un archivo regular, un *socket* también puede cerrarse mediante la llamada:

- Intclose (*intsock_descr*): también aquí la semántica es diferente con respecto a los archivos regulares. Cuando se efectúa un *close()* sobre un *socket* antes de efectivizarse se envían todos los datos pendientes y se pierde cualquier mensaje que aún no haya arribado. Después de una llamada *close()* los recursos asociados al *socket* se devuelven al sistema. Puede observarse que el sencillo esquema descriptor hasta ahora sólo permite que el servidor atienda un pedido y finalice.

En un entorno más sofisticado, podría diseñarse el mismo de manera que entre en un ciclo compuesto por las funciones *accept()*, *read()* y *write()* generando nuevos procesos para cada pedido. Aunque hasta aquí se ha utilizado un esquema de conexiones, los *sockets* pueden también utilizarse para

trabajar con servicios son conexión. En este caso, en el momento de crear el *socket* se debe indicar el deseo de obtener un servicio de datagramas en el segundo parámetro de la llamada *socket()*. En segundo lugar, los *sockets* para un servicio de datagramas no necesitan estar conectados antes de su uso dado que permiten un modo de transmisión en el que cada mensaje contiene todos los datos para alcanzar su destino y tampoco necesitan de la función *accept()*.

Los mensajes pueden enviarse y recibirse sin establecer ningún vínculo. Más aún, los *sockets* de datagramas permiten el envío a múltiples destinos desde un mismo origen y la recepción en un solo *socket* de información proveniente de varios sistemas remotos. Para la transferencia de datagramas se cuenta con el par de funciones *sendto()* y *recvfrom()*, simulares a la *send()* y *recv()*, pero suman dos parámetros más que identifican la dirección de la entidad par y el tamaño de tal dirección. Su sintaxis es la siguiente:

- intsendto (intsock_*descr, structmsghdr *msg, int length, int flags, sockaddr *dest,intdestlength*);

- intrecvfrom (*intsock_descr, structmsghdr *msg, int length, int flags, sockaddr*from,intfromlenght*);

El parámetro *msg* apunta a los datos a enviar o al lugar donde se guardarán los recibidos y *length* especifica la longitud del *msg*; *dest*y *from* son punteros a estructuras de direcciones y *dest length*y *from length* indican el tamaño de las mismas.

Independientemente del tipo de servicio, además de las funciones mencionadas, se disponen de varias rutinas de biblioteca que brindan servicio a los programas de aplicación traduciendo valores numéricos en nombres,

direcciones de red y denominaciones de protocolos en formatos legibles por humanos.

Por ejemplo, *gethostbyaddr()* devuelve el nombre de un host y algunos datos del mismo dado su dirección de red. *gethostbyname()* efectúa la operación inversa.

getnetbyname() y *getnetbyaddr()* dan resultados análogos con respecto a la red. Las funciones *getprotobyname()* y *getprotbynumber()* actúan sobre los protocolos disponibles en un *host* dado. *getservbyname()* y *getservbyport()* otorgan información sobre los servicios de red disponibles.

Los *sockets*, en definitiva, son una de las posibles interfaces entre las aplicaciones y los servicios de transporte que, por razones históricas y técnicas, se ha convertido en la interfaz de programación más importante en un entorno de red; está disponible en una cantidad importante de plataformas, lo que sugiere que permanecerá en vigencia un buen tiempo. El paradigma utilizado ha demostrado su eficacia en multitud de situaciones y el desarrollo de aplicaciones puede hacerse altamente portable si se toman las medidas adecuadas desde el comienzo. En este último punto, las mayores dificultades radican en la representación de los datos que se intercambian, por lo que debería recurrirse previamente a servicios de presentación.

Debe tenerse en cuenta que los *sockets* son un concepto de bajo nivel por lo cual su uso no resulta sencillo aunque lo parezca a primera vista, exigiendo un conocimiento preciso de los detalles de la red y de la implementación. Por otra parte, tiene la ventaja de un mayor control; por ejemplo, elegir el paradigma que utilizará la aplicación y la biblioteca de representación de datos que se usa en un momento dado; en definitiva, una mayor flexibilidad.

2. PROTOCOLO SMPP

2.1. Conceptos

Este protocolo cuyas siglas corresponden en inglés Short Message Peer-to-Peer Protocol, es un protocolo estándar de telecomunicaciones pensado en realizar el intercambio de mensajes SMS entre equipos que gestionan los mensajes como pueden ser los SMSC (Short Messages Service Center), o los GSM USSD (Unstrucured Supplementary Services Data server*),* y un sistema de solicitud de SMS como puede ser un servidor WAP o cualquier servidor *gateway* de mensajería.

SMPP es un protocolo de código abierto conocido mundialmente como *open source*, lo que se podría llamar una especie de estándar en la industria, desarrollado con la intención de proveer la flexibilidad de una interface en la comunicación de datos para el intercambio de mensajes cortos entre ESME (entidades externas de mensajes cortos), RE (entidades de ruteo) y MC (centros de mensajes).

Estas entidades se utilizan para el intercambio de mensajes cortos, así:

Centro de mensajes (MC o SMSC): término utilizado para la descripción de sistemas como un SMSC *(*Short Message Service Center*),* GSM USSD (Unstrucured Supplementary Services Data server) o el Cell Broadcast Center (CBC).

Entidades externas de mensajes cortos (ESME): representa comúnmente a una aplicación cliente configurado en la red, tal como lo sería un WAP Proxy Server, una puerta de enlace para el correo, o bien un servidor de *voice mail*. Un ESME también puede representar una Entidad de Broadcast por Celda o conocido en el medio como CBE (Cell Broadcast Entity).

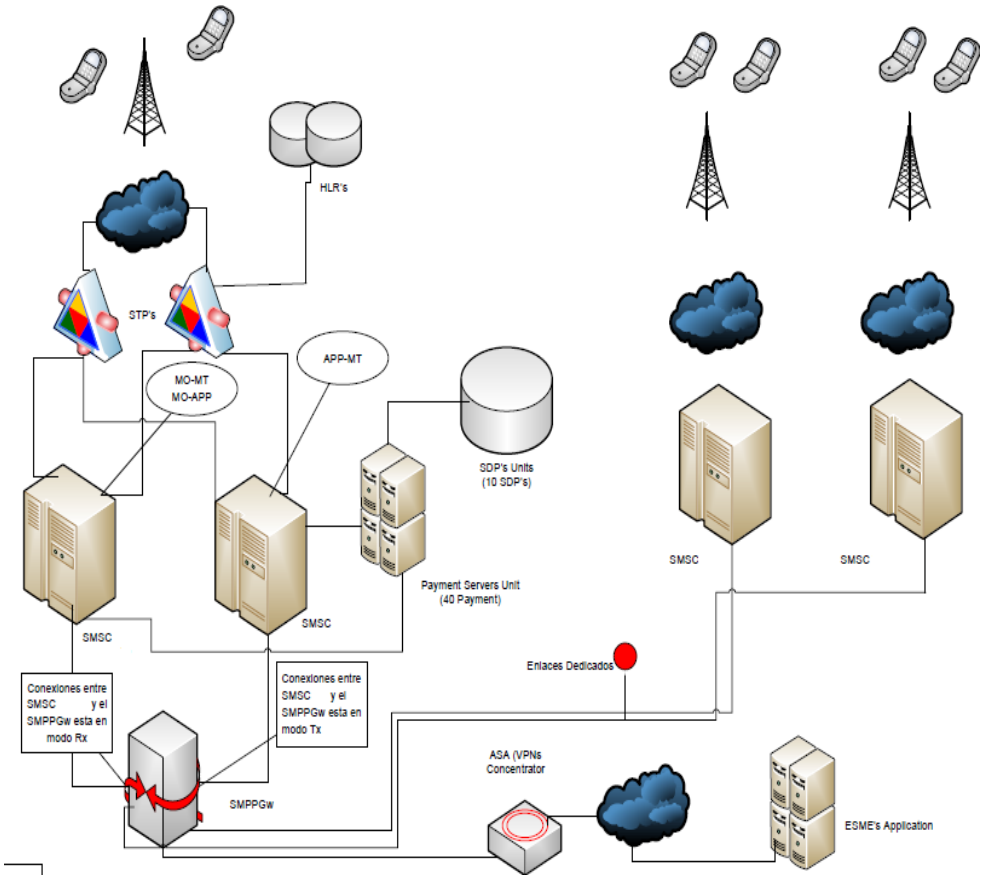
Entidades de ruteo (RE): término general utilizado para cualquier elemento de red que es utilizado por un centro de mensajería para enrutar tráfico SMS a otro centro de mensajería. También es utilizado para enrutar tráfico SMS de una ESME hacia un centro de mensajería.

Una entidad de ruteo tiene la capacidad de emular funcionalidad asociada con un centro de mensajería y una entidad externa de mensajes cortos (ESME). Para una ESME, una entidad de ruteo aparecerá como un centro de mensajería y para un centro de mensajería una entidad de ruteo aparentará ser una ESME.

Un proveedor de telefonía móvil puede utilizar entidades de ruteo para ocultar un centro de mensajería en la red, presentando solo las entidades de ruteo como interfaces de punto externas para entidades externas de mensajes cortos.

El siguiente diagrama ilustra los conceptos descritos arriba de SMPP en una red de móviles:

Figura 7. **Red de Telefonía móvil GSM con un SMSC y SMPP**



Fuente: elaboración propia, con Microsoft Office Visio 2007.

2.2. Sesiones SMPP

Dentro del estándar de uso de SMPP, una sesión SMPP deberá ser establecida entre el ESME y el centro de mensajería (MC) o la entidad de ruteo SMPP (RE) donde la necesidad técnica o negocio lo justifique.

La sesión establecida será basada sobre una capa de aplicación TCP/IP o una conexión X.25 entre el ESME y la entidad de ruteo o el centro de mensajería, inicialmente requerida o convocada por el ESME.

Las tres formas de inicializar una sesión con un ESME:

- Transmisor (TX): cuando se está autentificado como un TX, un ESME puede disparar mensajes cortos hacia el centro de mensajería para que este haga entrega a las estaciones móviles. Una sesión TX también permitirá a un ESME cancelar, consultar o reemplazar previamente los mensajes enviados. Los mensajes enviados en TX a menudo son llamados mensajes de terminación en móviles.

- Receptor (RX): una sesión de RX permite a un ESME recibir mensajes que vienen de un centro de mensajería. Estos mensajes son típicamente originados de estaciones móviles y son conocidos como mensajes que originan los teléfonos móviles en la red.

- Transceiver (TRX): una sesión TRX es la combinación de TX y RX, esto hace posible que una sesión sencilla de SMPP sea usada para disparar mensajes hacia un móvil y recibir los mensajes originados por el móvil.

Adicionalmente el centro de mensajería puede establecer una sesión SMPP para conectar al ESME. A esto se le conoce como una sesión de enlace hacia el exterior.

2.2.1. Descripción de la sesión SMPP

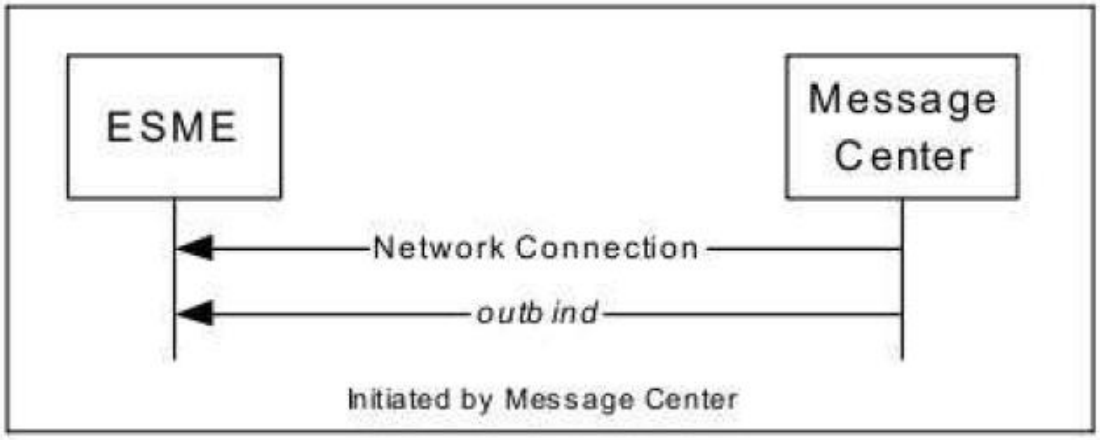
Una sesión SMPP entre un SMSC y una ESME es iniciada por el ESME, estableciendo una conexión de red con el SMSC que le enviará al ESME una respuesta a su solicitud de enlace (bind request) para que la sesión SMPP sea abierta.

Una ESME que desee enviar y recibir mensajes requiere establecer dos conexiones de red (TCP/IP o X.25) y dos sesiones SMPP, una de transmisión (TX) y otra de recepción (RX). Alternativamente, sobre esta versión del protocolo SMPP un ESME tiene la posibilidad de establecer una sesión SMPP *transceiver* sobre una única conexión de red. Durante una sesión SMPP, una ESME posiblemente realice una serie de solicitudes hacia el SMSC, mismas que este preponderá correspondientemente.

La sesión SMPP puede ser definida en términos de los siguientes posibles estados:

- CLOSED (fuera de límite y desconectado): estado proveniente del SMSC que indica una conexión de red cerrada. El SMSC puede también terminar la conexión que proviene del ESME.

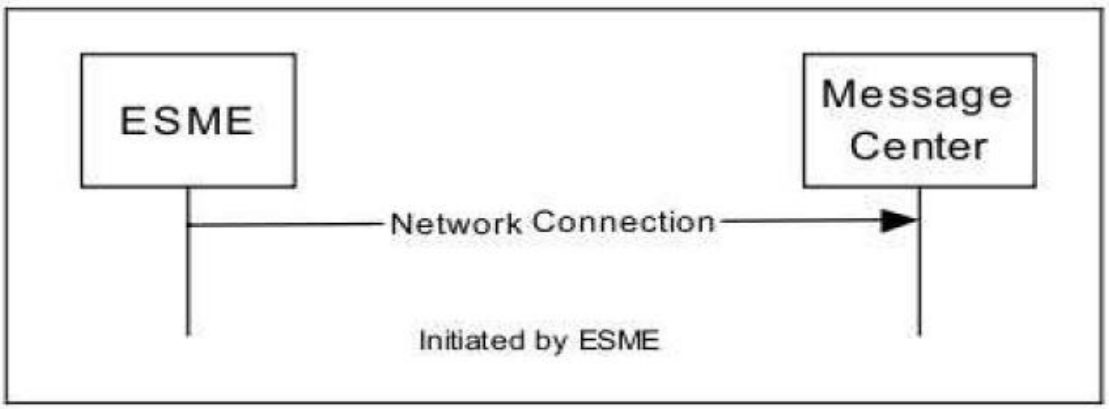
Figura 8. **Diagrama de flujo de sesión Closed SMPP**



Fuente: elaboración propia.

- OPEN (conectado y enlace pendiente): una ESME tiene establecido una conexión de red hacia el SMSC pero este aun no responde.

Figura 9. **Diagrama de flujo de sesión Open SMPP**

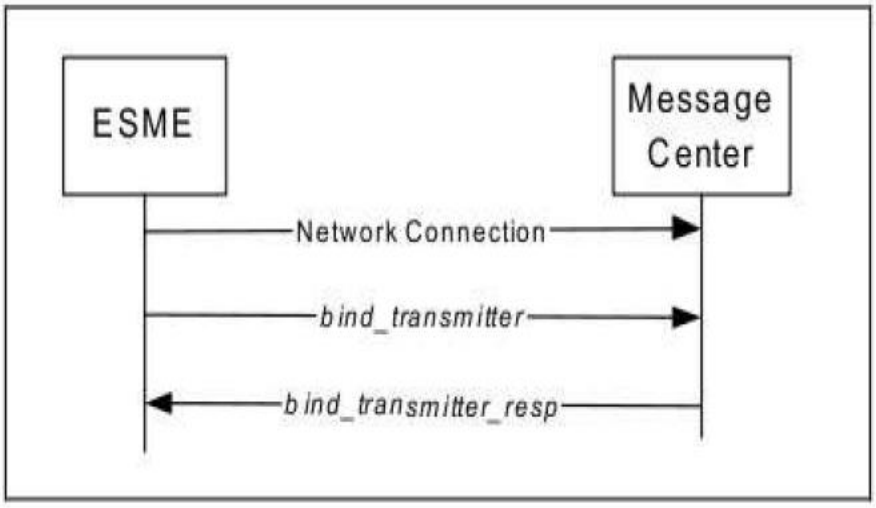


Fuente: elaboración propia.

- BOUND_TX: una ESME conectado tiene que solicitar al enlace como una ESME *transmitter* (razón de utilizar un PDU bind_transmitter) y tendrá que haber recibido una respuesta del SMSC autorizando la solicitud de enlace. Una ESME limitado como un transmisor podrá enviar mensajes cortos hacia un SMSC para que este los envíe a una estación móvil o a otro ESME.

El ESME puede también remplazar, consultar o cancelar previo mensaje corto enviado en el establecimiento de esta conexión.

Figura 10. **Diagrama de flujo de sesión ESME Transmitter SMPP**

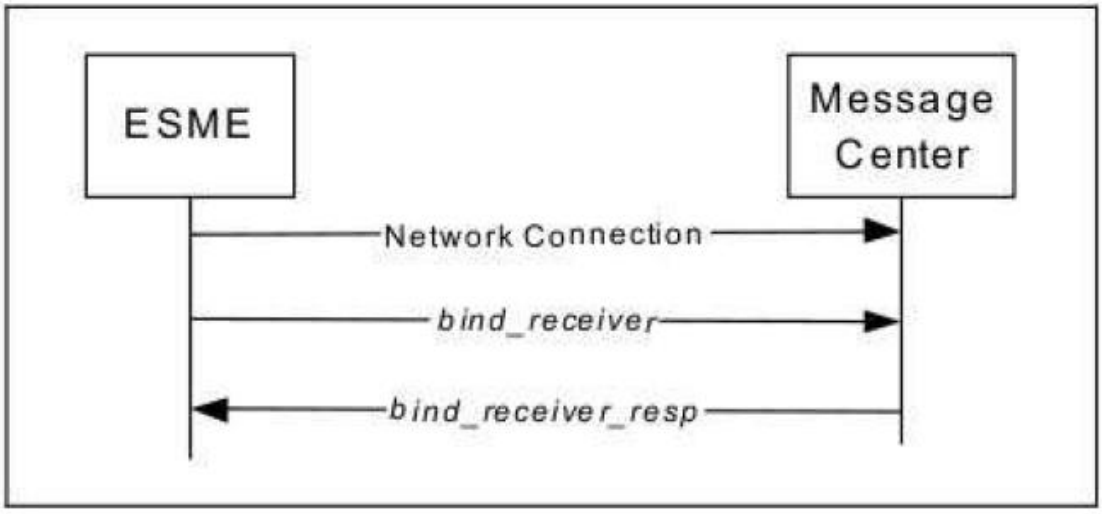


Fuente: elaboración propia,

- BOUND_RX: un ESME conectado debe haber solicitado conexión como un ESME Receiver (por utilizar un PDU bind_receiver) y debió haber recibido una respuesta proveniente del SMSC autorizando esta solicitud

de enlace (bind request). Una ESME limitado como receptor podrá recibir mensajes cortos provenientes de un SMSC los cuales pueden ser originados por una estación móvil, por otro ESME o por el centro de mensajería mismo.

Figura 11. **Diagrama de flujo de sesión ESME Receiver SMPP**



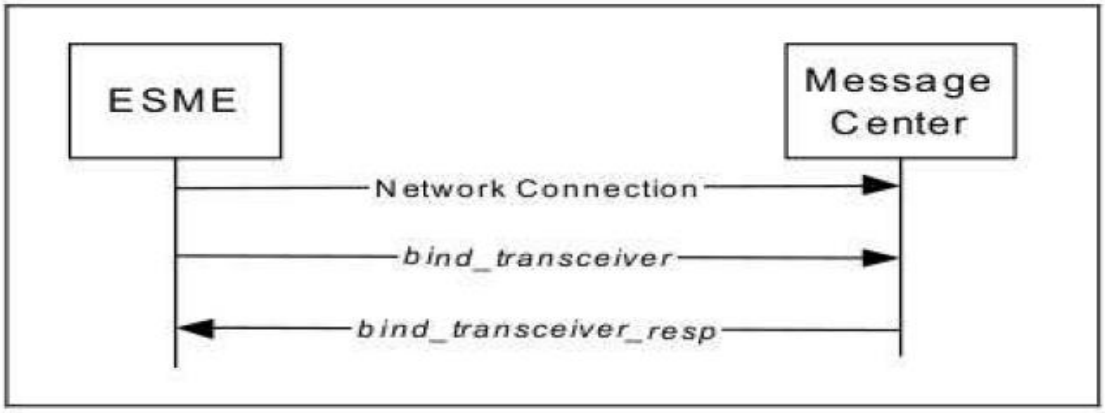
Fuente: elaboración propia.

- BOUND_TRX: una ESME conectado debe haber solicitado conexión como un ESME *transceiver* (por utilizar un PDU bind_transceiver) y tendrá que haber recibido una respuesta proveniente del SMSC autorizando esta solicitud de enlace (bind_request). Un ESME limitado como *transceiver* soporta el completo conjunto de operaciones soportadas por transmisor ESME y un receptor ESME.

Una ESME limitado como *transceiver* podrá enviar mensajes cortos a un SMSC para que este los envíe a una estación móvil o a otro ESME. El ESME

podrá también recibir mensajes cortos que provengan de un SMSC los cuales pudieron ser originados por una estación móvil, por otro ESME o por el SMSC mismo.

Figura 12. **Diagrama de flujo de sesión ESME Transceiver SMPP**



Fuente: elaboración propia.

2.3. Operación de Protocolos SMPP y la PDU

El protocolo SMPP es básicamente un conglomerado de operaciones, cada una tomando la forma de una solicitud y una respuesta PDU (Protocol Data Unit).

Por ejemplo, si una ESME desea enviar un mensaje corto, la ESME puede mandar un submit_sm PDU hacia el centro de mensajería. El centro de mensajería responderá con un submit_sm_resp PDU, indicando el éxito o la falla de la solicitud. De cierta manera, si un centro de mensajería desea enviar un mensaje hacia una ESME, el centro de mensajería puede mandar un

deliver_sm PDU hacia una ESME, el ESME al recibirlo responderá un deliver_sm_resp PDU como una forma de reconocer el envío del mensaje de texto.

Un ESME puede enviar un submit_sm hacia un centro de mensajería, solo si se tiene establecido una sesión TX o TRX con su centro de mensajería. Partiendo de lo anterior, un centro de mensajería puede enviar deliver_sm PDU solo a ESME que tengan establecidas sus sesiones a RX o TRX.

Las operaciones son comúnmente categorizadas dentro de los siguientes grupos:

- Administradoras de sesiones: estas son operaciones asignadas para permitir establecer sesiones SMPP entre un ESME y un centro de mensajería y proporcionar lo necesario para el manejo de errores inesperados.

- De mensajes introducidos: estas son operaciones asignadas específicamente para el envío de mensajes provenientes de los ESME hacia el centro de mensajería.

- De mensajes enviados: esta define operaciones que permiten al centro de mensajería enviar mensajes a una ESME.

- Mensajes de broadcast: operaciones que proveen servicio de celdas de broadcast dentro de un centro de mensajería.

- Operaciones varias: por último define las operaciones que proveen características reforzadas tales como la cancelación, consulta o reemplazo de mensajes.

Con el fin de ampliar a un mayor nivel de detalle se procederá a detallar cada operación documentada.

2.3.1. Administrador de sesiones

Con el fin de describir cada operación de administración de sesiones se detalla a continuación el nombre del PDU SMPP y su descripción:

- bind_transmitter: nombre que se le brinda al PDU SMPP de autenticación usado por un ESME transmisor (TX ESME) para enlazar y/o establecer comunicación con el centro de mensajería. El PDU contiene información de identificación y un *password* de acceso para el ESME. El estado de sesión requerido es abierto y este es ofrecido por la ESME y no por el centro de mensajería (SMSC).

- bind_transmitter_resp: nombre que se le brinda a la respuesta que brinda el centro de mensajería al PDU anteriormente descrito. Este PDU indica el éxito o falla del ESME tentativo a enlazar y/o establecer comunicación como un transmisor. El estado de sesión requerido es abierto y este no es ofrecido por la ESME y si por el centro de mensajería (SMSC).

- bind_receiver: nombre que se le brinda al PDU de autenticación usado por una conexión RX para enlazar y/o establecer comunicación con el centro de mensajería. El PDU contiene información de identificación, un

password de acceso para el ESME y puede también contener información de enrutado, especificando el rango de direcciones a las cuales se brinda el servicio por el ESME. El estado de sesión requerido es abierto y este es ofrecido por la ESME y no por el centro de mensajería (SMSC).

- bind_receiver_resp: nombre que se le brinda a la respuesta que brinda el Centro de Mensajería al PDU bind_receiver. Este PDU indica el éxito o falla del ESME tentativo a enlazar y/o establecer comunicación como un receptor. El estado de sesión requerido es abierto y este no es ofrecido por la ESME y sí por el centro de mensajería (SMSC).

- bind_transceiver: nombre que se le brinda al PDU de autenticación usado por un ESME *Transrecive*r (TRX *Transceiver*) a enlazar y/o establecer comunicación con el centro de mensajería. El PDU contiene información de identificación, un *password* de acceso para el ESME y puede también contener información de enrutado, específicamente el rango de direcciones a las cuales se brinda el servicio por el ESME.

El estado de sesión requerido es abierto y este si es ofrecido por la ESME y no por el centro de mensajería (SMSC).

- bind_transceiver_resp: nombre que se le brinda a la respuesta que brinda el centro de mensajería al PDU bind_transceiver. Este PDU indica el éxito o fracaso del enlace del ESME tentativo como un transceiver. El estado de sesión requerido es abierto y este no es ofrecido por la ESME y sí por el centro de mensajería (SMSC).

- Outbind: PDU de autenticación usado por un centro de mensajería para enlazar y/o establecer comunicación desde fuera a un ESME que informará a este que mensajes están presentes en el centro de mensajería. El PDU contiene identificación, un *password* de acceso para el ESME. Si el ESME autentifica la solicitud, este responderá con un bind_receiver o bind_transceiver para iniciar el proceso de enlazamiento dentro del centro de mensajería. El estado de sesión requerido es abierto y este no es ofrecido por la ESME y sí por el centro de mensajería (SMSC).

- Unbind: este PDU puede ser mandado por el ESME o un centro de mensajería como un mecanismo de terminación de la sesión SMPP. El estado de sesión requerido puede ser BOUND_TX, BOUND_RX y BOUND_TRX y este sí es ofrecido por la ESME y también por el centro de mensajería (SMSC).

- Unbind_resp: este PDU puede ser enviado por el ESME o el centro de mensajería como un mecanismo de reconocimiento a la recepción de una solicitud unbind, después, de enviar este PDU el centro de mensajería por lo general cerrara la conexión de red. El estado de sesión requerido puede ser BOUND_TX, BOUND_RX y BOUND_TRX y este sí es ofrecido por la ESME y también por el centro de mensajería (SMSC).

- Enquire_link: este PDU puede ser enviado por el ESME o el centro de mensajería, solo para probar la conexión de red. El estado de sesión requerido puede ser BOUND_TX, BOUND_RX y BOUND_TRX y este sí es ofrecido por la ESME y también por el centro de mensajería (SMSC).

- Enquire_link_resp: este PDU se utiliza para el reconocimiento de una solicitud enquire_link que fue enviada por un ESME o centro de mensajería. El estado de sesión requerido puede ser BOUND_TX, BOUND_RX y BOUND_TRX y este sí es ofrecido por la ESME y también por el centro de mensajería (SMSC).

- Alert_notification: un centro de mensajería envía alert_notification a una ESME como un mecanismo de alerta para verificar la disponibilidad de la misma. El estado de sesión requerido puede ser BOUND_RX y BOUND_TRX y este no es ofrecido por la ESME y sí por el centro de mensajería (SMSC).

- Generic_nack: este PDU puede ser mandado un ESME o un Centro de Mensajería como una manera de indicar la recepción de un PDU invalido. Con base en un criterio de su medida o contenido. El estado de sesión requerido puede ser BOUND_TX, BOUND_RX y BOUND_TRX y este sí es ofrecido por la ESME y también por el centro de mensajería (SMSC).

2.3.2. Operación de mensajes introducidos (Submit) en el protocolo SMPP

Con el fin de describir cada operación de mensajes introducidos se detalla a continuación el nombre del PDU SMPP y su descripción:

- Submit_sm: un ESME transmisor o transceiver, que desea ingresar un mensaje corto, puede usar este PDU para especificar el remitente, destinatario y texto del mensaje corto. Otros atributos incluirán la prioridad del mensaje, esquema de codificación, periodo de validez. El

estado de sesión requerido puede ser BOUND_TX, BOUND_TRX y este sí es ofrecido por la ESME y no por el centro de mensajería (SMSC).

- Submit_sm_resp: el centro de mensajería responderá al PDU submit_sm, indicando éxito o falla de la solicitud. Esta también incluye un message_id del centro de mensajería que puede ser usada en operaciones subsecuentes a consultas, cancelaciones o reemplazo de contenidos de un mensaje enviado. El estado de sesión requerido puede ser BOUND_TX, BOUND_TRX y este no es ofrecido por la ESME y sí por el centro de mensajería (SMSC).

- Submit_sm_multi: este es una variación del PDU submit_sm que soporta arriba de 255 recipientes para la entrega de mensaje. El estado de sesión requerido puede ser BOUND_TX, BOUND_TRX y este sí es ofrecido por la ESME y no por el centro de mensajería (SMSC).

- Submit_sm_multi_resp: el centro de mensajería responde al PDU submit_sm_multi. Esto es algo similar al PDU submit_sm_resp. La principal diferencia se encuentra en donde si algunos de los recipientes especificados fueron por inválidos o rechazados por el centro de mensajería, el PDU podrá especificar la lista de recipientes fallidos, agregando el código de error específico para cada uno, indicando la razón de porque el recipiente fue inválido. También es incluido un message_id por el centro de mensajería, el cual será usado en operaciones subsecuentes de consulta, cancelación o reemplazar el contenido de un mensaje no enviado. El estado de sesión requerido puede ser BOUND_TX, BOUND_TRX y este no es ofrecido por la ESME y sí por el centro de mensajería (SMSC).

- Data_sm: este es una versión flujo (*stream*) de la operación submit_sm, designada para basarse en el empaquetamiento de aplicaciones que no demanden funcionalidad extendida normalmente disponible en la operación submit_sm. Generalmente esta operación la utilizan ESMEs que implementan WAP en un SMS. El estado de sesión requerido puede ser BOUND_TX, BOUND_RX y BOUND_TRX y este sí es ofrecido por la ESME y también por el centro de mensajería (SMSC).

- Data_sm_resp: el centro de mensajería responde al PDU data_sm, indicando el éxito o fallo de la solicitud. También es incluido un message_id del centro de mensajería, el cual será usado en operaciones subsecuentes de consulta, cancelación o reemplazar el contenido de un mensaje no enviado. El estado de sesión requerido puede ser BOUND_TX, BOUND_RX y BOUND_TRX y este sí es ofrecido por la ESME y también a su vez por el centro de mensajería (SMSC).

2.3.3. Operación de mensajes enviados (delivery) en el protocolo SMPP

- Deliver_sm: esta operación es el opuesto simétrico al submit_sm y es usado por un centro de mensajería para enviar y recibir o ejecutar el rol de un ESME que está configurado como transmisor y receptor (*Transceiver*). El estado de sesión requerido puede ser BOUND_RX, BOUND_TRX y este no es ofrecido por la ESME y sí por el centro de mensajería (SMSC).

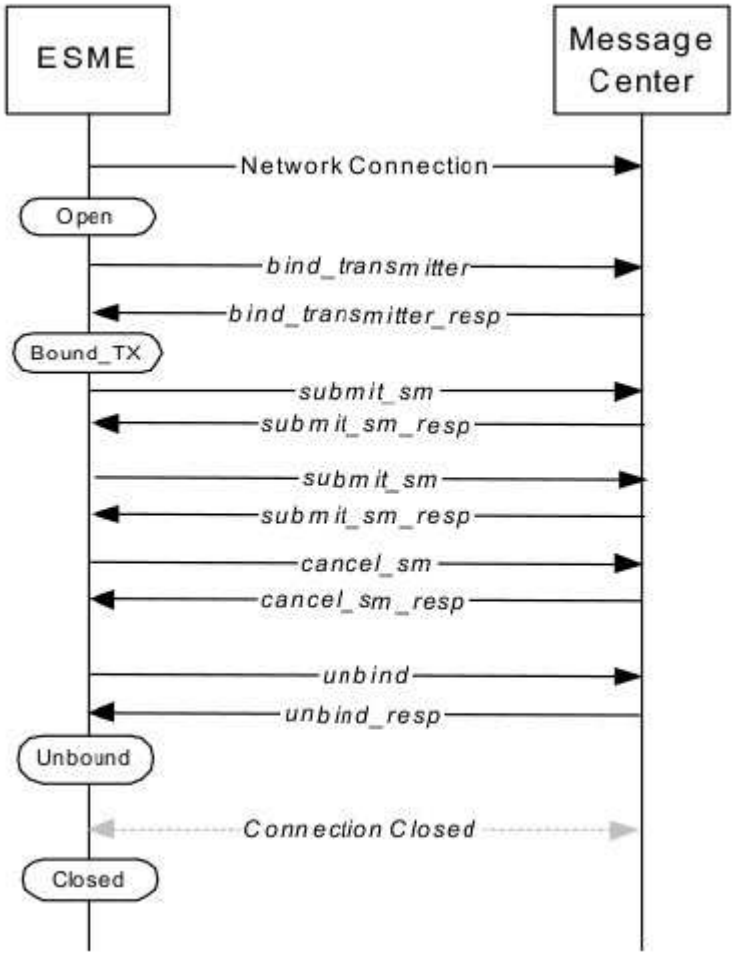
- Deliver_sm_resp: este PDU indica aceptación o rechazo de un mensaje enviado. El error regresado por el ESME puede causar que el mensaje sea revisado en algún momento posterior o rechazado. El estado de sesión requerido puede ser BOUND_RX, BOUND_TRX y este sí es ofrecido por la ESME y también por el centro de mensajería (SMSC).

- Data_sm: este es utilizado para enviar un mensaje desde el centro de mensajería hacia el ESME. Generalmente esta operación la utilizan ESMEs que implementan WAP en un SMS. El estado de sesión requerido puede ser BOUND_TX, BOUND_RX, BOUND_TRX y este sí es ofrecido por la ESME y también por el centro de mensajería (SMSC).

- Data_sm_resp: el ESME responde a un PDU de data_sm, indicando el éxito o el fracaso del envío iniciado por el centro de mensajería. El estado de sesión requerido puede ser BOUND_TX, BOUND_RX, BOUND_TRX y este sí es ofrecido por la ESME y también por el centro de mensajería (SMSC).

Con el fin de ejemplificar lo descrito y con fines explicativos se detalla a continuación las operaciones SMPP y sus estados relativos, los siguientes ejemplos ilustran diálogos típicos para 3 tipos de ESME:

Figura 13. **Diagrama de flujo de las operaciones SMPP**



Fuente: elaboración propia.

2.4. Versiones del protocolo SMPP

A continuación se explicarán las distintas versiones del protocolo.

2.4.1. Protocolo SMPP 3.3

Esta especificación fue la versión original definida, siendo su creador Aldiscon Ltd., actualmente llamada Lógica, creada de 1991- 1997 donde se realizaron la versión 1 hasta la actual 3.3.

Cuando se establece una comunicación en SMPP 3.3, se requieren dos conexiones virtuales. Una es usada para mensajes originados en el sistema ESME y el otro para responder los mensajes.

SMPP 3.3 agrupa sus transacciones en:

- Mensajes desde ESME a SMSC (*Transmitter*)
- Mensajes desde SMSC a ESME (*Receiver*)

2.4.2. Protocolo SMPP 3.4

La versión SMPP 3.4 soporta la tecnología de las siguientes tecnologías de red celular.

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

SMPP 3.4 agrupa sus transacciones en:

- Mensajes desde ESME a SMSC (*Transmitte*r)
- Mensajes desde SMSC a ESME (*Receiver*)
- Mensajes desde ESME a SMSC (*Transceiver*)
- Mensajes desde SMSC a ESME (*Transceiver*)

La diferencia entre ambos protocolos radica principalmente en la sesión de tipo *Transceiver* y la tecnología celular que es soportada por el protocolo 3.4.

2.4.3. Protocolo SMPP 5.0

Esta versión es una mejora de SMPP versión 3.4. El cambio de versión de 3.4 a 5.0 intentó evitar la confusión con 4.0 que ideó Lógica según su especificación.

SMPP 5.0 incorpora muchos cambios definidos por foros de SMS durante el tiempo de vida de la versión 3.4. Con este nuevo enfoque se proveen más detalles y descripciones de la funcionalidad del protocolo.

Para nombrar las nuevas funcionalidades están se describen a continuación:

- Operaciones adicionales de cell broadcast para uso de plataformas como el cell broadcast center.

- Nuevos registros para el modo de envío exitoso de mensajes recipiente en el centro de mensajería.

- Nuevos tipos de errores adicionales para la clasificación de código de errores en la red.

- Nuevos estados de congestión agregados para soportar nuevos flujos de control y evitar congestión en conexión de puertos.

- Nuevos códigos de errores para indicar una operación prohibida a una aplicación de contenido o ESME.

- Agregado de identificadores de cobro entre una ESME y el centro de mensajería.

- Soporte de portabilidad de números para ruteo con Inter-Carriers.

2.5. Command_Status en el Protocolo SMPP

Se presenta un cuadro con la explicación de los distintos códigos que existen en el estándar SMPP.

2.5.1. Explicación de los códigos de estatus del SMPP

La plataforma SMGateway interactúa esencialmente, con ESME y SMSC.

Esta interacción es hecha mediante el protocolo SMPP, el cual está formado por requerimientos y respuestas. En este capítulo se detallan los Command_Status que la plataforma puede generar a partir de un requerimiento junto con los motivos por los cuales pudo ser generado tal Command_Status.

Tabla II. **Tabla explicativa sobre los Command_Status en el protocolo SMPP**

cmd_status	Descripción
0x00000000 ESME_ROK:	Este *command status* indica que el requerimiento fue cumplido exitosamente. Desde el punto de vista de la plataforma, esto indica que el SM fue entregado exitosamente al sistema de destino.
0x00000001 ESME_RINVMSGLEN:	El largo del SM entregado por el sistema de origen es inválido.
0x00000002 ESME_RINVCMDLEN:	El largo del PDU SMPP es inválido. Ya sea el SMPP Client o SMPP Server detectó un PDU cuya especificación de largo es inválida.
0x00000003 ESME_RINVCMDID:	El requerimiento de comando SMPP corresponde a un comando no soportado por la plataforma.
0x00000004 ESME_RINVBNDSTS:	Un ESME que hizo *bind* en modo TRANSMITTER(TX) intentó enviar un SM con una operación SUBMIT_SM, lo cual es inválido en el estándar SMPP.
0x00000005 ESME_RALYBND:	Un ESME que ya hizo una operación *bind* exitosa intentó realizar otro *bind.*
0x00000006 ESME_RINVPRTFLG:	Un ESME o SMSC envió un requerimiento especificando el campo priority_flag con un valor no permitido por el estándar SMPP.
0x0000000A ESME_RINVSRCADR:	Un ESME o SMSC envió un requerimiento de entrega de SM especificando una dirección de origen inválida, ya sea porque superó el largo máximo establecido para las direcciones en el estándar SMPP, o porque no estaba terminada por el carácter nulo (‘\0’).
0x0000000B ESME_RINVDSTADR:	Un ESME o SMSC envió un requerimiento de entrega de SM especificando una dirección de destino inválida, ya sea porque superó el largo máximo establecido para las direcciones en el estándar SMPP, o porque no estaba terminada por el carácter nulo (‘\0’). El genera este error cuando un prefijo de numeración de destino no está configurado en los árboles de análisis de numeración.
0x0000000D ESME_RBINDFAIL:	Un ESME intentó realizar una operación de *BIND*, la cual fracasó, esto pudo ocurrir debido a que el ESME especificó datos de conexión (system_id, *password* o system_type) distintos a los configurados, ya sea porque se está intentando conectar desde una IP distinta a la configurada o porque superó el número máximo de conexiones que tenía permitido a la plataforma.

Continuación de la tabla II.

0x0000000E ESME_RINVPASWD:	El *password* especificado por el ESME en la operación BIND tiene un largo inválido, o no está terminado por el carácter nulo.
0x0000000F ESME_RINVSYSID:	El system_id especificado por el ESME en la operación BIND tiene un largo inválido, o no está terminado por el carácter nulo.
0x00000014 ESME_RMSGQFUL:	Se ha superado el límite global de TPS licenciado.
0x00000015 ESME_RINVSERTYP:	El campo service_type de un requerimiento tiene un largo inválido, o no está terminado por el carácter nulo.
0x00000043 ESME_RINVESMCLASS:	El campo esm_class de un requerimiento tiene un valor inválido.
0x00000048 ESME_RINVSRCTON:	El campo *type of number* (TON) de la dirección de origen de un requerimiento de entrega de SM tiene un valor inválido.
0x00000049 ESME_RINVSCRNPI:	El campo *numeric plan indicador* (NPI) de la dirección de origen de un requerimiento de entrega de SM tiene un valor inválido.
0x00000050 ESME_RINVDSTTON:	El campo *type of number* de la dirección de destino de un requerimiento de entrega de SM tiene un valor inválido.
0x00000051 ESME_RINVDSTNPI:	El campo *numeric plan indicator* de la dirección de destino de un requerimiento de entrega de SM tiene un valor inválido.
0x00000053 ESME_RINVSYSTYPE:	El campo system_type de un requerimiento *BIND* hecho por un ESME tiene un valor inválido, ya sea porque superó el largo permitido o porque no estaba terminado con un carácter nulo.
0x00000058 ESME_RTHROTTLED:	El ESME o SMSC que originó el requerimiento superó el límite de transacciones por segundo configurado para esa conexión.
0x00000061 ESME_RINVSCHED:	El ESME o SMSC que originó el requerimiento especificó un valor inválido para el campo schedule_delivery_time.
0x00000062 ESME_RINVEXPIRY:	El ESME o SMSC que originó el requerimiento especificó un valor inválido para el campo validity_period.

Continuación de la tabla II.

0x00000064 ESME_RX_T_APPN:	Error temporal. Este error puede ser generado debido a los siguientes motivos: la conexión de destino del SM no estaba conectada de manera que pudiera recibir el SM; ocurrió un problema puntual con la entrega del SM a la conexión de destino. Este error indica al sistema de origen que en ese momento el SM no pudo ser entregado, pero que debería reintentar posteriormente la operación.

Fuente: elaboración propia.

3. PARÁMETROS Y FORMATO PDU DEL PROTOCOLO SMPP

El protocolo SMPP está basado en un conjunto de varias operaciones, donde cada una toma la forma de una solicitud de respuesta llamada PDU (Protocol Data Unit) que en sus siglas en español significa unidad de datos del protocolo.

3.1. Concepto del PDU

Básicamente el concepto de PDU es donde cada operación toma la forma de una solicitud y respuesta del protocolo. Una forma sencilla de comprender el mismo es si una ESME desea realizar el envío de un mensaje *(submit)*, este de enviar un submit_sm PDU a un centro de mensajería. El centro de mensajería o SMSC responderá con un submit_sm_resp PDU, indicando si la solicitud de envío del mensaje fue de manera exitosa o fallida. Lo mismo sucede en el caso de si un centro de mensajería o SMSC deseara transmitir un mensaje a una aplicación ESME, esta puede enviar el mismo haciendo un deliver_sm PDU a una aplicación ESME, la cual como respuesta a la solicitud responderá con un deliver_sm_resp PDU para confirmar al centro de mensajería de la recepción de la solicitud enviada.

Algunas operaciones son específicas para aplicaciones ESME y otras son específicas para centros de mensajería.

El tipo de sesión se puede utilizar para el envío o la recepción de un mensaje de texto, una ESME puede renviar un submit_sm PDU a un centro de mensajería solamente si tiene establecida una sesión del tipo TX o TRX con su

centro de mensajería. Partiendo del mismo concepto lo mismo aplicaría para un centro de mensajería ya que para enviar un mensaje o realizar un deliver_sm PDU a una ESME debe tener establecida una sesión del tipo RX o TRX.

3.2. Secuencias en el PDU

El concepto importante a comprender es que el protocolo SMPP es una especie de protocolo de intercambio o en sus siglas en inglés (*handshake*) donde cada solicitud es conocida antes de la siguiente solicitud, debido a que los efectos de los conceptos anteriormente comprendidos brindan los detalles iniciales del funcionamiento se profundizará más en el funcionamiento del protocolo SMPP.

Número de secuencia PDU: cada PDU de solicitud SMPP posee un identificador llamado número de secuencia que será usado únicamente para la identificación del PDU dentro del contenido del mensaje, este número de secuencia lo estará generando la entidad y la actual sesión SMPP. La respuesta PDU resultante la cual debe regresar sobre la misma sesión SMPP es esperada para reflejar el número de secuencia de la solicitud original.

3.2.1. Síncrono vrs. asíncrono

El protocolo SMPP es un protocolo por definición asíncrono. Esto permite que una aplicación de contenido ESME o un centro de mensajería pueda enviar solicitudes al mismo tiempo que las dos entidades deseen. Aquí es cuando el número de secuencia PDU tiene un papel crítico, para poder soportar la naturaleza asíncrona de SMPP. Para comprender la naturaleza del funcionamiento básico del protocolo, anteriormente se presentó el protocolo SMPP como síncrono. SMPP tiene una naturaleza asíncrona, ya que así, este

puede mandar varios PDU, sin tener que esperar una respuesta de la primera solicitud que fue enviada.

3.2.2. Relojes SMPP (*Timers*)

Con la finalidad de realizar un intercambio eficiente de transacciones SMPP, se recomienda que cada sesión SMPP esté administrada usando relojes configurables en ambas entidades, que intervienen en la comunicación de la aplicación de contenido y el centro de mensajería.

Para poder nombrar que tipo de relojes se encuentran relacionados tanto en un centro de mensajería como en una aplicación de contenido ESME, se detallan a continuación y se explica su significado:

- Reloj de iniciación para una conexión SMPP: para asegurarse que cuando una aplicación de contenido ESME inicia una sesión SMPP, este debe ocurrir dentro de un especifico periodo después de abrir una conexión de red hacia el centro de mensajería.

- Reloj de sesión para una conexión SMPP: habilita cualquier solicitud de una aplicación de contenidos ESME o del centro de mensajería, el estatus de la sesión SMPP de la otra entidad SMPP se está comunicando vía el comando enquire_link para mantener la sesión establecida y operando.

- Reloj de inactividad para una conexión SMPP: esta especifica el máximo periodo de tiempo para que la sesión SMPP no se pierda, si los mensajes SMPP no son intercambiados, la sesión SMPP podría ser concluida.

- Reloj de transacción para una conexión SMPP: esta especifica el lapso de tiempo permitido entre una solicitud SMPP y la correspondiente respuesta SMPP a dicha solicitud.

Modos de mensaje

El estándar SMPP ofrece la opción de modos de mensaje, la cual si es soportada sobre el centro de mensajería, este permite a la aplicación de contenido seleccionar el mecanismo de envío. Típicamente los mecanismos de envío ofrecidos por un centro de mensajería son:

- Modo de almacenamiento y reenvío, este es utilizado por la aplicación de contenido
- Modo datagrama
- Modo de transacción

Modo de almacenamiento y reenvío, este es utilizado por la aplicación de contenido:

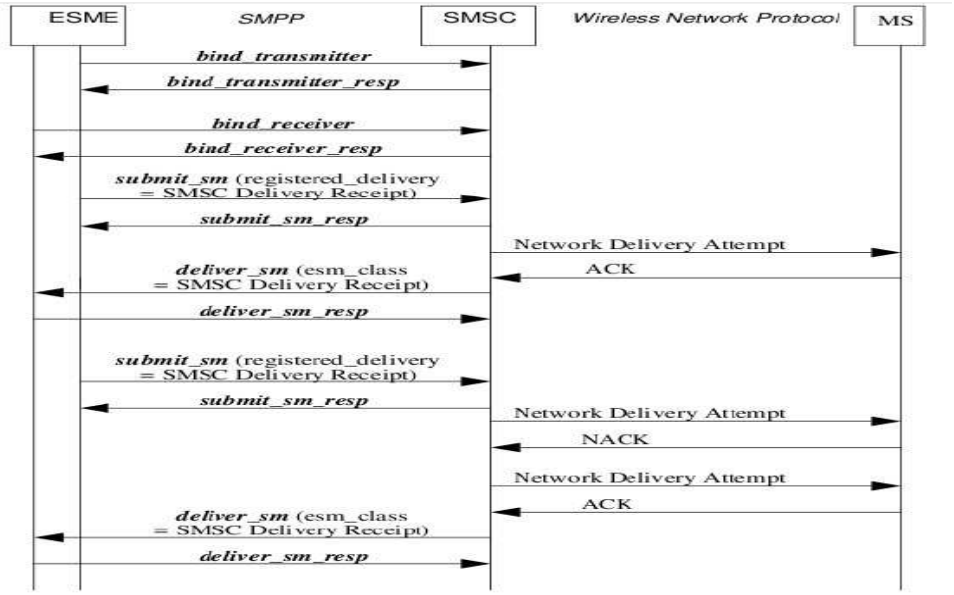
Normalmente los mensajes cortos son almacenados sobre un centro de mensajería antes de ser enviados a una aplicación de contenido. Con este modelo, el mensaje faltante seguramente almacenado en caso de que esto significará que todas las entregas fueran exitosamente realizadas por el centro de mensajería. A esta forma de manejar los mensajes se le conoce comúnmente como almacenaje y reenvío. El protocolo SMPP soporta el mecanismo de envio almacenaje y reenvío vía la operación submit_sm, la cual permite al ESME mandar un mensaje hacia el centro de mensajería donde este es almacenado, si no es así, la razón será que fue exitosamente enviado o

significa que el periodo de validación del mensaje ha expirado. El modo almacenaje y reenvío también esta soportado vía la operación data_sm.

El modo de mensajería almacenado y reenvío también facilita operaciones SMPP subsecuentes sobre el mensaje corto almacenado tal como query_sm, replace_sm y cancel_sm. El PDU submit_sm también facilita funcionalmente el reemplazo si está presente, lo cual requiere que el mensaje original se encuentre almacenado sobre el centro de mensajería.

La figura 14 ilustra el flujo para almacenar y reenviar mensaje donde el ESME está funcionando como un transmisor y como un receptor. El ESME solicitó un recipiente de envío SMSC.

Figura 14. **Diagrama de flujo para almacenar y reenviar un mensaje de texto entre una ESME y un SMSC**



Fuente: elaboración propia.

3.3. Definición de tipo PDU SMPP

Se explicará uno a uno las operaciones de administración de sesiones por su nombre de PDU en el protocolo SMPP.

Bind_transmitter: este PDU se utiliza para autenticación por una ESME Transmisora para poder establecer conexión con un centro de mensajería. El PDU contiene información sobre los parámetros de identificación y un *password* de autenticación para la aplicación de contenidos ESME.

Este tipo de operación es permitido para las aplicaciones de contenido ESME y no así para el centro de mensajería, el estado de sesión requerido para utilizarlo debe ser abierto.

Bind_transmitter_resp: este PDU se utiliza para que un centro de mensajería realice un mensaje de confirmación de respuesta al PDU bind_transmitter. Este PDU indica el éxito o fallo de la solicitud enviada por la aplicación de contenidos ESME que realizó el intento de conexión con el centro de mensajería. Este tipo de operación está permitido para el centro de mensajería y no para la aplicación de contenido. El estado de sesión requerido para utilizar esta función es también abierto.

Bind_receiver: este tipo de PDU de autenticación es utilizado por una aplicación Receptora de un proveedor de contenido ESME para establecer comunicación con un centro de mensajería. Este PDU contiene información de autenticación, también un *password* para la aplicación de contenido ESME, así como información de enrutamiento especificando el rango de direcciones a las cuales se brinda el servicio por la aplicación de contenido ESME. El estado de sesión requerido para utilizar esta función es también abierto. Este tipo de

operación es permitido para las aplicaciones de contenido ESME y no así para el centro de mensajería.

Bind_receiver_resp: el centro de mensajería responde a un PDU bind_receiver solicitado por una aplicación de contenido. Este PDU indica a la aplicación de contenido la respuesta brindada por el centro de mensajería con el resultado de si la comunicación fue exitosa o fallida para poder establecer comunicación con la aplicación de contenido ESME. Esta acción está permitida para centro de mensajería pero no para la aplicación de contenido ESME. También el tipo estado de sesión requerido para utilizar esta función es abierto.

Bind_transceiver: este tipo de PDU es utilizado para la autenticación y es utilizado para las aplicaciones de contenido que necesitan utilizar sesiones tipo RTX, para establecer comunicación con un centro de mensajería. Este PDU contiene información de autenticación, un *password* para que una aplicación de contenido o ESME pueda realizar la autenticación. Este tipo de estado de sesión requerido para utilizar esta función es también abierto. Este tipo de operación es permitido para las aplicaciones de contenido ESME y no así para el centro de mensajería.

En este ambiente, la telefonía móvil de tercera generación (3G) está siendo especificada por el proyecto mundial 3GPP (3G Partnership Project), con el objetivo de lograr una red global móvil con completa compatibilidad.

Bind_transceiver_resp: es utilizado para que un centro de mensajería responda a un PDU bind_transceiver. Este PDU indica a la aplicación de contenido la respuesta brindada por el centro de mensajería con el resultado de si la comunicación fue exitosa o fallida para poder establecer comunicación con la aplicación de contenido ESME. Esta acción está permitida para centro de

mensajería pero no para la aplicación de contenido ESME. El tipo estado de sesión requerido para utilizar esta función es abierto.

Outbind: este tipo de PDU de autenticación es utilizado por un centro de mensajería para comunicarse con una aplicación de contenido ESME, que informará a la misma que mensajes están presentes en el centro de mensajería. Este PDU contiene parámetros de autenticación tales como un usuario y *password* de acceso para que la aplicación de contenido pueda utilizar para establecer comunicación. Si la aplicación de contenido autentica la solicitud este responderá con un bind_receiver o bind_transceiver para con esto realizar el proceso de conexión dentro del centro de mensajería. El tipo estado de sesión requerido para utilizar esta función es abierto, esta acción está permitida para centro de mensajería pero no para la aplicación de contenido ESME.

Unbind: este tipo de PDU puede ser enviado tanto por la aplicación de contenido ESME y un centro de mensajería para poder indicar la finalización de la sesión SMPP. El tipo estado de sesión requerido para utilizar esta función puede ser bound_tx, bound_rx y bound_trx. Este PDU puede ser utilizado tanto para un centro de mensajería como para una aplicación de contenido ESME.

Unbind_resp: este tipo de PDU puede ser enviado desde un centro de mensajería o una aplicación de contenido ESME, el mismo es utilizado para dar la respuesta a la solicitud de desconexión (unbind). Cuando este PDU es enviado el centro de mensajería procederá por lo general a cerrar la conexión de red establecida. El tipo estado de sesión requerido para utilizar esta función puede ser bound_tx, bound_rx y bound_trx. Este PDU puede ser utilizado tanto para un centro de mensajería como para una aplicación de contenido ESME.

Enquire_link: este PDU puede ser enviado tanto por una aplicación de contenido ESME o un centro de mensajería, únicamente con el fin de probar la conexión de red y mantener viva una sesión SMPP. Este PDU puede ser utilizado tanto para un centro de mensajería como para una aplicación de contenido ESME y el tipo estado de sesión requerido para utilizar esta función puede ser bound_tx, bound_rx y bound_trx.

Enquire_link_resp: este PDU de sesión es utilizado para responder a la solicitud de enquire_link que fue enviada por una aplicación de contenido o un centro de mensajería. Este PDU puede ser utilizado tanto para un centro de mensajería como para una aplicación de contenido ESME, y el tipo estado de sesión requerido para utilizar esta función puede ser bound_tx, bound_rx y bound_trx.

Alert_notification: este tipo de PDU es utilizado cuando un centro de mensajería envía una alerta de notificación a una aplicación de contenido, como un mecanismo de alerta para así verificar la disponibilidad de una aplicación de contenido ESME. El tipo estado de sesión requerido para utilizar esta función puede ser bound_rx y bound_trx. Este tipo de funcionalidad es únicamente utilizado por el centro de mensajería y no por la aplicación de contenido ESME.

Generic_nack: este tipo de PDU puede ser enviado tanto por una aplicación de contenido como por un centro de mensajería para poder indicar la recepción de un PDU no valido. Con base en el estándar del contenido del PDU. Este PDU puede ser utilizado tanto para un centro de mensajería como para una aplicación de contenido ESME y el tipo estado de sesión requerido para utilizar esta función puede ser bound_tx, bound_rx y bound_trx.

3.4. Formato del PDU SMPP

Es posible explicar el formato de PDU en el protocolo SMPP según se puede detallar y ejemplificar por partes.

3.4.1. PDU SMPP por partes

Las operaciones que comúnmente son categorizadas dentro del siguiente grupo:

- Administradores de sesiones
- Envío de mensajes (*submit*)
- Entrega de mensajes (*deliver*)
- Envío masivo de mensajes
- Otras operaciones de funcionalidad del PDU

Para comprender el significado de estas categorías se explica su importancia y significado:

- Administración de Sesiones: esta operación está diseñada para habilitar el establecimiento de sesiones SMPP entre una *ESME* y un centro de mensajería, y proporcionar la información necesaria para el manejo de errores inesperados.

- Envío de mensajes (*submit*): estas son las operaciones asignadas específicamente para el envío de mensajes provenientes de las ESME hacia el centro de mensajería, es la única manera de poder realizar el envío de mensajes de texto desde una aplicación de contenidos ESME y que un centro de mensajería pueda recibirlos.

A continuación se explicará una a una las operaciones de envío de mensajes (*submit*) utilizadas en el protocolo SMPP:

Submit_sm: una aplicación ESME con tipo de sesión transmisora (TX) del tipo de transmisión recepción (TRX) que solicite el envío de un mensaje corto, puede utilizar este PDU para especificar los datos del origen y destino al cual desea enviar el mensaje corto. Existen otros atributos como la prioridad de envío del mensaje, su esquema de codificación, el periodo de validez, entre otros.

El tipo de estado de sesión requerido pueden ser bound_tx y bound_trx. Este servicio puede ser ofrecido por una aplicación de contenido ESME y no por un centro de mensajería.

Submit_sm_resp: el centro de mensajería responderá a un PDU submit_sm, con el cual le informará el éxito o falla de la solicitud enviada por la aplicación de contenido ESME. También este PDU incluye una message_id del centro de mensajería que puede ser utilizada en operaciones subsecuentes para poder realizar consultas, cancelaciones o reemplazo de contenidos de un mensaje de texto que aún no ha sido enviado. El tipo de estado de sesión requerido pueden ser bound_tx y bound_trx. Este servicio puede ser ofrecido por un centro de mensajería y no por una aplicación de contenido ESME.

Submit_sm_multi: este PDU es una variación del PDU submit_sm que puede soportar arriba de 255 recipientes para la entrega de mensajes múltiples en un solo envío de comunicación entre la aplicación de contenido ESME y un centro de mensajería. El tipo de estado de sesión requerido pueden ser bound_tx y bound_trx. Este servicio puede ser ofrecido por una aplicación de contenido ESME y no por un centro de mensajería.

Submit_sm_multi_resp: el centro de mensajería responde al PDU submit_multi para poder confirmar la recepción de la solicitud enviada por la aplicación de contenido ESME. Esto es similar al PDU submit_sm_resp. La principal diferencia se encuentra en donde algunos de los recipientes específicos fueron por cualquier razón inválidos o rechazados por el centro de mensajería, el PDU submit_sm_multi_resp puede especificar la lista de recipientes fallidos, agregando así el código de error específico para cada uno e indicando la razón de porque el recipiente fue inválido en la respuesta de este PDU se incluye un message_id del centro de mensajería el cual puede ser usado en operaciones subsecuentes para realizar consultas, cancelación o reemplazo de contenidos de un mensaje no enviado. El tipo de estado de sesión requerido pueden ser bound_tx y bound_trx. Este servicio puede ser ofrecido por un centro de mensajería y no por una aplicación de contenido ESME.

Data_sm: este tipo de PDU es una versión flujo de la operación submit_sm, pero destinada para basarse en el empaquetamiento de aplicaciones que no demandan funcionalidad extendida normalmente disponible en la operación submit_sm. Este tipo de PDU es utilizado por aplicaciones de contenido ESME que implementan WAP en un mensaje de texto SMS. Este servicio puede ser ofrecido por un centro de mensajería y por una aplicación de contenido ESME, el tipo de estado de sesión requerido pueden ser bound_tx, bound_rx y bound_trx.

Data_sm_resp: el centro de mensajería responde al PDU data_sm indicando el éxito o falla de la solicitud enviada por la aplicación de contenido. También en esta funcionalidad se incluye un message_id del centro de mensajería, el cual puede ser utilizado para operaciones subsecuentes de consulta, cancelación o reemplazo de contenido de un mensaje que aún no ha

sido enviado. Este servicio puede ser ofrecido por un centro de mensajería y por una aplicación de contenido ESME, el tipo de estado de sesión requerido pueden ser bound_tx, bound_rx y bound_trx.

Entrega de mensajes (*deliver*): esta operación es la que permite al centro de mensajería enviar mensajes a una aplicación de contenidos ESME, esta es la única manera de poder realizar el envío de mensajes de texto desde el centro de mensajería hacia una aplicación de contenidos.

Las operaciones de entrega de mensajes (*deliver*) utilizadas en el protocolo SMPP son:

Deliver_sm: este PDU es el opuesto simétrico a un submit_sm y es utilizado por un centro de Mensajería para enviar y recibir o ejecutar el rol de una aplicación de contenido ESME con sesión transmisión y recepción (TRX). El tipo de estado de sesión requerido pueden ser bound_rx y bound_trx. Este servicio puede ser ofrecido por un centro de mensajería y no por una aplicación de contenido ESME.

Deliver_sm_resp: este PDU indica la aceptación o rechazo de un mensaje enviado. El error que retorna una aplicación de contenido ESME puede causar que el mensaje pueda ser reprocesado en un momento posterior o inclusive ser rechazado. El tipo de estado de sesión requerido pueden ser bound_rx y bound_trx. Este servicio puede ser ofrecido por un centro de Mensajería y por una aplicación de contenido ESME.

Data_sm: este tipo de PDU data_sm también puede ser utilizado para enviar un mensaje desde el centro de mensajería hacia una aplicación de contenido ESME. Aplicaciones de contenido implementan WAP en SMS por lo

general para usar esta operación. Este servicio puede ser ofrecido por un centro de mensajería y por una aplicación de contenido ESME. El tipo de estado de sesión requerido pueden ser bound_tx, bound_rx y bound_trx.

Data_sm_resp: una aplicación de contenido responde a un PDU data_sm, indicando el éxito o fracaso del envio de un centro de mensajería que solicito este tipo de envío. El tipo de estado de sesión requerido pueden ser bound_tx, bound_rx y bound_trx. Este servicio puede ser ofrecido por un centro de mensajería y por una aplicación de contenido ESME.

Envío masivo de mensajes: esta operación está diseñada para proveer la funcionalidad del envío de mensajes masivos sin la necesidad de la utilización de un centro de mensajería.

Otras operaciones de funcionalidad del PDU: estas operaciones están diseñadas para proveer funcionalidades mejoradas tales como la cancelación del envío de un mensaje o solicitudes de consulta del envío o reemplazar los mensajes ya enviados.

3.4.2. Longitud de un PDU SMPP

El campo command_length que inicia el encabezado PDU SMPP, indica el número total de octetos contenidos en ese PDU SMPP. El campo command_length contiene un entero de 4 octetos transmitido en formato *Big Endian*. Para decodificar un PDU SMPP, el ESME o SMSC deberán leer primero el campo command_length (de 4 octetos) para determinar la longitud del PDU. La cantidad de datos es determinada sustrayendo la longitud del campo command_length que contiene (4 octetos) del total de la longitud del PDU como el valor de campos proveídos por el campo command_length.

4. EL PROTOCOLO SMPP APLICADO EN CENTROS DE MENSAJERIA SMSC O APLICACIONES ESMES

4.1. SMSC

Las siglas SMSC esencialmente corresponden a Short Message Service Center cuya traducción al español es la de centro de servicio de mensajes cortos, es básicamente un centro de servicio en una red de telefonía celular cuya función básica es hacer posible el envío y recepción de mensajes de texto.

4.1.1. Conceptos

En una red celular cuando un abonado A desea enviar un mensaje de texto en sus siglas en inglés SMS (Short Message Service) a un abonado B, se realiza una solicitud de envío mediante el aprovisionamiento de un Global Title (GT) en la terminal del abanado A. La red celular del suscriptor A envía la petición al centro de mensajería (SMSC), la cual realiza una búsqueda del suscriptor B para entregar dicho mensaje de texto. La función especial de un SMSC es que cuenta con la habilidad de almacenar dicho mensaje de texto y poder entregar el mensaje a un suscriptor B si este no se encuentra disponible en la red al momento de haberle sido enviado el mensaje de texto.

La configuración de tiempo de almacenaje de un mensaje de texto es configurado en el centro de mensajería (SMSC) al que pertenece el suscriptor. Este depende de la cantidad de usuarios y el tamaño de los mensajes, así como de la cantidad de reintentos que realizar el centro de mensajería durante un tiempo determinado para poder encontrar al abonado B.

El esquema de reintentos que se configura en un centro de mensajería para la entrega de un mensaje de texto, puede ser optimizado dependiendo de las respuestas a nivel de señalización que se reciben al momento de entregar el mensaje al abonado B. Con esto puede reducirse y optimizar recursos de red tales como señalización y de consulta.

Un centro de mensajería (SMSC) presenta conectividad contra una red completa de Core, la cual es la encargada de poder conectar al centro de mensajería a toda la infraestructura de red y hacer posible que esta pueda tener comunicación completa con todos los suscriptores de una red celular. Un centro de mensajería debe contar con comunicación a su vez con una unidad consulta de saldo, la cual en tiempo real puede informar al suscriptor si cuenta con saldo suficiente para poder enviar un mensaje de texto a un abonado B.

Un centro de mensajería (SMSC) también cuenta con conexión con otros centros de mensajería (SMSC), esto para poder realizar el envío de mensajes de texto entre operadores y poder realizar la comunicación con miles de suscriptores que están situados en otras coberturas de una red celular.

4.1.2. Arquitectura física y funcionamiento de un centro de mensajería (SMSC)

Un centro de mensajería (SMSC) debe ser capaz de comunicarse con la red para poder realizar consultas sobre la ubicación en la red de un suscriptor o abonado y poder entregar el mensaje de texto (SMS) al suscriptor. Así como poder consultar en línea el saldo del suscriptor o abonado que requiere el envío del mensaje de texto.

Un centro de mensajería (SMSC) básicamente está compuesto de varios equipos de cómputo con altas cualidades de procesamiento, memoria RAM y alta capacidad en discos duros. Esencialmente es un servidor poderoso a nivel de hardware, el cual permite por estas cualidades de alto performance de procesamiento poder atender una gran cantidad de peticiones en un tiempo relativamente corto. El software que tiene instalado cada equipo de cómputo utilizado en un centro de mensajería (SMSC) varía según su funcionalidad y acción tomada para poder procesar un mensaje de texto.

Para poder comprender a detalle la arquitectura física y funcionamiento de un centro de mensajería se procederá a detallar a continuación los nodos que lo componen.

Servidores MAP o señalización: las siglas MAP significan Mobile Application Part, el cual es un protocolo de SS7 el cual provee una capa de aplicación para varios nodos GSM y UMTS de una red móvil Core la cual se utiliza para comunicarse con otros nodos que proveen servicios de telefonía celular. MAP es una capa de aplicación utilizada para poder acceder al HLR (Home Location Register), VLR (Visitor Location Register), MSC (Mobile Switching Center), EIR (Equipment Identity Register), AUI (Authentication Centre) o al SGSN (Serving GPRS Support Node).

Estos servidores hacen posible la comunicación del centro de mensajería con la red completa Core que posee una red de telefonía celular. Dentro del código binario incluido en el software de este equipo este es capaz de interpretar la información y comprenderá la misma.

El lenguaje utilizado en señalización es el que utiliza la red Core para comunicarse entre ellos. El servidor MAP en el centro de mensajería (SMSC)

únicamente traduce el lenguaje de señalización a un lenguaje propio para poder comunicarse con los otros servidores internos que posee para poder realizar la petición de envío de mensaje de texto que proviene de la red de telefónica celular.

Servidores SFE: las siglas SFE significan Store and Forward Engine, en el centro de mensajería estos son los encargados de contar con la lógica de almacenamiento de un mensaje de texto que no fue entregado en el primer intento a un abonado. Estos equipos cuentan con un traductor interno para poder comunicarse con los servidores MAP, que utilizan el protocolo MAP, para interactuar con los nodos Core de la red celular. Estos servidores utilizan un protocolo único para poder realizar la lógica de validación de enrutamiento y verificación de saldo; así como almacenar y realizar con base en una respuesta de la red el reenvío de un mensaje de texto.

Dependiendo del fabricante, estos equipos cuentan con la capacidad de almacenar los mensajes en un formato de archivo de texto, el cual hace posible manejar una gran cantidad de mensajes y así contar con una gran capacidad de almacenamiento. Estos equipos además de contar con la lógica de almacenamiento, contienen las configuraciones de esquemas de reintentos de los mensajes que no fueron entregados al primer intento, el cual hace posible que un suscriptor pueda recibir el mensaje de texto horas después de que el mismo fue enviado.

Estos equipos contienen la lógica de enrutamiento de los mensajes. Estas son tablas de enrutamiento que contienen el detalle de conexión para cada caso de envío, es decir el caso de enrutamiento que debe seguirse para que el mensaje de texto enviado pueda ser enviado a su destinatario final; así como la realización del cobro en línea con algún sistema prepago o varios.

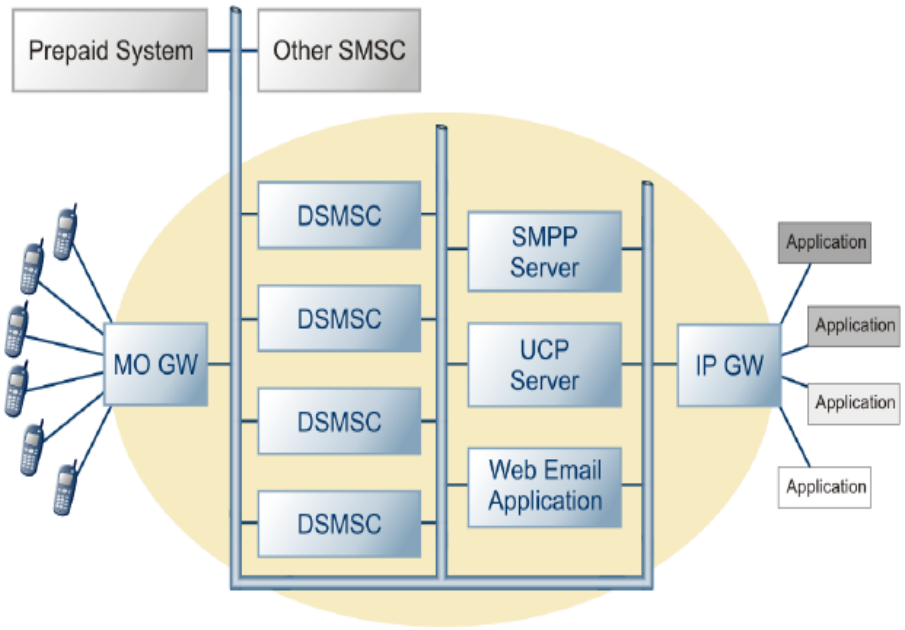
Servidores SMPP: las siglas SMPP significan Short Message Peer to Peer y este es un protocolo utilizado en la industria para el envío y recepción de mensajes de texto en una red GSM.

La función especial de estos servidores en un centro de mensajería, es poder traducir al protocolo SMPP un mensaje de texto. En la industria de telecomunicaciones este tipo de servidores son utilizados para poder comunicarse vía el protocolo SMPP con otros servidores de aplicaciones de contenido llamadas ESME, así como con otros centros de mensajería para poder así, realizar el envío de mensajes de texto no solamente a usuarios de una misma red, sino a varias redes o aplicaciones de contenido que cuentan con la necesidad de recibir mensajes de texto (SMS).

Estos equipos cuentan con un traductor interno para poder comunicarse con los servidores SFE y poder convertir la petición que proviene de los servidores de almacenamiento al estándar SMPP y así poder comunicarse con otros equipos que manejan este lenguaje.

La comunicación o solicitud de envío de mensaje de texto SMS puede ser originada también a través de un servidor SMPP, el cual se comunicará con los servidores de almacenamiento SFE y el servidor MAP si en el servidor de almacenamiento SFE se decidiera que lo requiere.

Figura 15. **Diagrama interno de un centro de mensajería SMSC**



Fuente: elaboración propia. Microsoft Visio 2007.

4.1.3. Arquitectura física y funcionamiento de una entidad de enrutamiento (RE)

Conocida mundialmente en la industria de telecomunicaciones como Routing Entity RE que en español es llamada entidad de enrutamiento.

Una entidad de enrutamiento o RE consta de servidores de cómputo los cuales cuentan con binarios que contienen la lógica de ruteo que hace posible el correcto enrutamiento de un solicitud de envío de mensaje de texto a otro centro de mensajería SMSC o a una aplicación ESME que en sus siglas en inglés significa External Short Messaging Entity.

Una entidad de enrutamiento es la encargada de establecer y mantener la comunicación entre un centro de mensajería o un ESME mediante el estándar SMPP.

Utilizando el protocolo TCP/IP y el estándar SMPP la entidad de enrutamiento puede comunicarse y mantener una conexión establecida, utilizando el protocolo SMPP para gestionar el tipo de sesión que debe establecerse para el envío de mensajes de texto.

Básicamente su topología se compone de un servidor capaz de realizar el enrutamiento bajo la definición de tablas de ruteo, configuradas manualmente para establecer comunicación y gestionar la misma.

Un servidor de entidad de enrutamiento no posee la habilidad de realizar almacenamiento. Su funcionamiento es especialmente utilizado en la industria únicamente para distribuir tráfico SMS, sin contar con una inteligencia propia sobre el éxito o fracaso del mensaje de texto enviado o recibido. Es comúnmente llamada terminal tonta y este ente únicamente se dedica a realizar tarea de enrutamiento, sin importarle el éxito o fracaso del envío del mensaje de texto solicitado.

4.1.4. Lógica de operación

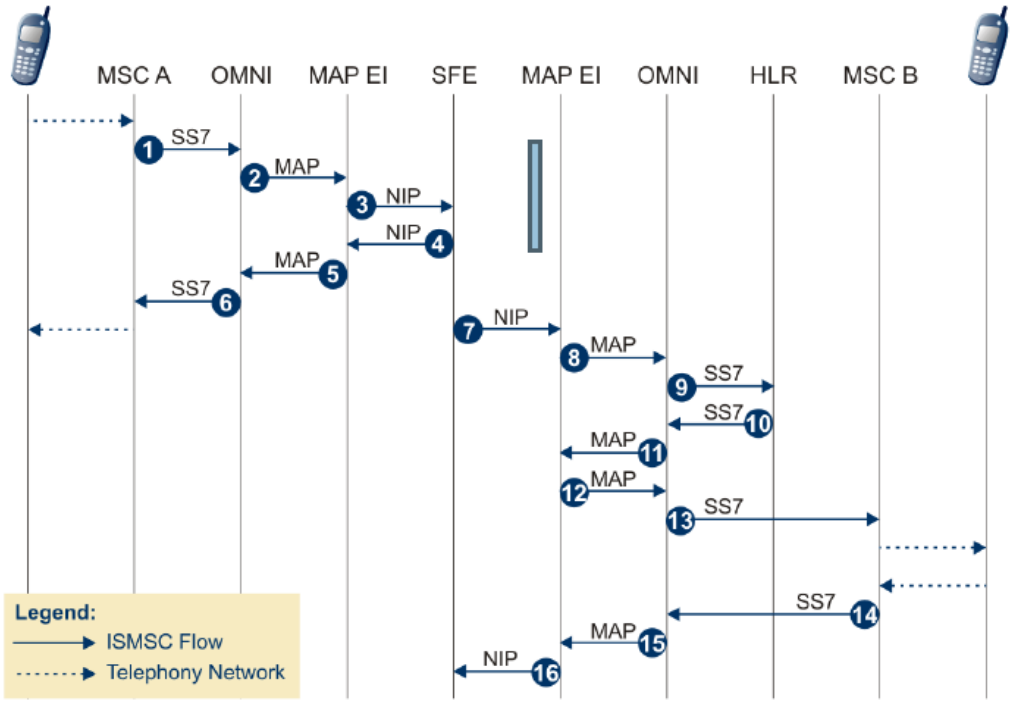
Para explicar la lógica de operación de un centro de mensajería SMSC, se procederá a utilizar escenarios de envío para poder comprender la interacción con todos los dispositivos.

Envío de mensaje de texto entre usuarios de una misma red GSM (MO – MT):

Las siglas MO significan Mobile Originator and MT Mobile Terminating son utilizadas en la industria de telecomunicaciones para denotar al suscriptor móvil que origina el mensaje o a quien se dirige el mismo (terminante). También el acrónimo MS (Mobile Suscriber) es usado para describir al suscriptor en una red de telefónica.

El siguiente escenario describe el flujo de MO a MT sobre una misma red. En este ejemplo ambos MS residen en una red GSM.

Figura 16. **Diagrama de flujo entre un MO a MT en una red de telefonía**



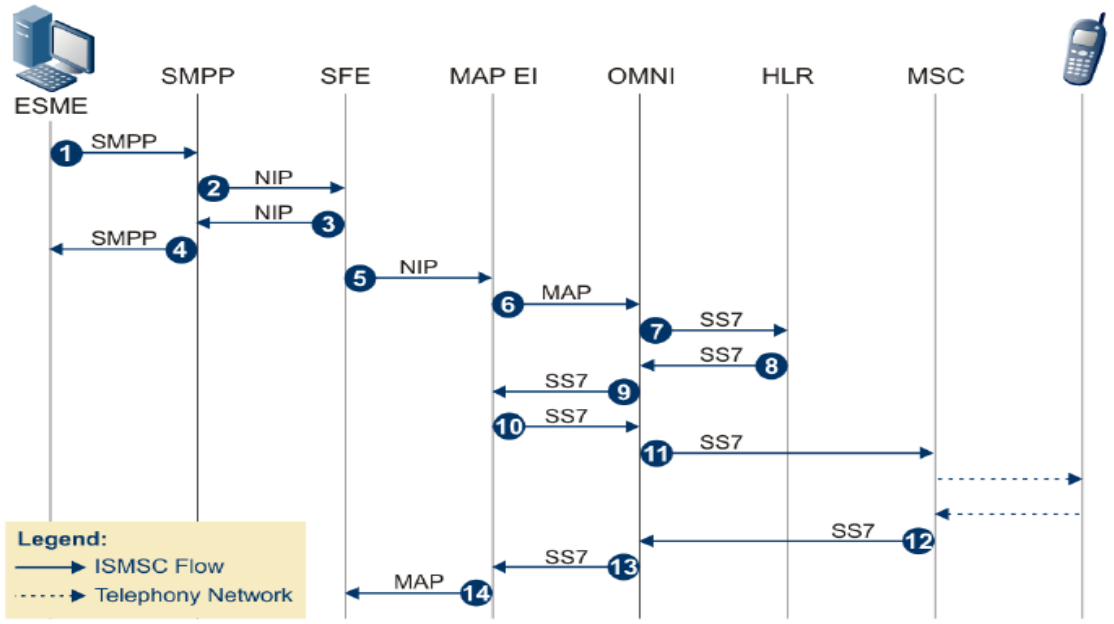
1	La MSC envía el mensaje al servidor MAP.
2	El servidor MAP envía el mensaje al servidor SFE.
3	El servidor SFE brinda la confirmación de recepción del mensaje, después verifica y lo almacena.
4	El servidor MAP envía la confirmación de recepción del mensaje de texto.
5	El servidor SFE envía el mensaje al servidor MAP.
6	El servidor MAP requiere al HLR que ejecute la consulta de ubicación del suscriptor B para así poder enviarle el mensaje de texto.
7	El HLR responde a la interrogación con la información detallada del ruteo que de realizarse para encontrar al suscriptor B.
8	Considerando la información enviada, el MAP entrega el mensaje a las MSC.
9	La MSC brinda la confirmación de recepción de entrega.
10	El servidor MAP brinda la confirmación de entrega de vuelta al servidor SFE.

Fuente: elaboración propia.

Envío de mensaje de texto entre una aplicación ESME y un móvil (AO – MT):

El siguiente escenario describe el flujo de envío de un mensaje de texto entre una Aplicación de Contenido (ESME) y un suscriptor. En este ejemplo, la aplicación utiliza el protocolo SMPP y el suscriptor abonado (MS) es parte de una red GSM.

Figura 17. **Diagrama de flujo entre una ESME a un suscriptor en una red de telefonía**



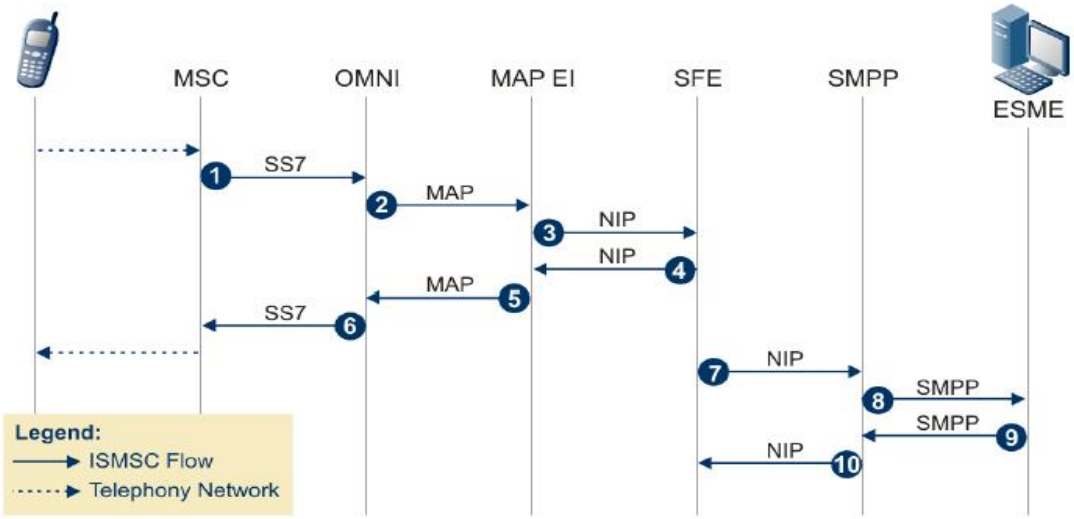
1	La ESME envía un mensaje al servidor SMPP.
2	El SMPP envía el mensaje al servidor SFE.
3	El servidor SFE brinda la confirmación de recepción del mensaje, después verifica y lo almacena en su base de datos.
4	El servidor SMPP brinda la confirmación de recepción del mensaje.
5	El servidor SFE envía el mensaje al servidor MAP.
6	El servidor MAP requiere e interroga al HLR para conocer la información de ruteo del suscriptor.
7	El HLR responde con la información de ruteo del suscriptor.
8	Considerando la información brindada por el HLR, el servidor MAP entrega el mensaje a la MSC.
9	La MSC brinda la confirmación de recepción del mensaje a entregar.
10	El servidor MAP brinda la confirmación de recepción del mensaje al servidor SFE para enterarlo que el mensaje ya fue recibido por la MSC.

Fuente: elaboración propia.

Envío de mensaje de texto entre móvil a una aplicación de contenido (MO - AT):

El siguiente escenario describe el flujo de envío entre un suscriptor móvil y una aplicación de contenido ESME, en este ejemplo el suscriptor forma parte de la red GSM.

Figura 18. **Diagrama de flujo entre una ESME a un proveedor de contenido (ESME) en una red de telefonía**



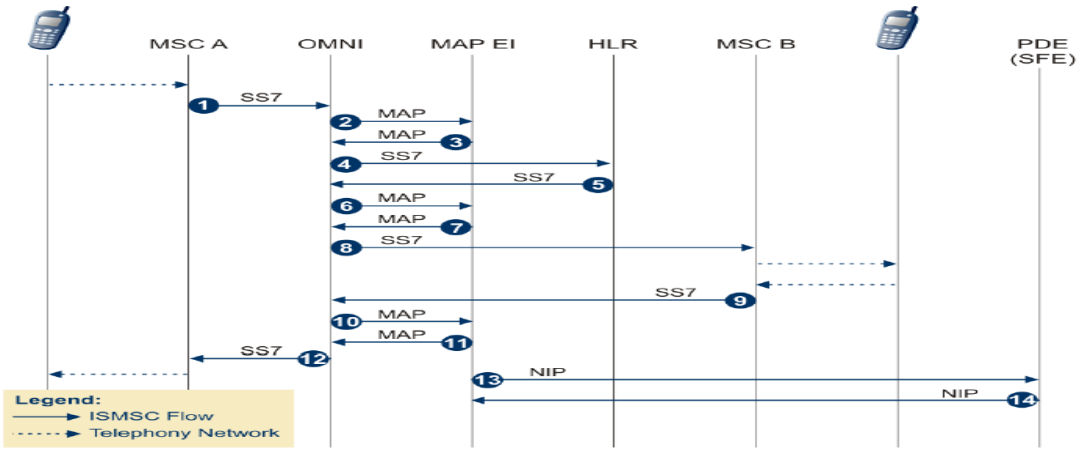
1	La MSC envía el mensaje al servidor MAP
2 y 3	El servidor MAP envía el mensaje al servidor SFE.
4 y 5	El servidor SFE brinda la confirmación de recepción del mensaje después de ser verificado y almacenado.
6	El servidor MAP brinda la confirmación de recepción del mensaje a la MSC que originó la solicitud.
7	El servidor SFE envía el mensaje al servidor SMPP.
8	El servidor SMPP entrega el mensaje a la aplicación de contenido ESME.
9	La ESME brinda la confirmación de recepción al servidor SMPP.
10	El servidor SMPP brindar la confirmación de recepción de vuelta al servidor SFE.

Fuente: elaboración propia.

FDA (First Delivery Attempt)

Este escenario describe el flujo para el envío de un mensaje de texto en el primer intento.

Figura 19. **Diagrama de flujo para el envío de mensajes de texto en el primer intento en una red de telefonía**



1, 2 y 3	La MSC "A" envíe un mensaje al servidor MAP.
4	Después de verificar el mismo, el servidor MAP requiere la ejecución del procedimiento de consulta de enrutamiento al HLR.
5, 6 y 7	El HLR responde con la información de enrutamiento del suscriptor al servidor MAP.
8	Considerando la información brindada el servidor MAP entregará el mensaje nuevamente a la red, en este caso al MSC "B".
9	La MSC "B" brinda la confirmación de recepción del mensaje entregado.
10, 11	El servidor MAP recibe la confirmación de recepción del mensaje enviado.
12	El servidor MAP solicita esta confirmación para poder así realizar acción de posentrega si el mensaje no fue entregado en el primer intento.
13 y 14	Se recibe la respuesta de confirmación del servidor de SFE.

Fuente: elaboración propia.

4.2. ESMES

Su significado en español es entidad externa de mensajeria corta, en sus siglas en inglés (External Short Messaging Entity), este es un término originalmente brindado por Aldiscon, compañía de telecomunicaciones fundada en Dublin, Irlanda en 1988. Aldiscon suplió productos de software a operadores de teléfonos móviles y se convirtió en un líder en suministrar al mundo centros de mensajería.

4.2.1. Conceptos

Aldiscon inventó el protocolo industrial de telecomunicaciones de mensajes de punto a punto llamado SMPP, el cual sirve para intercambiar mensajes cortos entre entes tales como centros de mensajerías.

Este se describe como una aplicación externa que se interconecta con un centro de mensajería para realizar el envío o recepción de mensajes cortos:

- SME es un término utilizado en muchos círculos de telefonía para describir una entidad de red de telefonía que es capaz de enviar y recibir mensajes cortos. Una ESME es esencialmente un ente capaz de conectarse vía TCP/IP o X.25. Bajo el estándar SMPP 3.4 una *ESME* se refiere únicamente a un ente externo capaz de enviar mensajes cortos como sistema de procesamiento de voz, servidores WAP Proxy. Un ejemplo típico de ESME son sistemas que envían automáticamente mensajes de texto a móviles de usuarios de una red de telefonía celular. Básicamente el término se utiliza para nombrar a una aplicación que no es catalogada como un móvil en una red de telefonía.

Representa típicamente a un cliente SMS (Sistema de mensajes cortos) configurado en la red, tal como lo sería un WAP Proxy Server, un puerta de enlace para el correo, o bien un servidor de *voice mail*. Un ESME también puede representar una entidad célula del broadcast (Cell Broadcast Entity en sus siglas en inglés CBE).

4.2.2. Parámetros básicos de configuración para el establecimiento de *bind* con un SMSC o RE

En la industria de telecomunicaciones es muy utilizado el SMGateway como Router Entity o Entidad de enrutamiento entre la ESME y el SMSC.

Una SMGateway es una plataforma diseñada con una arquitectura modular completamente basada en los recursos y estándares del sistema operativo UNIX. Cada uno de los bloques de software que componen el sistema atiende una función específica dentro de la operación de SMGateway y se intercomunican internamente utilizando recursos interproceso definidos en el sistema operativo.

Por otra parte, la interconexión de la plataforma con el resto del mundo se realiza a través de puertos de red Ethernet, sobre los cuales se implementa el conjunto de protocolos TCP/IP. Como puede notarse los recursos sobre los cuales se estructura el SMGateway son completamente convencionales, lo que se traduce en una mayor capacidad de integración de la plataforma.

El comando básico para realizar la conectividad en un centro de mensajería o en una ESME es el *bind.*

El tipo de *bind* que se puede realizar puede variar según el tipo de conexión o sesión establecida:

bind_receiver
bind_transmiter
bind_transceiver

Las estructuras de un PDU de SMPP para la realización de un *bind* se compone de:

- Header o en español encabezado
- Mandatory Parameters o en español parametros mandatorios
- Optional Parameters o en español parámetros opcionales
- Input Mode o en español modo de entrada

Los parámetros básicos con los cuales se compone un PDU de bind según su estructura son:

- Header

Command_id: este campo identifica el tipo de mensaje que representa el PDU, por ejemplo: Submit_sm o query_sm. Campo utilizado en el protocolo para determinar el tipo de comando que se realizará entre un centro de mensajería o una ESME. Este comando gobierna el tipo de conexión o sesión que se realizará entre estos entes y el mismo es el que indica que tipo de acción se realizará entre estos equipos.

Command_Status: campo del PDU que indica el éxito o fracaso de una petición de SMPP. Este es relevante solo en la respuesta de un mensaje SMPP y debe ser configurada a NULL en una solicitud de mensaje.

El listado completo de errores para este campo son los que se representan en la figura de abajo:

Figura 20. **Listado de errores Command_Status en el protocolo SMPP**

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
Reserved	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
Reserved	0x00000010	Reserved
ESME_RCANCELFAIL	0x00000011	Cancel SM Failed
Reserved	0x00000012	Reserved
ESME_RREPLACEFAIL	0x00000013	Replace SM Failed

Fuente: elaboración propia.

Sequence number: este campo es quien permite llevar la correlación de una respuesta al PDU que solicito una operación. La respuesta SMPP del PDU debe conservar este campo.

Según el estándar SMPP 3.4 el rango de sequence number es de 0x00000001 a 0x7FFFFFFF en hexadecimal y en números decimales de 1 a 2,147,483,647.

Este valor no debe ser sobrepasado ya que si se excede el mismo y el ente remoto atiende el estándar SMPP 3.4 dejará de procesar el tráfico SMPP enviado, causando problemas en el envío del tráfico.

Command length: este parámetro indica la longitud en octetos de un mensaje SMPP. Esta longitud incluye el encabezado del mensaje SMPP (incluyendo el campo command_length por sí mismo), los parámetros mandatorios y los parámetros opcionales.

Campos mandatorios y opcionales

System_id: este parámetro es usado para identificar una ESME o un SMSC al momento de realizar el bind o conexión. Una ESME se identifica a una ESME o un agente ESME a un SMSC. El system_id de un SMSC provee una identificación de un SMSC a una ESME.

Password: el parámetro de *password* es usado por la SMSC para autenticar la identidad de una conexión ESME. Un proveedor de servicios debe contar con ESME para proveer un *password* cuando realiza la conexión a la SMSC. El *password* normalmente es configurado con la finalidad que únicamente el administrador del sistema pueda conocerlo.

System_type: este parámetro es usado para categorizar el tipo de ESME que está requiriendo conexión a la SMSC.

Ejemplos de estos incluyen VMS (Voice Mail System) y OTA (Over-The-Air Activation System). La especificación del system_type es opcional, algunas SMSC o centros de mensajería no requieren a la ESME este detalle. En este caso la ESME puede configurar el system_type sin ningún valor, es decir dejar el campo vacío.

Interface_version: este parámetro es usado para indicar la versión del protocolo SMPP. Los valores de versión de interfaces se comparten a continuación:

Figura 21. **Listado de valores de versión de interfaces en el protocolo SMPP**

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

Fuente: elaboración propia.

Addr_ton, source_ton_dest_addr_ton, esme_addr_ton: los campos que definen el número de tipo o Type of Number (TON) en sus siglas en Inglés es usado en los parámetros de direcciones de una ESME. Los siguientes valores de TON se definen a continuación:

Figura 22. **Listado de valores de TON en el protocolo SMPP**

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

Fuente: elaboración propia.

Este campo define el número de indicador de plan o Numeric Plan Indicator (NPI) en sus siglas en Inglés a ser utilizados los parametros de una ESME. Los valores de NPI se definen a continuación:

Figura 23. **Listado de valores de NPI en el protocolo SMPP**

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

Fuente: elaboración propia.

Address_range: este parametro es usado en una conexión con comandos bind_receiver y bind_transceiver para especificar una configuracion de servicio de direccion de una ESME para el cliente ESME. Una única dirección ESME puede especificar un parámetro de address_range. Los mensajes direccionados a ningún destino en este rango pueden ser enrutados a la ESME. Es posible especificar una única dirección IP. Un rango de direcciones IP no es permitió. La versión IP 6.0 no es actualmente soportada en esta version de protocolo SMPP 3.4.

Descripción de parámetros opcionales SMPP: estos campos opicionales son campos, de los cuales los mismos pueden ser opcionales en un mensaje SMPP. Estos parametros opcionales deben siempre aparecer al final de un mensaje, en la seccion de Optional Parameters de un PDU SMPP. Sin embargo, estos siempre deben estar presentes en el PDU y el estandar SMPP pueda mantenerse y ser utilizados según se requiera según la necesidad de la aplicación.

Todos los parametros opcionales SMPP poseen parametros de 16 bits como identificador.

El protocolo SMPP define los siguientes bloques de parametros en la figura de abajo:

Figura 24. **Listado de bloques de parámetros en el protocolo SMPP**

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

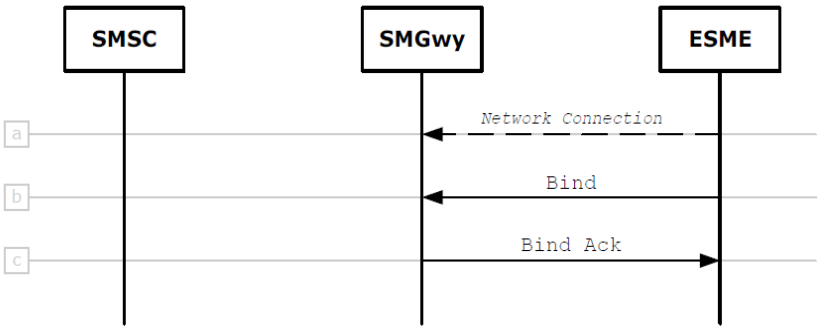
Fuente: elaboración propia.

Input Mode: este campo es utilizado en la aplicación para definir que todos los valores en el PDU que no sean hexadecimales por definicion lleven valores en hexadecimal o alfanuméricos (decimal). Este parametro es parte del aplicativo SMPP Tester el cual es de utilidad para validar el valor hexadecimal de los campos. La figura de abajo muestra de forma gráfica los campos de la estructura del PDU de una solicitud bind_transceiver.

4.2.3. Diagramas de flujo de información

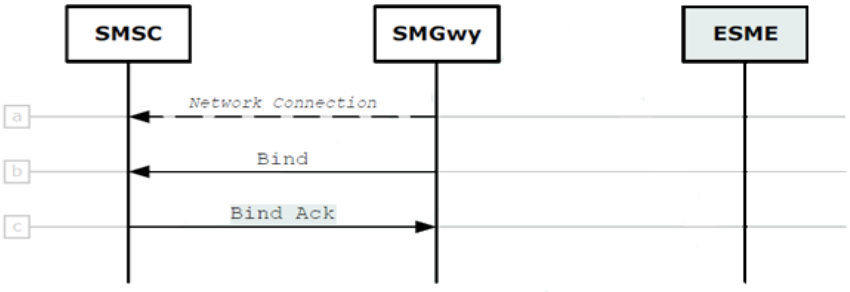
A continuación se muestran los diagramas de flujo de información de conexiones ESME o un SMSC desde y hacia la plataforma SMGateway.

Figura 25. **Diagrama de Flujo del diagrama de conexión de ESME hacia la plataforma SMGateway**



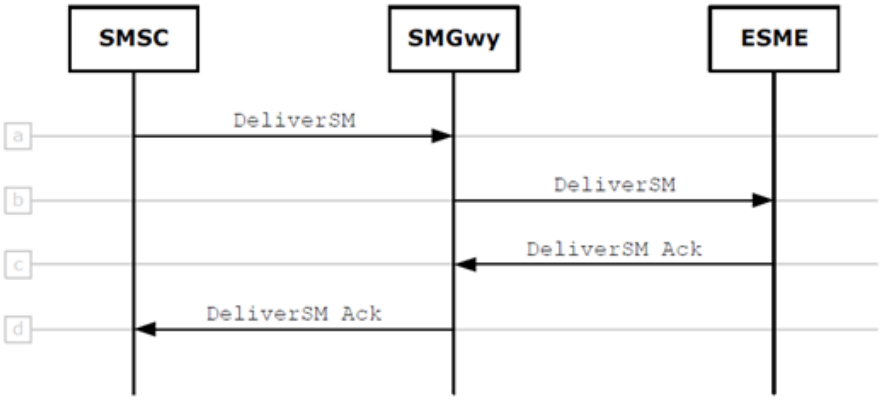
Fuente: elaboración propia.

Figura 26. **Diagrama de conexión de SMSC desde la plataforma SMGateway**



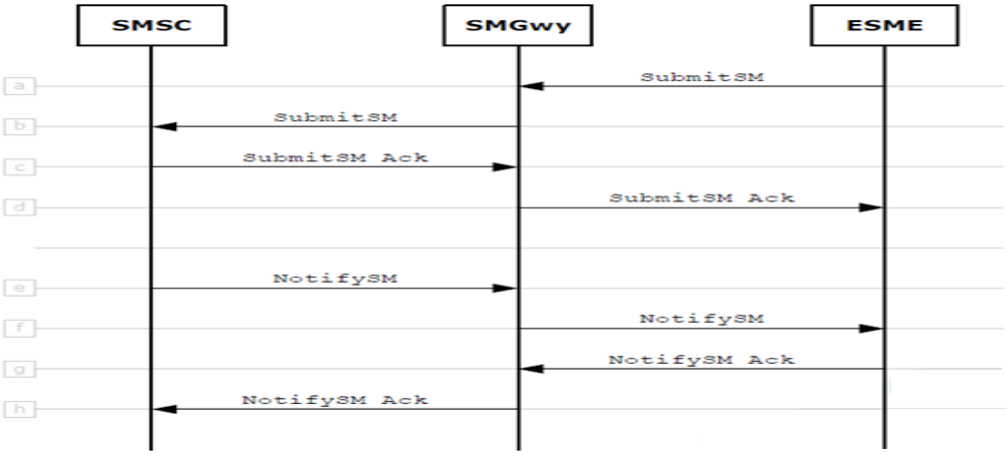
Fuente: elaboración propia.

Figura 27. **Diagrama de despacho de mensaje corto desde SMSC hacia ESME, a través de la plataforma SMGateway**



Fuente: elaboración propia.

Figura 28. **Diagrama de despacho de mensaje corto dese ESME hacia SMSC y su posterior notificación, a través de la plataforma SMGateway**



Fuente: elaboración propia.

4.3. Aplicación SMPP Tester

La herramienta SMPP Tester es un software capaz de simular una aplicación de prueba del protocolo SMPP. Capaz de ser conectada a un centro de mensajería o una entidad externa de mensajería corta. La misma cuenta con la estructura definida en el estándar SMPP 3.4, la cual permite simular el envío de mensajes de texto basado en el protocolo SMPP 3.4 el cual se explicó en los primeros capítulos de este trabajo de graduación.

4.3.1. Conceptos y descripción de la herramienta SMPP Tester

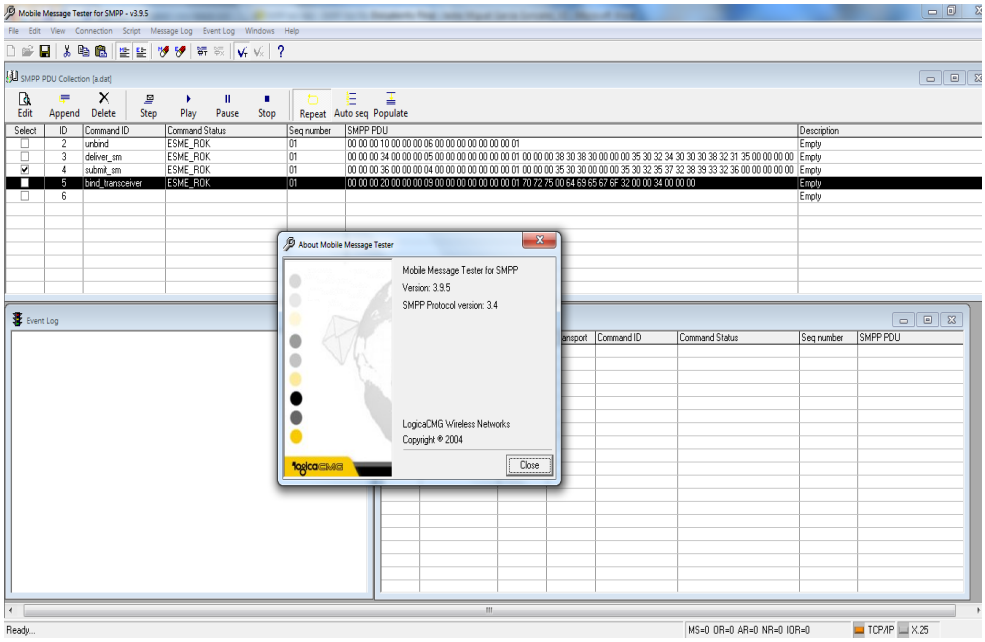
Esta aplicación es muy valiosa debido a que se emplea para la realización de actividades de *troubleshooting* al momento de fallas, o muy utilizada en el medio de telecomunicaciones para realizar pruebas de funcionalidad, y poder resolver inconvenientes de autenticación o problemas de envío de mensajes con entidades externas de mensajería consiguiendo simular y validar el correcto funcionamiento de ambos entes.

Dicha aplicación es capaz de soportar el estándar SMPP 3.4 y contar con la estructura PDU de este estándar. Siendo capaz de utilizar los siguientes command_id para realizar conectividad entre una ESME y una SMSC:

- BIND_TRANSMITTER / BIND_TRANSMITTER_RESP
- BIND_RECEIVER / BIND_RECEIVER_RESP
- SUBMIT_SM / SUBMIT_SM_RESP
- ENQUIRE_LINK / ENQUIRE_LINK_RESP

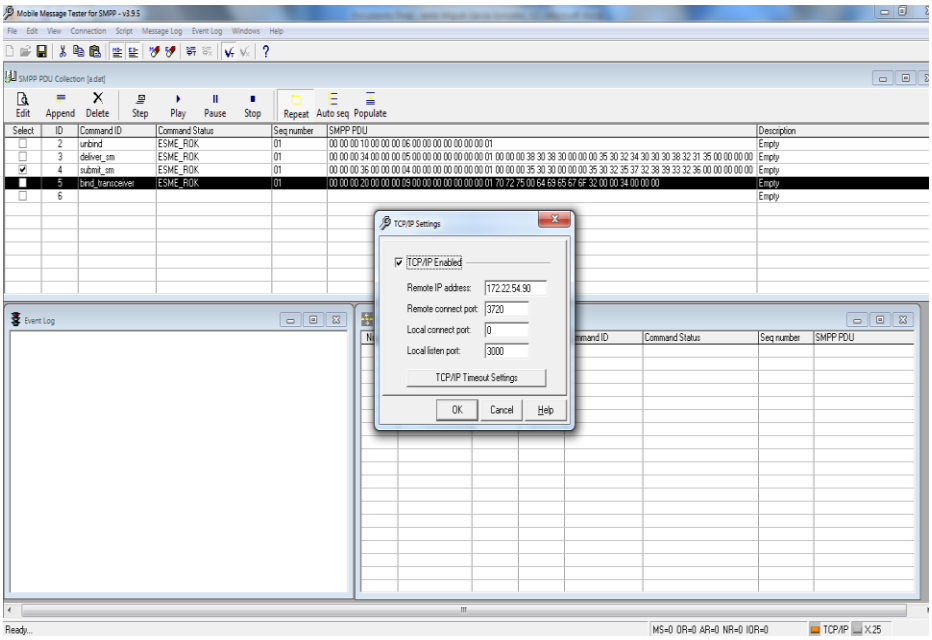
- QUERY_SM / QUERY_SM_RESP
- DELIVER_SM / DELIVER_SM_RESP
- BIND_TRANCEIVER / BIND_TRANCEIVER_RESP
- UNBIND / UNBIND_RES

Figura 29. **Vista de la aplicación SMPP Tester**



Fuente: elaboración propia. Printscreen the version de SMPP Tester.

Figura 30. **Vista de conexión TCP/IP de la aplicación SMPP Tester**



Fuente: elaboración propia, Printscreen the version de SMPP Tester.

Esta herramienta es útil para realizar las siguientes validaciones de prueba:

- Validar y simular la conectividad TCP/IP entre una ESME y un SMSC.

- Validar y simular la conectividad a nivel de tipo de sesión (Transmiter, Receiver, Transceiver) entre una ESME y un SMSC.

- Verificar que una ESME soporte o tenga configurado el carácter Latino ISO-8859 -1, donde el objetivo es que la ESME decodifique mensajes de texto con el estándar ISO-8859-1.

- Verificar que una ESME codifique mensajes de texto en UCS-2.

- Verificar que una ESME sea capaz de enviar mensajes concatenados a una unidad móvil de telefonía.

- Verificar que la ESME no pueda enviar mensajes de texto a teléfonos móviles que se encuentran en estado barred code en el SMSC. Se valida que la respuesta SMPP transmitida a la ESME se base en el estándar.

- Verificar que la ESME no tenga problemas al realizar una sesión de cerrado de sesión (*unbind*).

- Verificar que la ESME envía un *ENQUIRE LINK* cada 60 segundos después de establecer una conexión con un SMSC. Esta validación es importante para poder garantizar que la conexión no se vea interrumpida, crear algún script de validación para solicitar validar el aplicativo y realizar la conectividad nuevamente.

- Verificar que una operación de una ESME sea satisfactoria cuando el SMSC responda que el mensaje de texto no es posible enviar debido a que el cliente no cuenta con saldo suficiente para realizar el envío.

- Simular el envío de varios mensajes de texto a un SMS para realizar una prueba de estrés en un centro de mensajería.

CONCLUSIONES

1. El protocolo SMPP es un *open source* desarrollado con la intención de proveer la flexibilidad de una interface en las comunicaciones de datos para la transferencia de mensajes cortos entre ESME (entidad externa de mensajes cortos), RE (entidades de ruteo) y MC (centros de mensajes) utilizando para su interconectividad redes TCP/IP.

2. La tecnología que utilizan los centros de mensajería (ISMSC) utiliza el protocolo SMPP para comunicarse e intercambiar información entre usuarios y aplicaciones, las cuales pueden brindar cualquier tipo de información en tiempo real, brindando así beneficios a usuarios que deseen obtener información o inclusive realizar transacciones y pagos.

3. La versión más utilizada por los operadores es la versión SMPP 3.3 y 3.4. Estas versiones utilizan el modo *transceiver,* conexión que puede enviar y recibir información al mismo tiempo. La última versión SMPP disponible es la versión 5.0.

RECOMENDACIONES

1. Evaluación de los sistemas de comunicación que se utilizan en la actualidad, su versatilidad y adaptación a los nuevos protocolos y estándares establecidos bajo el protocolo SMPP.

2. Planificación y estructuración del diseño de red que se establecerá con la implementación del protocolo SMPP para aplicaciones de contenido.

3. Desarrollar y evaluar nuevas áreas de negocios, disponibles por la versatilidad del protocolo SMPP, incursionando en la generación, distribución y control de media sobre IP.

4. Generar de esta manera un cambio y ventaja en el mercado en los productos que se ofrecen y asegurando gracias a las características del protocolo SMPP e IP la calidad del servicio que se desee proveer.

BIBLIOGRAFÍA

1. CARR, Hugh; HEALY, Stephanie. *Short Message Peer to Peer (SMPP) Interface Specification.* [en línea] < http://opensmpp.org/specs/SMPP-IF-SPEC.v3.3.pdf > [Consulta: 20 de julio de 2012].

2. _____. *Short Message Peer to Peer (SMPP) Interface Specification Version 5.0* [en línea] < http://opensmpp.org/specs/smppv50.pdf > [Consulta: 20 de julio de 2012].

3. LAMMLE, Todd. *CCNA Cisco Certified Network Associate Study Guide*. 7th ed. New Jersey: Wiley, 2002. 864 p.

4. SENTH, Adev. *Sending SMS with SMPP, Kannel and Java*. [en línea] < http://mobiforge.com/developing/story/sending-sms-with-smpp-kannel-and-jav > [Consulta: 20 de julio de 2012].

5. SMPP DEVELOPMENT FORUM. *SMPP Protocol Specification v 3.4. Sending SMS with SMPP, Kannel and Java*. [en línea] < http://intetlab.com/docs/SMPP_v3_4_Issue_1_2.pdf > [Consulta: 22 de julio de 2012].

6. WEITZMANN, D. *A Standard for the Transmission of IP Datagrams on Avian Carriers* [en línea] < http:www.isi.edu/in-notes/rfc1149.txt > [Consulta: 10 de junio de 2012].