

Análisis e Implementación de un Servidor de Protocolo de Mensajería Escrita Punto a Punto (SMPP) versión 3.4 en Linux que interactúe con un Cliente SMPP en el envío y recepción de mensajes cortos (SMS's) y que genere archivos de registros de datos de llamadas, CDRs

César Macías ⁽¹⁾, Henry Vargas ⁽²⁾, José Menéndez ⁽³⁾
Facultad de Ingeniería en Electricidad y Computación ⁽¹⁾⁽²⁾⁽³⁾
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
cesar.alejandro.macias@gmail.com ⁽¹⁾, hdvargas@espol.edu.ec ⁽²⁾, jmenende@fiec.espol.edu.ec ⁽³⁾

Resumen

El presente proyecto está dirigido para cualquier estudiante o profesor de ingeniería en telecomunicaciones que desee entender el protocolo SMPP v3.4 de una forma didáctica, detallando los procesos principales, su funcionamiento y mostrando resultados puntuales. Se detalla el protocolo SMPP v3.4 desde el punto de vista del servidor, es decir una implementación limitada de una central de servicios de mensajes cortos (SMSC) y para realizar pruebas se interactúa con una entidad externa (ESME). Esta implementación se realiza en una máquina virtual VMware mediante lenguajes de programación: El Lenguaje Práctico de Extracción y de Informes (Practical Extraction and Report Language, Perl) para la implementación del protocolo SMPP y el lenguaje dinámico para Páginas Personales (Personal Home Page, PHP) para la aplicación web. La base de datos utilizada es MariaDB. Se detalla un análisis de los mensajes enviados entre ESME y SMSC observándose los resultados de toda la implementación limitada del SMSC mediante la herramienta Wireshark.

Palabras Claves: Protocolo, Servidor, Mensajes, SMPP, SMSC, ESME.

Abstract

The present project is directed to any student and professor in the field of Telecommunications Engineering who wishes to learn the SMPP v3.4 protocol in a didactic manner, detailing the principal processes, its functionality, and showing punctual results. The SMPP v3.4 protocol is detailed from a server's point of view, it is to say a limited implementation from centralized services of short messages (SMSC). To perform tests, it is interacted with an external entity (ESME). This implementation takes place in a virtual machine VMware through programming language: The Practical Extraction and Report Language (PERL) for the implementation of SMPP protocol and dynamic language for Personal Home Pages (PHP) for the Web application. The database used is MariaDB. An analysis of the messages sent between ESME and SMSC is detailed, observing the results of all the limited SMSC implementation through the Wireshark tool.

Keywords: Protocol, Server, Messages, SMPP, SMSC, ESME.

1. Introducción

El proyecto está basado principalmente en la explicación del funcionamiento del protocolo de Mensaje Corto Punto a Punto (Short Message Peer to Peer, SMPP) mediante la implementación de un Centro de Servicios de Mensajes Cortos (Short Message Service Center, SMSC) limitado, es decir, algo didáctico para un mejor entendimiento.

Se requiere de una Entidad Externa de Mensajes Cortos (External Short Message Entity, ESME) o un cliente SMPP para realizar las pruebas de los envíos y recepción de mensajes cortos. Sin embargo, este análisis se enfoca en el servidor SMSC y adicionalmente se ha realizado una simulación de un

sencillo ESME para probar la implementación del SMSC.

2. Objetivos

2.1 Objetivo Principal

El objetivo principal es dar una explicación didáctica del uso del protocolo SMPP como un servidor mediante la utilización de una plataforma SMSC limitada.

2.2 Objetivo Específico

Para que la explicación didáctica pueda cumplirse en este proyecto y para la demostración del

funcionamiento de los mensajes PDU del protocolo SMPP, se consideran los siguientes objetivos específicos:

- Describir el funcionamiento y operación del protocolo SMPP v3.4 de una forma sencilla para una comprensión rápida de cualquier estudiante o profesor de ingeniería en telecomunicaciones.
- Implementar una plataforma que actúe como Centro de Servicio de Mensajería Corta (SMSC) para la interacción con un ESME o cliente SMPP v3.4.

3. Justificación

La mensajería corta es una de las grandes fuentes de ingresos para las empresas telefónicas y entidades externas [1].

El protocolo SMPP sirve como solución para dar una buena administración del flujo de los mensajes enviados por entidades externas, tales como: proveedores de contenidos, notificaciones de la banca, noticias, entre otros.

4. Alcance y Limitaciones

El alcance de la implementación del SMSC está dado por la explicación del funcionamiento del protocolo SMPP empleándose sobre sistema operativo Ubuntu, una aplicación en Lenguaje Práctico de Extracción y de Informes (Practical Extraction and Report Language, PERL) y otra en lenguaje dinámico para Páginas Personales (Personal Home Page, PHP). Ambas aplicaciones están interconectadas mediante una base de datos MariaDB, que es una derivación de MySQL.

En las limitaciones encontramos principalmente lo siguiente:

- Solo se consideran los siguientes mensajes de Unidad de Datos del Protocolo (Protocol Data Unit, PDU) de SMPP: Transmitter, Receiver, Transceiver, Submit_sm, Deliver_sm, Unbind y Enquire_link.
- Los errores considerados por el ESME son los siguientes: no hay error, clave incorrecta y falla de estado de enlace (bind).

5. Metodología

Para la realización del presente trabajo se ha hecho una investigación descriptiva, mediante el cual la información obtenida proviene de fuentes de páginas web, ya que el Protocolo SMPP v3.4 es un protocolo abierto y estándar. Una herramienta fundamental para esta implementación es el uso de la librería "Net::SMPP" que se la puede descargar del sitio web <http://search.cpan.org/~sampo/Net-SMPP/SMPP.pm>.

Este recurso es muy útil y abarca la versión 3.4 del protocolo, en la que tiene funciones como por ejemplo: crear una sesión transmitter, receiver o transceiver. Para las pruebas del envío de los mensajes cortos se requiere de un aplicativo ESME llamado "SMSC client.NET", el cual puede ser descargado del sitio web <http://www.tops.com.pl/en/smscc/>. Los lenguajes de programación usados son: PERL y PHP. El lenguaje PERL para la administración de los mensajes PDU (implementación SMSC) y el lenguaje PHP para la parte visual de estos mensajes (aplicación web)

Con todas estas herramientas mencionadas y también recursos brindados por el tutor, son los métodos fundamentales para la realización de este proyecto, el cual tiene un fin didáctico y no busca cambiar ningún enunciado.

6. Descripción de SMPP

El protocolo de Mensajes Cortos Punto a Punto (Short Message Peer to Peer) más conocido como SMPP, es un protocolo abierto y estándar de las telecomunicaciones, para la comunicación de mensajes cortos fuera de la red móvil entre entidades, tales como, Entidades Externas de Mensajes Cortos (ESME) que se conectan con el Centro de Servicios de Mensajes Cortos (SMSC). El SMSC es el servidor y el ESME el cliente [2] [3].

Para que exista la comunicación, se la realiza mediante una conexión del Protocolo de Control de Transmisión/Protocolo de Internet (Transmission Control Protocol/Internet Protocol, TCP/IP) o similar, en cual el protocolo SMPP hace la función de intercambiar mensajes cortos entre ESME y SMSC con los llamados Unidad de Datos del Protocolo (PDU). Cada petición PDU que se envía de una entidad tiene su respectiva respuesta PDU [2] [3]. En la Figura 1 se muestra un bosquejo de lo explicado.

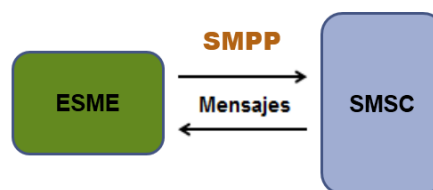


Figura 1. ESME, SMSC y Protocolo SMPP [4]

6.1 Sesiones SMPP

En la Figura 2 se muestra las tres formas en que el ESME inicia sesión y sus respectivos mensajes PDU básicos [2] [5]:

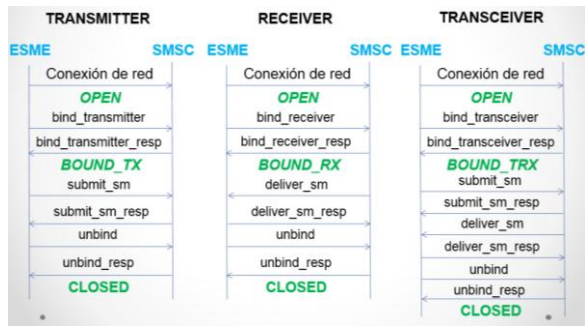


Figura 2. Sesión: Transmitter, Receiver, Transceiver

7. Implementación de un SMSC

Se requiere un ambiente de programación sobre un servidor Ubuntu 14.04. Este servidor es una máquina virtual instalada en una computadora.

El SMSC se implementó como un servicio (demonio) el cual realiza las funciones por medio del protocolo SMPP. Adicionalmente, se implementó una aplicación para la visualización de los mensajes SMPP y su administración. Ambas aplicaciones interactúan con la misma base de datos.

Los lenguajes de programación utilizados son: Lenguaje Práctico de Extracción y de Informes (Practical Extraction and Report Language, Perl) para la implementación del SMSC y el lenguaje dinámico para Páginas Personales (Personal Home Page, PHP) para la aplicación web que transmite y recibe los mensajes PDU. La base de datos utilizada es MariaDB.

7.1 Descripción General de la Implementación del SMSC

El sistema operativo Ubuntu se muestra en la Figura 3.

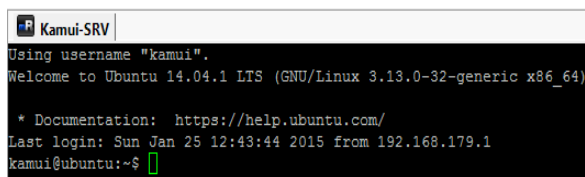


Figura 3. Ubuntu 14.04

La máquina virtual que se ejecuta sobre la plataforma de VMWare se muestra en la Figura 4 [6].

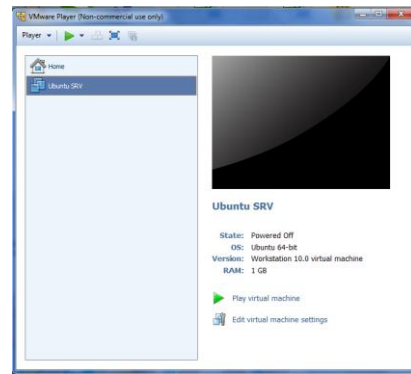


Figura 4. Máquina Virtual VMware

En la Figura 5 se presenta un gráfico en donde se tiene un ESME, un bloque de color rojo que representa el servicio SMSC, un bloque de color verde que representa la aplicación web y la base de datos MariaDB.

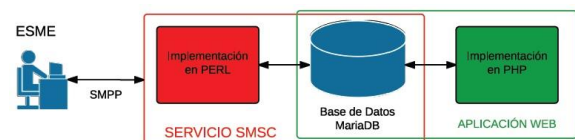


Figura 5. Descripción General del Proyecto

El ESME es cualquier cliente que se conecte a este SMSC vía SMPP. Se ha considerado el bloque del ESME en la Figura 5 con el fin de ilustrar la estructura de la comunicación entre ESME y SMSC.

El dominio total del SMSC lo tiene el servicio implementado en PERL, Figura 5. Para el efecto, se ha hecho uso de una librería en Perl para el manejo de los mensajes SMPP. Esta librería es Net::SMPP y se la encuentra en la página web de CPAN [7], donde se pueden encontrar alguna librerías de Perl para otros usos.

En la base de datos se guarda la información del número corto del ESME (short code), claves de ESME e interacción entre el programa de manejo del protocolo SMPP y el ambiente visual de gestión.

La decisión de qué ESME es el destinatario del SMS recibido y almacenado en el SMSC, se lo realiza por medio de la tabla *tb_sh_codes* en la base de datos, Figura 3.12.

id_sc	short_code	id_esme
1	2020	1
2	2080	2
3	5050	3
4	6060	4
5	4040	5
(Auto)	(NULL)	(NULL)

Figura 6. *tb_sh_codes*

Ahí se realiza una relación del número corto de destinatario al identificador numérico del ESME creado en la tabla *tb_esme*, Figura 7.

id_esme	system_id	password	Descripcion
1	abc	123	ESME Test1
2	abcd	1234	ESME Test2
3	telfonica	123456	
4	cnt	123456	CNT EP - Movil
5	claro	123456	Conecel - Claro
(Auto)	(NULL)	(NULL)	(NULL)

Figura 7. *tb_esme*

La aplicación web se implementó usando el lenguaje PHP, esta se conecta con la base de datos como se observa en el bloque de color verde en la Figura 5.

7.2 Funcionamiento del SMSC

El servicio implementado en Perl es la parte más importante del proyecto y se puede dividir en cuatro procesos o tres niveles detallados en la Figura 8.

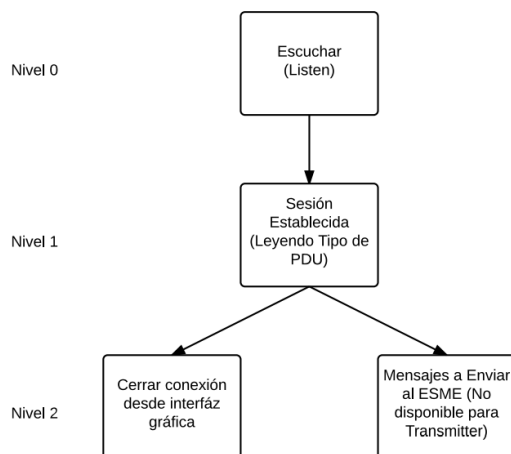


Figura 8. Diagrama del funcionamiento del SMSC

El proceso Escuchar (Listen) o también llamado Nivel 0, es el primer proceso en donde el SMSC está escuchando por el puerto 2251 esperando por una conexión de algún ESME esto se implementó haciendo uso de la librería para SMPP.

Se ha denominado también al proceso Escuchar como proceso Padre, porque en el momento en que un ESME hace conexión con el SMSC se derivan y crean más sub-procesos.

En el proceso Hijo o Nivel 1 ya se ha establecido una conexión con algún ESME específico. Este se encuentra leyendo el tipo de PDU, maneja el envío/recepción de mensajes en esa sesión y dependiendo del tipo de conexión del ESME crea hasta dos procesos.

Estos dos procesos están en el Nivel 2 y se detallan a continuación:

- Proceso Cerrar Conexión: Maneja la desconexión del ESME por medio del cambio de un estado en la base de datos, cambio que es realizado en la aplicación de administración.
- Envío de Mensajes: Maneja el envío de mensajes al ESME conectado como tipo receiver o transceiver. Estos mensajes han sido enviados por otro ESME ya sea previamente o durante la conexión del ESME destinatario.

Si el ESME se conecta como un transmitter, solo se desglosa el proceso de desconexiones. La razón se da porque el ESME como transmitter envía mensajes al SMSC y no las recibe. Por otro lado, si el ESME se conecta como un receiver o transceiver, se desglosa los dos procesos que están en el Nivel 2.

8. Análisis de Resultados

8.1. Análisis General

Para un análisis de resultados sobre la implementación del SMSC y su buen funcionamiento mediante el protocolo SMPP, se utiliza la herramienta Wireshark, para analizar el tráfico del protocolo SMPP.

Para el presente trabajo, el análisis se realizó con un ESME de prueba configurado en una computadora con dirección ip 192.168.179.1 y un SMSC implementado en una maquina virtual VMware instalado en la misma computadora donde se encuentra el ESME de prueba pero que tiene dirección ip 192.168.179.131.

8.2. Análisis de Conexión del ESME hacia el SMSC.

En la Figura 9 observamos que el ESME ha solicitado un inicio de sesión de tipo *bind_receiver* en el paquete 26 y su respuesta consecuentemente es un *bind_receiver_resp* en el paquete 28 enviado por el SMSC.

No.	Time	Source	Destination	Protocol	Length	Info
26	63.3462620	192.168.179.1	192.168.179.131	SMPP	85	SMPP Bind_receiver
28	63.3585650	192.168.179.131	192.168.179.1	SMPP	75	SMPP Bind_receiver - resp: "ok"
33	73.3469620	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
34	73.3572820	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"
36	83.3468130	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
37	83.3813260	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"
39	93.3473970	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
40	93.3553920	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"

Figura 9. Proyecto sesión modo *bind_receiver*

Luego podemos observar en la Figura 10, que el ESME ahora genera una conexión para hacer un inicio

de sesión como bind_transmitter en el paquete 56 y su respectiva respuesta bind_transmitter_resp está dada en el paquete 58 que emite el SMSC, lo cual indica un ESME como transmitter exitoso. Ahora en la misma Figura 10, en el paquete 63 se muestra un mensaje PDU de tipo submit_sm por parte del ESME. El SMSC recibe este mensaje submit_sm y le responde con un submit_sm_resp con el paquete 65. Este SMSC se encarga ahora de enviar el mensaje al mismo ESME mencionado (así se configuró para esta prueba) mediante el mensaje PDU deliver_sm con el paquete 67 y su respectiva respuesta deliver_sm_resp por parte del ESME es dada por el paquete 68. Como aún no se cierra la conexión, el ESME sigue verificando que aún haya dicha conexión con enquire_link en el paquete 70.

No.	Time	Source	Destination	Protocol	Length	Info
53	113.347427	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
54	113.355160	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"
56	113.719604	192.168.179.1	192.168.179.131	SMPP	83	SMPP Bind_transmitter
58	113.732292	192.168.179.131	192.168.179.1	SMPP	75	SMPP Bind_transmitter - resp: "ok"
63	116.927408	192.168.179.1	192.168.179.131	SMPP	107	SMPP Submit_sm
65	116.978057	192.168.179.131	192.168.179.1	SMPP	81	SMPP Submit_sm - resp: "ok"
67	117.477142	192.168.179.131	192.168.179.1	SMPP	107	SMPP Deliver_sm
68	117.593242	192.168.179.1	192.168.179.131	SMPP	71	SMPP Deliver_sm - resp: "ok"
70	126.927801	192.168.179.1	192.168.179.131	SMPP	70	SMPP Enquire_link
72	126.966278	192.168.179.131	192.168.179.1	SMPP	70	SMPP Enquire_link - resp: "ok"

Figura 10. Proyecto sesión modo bind_transmitter

8.3. Resumen de mensajes PDU del presente proyecto.

Como un breve resumen de lo detallado en este análisis, se hizo un diagrama explicativo de lo revisado en todo este subcapítulo, Figura 11.

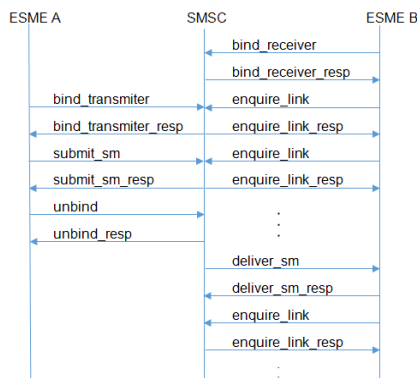


Figura 11. Resumen de mensajes PDU del proyecto

9. Conclusiones

1. El objetivo principal en la explicación del protocolo SMPP como un servidor (SMSC) se ha detallado de una forma entendible y rápida. Su explicación didáctica proporciona las herramientas básicas que ayudarán al entendimiento del protocolo SMPP. Este proyecto es útil para estudiantes de ingeniería relacionados en el campo de las

telecomunicaciones e interesados en el desarrollo de sistemas de envío y recepción de mensajes de texto por medio del protocolo SMPP.

2. Este trabajo sólo tiene un alcance didáctico y no comercial, debido a que el SMSC es parte de los equipos dentro de una operadora de telefonía celular. Para un completo funcionamiento requiere el manejo de otros protocolos de SMS para la recepción de los mensajes internos a la operadora, donde este trabajo solo se enfoca al protocolo SMPP y no cubre lo antes indicado.
3. La implementación de protocolos de comunicación con un enfoque didáctico, es viable usando lenguajes de programación que tengan una gran comunidad como soporte, como se lo realizó en este trabajo usando PERL y la librería Net::SMPP.
4. La herramienta WireShark, es de gran apoyo y utilidad para el entendimiento y análisis de tramas IP, esencial en la docencia para que los estudiantes visualicen los protocolos de comunicación revisados en clases. En el ámbito laboral, es de suma ayuda para el análisis de problemas más complejos donde se requiere revisar a nivel de paquetes el problema. Para el presente proyecto sirvió mucho en el análisis de los mensajes PDU y además demostrar que realmente se está usando el protocolo SMPP.

10. Referencias

- [1] lasprovincias.es, Acision es reconocido como el número uno del mundo en servicio de mensajería, <http://www.lasprovincias.es/20100209/mas-actualidad/tecnologia/acision-reconocido-como-numero-201002091906.html>, fecha de consulta noviembre 2014.
- [2] Protocolo SMPP, http://infotelecommil.webcindario.com/manuales/SMPP_Intro.pdf, fecha de consulta noviembre 2014.
- [3] SMPP Developers Forum, Short Message Peer to Peer Protocol Specification v3.4, <http://www.smstrade.de/pdf/smpp.pdf>, fecha de consulta noviembre 2014.
- [4] Nuruddin Ashr, SMPP, <http://es.slideshare.net/uudashr/smpp-presentation>, fecha de consulta diciembre 2014.
- [5] ActiveXperts software, SMPP SessionDescription, http://www.activeexperts.com/sms-component/smpp-specifications/overview/#smpp_session_description, fecha de consulta diciembre 2014.
- [6] Sagástegui Lescano Walter, Máquinas virtuales (VMWare, Virtual PC, Sandbox). Qué son y para que sirven. (DV00402A),

http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=139,
fecha de consulta febrero 2015.

- [7] CPAN, Net::SMPP,
<http://search.cpan.org/~sampo/Net-SMPP/SMPP.pm>, fecha de consulta enero 2015.