

PROYECTO DE FIN DE CARRERA
DOCUMENTACIÓN

AST
Application Service Tester

Sofía Depons, Marcelo Fiori,
Julio Fitipaldo, Sergio Martínez

Tutor: Ing. Fernando Fontán

30 de septiembre de 2008

Facultad de Ingeniería
Universidad de la República
Uruguay

Agradecimientos

En esta sección queremos reconocer la ayuda de personas e instituciones que aportaron en el desarrollo de este proyecto.

En primer lugar queremos agradecer particularmente el apoyo de nuestro tutor Fernando Fontán, quien desde el principio nos incentivó y guió, aportando especialmente la crítica constructiva.

Agradecemos a ANTEL, quien a través del LATTIA nos brindó el apoyo logístico necesario, y a todas las personas que dentro de ella han permitido que el proyecto saliera adelante: Eduardo Spremolla, Stella Bonino, Daniel Cabral, y especialmente a Manuel Mariño, quien nos apoyó en la parte operativa dentro del LATTIA.

Al mismo tiempo, queremos hacer extensivo el reconocimiento a Fabián Croce y a Cecilia Parodi, quienes aportaron con discusiones, sugerencias y correcciones.

Por otro lado, como este trabajo simboliza el final de todo un proceso, no queremos perder la oportunidad de poder expresar nuestra gratitud con nuestras familias y la Facultad en general, que nos han apoyado a lo largo del mismo.

Tabla de Contenidos

Tabla de Contenidos	v
I Descripción	1
1. Introducción	3
1.1. Motivación	3
1.2. Acercamiento al t3pico	4
1.3. Organizaci3n del documento	6
2. Descripci3n del proyecto	7
2.1. Objetivos	7
2.2. Actores que participan en el proyecto	8
2.3. Restricciones	8
2.4. Alcance del proyecto	9
2.5. Criterios de 3xito	9
3. Descripci3n primaria del sistema	11
3.1. Arquitectura general	11
3.2. Mediciones de calidad	16
II Base Te3rica	19
4. Modelado de usuarios	21
4.1. Definiciones b3sicas	22
4.2. Algunos modelos posibles	24
4.3. El modelo propuesto	29
5. Calidad de Servicio (QoS)	33
5.1. Indicadores de QoS	34
5.2. ¿Qu3 mide el AST?	36
5.3. Regulaci3n en Uruguay	39
6. Calidad de Experiencia (QoE)	41
6.1. Introducci3n	41
6.2. QoE-QoS en IMS	42
6.3. Implementaci3n en el AST	45

III	Diseño e implementación	47
7.	Diseño	49
7.1.	Mensajería	50
7.2.	Módulo de procesamiento de datos	53
7.3.	Módulo SNMP	55
7.4.	GUI	55
7.5.	Base de datos	57
8.	Implementación	59
8.1.	Aplicación en Java	59
8.2.	Interfaz Gráfica de Usuario (GUI)	65
8.3.	Base de Datos	65
8.4.	Proxy SMPP-HTTP	66
9.	Descripción del Hardware	67
9.1.	Estructura general del sistema	67
9.2.	Características necesarias de los móviles	67
9.3.	Requerimientos necesarios	69
IV	Análisis de resultados y Conclusiones	71
10.	Análisis de resultados	73
10.1.	QoS	73
10.2.	QoE	79
10.3.	Vulnerabilidades	82
11.	Conclusiones	83
	Bibliografía	85
V	Apéndices	89
A.	Introducción a la tecnología GSM	91
A.1.	Arquitectura de una red GSM	91
A.2.	Protocolos e interfaces	94
A.3.	Proceso de envío de SMS	95
B.	Manual del usuario	97
B.1.	Inicio	98
B.2.	Configuración	98
B.3.	Estadísticas	104
C.	Gestión del proyecto	107
C.1.	Riesgos	107
C.2.	Tiempo	108
C.3.	Costos	109
C.4.	Evaluación de la gestión	110

TABLA DE CONTENIDOS

Tabla de Contenidos

D. Contenido del CD	111
Glosario	115

Resumen

En este trabajo se presenta, en primer lugar, un estudio sobre las necesidades de un operador de una red de telefonía celular en cuanto al monitoreo del servicio de mensajería corta; en particular, contar con información sobre el desempeño global de la red, y cómo es ésta percibida por los usuarios.

En segundo lugar, se presenta el análisis de una solución, detallando los conceptos teóricos utilizados y las consideraciones generales del diseño. En este aspecto, el aporte fundamental consiste en la propuesta de un modelo probabilístico original para el comportamiento de los usuarios, a efectos de simular la experiencia de los mismos.

Finalmente, se realizó una implementación completa de la solución, que permite realizar mediciones de calidad *end-to-end* y su incorporación a los sistemas de monitoreo del operador, generando alarmas en situaciones de mal desempeño. El sistema implementado se incorporó a la red de ANTEL, obteniendo buenos resultados, los cuales son analizados al final de este documento.

Palabras clave: SMS, QoE, QoS, Application Service Tester, GSM.

Parte I

Descripción

Capítulo 1

Introducción

En este primer capítulo describimos el problema y lo ponemos en contexto. Luego realizamos una descripción a grandes rasgos de cuáles son los aspectos de dicho problema que ataca este proyecto.

1.1. Motivación

Las redes de telecomunicaciones móviles hoy están ampliamente extendidas y son el soporte de múltiples servicios de telecomunicaciones. Además del servicio de telefonía móvil, que es el principal servicio de las telecomunicaciones celulares, nuevos servicios de comunicación de datos han sido adoptados en forma masiva. En particular, el servicio de mensajería corta (SMS) se ha convertido en el principal medio de comunicación entre personas y el segundo generador de ingresos para los operadores móviles.

El SMS se desarrolla en dos modalidades, entre pares (*customer to customer* o *peer to peer*) y entre aplicaciones y consumidores (*customer to application* y *application to customer*). Si bien la más relevante en volumen de utilización es la primera, la segunda representa una parte importante de la facturación y los clientes son muy sensibles a su calidad. Las dos modalidades involucran diferentes plataformas de telecomunicaciones, siendo bastante complejo el soporte de la segunda modalidad.

Para un operador resultaría extremadamente valioso conocer en tiempo real cómo perciben sus clientes la calidad de los servicios que se le brindan. En el caso de SMS la percepción de calidad de extremo a extremo es prácticamente imposible de obtener a partir de la observación de cada plataforma individual que soporta el servicio.

Un autómata que replica el comportamiento de un consumidor, utilizando la red de servicios y observando los parámetros de calidad que le caracterizan, puede brindar información muy valiosa sobre la percepción de uso del servicio, favoreciendo la detección de problemas de desempeño antes de que se acumulen reclamos de los clientes. Esta función es precisamente la de un *service tester*, y se complementa con las alarmas que puedan generar las diferentes plataformas involucradas en el soporte de cada servicio.

1.2. Acercamiento al t3pico

De los p3rrafos que siguen a continuaci3n y del ap3ndice A¹ se desprende que son varios los lugares de la red del operador en los cuales pueden ocurrir problemas. Actualmente la detecci3n de fallas y problemas en el servicio se limita al monitoreo de fallas de los distintos elementos de manera independiente. Esto en general es insuficiente, y constituye por lo tanto un problema para el operador.

En la b3squeda de soluciones existentes en el mercado se constat3 que las mismas son escasas y en general forman parte de paquetes de mayor porte.

La soluci3n presentada por el grupo de proyecto ataca dos necesidades de la red GSM del operador: en primer lugar, la detecci3n de fallas y problemas de configuraci3n que afecten el servicio, y en segundo lugar, la falta de informaci3n acerca de la calidad de servicio otorgada al cliente final y la calidad de experiencia que 3ste percibe.

Muchas veces se hace dif3cil detectar las consecuencias (tanto qu3 afectaci3n como el grado de la misma) que trae una falla o un error humano en la configuraci3n de un equipo. Esto se hace a3n m3s dif3cil cuando el servicio involucra distintas 3reas del operador, que pueden significar distintas redes, con distintos equipos, y especialmente con distintos administradores. La complejidad de actores que aqu3 describimos puede generar tiempos altos de respuesta a fallas en el servicio. Para evitar este problema, la soluci3n que planteamos prueba el servicio de punta a punta, soslayando la complejidad de la red atraves3ndola horizontalmente (figura 1.1).

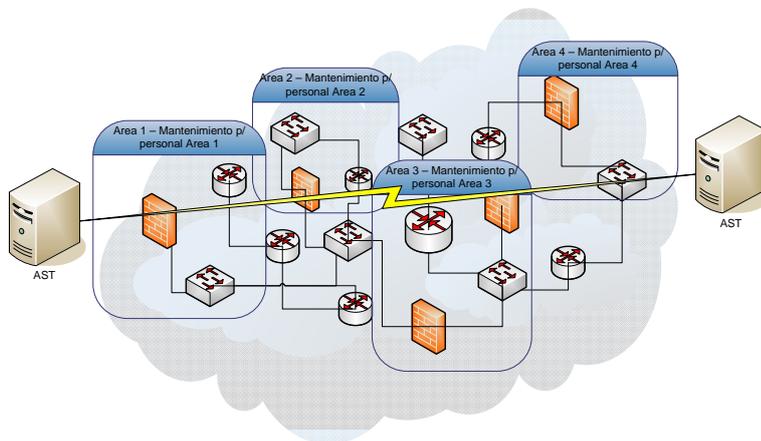


Figura 1.1: Complejidad de la red GSM, con distintas 3reas (con distintas redes, distintos equipos, y con distintos administradores)

De esta forma podemos saber cu3ndo el servicio es afectado por una falla en tiempos mucho menores que los habituales. Ser3 responsabilidad del operador encontrar el lugar exacto de la falla dado que las pruebas realizadas s3lo proveen informaci3n parcial de ubicaci3n. Es com3n que el operador no perciba la falla a trav3s de la plataforma de monitoreo, sino a trav3s de la plataforma de atenci3n al cliente. Con este sistema

¹Referimos al lector a este ap3ndice a efectos de repasar conceptos de la tecnolog3a GSM.

simulamos a los propios clientes por lo que sabremos de su descontento antes de que 3stos reclamen.

Para ello es importante que nuestro sistema pueda integrarse a la plataforma ya existente de monitoreo, incorpor3ndose de forma r3pida, sencilla y econ3mica a los procesos del operador. Es por esto que se desarrollaron dos interfaces de gesti3n del sistema, una SNMP y otra WEB. M3s adelante profundizamos en este tema.

Por otro lado, para resolver el problema de la falta de informaci3n sobre la calidad de servicio² otorgada, y la calidad de experiencia³ de los usuarios, se prueba el servicio de punta a punta como ya comentamos, y se registran las medidas. Sobre estos datos se construyen dos m3tricas, la calidad de servicio (que resulta casi directa de las mediciones) y la calidad de experiencia (que se calcula a partir de un buen modelo de comportamiento de los usuarios).

Esta informaci3n permite al operador tomar decisiones de planificaci3n con mucha mayor precisi3n. Existe una realimentaci3n entre las decisiones tomadas, y la calidad de experiencia de los usuarios; conocer esta 3ltima permite evaluar a niveles m3s altos los impactos de dichas decisiones.

Adem3s, se pueden ingresar distintos perfiles de usuario al sistema, pudiendo reconocer la calidad percibida de los distintos tipos de clientes, cada uno con su perfil de uso, y con sus requerimientos de calidad. Esto permite una pol3tica de comercializaci3n y marketing consciente de la calidad que cada segmento de mercado est3 percibiendo.

Vale destacar que las mediciones que realizamos no abarcan la red de acceso del operador. Se buscan problemas en el NSS (Network Switching System), que afectan a toda la red. Si bien uno de los puntos de conexi3n con la misma es a trav3s de la red de acceso est3ndar (utilizando un m3vil, conectado a trav3s de una radiobase), no intentamos con ello relevar los problemas que aqu3 ocurren. De hecho, para las pruebas y mediciones realizadas, se busc3 un punto sin problemas de propagaci3n, y sin saturaci3n de los canales de se3alizacion de las interfaces *Um*, *A* y *Abis*, para que las mediciones realizadas no fueran afectadas por estos factores. El objetivo de la medici3n realizada es detectar problemas a nivel de aplicaci3n y *core* de la red.

Si bien en principio el uso primordial que se le podr3a dar a este sistema est3 orientado al relevamiento por el operador mismo, otros actores pueden encontrar interesante conocer la calidad de servicio y experiencia del servicio. Por ejemplo:

- Reguladores. Como vemos m3s adelante, mundialmente hay una tendencia a la regulaci3n de la calidad de servicio para los servicios de telecomunicaciones; este sistema puede llegar a ser una herramienta muy 3til a la hora de evaluar y regular la calidad de los distintos operadores del mercado.
- Clientes, Proveedores de servicios de valor agregado. Puede ser 3til para clientes que tienen un contrato con SLA (Service Level Agreement), poder relevar la calidad de servicio que el operador est3 brindando. De esta forma puede hacer los reclamos y cobrar las indemnizaciones acordadas por contrato.
- Operadores de la competencia. Puede ser interesante para la competencia conocer la calidad de servicio otorgada por los competidores y la suya propia. De esta manera puede posicionarse con respecto a los dem3s.

²Un conjunto de medidas objetivas sobre el desempe3o de la red.

³Un conjunto de medidas subjetivas sobre la percepci3n del usuario respecto del servicio.

1.3. Organización del documento

El documento está compuesto por cuatro partes que enumeramos a continuación.

Parte I

La primera parte es introductoria, y consta de los siguientes capítulos:

Capítulo 1. Se describe el problema en un contexto general y algunos lineamientos de la solución.

Capítulo 2. Se incluyen los objetivos particulares del proyecto, aspectos organizativos, qué hace y qué no hace el sistema.

Capítulo 3. Se detalla la arquitectura y el funcionamiento del sistema, visto externamente. Se explican las funcionalidades del sistema, pero no su diseño interno ni implementación.

Parte II

La segunda parte es mayoritariamente teórica, y sienta las bases para las siguientes partes. Está conformada por los capítulos:

Capítulo 4. Se definen algunos conceptos básicos de probabilidad y se ejemplifican algunos procesos estocásticos ampliamente utilizados para modelar arribos. Finalmente, se expone el modelo propuesto para modelar el comportamiento de los usuarios.

Capítulo 5. Se introduce al concepto de calidad de servicio (QoS) en general, y la calidad del servicio de mensajería corta en particular.

Capítulo 6. Se presenta la noción de calidad de experiencia (QoE), y su interpretación y aplicación al caso de SMS.

Parte III

La tercera parte se centra en el software en sí, tanto en su diseño como en la implementación.

Capítulo 7. Se describe la partición del problema en módulos, y el diseño de cada uno de estos bloques.

Capítulo 8. Se detallan algunos aspectos de la implementación de los módulos y sus interacciones.

Capítulo 9. Se enuncian los requerimientos y limitaciones de hardware.

Parte IV

Capítulo 10. Se analizan los resultados de las mediciones obtenidas en el período de pruebas.

Capítulo 11. Se presentan las conclusiones generales del proyecto.

Finalmente hay una sección de apéndices donde se encuentra un capítulo acerca de la tecnología GSM, un Manual de Usuario, aspectos organizativos del grupo y un glosario.

Capítulo 2

Descripción del proyecto

2.1. Objetivos

El objetivo del proyecto es proporcionar un sistema completo y autónomo que brinde una visión del desempeño de extremo a extremo del servicio de mensajería corta (SMS).

En particular, los objetivos generales del mismo son:

1. Lograr un nivel de conocimiento del problema y del estado del arte, suficientes como para realizar un análisis del problema y de sus posibles soluciones, comprendiendo una especificación de la solución elegida. En particular:
 - Estudiar el funcionamiento general de la red GSM, y particularmente los equipos, interconexiones y sus problemas
 - Estudiar las aplicaciones y elementos de la red que intervienen en el servicio de Mensajería particularmente
 - Estudiar protocolos para el servicio de mensajería corta utilizados actualmente en la industria
 - Estudiar el estándar utilizado para la comunicación entre computadores y móviles celulares GSM
 - Estudiar parámetros específicos de calidad de servicio y calidad de experiencia en redes de mensajería
 - Estudiar herramientas estadísticas utilizadas para modelar comportamientos de usuarios en redes de telecomunicaciones
 - Generar un modelo del comportamiento de usuarios en servicios SMS, para poder analizar la percepción del servicio, pero también para segmentar la población con otros fines
2. Diseñar e implementar un sistema que sea capaz de:
 - Medir en tiempo real la calidad de servicio (QoS)
 - Medir en tiempo real la calidad de experiencia de los usuarios (QoE)
 - Generar alarmas ante el incumplimiento de ciertos niveles de QoS y de QoE
 - Generar análisis estadísticos de ambos niveles
 - Obtener un producto terminado en los plazos estipulados

Es deseable además que dicho sistema cumpla también con una serie de objetivos más específicos, aunque menores, como ser:

- Facilidad en la incorporación a la plataforma de monitoreo existente del operador
- Uso intuitivo y sencillo para el usuario
- Gestión remota
- Realización con hardware de bajo costo, el sistema debe funcionar con cualquier móvil GSM que implemente interfaz AT
- La seguridad debe ser suficiente como para evitar acceso a extraños

2.2. Actores que participan en el proyecto

Existen 4 actores principales en el desarrollo del proyecto:

1. Integrantes del grupo de proyecto
2. Tutor: Ing. Fernando Fontán
3. ANTEL
4. Facultad de Ingeniería, Universidad de la República

Cada actor se involucra en el proyecto de distinta forma, el grupo de proyecto lleva adelante el mismo, mediante la guía del tutor. La Facultad de Ingeniería exige que se cumplan los plazos y requerimientos, haciendo un seguimiento del avance del proyecto desde el comienzo del mismo. ANTEL, como interesado en el proyecto, provee parte del hardware necesario y el acceso a su red.

2.3. Restricciones

Existen una serie de restricciones, impartidas principalmente por los actores mencionados en la sección anterior.

1. La Facultad de Ingeniería exige que el proyecto sea realizado en el período que va desde el mes de julio de 2007, al mes de setiembre de 2008. Esto restringe a un trabajo de una extensión menor a poco más de un año.
2. El proyecto debe ser realizado con muy bajos costos. Esto es uno de los objetivos del proyecto, que simplemente refleja la indisponibilidad de mayores recursos para realizarlo.
3. Utilización de hardware básico. En particular, uso de un PC normal y teléfonos móviles de uso estándar. Esta restricción está directamente relacionada con la anterior.
4. Restricciones de seguridad de ANTEL. ANTEL impone mediante la Gerencia de Seguridad de la Información ciertos requerimientos y limitaciones sobre cómo debemos interconectarnos con su red. Más adelante se plantean con profundidad estos requerimientos, y cómo son alcanzados.

5. Indisponibilidad de información acerca del comportamiento de los usuarios del servicio. Como veremos más adelante, es fundamental para realizar el modelado estadístico de los usuarios, contar con información acerca de cómo se comportan. Esta información no está disponible en el convenio firmado con ANTEL. Esta restricción se verá reflejada en el alcance del proyecto, aunque de todas maneras no compromete ninguno de los objetivos iniciales.

2.4. Alcance del proyecto

Se pretende alcanzar los objetivos especificados en este capítulo, con las siguientes limitaciones:

- Se especifica como parte del proyecto el modelo de usuario que se utiliza para medir la QoE, pero no se incorpora la información de perfiles de usuario. Esta información sólo se puede obtener analizando comportamientos de usuarios reales, información que no fue facilitada por ANTEL.
- Dicho modelo de comportamiento de usuarios es, naturalmente, paramétrico, y son éstos los parámetros que se deben completar con la información faltante comentada. No se estudia en este documento cómo se pueden obtener los mejores juegos de parámetros (aquellos que representen más fielmente el comportamiento real de los usuarios) a partir de la información de ANTEL. Es decir, no se dan especificaciones precisas sobre cómo obtener estos parámetros a partir de dichos datos, sino que se presentan sólo algunas ideas a tener en cuenta. Cabe observar que este aspecto no es trivial, y en todo caso requiere un estudio profundo.
- Si bien el sistema debe funcionar con una amplia variedad de terminales móviles (listados en la sección 9.2), no se pretende hacer esa lista extendida a todos los terminales existentes.

2.5. Criterios de éxito

El criterio de éxito fundamental es el cumplimiento de los objetivos explicitados en este capítulo, en tiempo y forma. Para asegurar que estos objetivos sean cumplidos, se realizaron una serie de ensayos, exigiendo que los resultados cumplan los criterios aquí mencionados.

2.5.1. Primer ensayo: Estabilidad y confiabilidad

Durante un período de 20 días, el sistema debe realizar las mediciones automáticamente, sin requerir la acción humana. Las mediciones no deben ser interrumpidas en ningún momento, y bajo ningún concepto se podrá perder información de las mismas. En particular:

1. Debe soportar cortes de energía, retomando las tareas y los datos al reintegrarse
2. Deben funcionar todas las interfaces de hardware durante todo el tiempo de prueba

De cualquier forma, de ser interrumpida la medición por algún factor externo al sistema, éste debe avisar a la plataforma de monitoreo, para que se realice la acción humana necesaria, de forma que el sistema pueda volver a estar en línea.

2.5.2. Segundo ensayo: Precisión y fiabilidad

Se deberán comparar los resultados y alarmas del AST en situaciones conocidas de la red. De esta forma se podrá verificar la precisión y fiabilidad del mismo. En particular, se deberán realizar las siguientes comparaciones:

1. Verificar que no existan falsas alarmas. En casos en que está probado el buen desempeño de la red, verificar que el sistema no detecta problemas. Esto es de suma importancia, dado que el disparo de falsas alarmas es inadmisibles para un sistema de este tipo.
2. Verificar que bajo la ocurrencia de fallas, el sistema perciba las mismas.

Capítulo 3

Descripción primaria del sistema

En la primera sección de este capítulo profundizamos en la arquitectura general del AST y en cuáles son los distintos actores que interactúan con él, pero sin detallar aspectos de diseño interno (que serán tratados en el capítulo 7), sino concentrándonos en una visión externa.

En la segunda sección, describimos la estructura lógica considerada para obtener valores de calidad del servicio y de calidad de experiencia .

3.1. Arquitectura general

Como fuera comentado en los capítulos anteriores, la propuesta consiste en un autómata que se dedica a enviar y recibir mensajes. Para ello, naturalmente, el sistema debe estar conectado a una red que brinde el servicio de mensajería corta.

Específicamente, el sistema tiene conectividad mediante móviles celulares y mediante una conexión sobre TCP/IP, que como veremos más adelante, abre la posibilidad de conectarse a la red GSM de varias formas. El diagrama 3.1 ilustra esta idea, mostrando de forma simplificada la interacción del AST con estos actores.

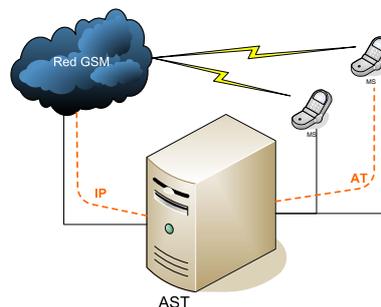


Figura 3.1: Esquema simplificado de la arquitectura

Profundizamos ahora en estas interfaces y en otros actores que interactúan con el sistema.

Terminales Móviles

El AST puede conectarse prácticamente a un número ilimitado¹ de terminales móviles GSM, utilizando el protocolo *AT* para comunicarse con cada uno de ellos. Dado que nos concentramos exclusivamente en el curso interno de los mensajes y no en la red de acceso, el hecho de tener los terminales móviles en el mismo lugar físico no influye en absoluto sobre los resultados buscados.

Los móviles (formalmente las SIM) no tienen por qué pertenecer a un operador en especial, lo que permite realizar mediciones del servicio entre operadores, inclusive sin su previa autorización, dado que simplemente estaremos utilizando el servicio como un cliente más para realizar las mediciones.

Interfaz IP

Por otro lado, el AST utiliza el protocolo SMPP (Short Message Peer-to-Peer Protocol) para las conexiones sobre TCP/IP. Este protocolo es un estándar de la industria y es el que nos permite conectarnos con el SMS-C.

Nuestro sistema se conecta a dos puntos importantes de la red a través de conexiones sobre TCP/IP, en primer lugar al SMS-C, y en segundo lugar al SMS-Gateway. Este último es quien recibe las conexiones de los proveedores de servicios de valor agregado (el sistema puede simular entonces ser un proveedor de servicios).

De esta manera, podremos monitorear el servicio de mensajería entre varios puntos de la red, ya sean terminales o internos a la misma.

Por lo tanto no sólo podemos detectar fallas si no que podemos identificar en qué sección de la red se encuentra y qué servicio se ve afectado.

En resumen, en la implementación actual del AST en ANTEL, el sistema se conecta a la red a través de tres puntos: terminales móviles, el SMS-C y el SMS-Gateway. En la figura 3.2 se ilustra este esquema.

¹Salvo por restricciones de hardware que comentamos en el capítulo 9

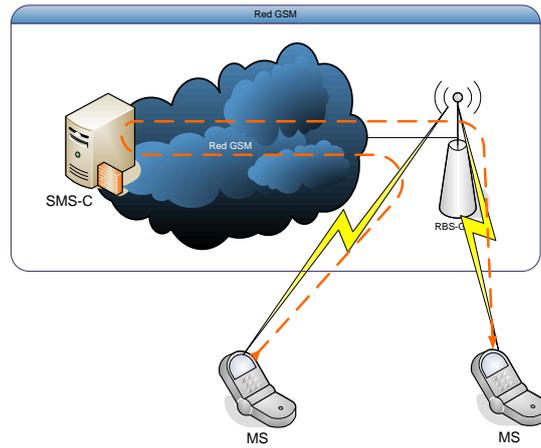


Figura 3.3: Camino 1: MS - MS

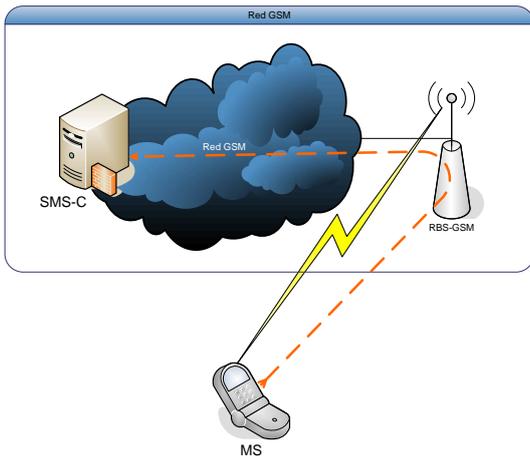


Figura 3.4: Camino 2: MS - SMS-C

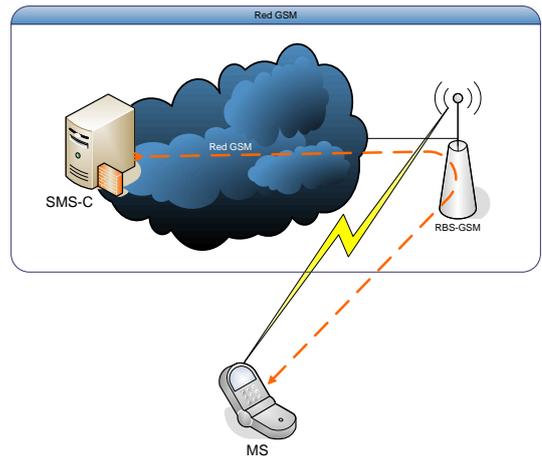


Figura 3.5: Camino 3: SMS-C - MS

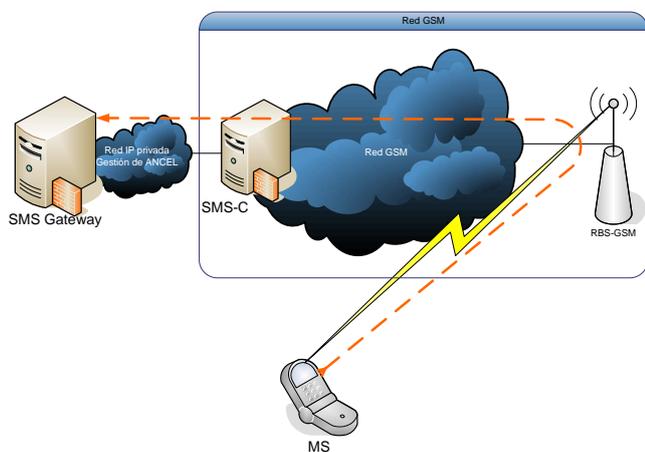


Figura 3.6: Camino 4: MS - SMS-GW

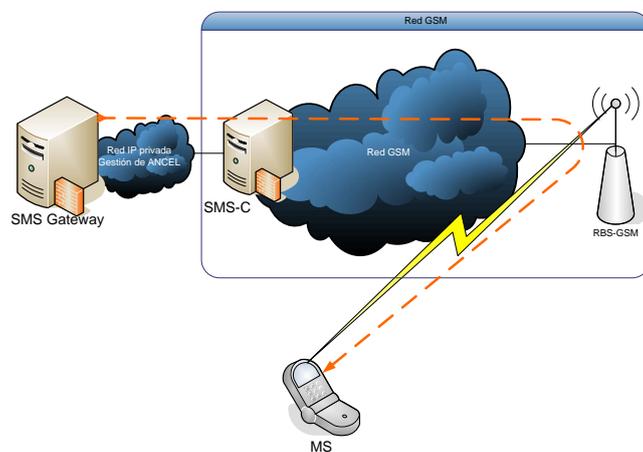


Figura 3.7: Camino 5: SMS-GW - MS

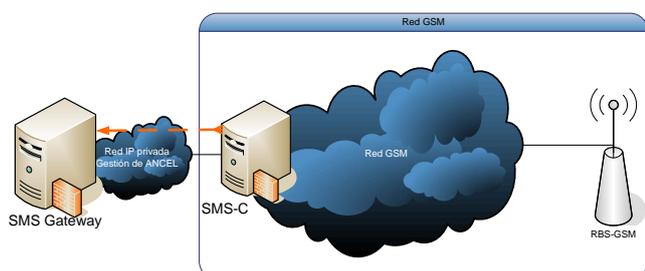


Figura 3.8: Camino 6: SMS-C - SMS-GW

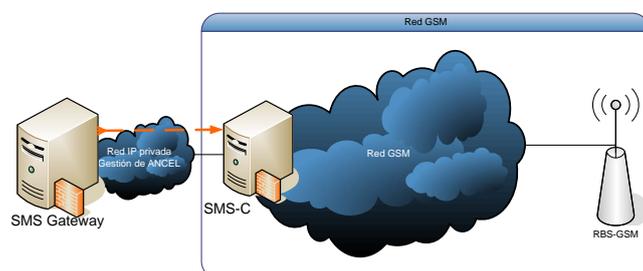


Figura 3.9: Camino 7: SMS-GW - SMS-C

Describimos a continuación las interfaces que permiten al usuario configurar el sistema y recibir alarmas y estadísticas del funcionamiento del mismo.

Interfaz de Usuario

Identificamos dos funcionalidades deseables en cuanto a la interacción del sistema con los usuarios del mismo: una interfaz que permita configurar los parámetros y actividades, y una segunda interfaz para que el sistema pueda enviar alarmas ante incumplimientos en el desempeño de la red.

Para la primera de las mencionadas, implementamos una interfaz web, de manera que se pueda configurar remotamente.

Por otro lado, utilizamos el estándar SNMP para el manejo de alarmas y otro tipo de consultas.

Finalmente, en la figura 3.10 se puede ver un diagrama completo de la red, que contiene todos los elementos mencionados en este capítulo con sus respectivas interacciones.

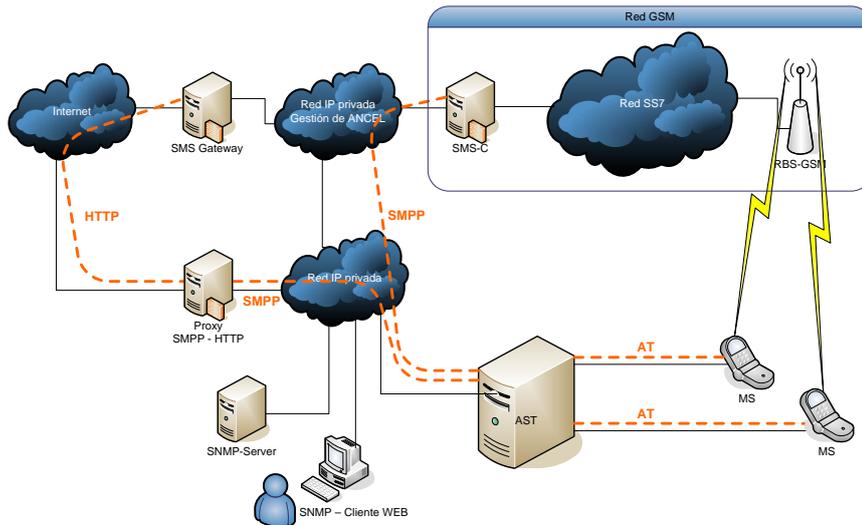


Figura 3.10: Esquema completo de la arquitectura

3.2. Mediciones de calidad

En esta sección presentamos la estructura lógica para extraer valores de calidad de servicio y calidad de experiencia, haciendo uso de la arquitectura presentada.

El planteo original de las funcionalidades del sistema, tal como fue propuesto al inicio, consistía en obtener exclusivamente la calidad percibida por los usuarios. Con ese objetivo, la solución más acorde era enviar mensajes (utilizando las interfaces descritas en la sección anterior) en aquellos momentos en los cuales un usuario real lo haría. Para esto es necesario contar con un modelo para simular el comportamiento de los usuarios. Luego, el resultado de la calidad percibida se obtendría teniendo en cuenta el desempeño de la red en los mensajes enviados.

Como alternativa, otra solución considerada fue enviar mensajes periódicamente mediante las interfaces físicas recién mencionadas, realizar también un modelo de comportamiento de usuarios como en el caso anterior, y luego superponer la información sobre la simulación de los usuarios. Específicamente, dada una simulación de un usuario, se toman los instantes correspondientes a los envíos de mensajes y se busca el mensaje real más cercano en el momento de envío. Nuevamente la calidad percibida se obtiene a través del desempeño de la red en estos mensajes.

Este segundo enfoque fue el elegido, debido principalmente a dos consideraciones importantes:

- Se logra una mayor independencia entre la simulación de usuarios y el envío físico de mensajes.

- Se obtiene además un muestreo periódico del comportamiento de la red, lo que permite extraer valores objetivos de calidad de servicio.

Detallamos en lo que sigue algunos aspectos sobre este enfoque.

Cada usuario podrá estar asociado a cierto tipo de utilización del servicio. Por ejemplo, se puede segmentar la población según el grado de utilización (usuarios muy activos, muy pasivos, etc), o también según el destino de los mensajes (mayoritariamente mensajes del mismo proveedor de servicios, hacia otro proveedor, o hacia aplicaciones de valor agregado²). Esta última clasificación no es particular de los usuarios sino que es en realidad más general, y por ello creamos el concepto de *modo*, que refiere a una asociación origen-destino (por ejemplo ANCEL-ANCEL, ANCEL-Movistar, ANCEL-SMS-GW, etc).

Tenemos entonces un bloque que se ocupará de enviar mensajes cada cierto intervalo de tiempo, para cada uno de los modos de utilización configurados, y medirá el desempeño de dichos mensajes. Estos datos se irán guardando en una base de datos.

Con esta información, podemos obtener directamente mediciones de la calidad de servicio de la red, ya sea de punta a punta o entre cualquier par de interfaces a las cuales se puede conectar el sistema.

De manera independiente a lo recién expuesto, pretendemos encontrar un modelo que represente el comportamiento de los distintos tipos de usuarios (desarrollo que se presenta en el capítulo 4). Combinando la información del muestreo de la red con las simulaciones de este modelo de usuarios, podemos extraer un valor sobre la calidad de experiencia.

El diagrama de la figura 3.11 muestra en qué se fundan la calidad de servicio y de experiencia, sintetizando lo expuesto en esta sección.

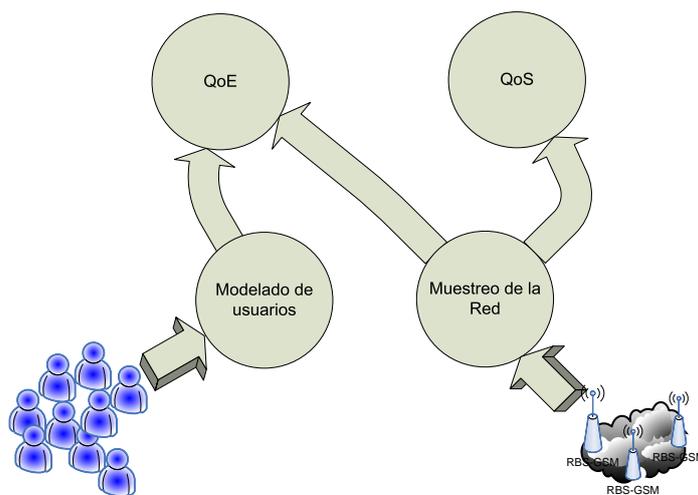


Figura 3.11: Diagrama de flujo conceptual de procesamiento

²Por ejemplo, aplicaciones de preguntas y respuestas, servicios de información (Horóscopo, informe meteorológico, guía de clasificados, etc), votaciones en concursos, etc.

Parte II
Base Teórica

Capítulo 4

Modelado de usuarios

Un elemento de suma importancia para el estudio de la Calidad de Experiencia tal como es planteado en este trabajo (es decir, simular usuarios y medir la percepción del servicio) es un buen modelado de la actividad de los usuarios individualmente.

Específicamente, el comportamiento que deseamos comprender y describir abarca los momentos de envío de mensajes de texto, y la importancia que tiene cada mensaje particular para el individuo.

Con modelo nos referimos a un modelo probabilístico para el tiempo que transcurre entre los disparos de mensajes, y la naturaleza de cada mensaje en sí.

Nos proponemos entonces encontrar una clase de modelos (por ejemplo: sin memoria, Markov de orden k , máquinas de estados finitos, etc) que se ajuste a los elementos citados anteriormente. Por supuesto que esta clase debe ser lo suficientemente rica como para describir razonablemente el fenómeno, pero a la vez lo suficientemente simple como para que la complejidad no impida trabajar con comodidad. Dicho compromiso es trivial y conocido, pero lo remarcamos por ser argumento para nuestra decisión (así como lo es en general).

La forma clásica de encarar este problema en telefonía, es suponer que cada usuario realiza llamadas acorde a una distribución exponencial, de manera independiente (sin memoria). Este primer acercamiento es simplista para el caso del servicio de mensajería SMS, principalmente por dos razones: no describe la interacción (conversaciones) mediante mensajes, y supone una misma distribución (e importancia) para todos los eventos. Por otro lado, como en general los modelos clásicos se utilizan para representar la forma en que la red ve el comportamiento del resto de los usuarios en su conjunto y no a cada usuario en particular, se pierde la posibilidad de modelar la diferencia de actividad de las personas en distintos momentos del día.

Basados en estas tres debilidades, proponemos un modelo con memoria que intenta salvar las deficiencias mencionadas.

4.1. Definiciones básicas

Presentamos a continuación algunos conceptos básicos, a efectos de formalizar el estudio de los modelos propuestos. Intentamos intercalar los aspectos técnicos con las respectivas aplicaciones que se encuentran en este trabajo.

El presente capítulo tiene, además del objetivo descripto relacionado con la necesidad de medir la calidad de experiencia, la ambiciosa pretensión de ser una introducción razonable a las herramientas probabilistas aplicadas al modelado de tráfico.

En este sentido, con miras a la completitud de la presentación, comenzamos repasando las definiciones de espacio de probabilidad y variable aleatoria, desde un punto de vista de teoría de la medida.

El nivel de rigurosidad matemática es intermedio, remitimos al lector a la bibliografía para profundizar (ver [1] [2] [3]).

Definición 4.1.1. *Un Espacio de Probabilidad (Ω, \mathcal{B}, P) es un espacio de medida, donde la medida total es 1. Esto es, un conjunto Ω , una σ -álgebra \mathcal{B} sobre Ω , y una función $P : \mathcal{B} \rightarrow [0, 1]$ tal que:*

- $P(\Omega) = 1$,
- $P(\emptyset) = 0$,
- Si $\{A_j\}$ es una sucesión de conjuntos disjuntos dos a dos, entonces $P(\bigcup_1^\infty A_j) = \sum_1^\infty P(A_j)$.

Al conjunto Ω se le denomina espacio muestral o espacio de sucesos elementales, y a los elementos de \mathcal{B} eventos o sucesos. Cada elemento ω representa una realización del azar.

Definición 4.1.2. *Una Variable Aleatoria es una función $X : \Omega \rightarrow \mathbb{R}$ medible.*

Definición 4.1.3. *La Función de Distribución de X es*

$$F(x) = P(X \leq x).$$

Si dos variables aleatorias tienen la misma función de distribución decimos que son Idénticamente Distribuidas.

Si existe $f : \mathbb{R} \rightarrow \mathbb{R}$ positiva e integrable tal que $F(x) = \int_{-\infty}^x f(t)dt$, decimos que la distribución F es absolutamente continua, y se le denomina función de densidad a f .

Las distribuciones utilizadas aquí son todas absolutamente continuas, por lo que trabajamos con la función de densidad f , pero de todas maneras también es útil la función de distribución F , sobretodo en la implementación.

Definición 4.1.4. *La Esperanza de una variable aleatoria real X es $E[X] = \int_{\Omega} XdP$.*

En el caso de nuestro trabajo, las variables aleatorias representan generalmente tiempos o intervalos de tiempos entre mensajes, por lo tanto son variables aleatorias con valores en \mathbb{R}^+ , y naturalmente su esperanza es también positiva.

Ejemplificamos algunas distribuciones, que utilizamos más adelante en este capítulo.

Ejemplo 4.1.1. *Una variable aleatoria es Normal o Gaussiana estándar si tiene densidad dada por*

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

Ejemplo 4.1.2. Una variable aleatoria tiene Distribución Uniforme en $[a, b]$ si tiene densidad dada por

$$f(x) = \frac{1}{b-a} \mathbf{1}([a, b])(x).$$

La esperanza de una variable aleatoria con esta distribución es $\frac{a+b}{2}$. Utilizamos esta distribución también en la implementación de distintas variables aleatorias (sección 8.1.3).

Como fue adelantado en la introducción, la distribución más utilizada en el modelado de tráfico es la exponencial. Continúa entonces su definición, seguida de algunos resultados que justifican su liderazgo, y algún comentario lateral.

Ejemplo 4.1.3. Una variable aleatoria tiene Distribución Exponencial con parámetro λ si tiene densidad dada por

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

Una variable aleatoria con esta distribución tiene esperanza $\frac{1}{\lambda}$.

Quizás la característica más importante en este contexto de las variables aleatorias con distribución exponencial sea la denominada “falta de memoria”. La siguiente proposición formaliza esta descripción, y además establece que ésta caracteriza a la distribución exponencial, es decir, es la única distribución con esta propiedad.

Proposición 4.1.1. Una variable aleatoria T tiene distribución exponencial si y sólo si $\forall t > 0$ y $\forall h > 0$ verifica

$$P(T > t + h | T > t) = P(T > h).$$

Si la variable aleatoria T representa el tiempo que transcurre hasta la ocurrencia de un evento, la propiedad enunciada significa lo siguiente: el tiempo en el cual puede ocurrir el evento a partir de cierto instante h que ya transcurrió, es independiente del tiempo h transcurrido¹. En general, es razonable que esto ocurra en el caso de los mensajes de texto: el hecho de que el usuario no haya mandado un mensaje hasta ahora, no influye en el tiempo que tardará en enviar el próximo mensaje. Es decir, el proceso real subyacente es intrínsecamente sin memoria.

Desde otro punto de vista, si suponemos que no tenemos información a priori acerca del momento de envío del próximo mensaje (más allá del valor esperado), es razonable adoptar una distribución que tenga gran “incertidumbre”. Si consideramos la entropía diferencial como medida de incertidumbre, esta distribución buscada es la exponencial. En efecto, de todas las distribuciones de probabilidad continuas con soporte en $[0, \infty)$ y esperanza μ , la exponencial (con $\lambda = 1/\mu$) es la que maximiza la entropía ([4]).

Hasta ahora definimos y ejemplificamos variables aleatorias, las cuales describen una realización del azar de un experimento puntual. Es necesario para poder modelar un fenómeno continuo, como lo es el que nos proponemos modelar, pasar a la noción de Proceso Estocástico, que es además la continuación natural de lo expuesto.

Definición 4.1.5. Un Proceso Estocástico es un conjunto de variables aleatorias indexadas $\{X_t : t \in T\}$.

¹Esto no quiere decir que los eventos $T > t + h$ y $T > h$ sean independientes (de hecho no lo son).

Dicho de otra manera, un proceso estocástico es una función $X : \Omega \times T \rightarrow E$.

Recordemos que los $\omega \in \Omega$ representan la aleatoriedad, mientras los $t \in T$ representan el tiempo. Entonces, fijando un elemento $\omega \in \Omega$ se obtiene una función particular² $X(\omega) : T \rightarrow E$ resultado del azar y, por otro lado, fijando un tiempo $t \in T$ se obtiene una variable aleatoria $X_t : \Omega \rightarrow E$.

Al espacio T se le llama *espacio de parámetros*, y en este trabajo nos concentramos en el caso $T \subset \mathbb{R}$, e identificamos a $t \in T$ con el tiempo, a pesar de que todas las definiciones son válidas en espacios más generales. Cuando T es discreto (por ejemplo $T = \mathbb{N}$), decimos que el proceso es de *tiempo discreto*, y usualmente se asocia a observaciones temporales de un suceso (por ejemplo un muestreo). Si por el contrario T es un intervalo, decimos que el proceso es de *tiempo continuo*.

El conjunto de valores que puede tomar X_t (en este caso E) se denomina *espacio de estados*, y también distinguimos cuando el espacio E es discreto o continuo.

Tenemos por lo tanto una clasificación de los procesos según la topología de los espacios T y E . A saber:

- Procesos estocásticos de tiempo discreto con espacio de estados discreto.
- Procesos estocásticos de tiempo discreto con espacio de estados continuo.
- Procesos estocásticos de tiempo continuo con espacio de estados discreto.
- Procesos estocásticos de tiempo continuo con espacio de estados continuo.

Una clase particular de procesos estocásticos de especial interés en las telecomunicaciones, es la clase de procesos estacionarios. Estos son aquellos para los cuales las características no dependen del tiempo.

Definición 4.1.6. *Un proceso estocástico es Estacionario si y sólo si la distribución de X_t es invariante bajo traslaciones (en t).*

En lo referente a este trabajo, esperamos que nuestro modelo no sea estacionario, pues hay una dependencia temporal evidente de la actividad de los usuarios³.

4.2. Algunos modelos posibles

Presentamos a continuación un breve descripción de algunos modelos posibles, en orden ascendente de complejidad.

4.2.1. Proceso de Poisson

Comenzamos describiendo algunos procesos denominados *puntuales*. Luego eliminamos restricciones para ganar en generalidad, haciendo un análisis sobre las posibilidades que nos brinda cada modelo.

Es de particular interés en muchas aplicaciones, contar la cantidad de ocurrencias de un determinado suceso. La definición que sigue, engloba pues una gran cantidad de procesos muy utilizados, que se pueden ver como caso particular de éste.

En algunas oportunidades, utilizamos la notación $N_t = N([0, t])$ (esto es, la cantidad de eventos que ocurrieron en el intervalo de tiempo $[0, t]$).

²Denominada también *trayectoria*

³Por ejemplo, los usuarios duermen.

Definición 4.2.1. *Considérese una sucesión de variables aleatorias T_1, T_2, \dots denominadas incrementos, estrictamente positivas, que representan la duración de cierto evento, y las variables aleatorias S_n que representan el tiempo total transcurrido hasta el evento n :*

$$S_n = \sum_{i=1}^n T_i$$

Entonces el proceso definido como

$$N_t = \max\{n \geq 0 : S_n \leq t\}$$

se denomina *Proceso de Conteo (o puntual)*.

En este caso, el espacio de estados es $E = \mathbb{N}$, y el espacio de tiempo es $T = [0, \infty)$, o sea, el proceso es $N : \Omega \times [0, \infty) \rightarrow \mathbb{N}$.

Aplicado a nuestro problema, la variable aleatoria T_i podría representar el tiempo que transcurre a partir del momento en el cual el usuario envió el mensaje $i - 1$, hasta el momento en que envió el i -ésimo mensaje (es decir, el tiempo entre mensajes). De esta manera, el proceso N_t cuenta la cantidad de mensajes enviados hasta el instante t .

Mirando una realización del proceso (para un ω específico), se desprenden los tiempos de envío de cada mensaje (en la gráfica, mirando los escalones).

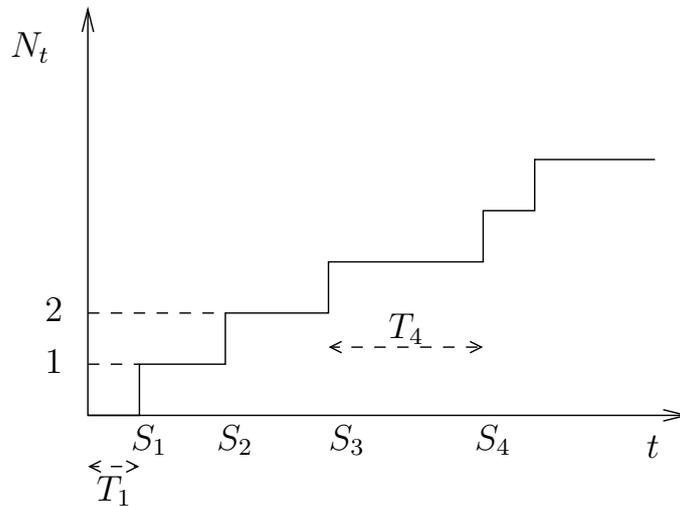


Figura 4.1: Ejemplo de una realización (ω fijo) de un proceso de conteo N_t .

En la figura 4.1 podemos ver un ejemplo de una realización de un proceso de conteo. En nuestra interpretación, la altura representa la cantidad de mensajes enviados, los escalones están situados sobre los tiempos de envío, y las distancias entre los escalones corresponden a las variables aleatorias T_i .

Un caso particular es el Proceso de Poisson, que proviene de suponer una distribución exponencial de las variables aleatorias T_i . Existen varias definiciones equivalentes de este proceso, tomamos aquí dos: la primera como el caso particular recién comentado, y la segunda, como base para luego hacer una generalización natural.

Definición 4.2.2. Cuando las variables aleatorias T_i son i.i.d. con distribución exponencial de parámetro λ , al proceso de conteo $\{N_t\}$ se le denomina *Proceso de Poisson de parámetro λ* .

Recordemos la notación introducida $N_h = N([0, h]) = N([t, t + h])$, donde la última igualdad es cierta si los incrementos son independientes. Pasemos entonces a la segunda definición.

Definición 4.2.3. Un proceso de conteo N_t es de Poisson con tasa λ si y sólo si

- $N(0) = 0$,
- los incrementos son independientes y estacionarios,
- $P(N_h = 1) = \lambda h + o(h)$,
- $P(N_h \geq 2) = o(h)$.

Este modelo se usa tradicionalmente en redes de datos (para modelar tráfico) o en telefonía (para modelar arribos de llamadas), y los resultados que se obtienen son sumamente alentadores, en el sentido que describen ciertamente la realidad. En este contexto, los resultados más famosos son los de Erlang, que vienen siendo utilizados con éxito para planificar redes de telecomunicaciones.

Creemos, sin embargo, que este modelo no se ajusta de manera fiel al tráfico cuando se trata de SMS. Por supuesto que esta apreciación es completamente subjetiva, sobre todo sin poseer una gran cantidad de datos como para realizar evaluaciones, además de la dificultad intrínseca acerca de definir el concepto de “buen modelo”, o “mejor modelo” que otro. En cualquier caso, las razones incluidas en la introducción de este capítulo (a saber, el presente proceso supone independencia entre un mensaje y el siguiente, y no hay manera de discriminar entre distintos tipos de mensaje) sugieren un modelo un tanto más complejo.

Una generalización inmediata del proceso de Poisson es permitir a la tasa λ que varíe con el tiempo. Esto permite modelar procesos no estacionarios, que es una característica buscada en aplicaciones donde la actividad depende fuertemente de, por ejemplo, la hora del día o del día del mes. Nos basamos entonces en la definición 4.2.3, como sigue:

Definición 4.2.4. Dada una función $\lambda : \mathbb{R} \rightarrow \mathbb{R}$ derivable a trozos, decimos que un proceso N_t es de Poisson no-homogéneo con tasa instantánea $\lambda(t)$ si y sólo si

- $N(0) = 0$,
- los incrementos son independientes,
- $P(N([t, t + h]) = 1) = h \cdot \lambda(t) + o(h)$,
- $P(N([t, t + h]) \geq 2) = o(h)$.

Naturalmente, si $\lambda(t) = \lambda \forall t$ recuperamos el proceso de Poisson estacionario.

Este proceso resuelve el problema de la estacionariedad (o sea, agrega la dependencia temporal deseada, eliminando la estacionariedad), pero conserva los defectos principales mencionados. A continuación, describimos otra generalización del proceso de Poisson.

Definición 4.2.5. Sea una sucesión de variables aleatorias i.i.d. Z_1, Z_2, \dots , y un proceso $\{N_t\}$ de Poisson como en 4.2.2. Se llama *Proceso de Poisson Compuesto* al que definen las variables aleatorias

$$Y_t = \sum_{k=1}^{N_t} Z_k$$

Nuevamente recuperamos el proceso de Poisson cuando todas las Z_k valen 1. En el caso general, este proceso modela sucesos que ocurren cada tiempos exponenciales, pero que tienen distinta incidencia (incidencia representada mediante Z_k). Este modelo es ampliamente usado en aplicaciones financieras (ver [3]), pero en el contexto de este trabajo le podemos dar dos interpretaciones:

- Puede modelar mensajes múltiples (de una persona hacia varios). En este caso, los Z_k son naturales, que cuentan la cantidad de mensajes que está enviando el usuario en el momento.
- Si nos permitimos suponer que los mensajes de una misma conversación no están muy alejados en el tiempo, podemos modelar con Z_k la cantidad de mensajes de una conversación.

Si bien estamos generalizando el proceso de Poisson, este nuevo modelo no parece agregar mejoras significativas, por lo que seguimos avanzando en busca de una solución más satisfactoria.

4.2.2. Cadenas de Markov de Tiempo Continuo

El otro protagonista fundamental en el modelado de tráfico son las Cadenas de Markov. Estos son procesos estocásticos X_t , donde el espacio de parámetros es continuo, el espacio de estados es discreto, y además tienen la siguiente propiedad (llamada propiedad de Markov) interesante: la distribución de probabilidad en el instante $t + s$, dada la historia completa del proceso hasta el instante s , depende sólo del estado del proceso en el instante s . En efecto, la definición es exactamente esa:

Definición 4.2.6. Un proceso estocástico X_t es una Cadena de Markov de Tiempo Continuo si para todo $i, j, x(u) \in E$, $u \in [0, s)$ se cumple

$$P(X_{t+s} = j | X_s = i, X_u = x(u), 0 \leq u < s) = P(X_{t+s} = j | X_s = i)$$

Si además esta probabilidad condicional depende únicamente del valor de X_s pero no del instante s , decimos que la cadena es homogénea.

Una definición equivalente a 4.2.6, y que es quizás la más práctica en el contexto de este trabajo, es la siguiente:

Definición 4.2.7. Un proceso estocástico X_t es una Cadena de Markov de Tiempo Continuo si:

1. el tiempo de permanencia del proceso en el estado i es una variable aleatoria que tiene distribución exponencial de parámetro λ_i , independiente de las variables aleatorias correspondientes a los otros estados;

2. cuando el proceso abandona un estado i , la elección del siguiente estado se hace de acuerdo a las probabilidades p_{ij} (probabilidad de pasar del estado i al estado j), donde $p_{ii} = 0$ y $\sum_j p_{ij} = 1$.

Notar que las transiciones entre estados se realizan de acuerdo a una Cadena de Markov de Tiempo Discreto, pero el tiempo de permanencia en cada estado es una variable aleatoria exponencial, en lugar de ser constantes. A esta cadena discreta se la denomina *embedded Markov chain*.

Usualmente se representan las Cadenas de Markov de Tiempo Continuo mediante un grafo, donde los vértices simbolizan los estados (y la esperanza del tiempo de permanencia $1/\lambda_i$), y las aristas las transiciones (con sus probabilidades p_{ij}). En la figura 4.2 se puede ver un ejemplo de esta representación.

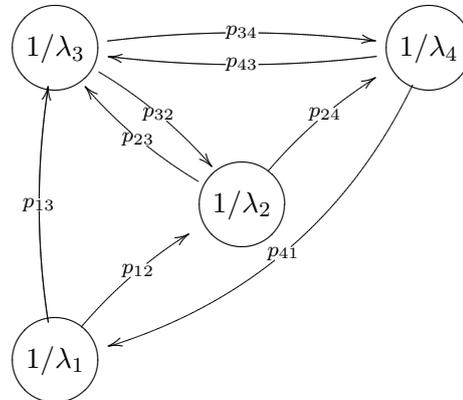


Figura 4.2: Ejemplo de CMTC, con los parámetros λ_i y p_{ij}

Otro posible conjunto de parámetros que define un proceso (de los cuales se podría decir mucho más de lo que aquí comentamos), son las tasas de transición, que cumplen las siguientes relaciones con los parámetros de arriba:

$$q_{ij} = \lambda_i \cdot p_{ij} \quad \text{si } i \neq j$$

$$q_{ii} = -\lambda_i$$

Entonces, también se pueden representar simplemente los estados en los vértices, y etiquetar las aristas con las tasas de transición q_{ij} . En la figura 4.3 se puede ver la cadena del ejemplo anterior, con estos nuevos parámetros.

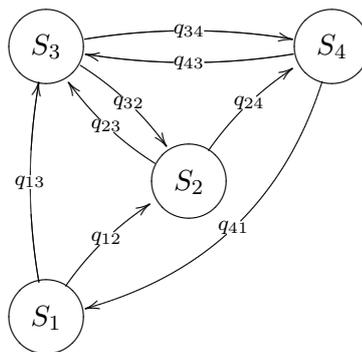


Figura 4.3: Ejemplo de CMTC, con los parámetros q_{ij}

Un caso particular de estas cadenas lo constituyen los Procesos de Nacimiento y Muerte, donde los estados son $E = 0, 1, \dots, n$, y las transiciones desde el estado i pueden ser sólo hacia el estado $i + 1$ o $i - 1$ con tasas λ_i y μ_i respectivamente.

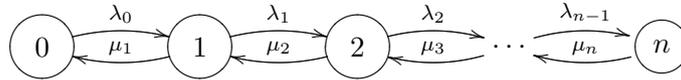


Figura 4.4: Proceso de Nacimiento y Muerte

Más aún, si además los μ_i son nulos y $\lambda_i = \lambda$ para todo i , recuperamos el proceso de Poisson.

Evaluemos las potenciales aplicaciones de este tipo de procesos a nuestro problema concreto. El caso general de las cadenas de Markov nos brinda la posibilidad de tener distintos estados, y movernos entre ellos sin restricciones. En particular, los pasajes entre estados pueden ser efectivos en los momentos de envío de mensajes, y podemos interpretar cada estado de manera diferente. Esto subsana la dificultad de diferenciar los mensajes, pues se pueden asociar los estados a alguna clasificación deseada de los mensajes. De todos modos, todas las distribuciones deben ser exponenciales, y cuando se envía un mensaje se debe abandonar necesariamente el estado (de todas maneras, esto último se puede solucionar agregando un estado “fantasma”). Por otro lado, es difícil, por lo menos a primera vista, diferenciar el comportamiento según, por ejemplo, la hora del día.

4.3. El modelo propuesto

Luego de repasar y evaluar los potenciales modelos a utilizar, continuamos con descripciones, tanto formales como informales, de los consecutivos modelos que emergieron, y que fueron en definitiva pasos intermedios en el camino hacia nuestro modelo final.

La idea principal es establecer la dependencia del envío de un mensaje con los mensajes anteriores. Esta conexión dependerá de cada usuario en particular, y es justamente esto lo que permitirá modelar distintos tipos de usuario, para luego segmentar la muestra.

Separamos esta parte en tres subsecciones, coincidentes con tres niveles que alcanzó la idea en su proceso. Presentamos primero una descripción informal del modelo (en cada parte), puramente heurística, para luego formalizar los aspectos teóricos.

4.3.1. Primer acercamiento

El concepto original consistió en separar los mensajes “aislados” de los mensajes con respuestas. Los mensajes “aislados” son aquellos que el usuario envía sin esperar una contestación⁴, en cambio los segundos corresponden a interacciones (combinaciones para determinada actividad, preguntas y respuestas, etc).

Esta separación ataca directamente las dos deficiencias, comentadas anteriormente, que el modelo clásico para telefonía presenta en una presunta aplicación al servicio de mensajería corta. En primer lugar, modela de forma más precisa el comportamiento de los usuarios. Esto será, además de intuitivamente previsible, formalmente claro, ya

⁴Típicamente avisos, como por ejemplo “Estoy saliendo”, “Voy a demorar”, etc

que este nuevo modelo contiene a varios modelos anteriores como casos particulares. En segundo lugar, permite tratar diferencialmente los distintos tipos de mensajes en términos de importancia. Mientras en un mensaje “aislado” es más difícil darse cuenta de un retardo grande en el envío (incluso probablemente el usuario nunca se entere del retardo), en un escenario de conversación es, no sólo más fácil detectar retardos grandes, sino además más ofuscante para el usuario cuando éstos ocurren.

El modelo de dos estados propuesto se ilustra en la figura 4.5, y se detalla a continuación.

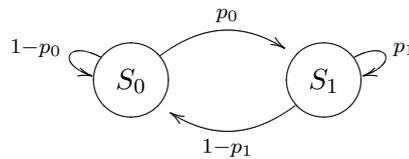


Figura 4.5: Modelo de dos estados

El estado S_0 representa al usuario cuando está “esperando” para enviar un mensaje (la mayoría del tiempo), y coincide con el modelado clásico de comportamiento en telefonía (y otras áreas). En este estado por lo tanto, el usuario envía un mensaje cada un intervalo de tiempo modelado con una variable aleatoria exponencial. Cuando ocurre el envío del SMS, se bifurcan dos posibilidades: si el SMS es “aislado” (en el sentido explicado anteriormente) el usuario permanece en el estado S_0 . En particular el próximo mensaje ocurrirá en un lapso de tiempo aleatorio gobernado por la distribución exponencial mencionada. Si en cambio el SMS origina una conversación, el estado pasará a ser S_1 , que simboliza un estado de interacción. En dicho estado, la distribución de la variable aleatoria que representa el tiempo entre mensajes será distinta que la mencionada para S_0 (naturalmente será “más rápida” que esta última). De hecho en principio no necesariamente es exponencial.

En el grafo de la figura 4.5, los mensajes enviados en S_0 tienen una probabilidad p_0 de originar una conversación, y una probabilidad $1 - p_0$ de ser un evento aislado. De similar manera, en el estado S_1 (“en conversación”), los mensajes tienen una probabilidad $1 - p_1$ de ser los finalizadores de la conversación, y p_1 de ser mensajes intermedios en la interacción.

La descripción se asemeja considerablemente a la realizada sobre las Cadenas de Markov de Tiempo Continuo, por lo que sería natural intentar formalizar este modelo desde esa perspectiva, pero incluso en esta versión simplificada del modelo hay por lo menos dos características que impiden verlo como una Cadena de Markov de Tiempo Continuo. A saber, en las CMTC no puede haber lazos (debe ser $p_{ii} = 0$), y las distribuciones en cada estado deben ser necesariamente exponenciales.

Formalizando, el proceso tiene como espacio de estados $S = \{S_0, S_1\}$. Sin embargo, si sólo miramos el estado en que se encuentra, en función del tiempo, no podemos distinguir los mensajes que se envían sin cambiar de estado (los *loops*). Es simplemente un tecnicismo que se puede solucionar de dos maneras: la primera es duplicando los estados, la segunda, agregando un contador de mensajes. Consideremos ésta última opción, por ser más simple.

Es entonces, $X : \Omega \times T \rightarrow S \times \mathbb{N}$, con las propiedades definidas anteriormente.

4.3.2. Agregando estados

Describimos ahora la generalización natural del modelo anterior. Consiste en agregar más estados, que cuenten la cantidad de mensajes enviados en una conversación. Es decir, el estado S_0 sigue representando lo mismo (al usuario en estado de espera), pero ahora el estado S_1 significa que el usuario ha enviado un mensaje “de conversación”, y enviará el segundo próximamente. Desde este estado entonces, el usuario podrá volver al estado S_0 (en caso que ese segundo mensaje haya sido el último de la conversación), o podrá pasar a un estado S_3 , desde el cual enviará un tercer mensaje de la conversación. Tomamos en este caso un modelo con una cantidad finita de estados, y por lo tanto desde el estado S_n se puede volver al estado S_0 cuando la conversación finaliza, o mantenerse en el estado S_n , que significa entonces que se han enviado n o más mensajes.

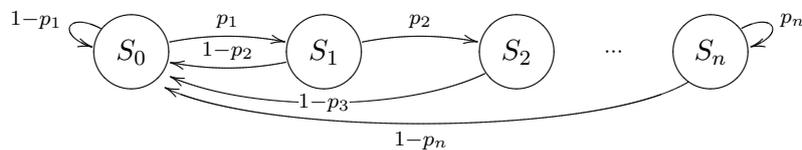


Figura 4.6: Modelo de n estados

Frente al modelo inmediato anterior, ofrece la ventaja de poder diferenciar los distintos mensajes de una conversación, tanto su importancia⁵ como la distribución que modele los tiempos entre mensajes.

En este caso el espacio de estados es entonces $S = \{S_0, S_1, \dots, S_n\}$, y el proceso nuevamente es $X : \Omega \times T \rightarrow S \times \mathbb{N}$.

De las tres objeciones principales a los modelos clásicos (nuevamente: igual distribución en todos los mensajes, mensajes indistinguibles, difícil de imponer dependencia horaria), este modelo salva las dos primeras, y lo hace sin agregar una complejidad excesiva.

Busquemos entonces, para finalizar el capítulo, levantar la última restricción.

4.3.3. Agregando dependencia temporal

Sobre la base del ltimo modelo, proponemos tres posibles caminos.

El primero de ellos, y más simple de todos, consiste en considerar dos tasas λ_1 y λ_2 distintas (o en un caso general, m tasas) para la distribución exponencial del estado S_0 . Entonces, durante un período del día (digamos, de 8hs a 22hs para fijar ideas) la distribución de S_0 es exponencial con parámetro λ_1 , y en el resto del día (de 22hs a 8hs), exponencial con parámetro λ_2 . Nótese que al tener la distribución exponencial la propiedad de pérdida de memoria, el momento del sorteo es indiferente, y se puede realizar la transición entre estos dos períodos del día, sin perder información y sin modificar la naturaleza de la distribución exponencial.

Puede resultar no del todo convincente la transición de λ a horas determinadas, pero es claro que si buscamos que queden representados en el modelo los períodos donde el usuario está despierto y dormido, necesariamente tendremos que agregar un factor determinista, o casi determinista.

⁵Por ejemplo, podría no ser de igual impacto el retardo en el inicio de una conversación que en el 4to mensaje de la misma.

La segunda forma de encarar el problema⁶ es similar a la primera, pero “suaviza” el switch entre las distribuciones con λ distinto. Consiste en agregar otro estado similar a S_0 , llamémosle S'_0 , con una distribución exponencial de parámetro λ_2 . Este proceso ahora se modula con otro proceso. Es decir, se pasa del estado S_0 a S'_0 (y viceversa) en aquellos momentos que resulten de otro proceso independiente (por lo tanto, esta transición no será en tiempos de envío de mensajes). Claramente, si queremos que represente de manera razonable los intervalos del día de una persona, este proceso modulante estará cerca de ser determinista.

Por último, recurrimos al proceso de Poisson no-homogéneo, definido en 4.2.4. Viemos en su momento que este proceso introduce una tasa variable con el tiempo $\lambda(t)$. Como lo que precisamos para el modelo con el que ya contamos es simplemente que la distribución del estado S_0 cuente con esta variabilidad en el tiempo, podemos adoptar el primer arribo de un proceso de Poisson no-homogéneo como la distribución de S_0 .

Sin lugar a duda, el último es el más elegante, y brinda más flexibilidad, puesto que se puede elegir cualquier función $\lambda(t)$ derivable a trozos, y no necesariamente una cantidad finita de valores de λ como en los casos anteriores. Por otro lado, existen algoritmos ([5]) que implementan la generación de eventos en procesos de Poisson (homogéneos y no-homogéneos), por lo tanto es viable la implementación de éste último modelo⁷.

⁶Inspirada en las denominadas MMPP (Markov-Modulated Poisson Processes)

⁷Viabilidad que no es clara a partir de la definición 4.2.4, que es para nada constructiva.

Capítulo 5

Calidad de Servicio (QoS)

La calidad en general es el grado en el que un conjunto de características satisface ciertos requerimientos. Al hablar de calidad en redes de telecomunicaciones el concepto más difundido es el de calidad de servicio, también conocido por QoS (Quality of Service). La ITU-T lo define como “el efecto colectivo de funcionamiento del servicio que determina el grado de satisfacción del usuario” ([6]).

La QoS de un servicio brindado sobre una red de telefonía móvil está determinada por tres aspectos fundamentales: accesibilidad de la red, accesibilidad del servicio y la integridad del mismo.

- *La accesibilidad de la red* se refiere a la disponibilidad de recursos de red para conectarse a un servicio. Cuando un usuario intenta acceder a un servicio, puede ser que la red atienda esta petición y provea el servicio solicitado, o por el contrario puede que la petición no sea atendida. Las posibles causas del fracaso de la atención de la red a la petición son entre otras: congestión en la red, falta de cobertura temporal o permanente en una determinada zona, etc.
- *La accesibilidad del servicio* incluye los aspectos relacionados con la disponibilidad del servicio. La respuesta a la petición de acceso al servicio puede ser la provisión del mismo o la indicación de que éste no está disponible. Las indisponibilidades del servicio pueden deberse a la congestión de recursos, desactivación temporal, servicio caído por avería, etc.
- *La integridad del servicio* se refiere a la calidad ofrecida durante el uso del servicio. Incluye los siguientes aspectos:
 - La calidad de la señal de voz y de la transmisión de datos.
 - Caída del servicio. Significa la imposibilidad de continuar accediendo a él tras haber establecido la comunicación en un primer momento, siempre y cuando la imposibilidad sea motivada por cualquier causa ajena a la voluntad de sus usuarios y siempre que éstos se encuentren en todo momento en la zona de cobertura de la red.
 - La efectividad del servicio. Es el porcentaje de accesos al servicio realizados y completados satisfactoriamente, frente a la totalidad de los accesos realizados.

En este trabajo lo que nos ocupa concretamente es medir la disponibilidad del servicio y la integridad del mismo para SMS.

5.1. Indicadores de QoS

Hay varios enfoques posibles para medir la calidad de servicio de SMS, dependiendo básicamente de dónde o cómo se quiera o se pueda medir. Sin lugar a dudas se pueden medir diferentes parámetros conociendo los mensajes de señalización transmitidos, o directamente en el SMS-C que al hacerlo de extremo a extremo. Con esto en consideración se pueden ver dos grandes enfoques. El primero se basa mayoritariamente en la tecnología empleada, estudiando el tipo de mensajes intercambiados entre los diferentes equipos, el tiempo que demoran, etc, y con ello se pueden identificar los diversos tipos de errores y su frecuencia. Por otra parte, el otro enfoque se basa fundamentalmente en la percepción del usuario y supone que la calidad en los elementos intermedios, sus interconexiones, etc, está implícita en la QoS global del servicio. Este enfoque es el más estandarizado y el que estudiamos a continuación.

La GSM Association y la ETSI elaboraron una serie de indicadores de QoS conocidos como parámetros estándar de QoS para SMS. Si bien se sugieren indicadores, conocidos también como KPI (Key Performance Indicators), no se indican valores aceptables u objetivos para ellos. Esto queda a criterio de la regulación del país si la hubiere o a criterio del operador en otro caso.

En lo que hace a la accesibilidad del servicio se definen dos KPI a continuación.

Definición 5.1.1. *Se llama porcentaje de éxito de acceso al servicio a la probabilidad de que el usuario pueda acceder al servicio de mensajería corta, siempre que el terminal indique que detecta señal de la red.*

Este indicador también se conoce como SA-SMS-MO (Service Accessibility SMS Mobile Originated)

Es entonces,

$$\text{SA-SMS-MO}(\%) = \frac{N_{\text{AccExitosos}}}{N_{\text{TotIntentos}}} * 100 (\%)$$

donde $N_{\text{AccExitosos}}$ es el número de accesos exitosos al SMS y $N_{\text{TotIntentos}}$ refiere al número total de intentos de acceso al SMS.

Se contabiliza un intento de acceso al servicio SMS cuando el usuario presiona *send* para enviar el mensaje y un acceso exitoso al recibir el *ack* del SMS-C.

Definición 5.1.2. *Se llama tiempo de acceso al intervalo que transcurre entre el envío del mensaje al SMS-C y la recepción de la notificación de éste.*

Este indicador también se conoce como AD-SMS-MO (Access Delay SMS Mobile Originated), y se calcula entonces de la siguiente manera:

$$\text{AD-SMS-MO} = t_{\text{rec}} - t_{\text{env}}$$

donde t_{rec} es el instante en el cual el móvil recibe la confirmación del SMS-C y t_{env} es el instante en el cual el usuario envía un SMS al SMS-C.

En lo que hace a la integridad del servicio, se definen otras dos KPI:

Definición 5.1.3. *Se llama porcentaje de compleción exitosa de SMS al porcentaje de mensajes recibidos correctamente entre dos terminales excluyendo aquellos que lleguen duplicados y/o corruptos. El mensaje es considerado válido si es entregado correctamente en un período de tiempo prefijado.*

El porcentaje de compleción exitosa de SMS también se conoce como CR-SMS-CS (Completion Rate SMS Circuit Switched).

Específicamente,

$$\text{CR-SMS-CS (\%)} = \frac{N^{\circ}_{rec} - N^{\circ}_{dup} - N^{\circ}_{corr}}{N^{\circ}_{env}} * 100 (\%)$$

donde N°_{rec} es el número de SMS recibidos, N°_{dup} el número de SMS duplicados, N°_{corr} el número de SMS corruptos y N°_{env} el número de SMS enviados.

Definición 5.1.4. *Se llama tiempo de entrega end to end al intervalo que transcurre desde que se envía un mensaje al SMS-C y éste se recibe en el móvil de destino*

A este indicador también se le conoce como DT SMS (Delivery Time SMS):

$$\text{DT SMS} = t_{rec} - t_{env}$$

donde t_{rec} es el instante en el cual el móvil destino recibe el SMS y t_{env} el instante en que el usuario envía el SMS al SMS-C.

5.1.1. Procedimiento de medida

El procedimiento de medida de los KPI también es detallado por la GSM Association. Describimos aquí los principales aspectos a tener en cuenta.

- Se recomienda eliminar cualquier incertidumbre causada por el acceso a la red, para ello los móviles empleados deben permanecer en un lugar fijo con probabilidad del 100 % de acceder a la red.
- Elementos a mantener constantes durante el procedimiento:
 - Formato del SMS (PDU o texto)
 - Largo del mensaje
 - SMS-C empleado
 - Intervalo entre SMSs
- La memoria del dispositivo de destino debe vaciarse antes de comenzar la medición para evitar errores debidos a falta de espacio en la memoria.
- Definir la pérdida del mensaje. Dado que el servicio de mensajería corta es de *store-and-forward* se necesita una clara definición para la pérdida de un mensaje. Desde el punto de vista del usuario se espera que el SMS sea entregado dentro de un cierto período de tiempo. Aún cuando sea entregado correctamente, si esto ocurrió fuera del tiempo esperado el mensaje debe considerarse perdido. Para eliminar la influencia de estos mensajes en otras medidas conviene setear un tiempo de vida razonable al mensaje.

5.1.2. Estadísticas y resultados

Todos los KPI antes mencionados anteriormente deben ser calculados a partir de medidas tomadas de:

- tráfico real
- una muestra representativa del tráfico real
- tráfico de prueba
- una combinación de los anteriores

Las medidas deben reflejar las variaciones a lo largo del día, de los días de la semana, etc. Para el monitoreo de SMS, puede monitorearse cada n-ésimo SMS, siendo n calculado a partir del total de SMSs esperados y del número necesario de observaciones.

Los resultados a evaluar y/o presentar consisten en:

- Para aquellos KPI medidos en porcentajes:
 - el porcentaje correspondiente
 - el n° de observaciones empleadas
 - los límites del intervalo de confianza al 95 %.
- Para aquellos KPI medidos en unidades de tiempo:
 - la media correspondiente
 - el n° de observaciones empleadas
 - el tiempo para el 95 % de los mensajes más rápidos.

5.2. ¿Qué mide el AST?

Habiendo descrito las generalidades de la QoS para SMS nos enfocamos ahora en nuestro sistema. El AST mide los parámetros referidos a la integridad del servicio, es decir, el tiempo de entrega end to end y una variante del porcentaje de compleción exitosa que no contempla los mensajes duplicados ni los corruptos, cuya fórmula se puede ver a continuación:

$$\text{CR-SMS-CS } (\%)_{AST} = \frac{N^{\circ}_{rec}}{N^{\circ}_{env}} * 100 (\%)$$

Recordando la arquitectura del sistema descrita en el capítulo 3 el AST está conectado al SMS-C, al menos a dos terminales móviles y al SMS Gateway. De esta manera al observar problemas en el envío de mensajes móvil-móvil podemos determinar en qué segmento del recorrido es que se producen observando los KPI en los modos que involucran al móvil y al SMS-C. La misma idea se aplica para los SMS enviados o terminados en el SMS-Gateway.

5.2.1. Reportes de QoS

Se implementaron dos tipos de reportes de QoS a través de la GUI, uno estándar y otro personalizado. En el reporte estándar se proveen los datos para un modo específico a lo largo del día, la semana, el mes en curso y el año. El DT-SMS se presenta en gráficas lineales y logarítmicas para poder apreciar grandes variaciones. Un ejemplo del encabezado del informe estándar, de los valores calculados y de sus gráficas, se pueden ver en las figuras 5.1, 5.2 y 5.3. Para todos los intervalos de tiempo mencionados anteriormente se proveen gráficos y datos del mismo tipo.

<p>Información del modo</p> <p>Origen:</p> <ul style="list-style-type: none">▪ Tipo: Gateway▪ Número: 59890001018▪ Service Provider: ANCEL SMS-C▪ IP: 10.35.33.249▪ Puerto: 9201 <p>Destino:</p> <ul style="list-style-type: none">▪ Tipo: Terminal móvil▪ Número: 59898390757▪ Service Provider: ANCEL▪ Marca: ERICSSON▪ Modelo: 0130101-BVF221m▪ IMSI: 748010030019314▪ IMEI: 350379020772244
--

Figura 5.1: Encabezado de un informe

<p>Informe de la última semana</p> <ul style="list-style-type: none">▪ Cantidad de observaciones: 3643▪ Retardo Máximo: 12103ms▪ Retardo Mínimo: 3595ms▪ Retardo Promedio: 4935ms▪ El 95 % de los mensajes tardó menos de : 5689ms

Figura 5.2: Valores de QoS de una semana

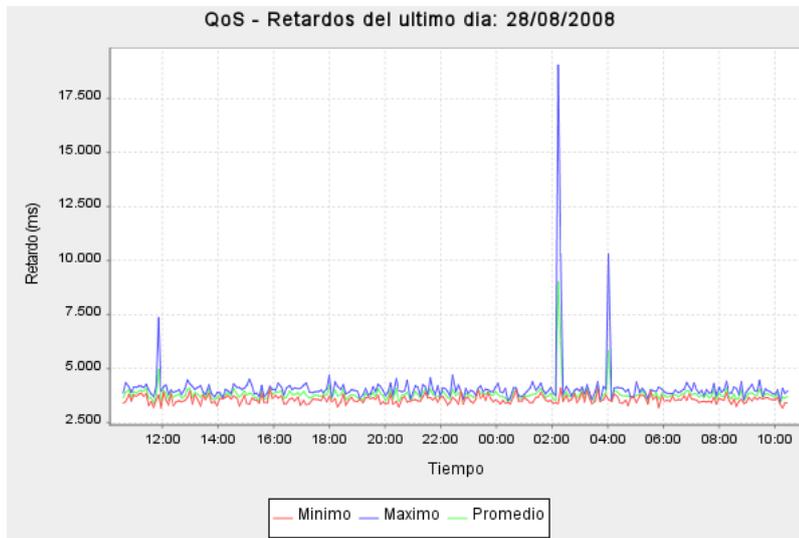


Figura 5.3: Gráfica en escala lineal de los retardos a lo largo de un día

En los histogramas se puede ver la distribución de los retardos de los mensajes. En la figura 5.4 se presenta un ejemplo de esta representación.

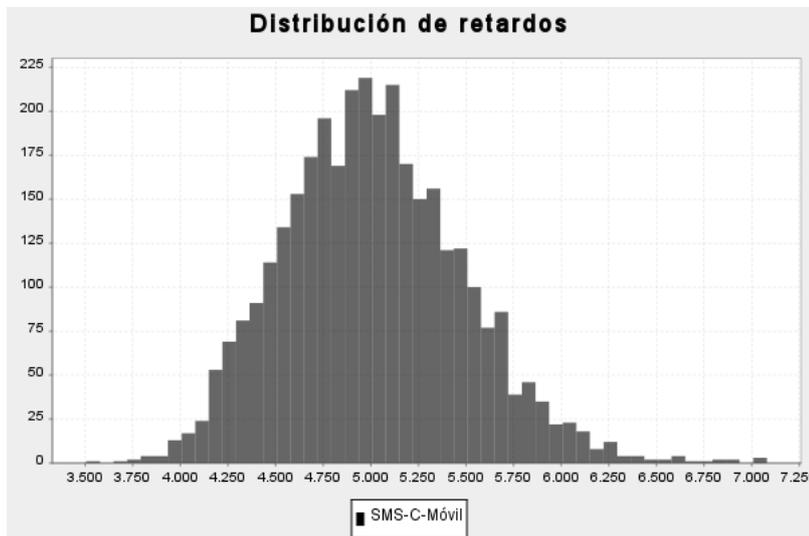


Figura 5.4: Histograma de los retardos de mensajes

El reporte personalizado brinda la posibilidad de hacer gráficas comparativas del DT-SMS entre diferentes modos, en un rango de fechas a elegir, y resulta particularmente útil para acotar el lugar de una falla. Por ejemplo, en la figura 5.5 vemos los retardos de los mensajes entre dos móviles, de un móvil al SMS-C, y del SMS-C a un móvil. Se desprende de la figura que el problema se produjo en el camino desde el SMS-C al terminal móvil. Analizaremos en profundidad esta situación en el capítulo 10.

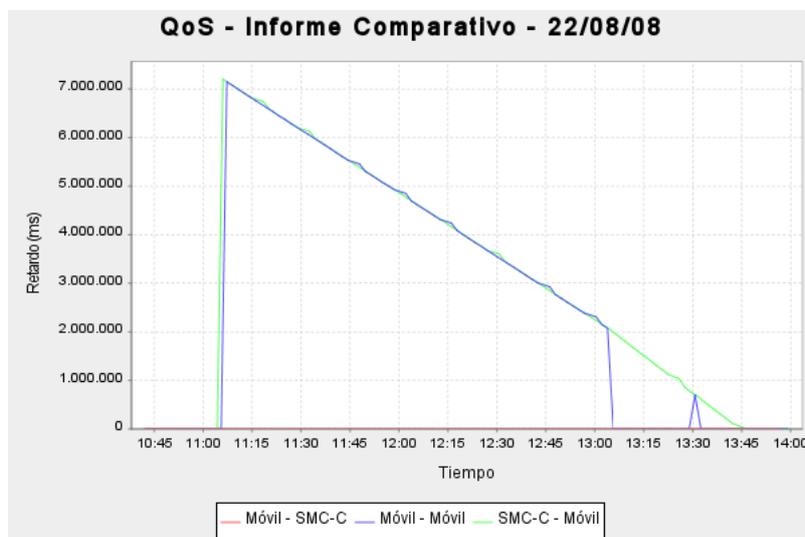


Figura 5.5: Comparación de retardos de varios modos

5.2.2. Alarmas

El objetivo del proyecto es por un lado poder generar informes como vimos en el capítulo anterior, pero también poder alertar al área de operación y mantenimiento en caso de situaciones de mal desempeño. Con este objetivo, se incorporó el protocolo SNMP. El operador al configurar los modos de funcionamiento debe introducir para cada modo, los umbrales de QoS para los que quiere generar alarmas. El sistema para cada modo activo verifica que los tiempos de respuesta sean menores que los umbrales, en caso contrario dispara una alarma. El AST permite la configuración de 2 umbrales distintos de forma que se puedan generar 2 alarmas con distinto nivel de gravedad.

5.3. Regulación acerca de QoS en Uruguay

Si bien actualmente no existe reglamento alguno en lo que se refiere a la QoS en servicios prestados sobre redes móviles, la URSEC ya se encuentra trabajando en ello. El proyecto de reglamento se encuentra hace algunos meses en la sección *Jurídica* de la URSEC¹.

El proyecto consiste en realizar un marco general respecto a calidad de servicio y luego agregar un anexo para cada servicio en particular, incluyendo el servicio de mensajería corta. La finalización de este reglamento estaba originalmente prevista para mediados de este año, pero se vio postergado básicamente por dos razones;

En primer lugar, a pesar de que la URSEC dispone de un estudio técnico ya realizado del equipamiento a adquirir para medir la QoS de los diferentes servicios, por razones presupuestales sólo se pudo adquirir una pequeña parte de éstos. Específicamente, sólo se ha adquirido el equipo para realizar mediciones sobre la red 3G. Esto implica que es imposible cotejar los datos provistos por los operadores con los obtenidos por un organismo neutral, por lo menos a corto plazo.

En segundo lugar, más allá de las decisiones técnicas, y de la posibilidad o no de medir la QoS, existen decisiones políticas importantes en el marco regulatorio. Existen

¹Información brindada extraoficialmente por fuentes de la URSEC

distintas políticas posibles, en países maduros en materia de regulación se exige a los operadores la publicación de un informe de calidad, con mediciones auditadas por el organismo regulador, este es el caso de países como el Reino Unido, España y Australia ([7]). Otros países están comenzando a plantear normativas similares, y están en proceso de puesta en práctica, como ser Colombia ([8]) y Nigeria ([9]); esto permite a los usuarios y clientes elegir el operador contando con información sobre la calidad del servicio prestado por cada uno de ellos. Otros países optaron por realizar mediciones directamente a través de su ente regulador, y aplicar sanciones a los operadores que no logran los objetivos impuestos, como es el caso de Chile. En Uruguay se adelantó que en principio no se prevé incluir valores objetivo en la reglamentación (al menos no para SMS).

Capítulo 6

Calidad de Experiencia (QoE)

6.1. Introducción

Hace ya unos pocos años, hemos empezado a ver un cambio de paradigma en el mundo de las telecomunicaciones. Pasamos de una ingeniería de protocolos y arquitecturas, a un campo de negocios orientado a los servicios y a los usuarios. Esto se manifiesta aún más con el IP Multimedia Subsystem (IMS), como plataforma para las Next Generation Networks (NGN). En estos casos, se pierde un poco de control sobre la red, y sobre los servicios que pasan sobre ella. Pasa a ser muy importante cómo el usuario final percibe la calidad del servicio que se le está brindando. Una pobre QoE (calidad percibida o calidad de experiencia) redundará en clientes insatisfechos, lo que a largo plazo lleva a una mala percepción de la empresa.

Encuestas indican que cerca del 82% de las deserciones de clientes son debido a la frustración sobre el producto o servicio y la incapacidad del proveedor para hacer frente a esta problemática¹. Por otra parte, esto lleva a una reacción en cadena en la cual, en término medio, cada cliente frustrado transmitirá a otros 13 su mala experiencia.

El operador no debe darse el lujo de esperar las quejas del cliente para evaluar el nivel de calidad de sus servicios. Las encuestas han demostrado que por cada persona que llama a su operador por un problema, hay otros 29 que nunca lo harán.

Es así que la única forma de prevalecer en esta situación es formular una estrategia para constantemente medir la QoE y mejorarla cómo y cuándo sea necesario.

Hay muchos factores que afectan en general la percepción del usuario: el costo, la fiabilidad, la disponibilidad, la facilidad de uso, la utilidad y la fidelidad ([10]). En nuestro caso, sólo veremos como parámetros la disponibilidad y fiabilidad, y no se considerará la disponibilidad geográfica, sino la temporal.

El concepto es fácil, si la QoE es buena, el cliente está contento y satisfecho, en cambio si no lo es, entonces el cliente estará insatisfecho. Esto se aplica para cualquier tipo de red y servicio.

En nuestro caso, nos enfrentamos a un servicio que a pesar de no estar dentro del grupo de tecnologías que tiene más desarrollo en esta área, tiene algunas cosas en común. La red celular no estaba originalmente pensada para transportar mensajes, éstos se transportan sobre la red de señalización, que tampoco estaba pensada originalmente para este tipo de utilización. Es muy difícil tener una idea de la calidad de servicio desde dentro de la red, y es más difícil aún cuando en el curso de los mensajes intervienen

¹Fuente: Accenture

varios elementos distintos.

Aprovechamos el desarrollo en el estudio de la percepción de calidad de los usuarios para distintos servicios, adaptándolo para el servicio SMS, e incorporándolo como parte de los resultados que entrega nuestro sistema. En particular exponemos en las siguientes secciones, los principales conceptos de calidad de experiencia analizados en el *Technical Report 126* del *DSL Forum* ([11]); muchos de estos conceptos fueron los que nos permitieron especificar un mecanismo para relevar la QoE para el servicio de mensajería corta.

6.2. QoE-QoS en IMS

En las plataformas donde se pretende dar al usuario servicios como voz, video a demanda, video broadcast, best effort data, etc, pasa a ser muy importante que los ingenieros diseñen teniendo en cuenta cómo es percibido el servicio por el usuario. En este sentido es que se define la QoE como una medida *end-to-end* de la performance del sistema desde el punto de vista del usuario y cómo éste satisface sus necesidades. A diferencia de QoE, QoS refiere a la performance del sistema desde el punto de vista de la red usada para brindar dicho servicio.

La calidad de experiencia y la calidad de servicio a menudo se utilizan indistintamente, pero en realidad son dos conceptos distintos. La calidad de experiencia es la performance global de un sistema desde el punto de vista del usuario, es una medida extremo a extremo del desempeño del nivel de servicio desde la perspectiva de los usuarios y una indicación de cuan bien el sistema cumple con las necesidades del usuario. Una métrica para la QoE es la “Mean Opinion Score” (MOS), ésta normalmente se utiliza como una medida subjetiva para cuantificar la percepción del impacto de diversas formas de degradación del servicio. En este sentido, la experiencia se expresa en cinco términos humanos: excelente, muy buena, buena, regular o mala.

Excellent
Very Good
Good
Fair
Poor

Figura 6.1: Niveles de QoE ([10])

En general existe una relación no lineal entre la QoE subjetiva, medida por el MOS y varios parámetros objetivos de rendimiento de los servicios (por ejemplo, la codificación de velocidad de bits, la pérdida de paquetes, *delay*, disponibilidad, etc), como se muestra en la figura 6.2.

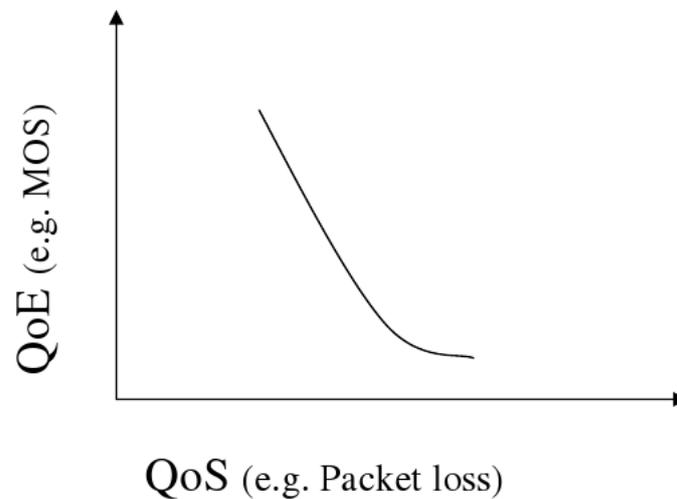


Figura 6.2: Relación entre QoE y QoS ([11])

Normalmente hay múltiples parámetros que indican la performance del nivel de servicio (QoS), y la relación entre los MOS QoE y los parámetros de QoS suelen ser derivados empíricamente. Una vez identificada dicha relación, puede ser utilizada de dos maneras:

- Dada una medición de QoS, se podría predecir la QoE esperada para un usuario
- Dado un objetivo QoE para un usuario, se puede deducir la performance requerida de la red para el servicio en cuestión.

6.2.1. Ingeniería basada en QoE

Para ilustrar la importancia de la calidad de experiencia en diversos sistemas multimedia, presentamos como ejemplo una aplicación al diseño de una red IMS. De esta manera, el proceso de ingeniería de una red IMS incluye:

- Análisis de los requerimientos finales del usuario.
- Definición de los requerimientos de QoE de la capa de aplicación.
- Mapeo de los requerimientos de QoE a los requerimientos de QoS de la red y a la capa de aplicación.
- Implementación de los requerimientos de QoS en los protocolos de capas, nodos y segmentos de red.

En la figura 6.3 se muestra un ejemplo de esta metodología, donde se observa que la ingeniería es basada en un enfoque *top-down*, comenzando desde los requerimientos finales del usuario.

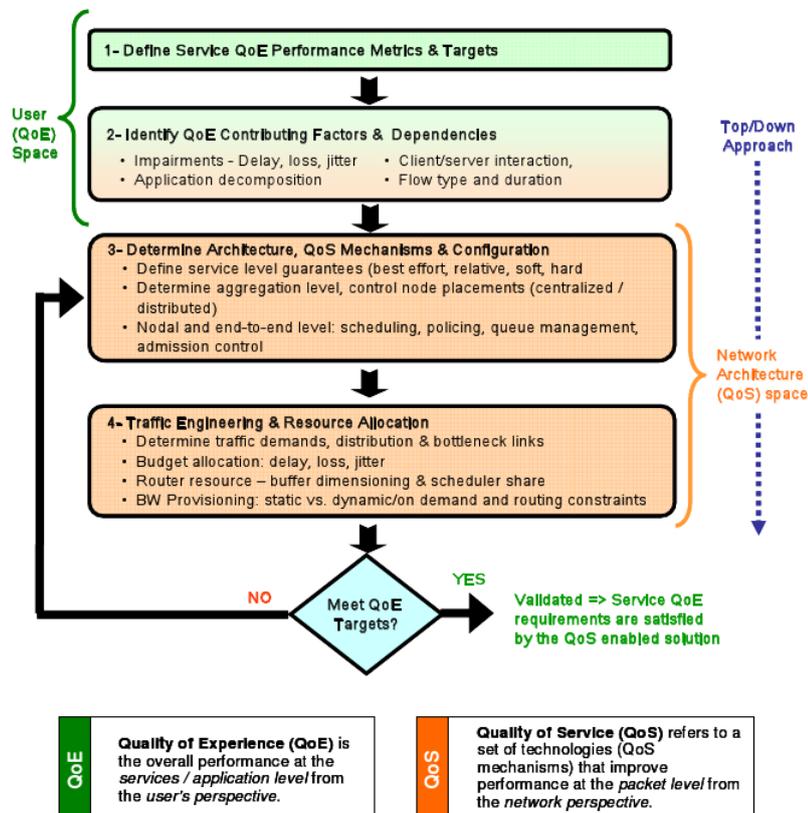


Figura 6.3: Ingeniería basada en QoE ([11])

6.2.2. Métrica QoE en función de parámetros de QoS

Para identificar los factores que subyacen la QoE para un servicio, hay tres aspectos de la prestación de servicios que deben considerarse:

1. El proceso de establecer un sesión (login on, marcando una llamada, etc)
2. Cómo funciona el servicio una vez que el período de sesiones se establece
3. Cierre de sesión (login off, etc.).

En cada uno de estos aspectos de la utilización de un determinado servicio o aplicación, hay múltiples facetas que contribuyen a la QoE. Entre ellas se incluyen:

- Esfuerzo requerido por el usuario
- Receptividad de la aplicación
- Fidelidad de la información
- Seguridad y fiabilidad
- Disponibilidad

El análisis de un determinado servicio con estos conceptos determina requerimientos que varían dependiendo el tipo de servicio (voz, video a demanda, etc) como delay, perdida de paquetes, jitter, etc. En la *TR-126* se analizan estos tipos de servicio en detalle.

6.3. QoE en SMS e implementación en el AST

Si bien las aplicaciones de IMS incluyen una variedad de servicios y parámetros, el estudio del servicio de mensajería corta se hace más sencillo, dado que consta de pocos parámetros. El concepto a utilizar es similar: se generan parámetros de QoE (también llamados KPIs), que por cierto serán mucho más simples que los KPI para los tipos de servicio mencionados anteriormente. Sin embargo, por las características mencionadas en la introducción, el mapeo de sus necesidades sobre la red GSM no es una tarea trivial, y aquí es donde entra en juego principalmente el conocimiento sobre el comportamiento de los usuarios.

El KPI principal considerado en este caso es el retardo del SMS enviado por el usuario (la pérdida del SMS se considera como un caso particular con retardo infinito). El modelo del cálculo de la calidad de experiencia para un usuario dado se basa en el modelo del comportamiento del usuario explicado en el capítulo 4. Con el fin de evaluar el impacto que tiene el retardo de un SMS teniendo en cuenta la importancia de dicho SMS para el usuario, se definen cuatro umbrales para cada estado S_i , el cual representa la tolerancia del usuario respecto al retardo que tenga el SMS. De esta forma se divide el retardo con el cual el SMS llega a destino en cinco regiones, las cuales determinan la experiencia del usuario (excelente, muy buena, etc.).

Además, en cada momento se tiene en cuenta la historia reciente de la experiencia del usuario. Específicamente, se toman los últimos n mensajes enviados por el usuario en los últimos m minutos², y se calcula la experiencia del usuario en cada uno de esos mensajes. Se decide la calidad de experiencia del usuario en el instante en función de estas n experiencias, teniendo en cuenta algún criterio. En la implementación actual, se toma (arbitrariamente) el mínimo de las experiencias (la peor de ellas), considerando que el malestar por una mala percepción es más persistente que una experiencia “normal”.

Se desprende de lo anterior que el resultado de la experiencia del usuario es una función no lineal de los retardos de los SMS. En efecto la no linealidad está dada por dos factores; por un lado se tiene en cuenta el estado del usuario correspondiente a cada mensaje, y por otro lado, dadas las últimas experiencias de cada usuario, éstas se combinan de una forma no lineal.

A efectos de disparar alarmas en tiempo real, se prevé que el operador configure varios usuarios de distintos tipos que virtualmente envíen mensajes, para que sean procesados y calculadas sus experiencias. Luego, se combina la información de todos estos usuarios, para decidir si se envía una alarma o no. En la implementación realizada, se cuenta la cantidad de usuarios con cada nivel de experiencia, se definen 5 ponderadores (configurable por el usuario desde la interfaz gráfica) p_i , y se efectúa el siguiente cálculo:

$$QoE_{pond} = \sum_{i=1}^5 p_i |U(i)|$$

²Ambos parámetros, n y m , son configurables

donde $|U(i)|$ es la cantidad de usuarios con experiencia i (codificando con un valor entero entre 1 y 5 los 5 tipos de experiencia: excelente se mapea a 5, y *mala* a 1).

Si el valor de QoE_{pond} supera cierto umbral predefinido por el operador, se dispara una alarma correspondiente a una mala calidad de experiencia.

Por otro lado, en los informes de QoE se puede observar el desempeño de la red desde el punto de vista de los usuarios para un determinado intervalo de tiempo. En estos informes, por un lado se explicitan las cantidades $|U(i)|$, y por otro lado se grafican las mismas en función del tiempo. Esto resulta en 5 curvas que ayudan a visualizar la percepción de los usuarios en cada momento. Ejemplos de estas curvas se encuentran en el capítulo 10, donde se analizan los resultados.

Parte III

Diseño e implementación

Capítulo 7

Diseño

En este capítulo tratamos específicamente el diseño de software realizado. El buen diseño de software, en conjunto con una estructura de orientación a objetos, permite no sólo un trabajo ordenado, sino que facilita la futura modificación y expansión del software. Además, permite la reutilización de código, favoreciendo el desarrollo en menor tiempo.

Si bien en este proyecto se dedicó una parte muy importante de los recursos de tiempo al buen diseño, presentamos aquí solamente una versión muy simplificada a efectos de que la lectura no sea tan ardua.

Inicialmente se plantearon casos de uso, modelo de dominio, diagramas de secuencia, diagramas de clases, etc. Una vez claros los objetos participantes, y las funciones que había que realizar, se comenzaron a dividir las funciones más importantes.

A partir de éstas, se separaron las funcionalidades en los bloques más importantes:

- El módulo de **Mensajería**, que incluye lo relativo a la administración de las interfaces (Móviles, Gateways, etc) y al envío, recepción y almacenamiento de mensajes de texto a través de las mismas.
- El módulo de **Procesamiento de Datos**, que implementa la generación de usuarios, mediciones de calidad, y generación de informes.
- El módulo **SNMP**, encargado de gestionar las conexiones mediante este protocolo.
- La **Interfaz Gráfica de Usuario (GUI)**. Es el nexo entre el operador y el sistema. A través de ella el operador accede al sistema pudiendo configurarlo y gestionarlo.
- La **Base de Datos**, que es donde se almacenan los mensajes y los datos de configuración del sistema.

En la figura 7.1 se ilustran las interacciones entre los módulos, y del usuario con el sistema.

En particular, los cuatro primeros bloques son implementados en lenguajes de alto nivel orientados a objetos. Los tres primeros módulos son implementados como parte de un bloque más grande en un lenguaje, y el cuarto está implementado en otro bloque independiente, en otro lenguaje. El quinto módulo es naturalmente independiente de los anteriores.

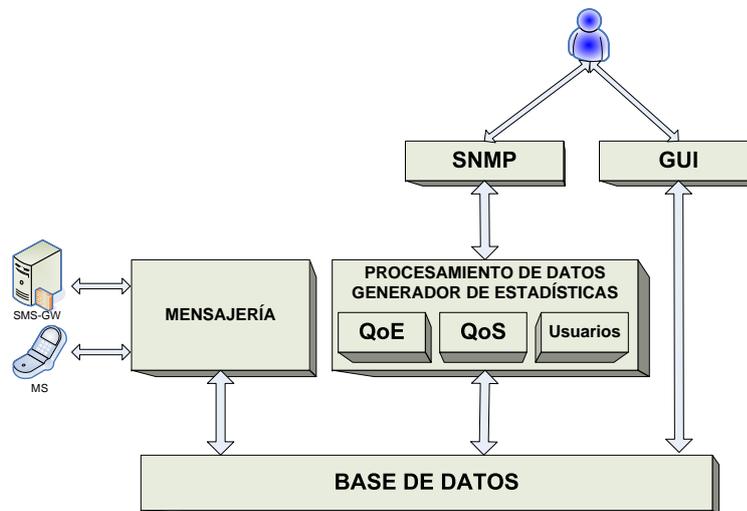


Figura 7.1: Módulos principales del sistema

Una vez claras las funcionalidades de cada bloque, se diseñó la interfaz entre cada uno de ellos, y luego se llevó a cabo diseño interno de cada módulo. Si bien en un principio se trabajó con intensidad en el diseño, se realizó un trabajo iterativo entre el diseño y la implementación.

Describimos a continuación los aspectos generales más importantes del diseño de cada módulo, e incluimos en el CD un diagrama de clases completo del sistema.

7.1. Mensajería

El módulo de mensajería comprende todo lo relativo al envío y recepción de mensajes. Listamos en lo que sigue los principales conceptos que se reconocen, y cómo se relacionan entre sí.

Interfaces

El primer concepto fundamental que identificamos es el de Interfaz¹ con la red GSM. Nuestro sistema básicamente cumple la función de enviar y recibir mensajes. Para ello debe estar interconectado a la red GSM por distintos puntos, cada uno con distintos protocolos de comunicación. Para poder interconectarnos a estos puntos, debemos construir las interfaces específicas para cada uno de ellos. Si bien los protocolos son distintos, las funciones básicas de cada interfaz son comunes. Construimos entonces el concepto general de interfaz, que contiene estos elementos comunes, de la que cada interfaz específica hereda. Dicha clase la denominamos *InterfazGeneral*.

Las funcionalidades comunes nombradas anteriormente son básicamente las de envío y recepción de mensajes, y los atributos comunes son por ejemplo el número de telefonía asociado.

Los dos protocolos implementados son *AT* para la comunicación con terminales móviles, y *SMPP* para la comunicación con otros equipos vía TCP/IP. Cada uno de

¹En este capítulo hacemos referencia al término *interfaz*, como interfaz con la red GSM, y no al concepto de diseño utilizado en orientación a objetos

los protocolos se implementó en una clase que hereda de la clase *InterfazGeneral*, denominadas *InterfazMS* e *InterfazGW*.

La conexión con los terminales móviles es mediante el puerto serial USB, por lo tanto, algunas de las particularidades de la interfaz con los terminales móviles son el puerto físico al cual está conectado (tty), además de generalidades del MS como el IMSI, el IMEI, el modelo del terminal, etc.

La conexión con los equipos sobre TCP/IP tendrá naturalmente una dirección IP y un puerto, además de un usuario y una contraseña.

Los métodos que implementan cada una de ellas son los mismos, por ejemplo enviar o recibir un mensaje, cambiando naturalmente cómo es implementado en cada una de ellas.

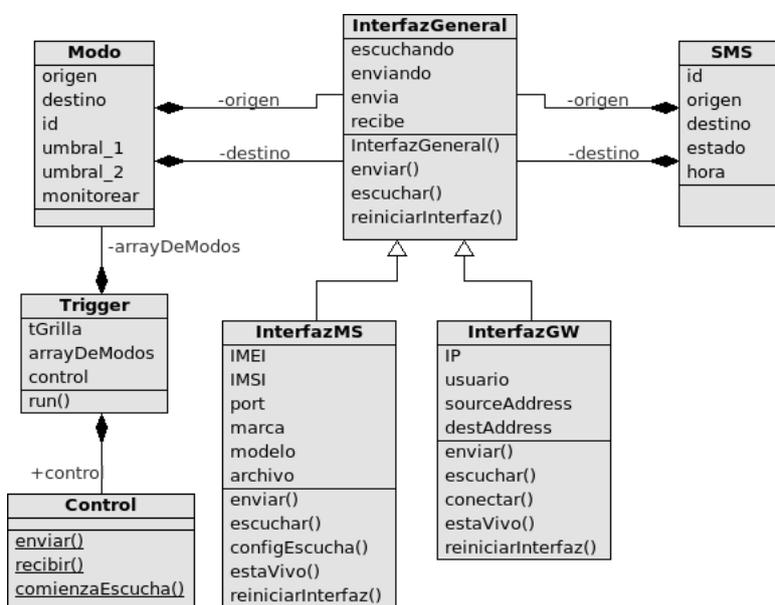


Figura 7.2: Diagrama de clases (simplificado) del paquete mensajería

Modos

Como fuera anticipado en el capítulo 3, el concepto de modo refiere a una asociación de interfaces origen y destino. Para implementar este concepto, se creó una clase *Modo*, que tiene como atributos dos objetos *InterfazGeneral*, una de origen, y otra de destino. Este concepto pasa a ser fundamental, al punto que para configurar el monitoreo que realizará el sistema se configura el o los modos a utilizar, ya que cada modo es, en definitiva, un camino a testear, de los comentados en la sección 3.1.

SMS

Surge naturalmente la necesidad de contar con una clase que represente los mensajes; a ésta, la denominamos *SMS*. Incorpora, entre otros, atributos como el origen y el destino del mensaje, un identificador, etiquetas de tiempo, etc.

El contenido del mensaje (el texto en sí) será un número de secuencia.

Control

Se creó una clase que se encarga de controlar el tránsito interno de mensajes, y de dejar un registro de éstos en la base de datos. Esta clase se llama *Control*, y es la que da la orden a las interfaces de enviar un mensaje. Además, las interfaces al

recibir un mensaje se lo transmiten a *Control*, quien centraliza la administración y el almacenamiento.

Trigger

Contamos entonces con interfaces que permiten enviar y recibir mensajes a través de la red, modos que indican origen y destino de cada mensaje y un administrador que centraliza los mensajes. Se debe asignar ahora la responsabilidad de originar los mensajes a enviar. Para cumplir esta función se creó la clase *Trigger*, la cual periódicamente (según la configuración en vigencia) creará los mensajes y por medio de *Control* hará efectivo el envío. Básicamente tiene como atributos el tiempo entre disparo de mensajes (llamado *tGrilla*), y un array con todos los modos activos.

Los objetos aquí descriptos cumplen las funcionalidades básicas que busca este paquete. Repasemos entonces cómo es el flujo de acciones normal en este paquete:

- El objeto *Trigger* verifica en su array de modos cuál es el siguiente *Modo* a ejecutar
- Crea un nuevo *SMS*, cuyo origen y destino coinciden respectivamente con el origen y destino del *Modo*
- *Trigger* envía ese mensaje a *Control*
- *Control* registra el mensaje en la base de datos y la envía a la interfaz de origen para que sea enviada a la red
- La interfaz pone el mensaje en la red GSM
- El mensaje es recibido por la interfaz de destino y se pasa a *Control*
- *Control* registra la llegada del mensaje en la base de datos
- Cada un tiempo *tGrilla* se vuelve al principio

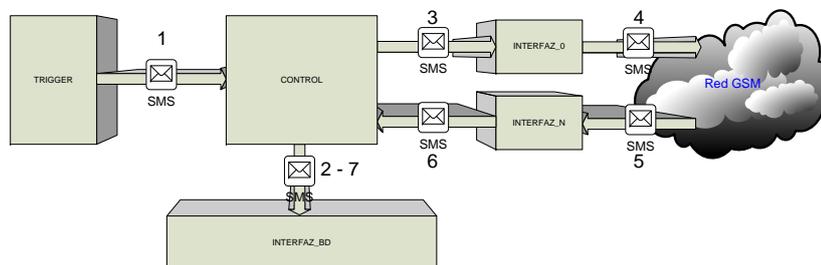


Figura 7.3: Diagrama de bloques del módulo de mensajería

En la figura 7.3 se puede ver el funcionamiento general del módulo mensajería, y en la figura 7.4 se ilustra el diagrama de secuencia correspondiente, que se repite para cada *Modo* en el array.

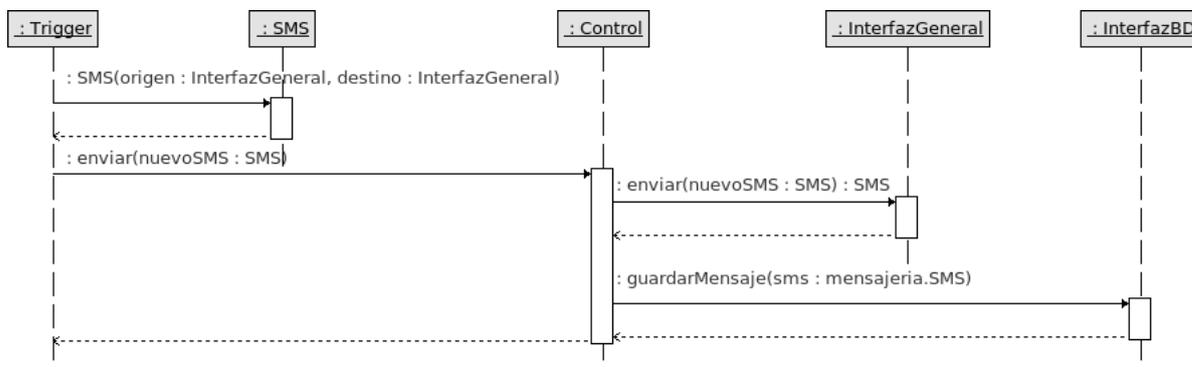


Figura 7.4: Diagrama de secuencia

7.2. Módulo de procesamiento de datos

En este bloque nos independizamos de cómo fue el proceso de muestreo de la red, y suponemos que en la base de datos ya se encuentra la información. De hecho, este módulo es independiente del módulo de mensajería, dado que el pasaje de información entre ambos módulos se limita simplemente al registro de mensajes en la base de datos.

Describimos primero el bloque correspondiente de implementar el modelo de usuarios del capítulo 4, y luego el bloque correspondiente a realizar las mediciones de calidad de los capítulos 5 y 6.

7.2.1. Usuario

El sistema será configurable para poder ingresar distintos tipos de usuarios, cada uno con un perfil distinto de utilización de la red. Tenemos entonces una clase *Tipo*, que representa un conjunto específico de parámetros del modelo desarrollado.

La clase *Tipo* tendrá entonces como atributos un conjunto de *Estados* y una matriz de transición.

Cada *Estado* por otra parte, tendrá una distribución asociada (por ejemplo exponencial, uniforme, etc), un parámetro para esa distribución, y un conjunto de umbrales de QoE.

Luego tenemos *Usuarios*, cada uno con un determinado *Tipo*, y con una determinada calidad de experiencia instantánea (que será producto del desempeño de la red en los mensajes que “envíe” el usuario²).

Estos usuarios serán manejados centralmente por una clase *ControlDeUsuarios*, encargada de crearlos y proporcionar una interfaz para consultar las calidades de experiencia de cada uno.

7.2.2. Medidores de calidad

Las mediciones de calidad de servicio y experiencia se realizan desde dos clases, *MedidorQoS* y *MedidorQoE* respectivamente. Ambas clases deben disparar una alar-

²Como fue explicado anteriormente, el usuario sortea (según el modelo) los tiempos en los cuales envía mensajes, y el desempeño de éstos es tomado del muestreo de la red realizado extrayendo la información de la base de datos. Esta información se mezcla con los parámetros propios del Usuario (umbrales), y resulta un valor de calidad de experiencia.

ma ante incumplimientos en umbrales predefinidos, y por lo tanto hacen uso de esta funcionalidad que brinda el módulo *SNMP*.

La clase correspondiente a la medición de la calidad de servicio simplemente se encargará de consultar periódicamente la base de datos, obteniendo resultados acerca de los últimos mensajes de cada modo. Es por lo tanto, una clase bastante independiente.

Por otro lado, la clase que brinda las mediciones de calidad de experiencia tiene acceso a la instancia de *ControlDeUsuarios* a efectos de interrogar a cada uno de los *Usuarios* acerca de su experiencia (cada *Usuario* tendrá un nivel de experiencia, de acuerdo a los parámetros del modelo correspondientes al *Tipo* al que pertenece, y al desempeño de la red en los mensajes). La información de la experiencia de todos los *Usuarios*, además de poderse consultar, se compara con los umbrales definidos y se generan alarmas según corresponda.

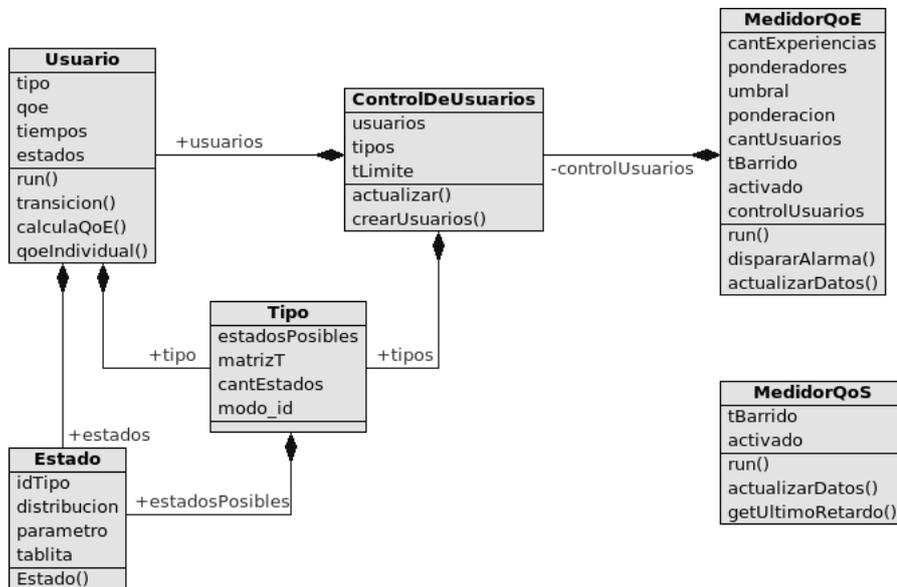


Figura 7.5: Diagrama de clases del paquete Usuario

7.3. Módulo SNMP

Una de las principales tareas que efectúa el AST es el cálculo estadístico de los parámetros de QoS y QoE, que se obtienen de los retardos de SMS y de la simulación de la interacción de los diferentes tipos de usuarios con la red GSM. Estos datos son registrados, calculados en tiempo real y almacenados en la base de datos del sistema. Otro punto importante, es la generación de alarmas ante el incumplimiento de ciertos umbrales en los parámetros de QoS-QoE.

Como características planteadas en el diseño, cualquier usuario del sistema debe tener acceso remotamente a las estadísticas obtenidas en tiempo real, así como la obtención de informes históricos de la QoS de la red y de la percepción del servicio por parte de los usuarios (QoE). Además, el sistema debe generar las alarmas correspondientes ante el incumplimiento de parámetros de QoS-QoE hacia algún sistema de monitoreo. Con tal objetivo, hemos implementado como parte de la aplicación, un agente SNMP el cual envía notificaciones de eventos y permite mediante un cliente SNMP consultar los datos que interesen.

SNMP (Simple Network Management Protocol) es un protocolo de capa de aplicación de la familia TCP/IP el cual permite la gestión y administración de dispositivos de red, por defecto utiliza los puertos UDP 161 y 162, y es un estándar de la IETF, RFC 1157. El mismo funciona como una aplicación del tipo cliente-servidor. El Agente (servidor) tiene almacenada su información en una estructura jerárquica tipo árbol llamada MIB (Management Information Base), la misma contiene las variables a ser consultadas (OIDs, Object Identifiers) y las que se pueden enviar en Traps (mensajes que pueden ser enviados por el agente en cualquier momento).

En el capítulo 8 se explican los detalles de implementación y la MIB del sistema.

7.4. GUI

Como ya vimos en la estructura general del software, la GUI corre como un proceso independiente, y se comunica con el resto del software mediante la base de datos.

La GUI se estructuró en cuanto a su layout, conteniendo objetos distintos para el encabezado de la página, menú lateral, pie de página, y contenido en sí. Esto permitió concentrarse en el desarrollo del contenido, sin preocuparnos por los menús, y encabezados.

A su vez se generó una estructura de menús, que dan una estructura natural a la gestión del software, dividiéndolo en 4 secciones importantes: *Estado del sistema*, *Configuración*, *Estadísticas*, y *Ayuda*.

En la sección Estado del Sistema, el usuario puede visualizar el estado del hardware asociado, y los recursos consumidos; además, puede ver el estado del software, esto implica ver si todos sus procesos están operativos, y en qué condiciones está corriendo. Es básicamente un pantallazo rápido de cómo está funcionando el sistema.

En la sección Configuración se establecen los tiempos involucrados (timeOuts, tiempos de grilla), los móviles, se configuran y se activan los modos, se parametriza la simulación de usuarios, así como los umbrales elegidos para QoE y QoS, y las configuraciones básicas de SNMP.

La sección de Estadísticas permite al usuario recopilar estadísticas de las mediciones realizadas, y en la sección de ayuda, naturalmente el usuario puede informarse sobre la utilización de la GUI y del software en general.

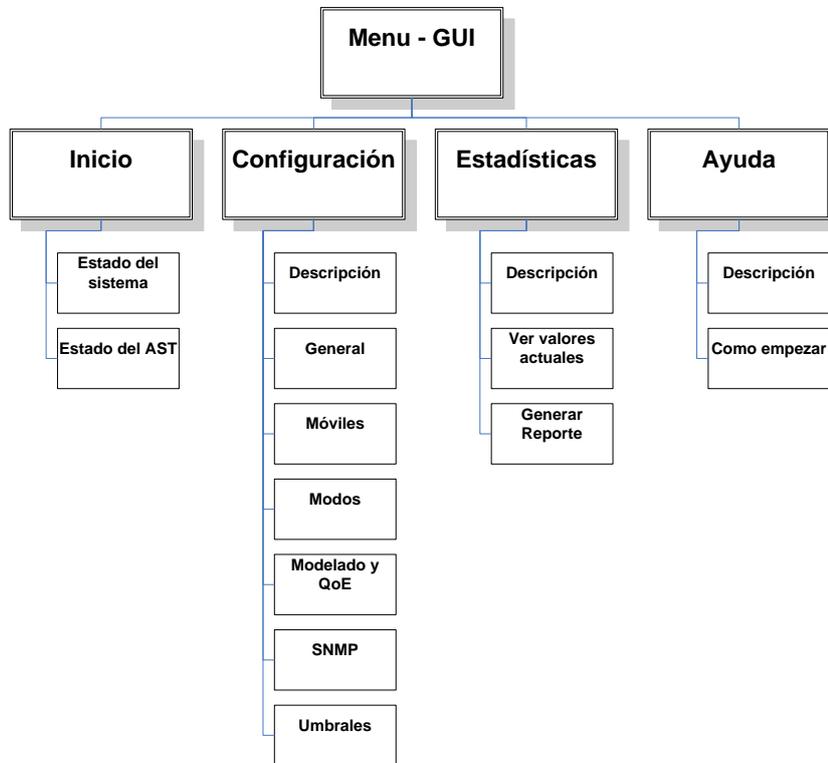


Figura 7.6: Esquema de secciones de la interfaz gráfica

Referimos al lector al apéndice B, donde son detallados tanto el contenido como el funcionamiento de la interfaz.

7.5. Base de datos

La base de datos sigue las líneas de diseño del resto del software, incorporando tablas que contienen a los distintos objetos ya mencionados en secciones anteriores.

Mensajes

La tabla más importante de la base de datos es la tabla de *mensajes*. En ella, se guardan todos los mensajes que han sido procesados por el sistema, estén recibidos o no. Para cada mensaje, que corresponde a un registro en esta tabla, se guarda su estado (Enviado, recibido, con error), la hora de enviado y de recibido, el retardo total de ida y vuelta, y cuál fue su origen y destino.

CelularesGW

En esta tabla se guardan los datos correspondientes a las interfaces que dispone el sistema para realizar el envío y recepción de mensajes. Se utiliza la misma tabla para los dos tipos disponibles de interfaces, para móviles y para conexiones SMPP. En ella se guarda información como ser sus capacidades, y datos característicos de la misma, como ser el IMSI, y el IMEI en el caso de los móviles, o la IP y el usuario en el caso de las conexiones SMPP.

Modos

Cada registro de la tabla de *modos*, representa el concepto que ya mencionamos en capítulos anteriores, el modo. La idea es simplemente guardar cuál es la interfaz de origen y de destino para cada modo. Además se guarda información como ser si el modo está activo, si hay que realizar monitoreo para este modo, etc.

Usuarios, Estados y TiposMatriz

La tabla de *usuarios* hace referencia a los usuarios que están corriendo instantáneamente para medir la calidad de experiencia. Este concepto ya lo analizamos en el sección de usuarios. A su vez, cada usuario tiene asociado un tipo y un modo, referenciados mediante identificadores a las respectivas tablas. En la tabla *tiposMatriz* se registra la matriz de transición de estados para los distintos tipos de usuarios, y en la tabla de *estados* se registran los estados posibles y sus parámetros.

DatosNuevos, Configuración, ConfQoES

Estas tablas no representan en sí un objeto de diseño, pero hacen referencia a configuraciones que quedan registradas.

En el diagrama 7.7 se representan las tablas que aquí describimos y sus relacionamientos.

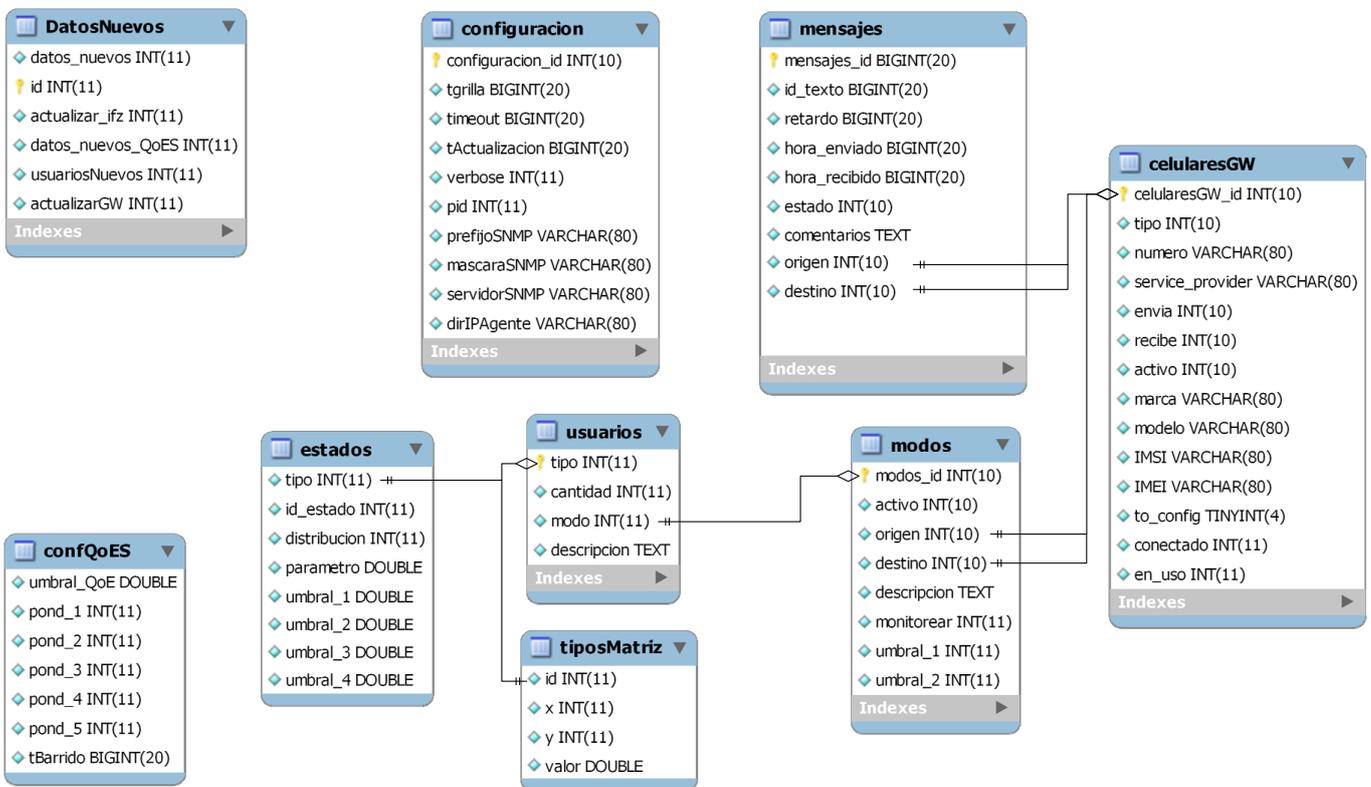


Figura 7.7: Tablas de la Base de Datos, y sus relacionamientos.

Capítulo 8

Implementación

En el capítulo anterior describimos algunos aspectos generales acerca del diseño del software. En este capítulo, comentamos algunas consideraciones que se tuvieron a la hora de la implementación, así como los problemas que surgieron y las soluciones empleadas.

Una característica general, tanto de nuestra implementación como de las librerías utilizadas, es el espíritu del licenciamiento libre. En efecto, utilizamos tanto para el desarrollo como para la ejecución de la versión final, un sistema operativo Linux, lenguajes y entornos de programación libres, y otras herramientas de igual característica; por otro lado, todos los paquetes externos tienen algún tipo de licencia libre.

Respecto a los bloques lógicos, conservamos el orden seguido en el capítulo 7, pero englobamos los primeros módulos en una sección debido a que están implementados en un mismo lenguaje, y son parte de un proceso más grande. Específicamente, los bloques de *Mensajería*, *Procesamiento de Datos* y *SNMP* fueron programados en Java, además de otros módulos adicionales, que no fueron mencionados en el capítulo anterior debido a su importancia menor.

Empezamos describiendo la aplicación en Java, luego la Interfaz Gráfica (implementada en php), la Base de Datos (que utiliza *MySQL*), y por último el Proxy SMPP-HTTP descrito en la sección 3.1.

8.1. Aplicación en Java

La parte principal del software corre en Java. Elegimos este lenguaje ya que brinda algunas comodidades a la hora de programar, teníamos experiencia con él, es orientado a objetos y no teníamos requerimientos de performance que nos hicieran inclinar por un lenguaje más eficiente como C++.

Dado que en esta aplicación se necesita procesar varias actividades en forma simultánea, se hizo uso de Hilos de Ejecución (o *Threads*) para administrar la concurrencia de procesos. A pesar de no ser particularmente crítico, manejamos con cierto cuidado las prioridades de ejecución de los hilos.

8.1.1. Generalidades

La aplicación se inicia desde la clase *Main*, que es la que se ocupa, además de la inicialización, de articular y administrar los distintos módulos.

Esta clase crea varios *Threads*, de los cuales sobresalen: *Trigger*, *MedidorQoS* y *MedidorQoE*. *Trigger* carga la configuración de *modos*, y se encarga de enviar periódicamente los mensajes correspondientes a cada modo¹, en una estructura de loop. *MedidorQoS* y *MedidorQoE* monitorean respectivamente la calidad de servicio y de experiencia, en tiempo real, y generan alarmas SNMP en los casos correspondientes.

8.1.2. Mensajería

Comentamos en esta sección algunos aspectos que tuvimos en cuenta en la implementación del bloque de *Mensajería*, que fue el que insumió la mayor cantidad de horas de programación y testeó.

Terminales Móviles

Los terminales móviles se pueden conectar al sistema mediante un puerto USB o mediante un puerto serial RS-232. Para poder mantener comunicaciones desde Java con los puertos seriales, utilizamos una librería denominada *RxTx* ([12]), que implementa esta funcionalidad para sistemas Linux bajo una licencia GPL. En dicha comunicación intervienen aspectos finos de hardware, y depende fuertemente de cómo el sistema operativo maneja los puertos, y por lo tanto requiere cierto esfuerzo realizar un sistema multi-plataforma. Es por esto que nos limitamos a trabajar en sistemas Linux para esta parte de la aplicación.

La librería *RxTx* nos brinda la posibilidad de escanear los puertos seriales (archivos *tty* en el caso del sistema operativo Linux) en busca de hardware conectado, setear la velocidad de trabajo y abrir el puerto para obtener streams de entrada y salida.

Encontramos que algunos móviles distintos conectados por puerto USB creaban distintos tipos de archivos en el sistema. Por ejemplo, los modelos V3 de Motorola creaban archivos “*ttyACM**”, mientras que terminales de Sony Ericsson creaban archivos “*ttyUSB**”. Pero la librería *RxTx* no escaneaba los puertos “*ttyACM*” por ejemplo, por lo que tuvimos que modificar el código fuente de la misma, e incluir cambios para que la librería se ajustara a nuestras necesidades.

Al iniciar el sistema se realiza un escaneo de todos los puertos, y si se reconoce el dispositivo conectado como un terminal móvil se crea una instancia de *InterfazMS* y se almacena en un array.

Utilizamos el protocolo AT ([13],[14]) para enviar comandos a los terminales, enviando y recibiendo las cadenas de caracteres mediante los streams mencionados. Creamos entonces en la clase *InterfazMS* un método *void escribir(String s)*, que envía la cadena contenida en “*s*” al dispositivo asociado. Implementamos también un método para leer el flujo proveniente del terminal.

Cuando se crea la instancia de *InterfazMS* se interroga al dispositivo sobre los números identificadores del equipo y de la tarjeta SIM. A modo de ejemplo, la sección del código correspondiente luce como sigue:

¹Como fuera descrito anteriormente, *Trigger* crea el objeto SMS, y utiliza un método de *Control* que se encarga del resto.

```
escribir("AT+CGMI");
imsi = leer();
escribir("AT+CGSN");
imei = leer();
```

Para enviar un mensaje de texto se utiliza el comando *AT+CMGS*, y hay varios comandos para configurar la recepción de mensajes. Existen dos modos para enviar y/o recibir mensajes, uno en texto plano y otro en hexadecimal, éste último con una estructura compleja². Dado que algunos terminales poseen solamente el modo hexadecimal, implementamos, con mucho trabajo, los dos modos.

El método cuyo encabezado es *SMS enviar(SMS mensaje)*, hace efectivo el envío del mensaje de texto a la dirección *destino* del objeto SMS, y devuelve un SMS con algunos atributos modificados (como la hora de envío y el estado del mensaje).

Para la recepción de mensajes creamos un *Thread*, ya que se debe consultar permanentemente si el stream de entrada tiene datos esperando a ser leídos. En este caso, se procesan los datos y se crea un objeto SMS, con el cual se llama a un método de la clase *Control* que se encarga de procesarlo.

Interfaces SMPP

Para implementar la interfaz SMPP se utilizó el paquete SMPP de *Logica* ([16]), que se encuentra bajo la licencia *Logica Open Source License*. Sobre el paquete de *Logica* se creó una interfaz con el resto del paquete *Mensajería*, implementando además rutinas de chequeo de la interfaz para asegurar la estabilidad en forma autónoma.

A continuación realizamos un pequeño resumen de algunas de las características de este protocolo.

SMPP es un protocolo de capa de aplicación diseñado exclusivamente para el envío y recepción de SMS. Su origen proviene de una empresa irlandesa (*Aldiscon*), compañía que luego fuera comprada por *Logica*. El protocolo luego fue tomado por el *SMS forum* para ser estandarizado.

Fue diseñado en particular para la comunicación entre centros de mensajería y ESMEs (External Short Message Entities). Los ESMEs pueden ser por ejemplo proveedores de servicios de valor agregado.

Existen varias formas de establecer una sesión de comunicación SMPP:

- Sesión iniciada por el SMSC. Este tipo de sesión es llamada OutBind Session y es poco común.
- Sesión iniciada por el ESME. Este tipo de sesión es la implementada normalmente. En particular, se puede establecer la sesión de tres maneras distintas:
 - TX - Transmitter. Cuando el ESME inicia este tipo de sesión puede realizar envío de mensajes a móviles. Además puede cancelar y consultar el estado de mensajes enviados anteriormente.
 - RX - Receiver. Este tipo de sesión sólo permite al ESME recibir mensajes.

²Por ejemplo, para codificar/decodificar mensajes hay que desarmar las tiras de 8 bits, agrupar de a 7 bits no consecutivos, simetrizar, etc ([15])

- TRX - Transceiver. Este tipo de sesión permite tanto el envío como la recepción de mensajes. Este es el tipo de sesión que establece nuestro software con el SMSC.

Ambas entidades de la sesión manejan el mismo tipo de PDU (Protocol Data Unit). Cuando una entidad envía un PDU, la otra envía otro confirmando la recepción del mismo, e informando el resultado de la petición. Por ejemplo, si un EMSE envía una petición de envío de mensajes *submit_sm*, el SMSC contesta con un *submit_sm_resp* informando sobre el éxito o no del pedido.

Los distintos tipos de PDU se dividen en grupos;

- Manejo de sesión. Conjunto de PDUs que permite el control sobre la sesión, como ser el inicio de sesión y finalización de la misma.
- Envío de SM. Conjunto de PDUs utilizados para el envío de mensajes
- Entrega de SM. Utilizados para la recepción de SM.
- Broadcast de SM. Utilizados para hacer envío de mensajes tipo Broadcast. En particular no implementamos este tipo de PDUs.
- Operaciones auxiliares. Permiten reordenar envío de mensajes, cancelar mensajes todavía no enviados ya solicitados, consultar el estado del mensajes, etc. Estas operaciones tampoco fueron implementadas en nuestro software.

Hubo un trabajo de adaptación del paquete de *Logica* con la interfazGW, en particular se debió codificar el PDU en la *InterfazGW*, para ser enviado y decodificar al recibirlo. En la etapa de codificación, se debieron setear algunos campos adecuadamente con los bits correspondientes, implicando un análisis detallado del protocolo para su implementación. Por otro lado, la decodificación del PDU por parte del paquete de *Logica* es confusa, sobretodo su implementación en objetos, por lo que tuvimos que re-implementar algunas secciones de la decodificación.

8.1.3. Procesamiento de Datos

Esta sección del sistema se encarga básicamente de procesar la información obtenida, de proveer una interfaz donde consultar en tiempo real los resultados, y de generar informes específicos.

Los cálculos que intervienen son simples, pero en algunos casos la cantidad de datos puede ser grande, por lo que intentamos aprovechar las facilidades que nos proveen las consultas MySQL para optimizar el proceso. Recordemos que los datos a procesar se encuentran en una base de datos *MySQL* (básicamente la información de mensajes enviados y recibidos). Todas las tareas aquí descritas fueron probadas con una cantidad de datos razonablemente grande, y el tiempo de respuesta del sistema permaneció en el orden de segundos, o algún minuto en un caso extremo.

Este paquete debe ofrecer los resultados de medidas de QoE y QoS, por lo tanto una parte importante del procesamiento consiste en implementar el modelo de comportamiento de usuarios tratado en el capítulo 4, y diseñado en el capítulo 7. La implementación de la estructura no ofrece mayores dificultades, por lo que solamente dedicamos algunas líneas a la simulación de las variables aleatorias involucradas.

Simulaciones de Variables Aleatorias

Java cuenta con un generador de números pseudo-aleatorios en el paquete *java.util.Random*. En general, estos generadores (tanto el de Java como el de C o C++) son de calidad cuestionable³, pero dado que en nuestro sistema la calidad del generador no es crítica (como sí lo sería en un sistema criptográfico por ejemplo), estimamos que no vale la pena implementar por cuenta propia un generador fiable.

Con este generador entonces, obtenemos números pseudo-aleatorios con distribución uniforme en el intervalo $[0, 1]$. Obtener a partir de éste, una simulación de una variable aleatoria con distribución uniforme en $[a, b]$ (una de las previstas en el modelo) es trivial.

Consideremos ahora el problema de generar una simulación de una variable aleatoria con distribución exponencial, a partir de una uniforme $[0, 1]$. Para esto utilizamos el siguiente resultado:

Proposición 8.1.1. *Sea F una función de distribución de probabilidad, F^{-1} su inversa definida como sigue:*

$$F^{-1}(u) = \inf\{x : F(x) = u, 0 < u < 1\}$$

y U una variable aleatoria uniforme $[0, 1]$. Entonces la nueva variable aleatoria $F^{-1}(U)$ tiene función de distribución F .

En particular, en el caso de la distribución exponencial la inversa de la función $F(x) = 1 - e^{-\lambda x}$ es $F^{-1}(u) = \frac{-\log(1-u)}{\lambda}$. Por lo tanto, a partir de una variable aleatoria U uniforme $[0, 1]$, la variable aleatoria $X = \frac{-\log(1-U)}{\lambda}$ tiene distribución exponencial.

En este caso particular cabe una simplificación: si U es uniforme $[0, 1]$, entonces $1 - U$ también es uniforme $[0, 1]$, por lo que podemos generar una variable X con distribución exponencial como $X = \frac{-\log(U)}{\lambda}$

En virtud de esto, una posible implementación sería la siguiente:

```
public static long exponencial(double lambda)
{
    double u = r.nextDouble();
    return (long) -Math.log(u)/lambda;
}
```

Generación de informes

Se eligió el formato estándar *pdf* para la elaboración de informes. En particular, éstos son generados con el lenguaje \LaTeX . Por lo tanto, el PC sobre el cual corre el sistema debe tener instalado un compilador para este lenguaje. En nuestra implementación, utilizamos el paquete *texlive-latex-base*, disponible en los repositorios.

Se genera un archivo *.tex* desde Java, a medida que se van procesando los datos, y luego se ejecuta el comando para compilar el archivo y guardarlo en un directorio determinado. Los comandos son ejecutados también desde Java, utilizando la clase *Runtime* para acceder al intérprete de comandos *shell* en */bin/sh*.

Utilizamos, además, el paquete *jfreechart* ([17]) para generar gráficas. Las mismas son creadas en el momento del procesamiento, y son guardadas en formato *png* para ser incluidas en el archivo *.tex*.

³en cuanto a su predictibilidad

Se crea entonces un archivo *pdf* que contiene todos los datos numéricos relevantes, y las gráficas correspondientes, de manera que sea accesible remotamente desde la interfaz gráfica. De hecho, un caso de uso normal sería el siguiente: el usuario consulta a través de la interfaz gráfica sobre el desempeño de la red en determinado período, y para determinados *modos*; la interfaz gráfica consulta a la aplicación Java, quien procesa los datos, genera el pdf y le responde con la dirección del archivo, el cual es mostrado al usuario en forma de *link* para descargar el informe.

8.1.4. Módulo SNMP

Este módulo consiste en un agente SNMP, el cual se implementó compatible con la versión 1, todos los OIDs los hemos implementado como sólo lectura (read only). La IANA⁴ ha designado para el AST el identificador 31217 bajo la rama *private* (1.3.6.1.2.1) de la base común para gestión de equipos en internet (MIB-II)([18], [19])

A continuación mostramos la MIB del AST donde se muestran las variables de QoS y QoE además de otros datos básicos.

OID	Descripción
1.3.6.1.4.1.31217.1.1.0	Memoria del sistema
1.3.6.1.4.1.31217.1.2.0	Procesos del cpu
1.3.6.1.4.1.31217.1.3.0	Uso disco duro
1.3.6.1.4.1.31217.2.1.0	Número de interfaces conectadas al sistema
1.3.6.1.4.1.31217.3.1.0	Número de modos activos
1.3.6.1.4.1.31217.3.2.x	Último retardo del modo x
1.3.6.1.4.1.31217.3.3.x	QoE para el usuario x
1.3.6.1.4.1.31217.4.1.0	Pedido de informe estadístico para el modo configurado en GUI
1.3.6.1.4.1.31217.4.2.0	Informe múltiple
1.3.6.1.4.1.31217.4.3.0	Informe de QoE
1.3.6.1.2.1.1.1.0	sysDescr
1.3.6.1.2.1.1.2.0	sysObjectID
1.3.6.1.2.1.1.3.0	sysUptime
1.3.6.1.2.1.1.4.0	sysContact
1.3.6.1.2.1.1.5.0	sysName
1.3.6.1.2.1.1.6.0	sysLocation

Cuadro 8.1: MIB del AST

La implementación del agente en java se hizo con la librería *libsnmp1.4-java* ([20]), la misma es una implementación Open Source de SNMP en Java. Entre otras cosas provee soporte para las operaciones básicas de agentes para las versiones 1 y 2 de SNMP.

8.1.5. Interfaz con la Base de Datos

Para la interfaz con la base de datos *MySQL*, se utilizó el paquete *libmysql-java*. Se implementó una clase denominada *InterfazBD*, en la cual se encuentran los métodos que

⁴Internet Assigned Numbers Authority: www.iana.org

permiten hacer las distintas consultas a la base. Los métodos cumplen varias funcionalidades, por ejemplo permiten guardar un objeto como un SMS en la base de datos, saber el estado de una bandera registrada en la misma y cualquier otra funcionalidad que exija comunicarse con la base de datos.

8.2. Interfaz Gráfica de Usuario (GUI)

Uno de los objetivos iniciales del proyecto era que el sistema pudiera ser gestionable remotamente, con facilidad, y sin la necesidad de instalar un cliente propietario en el PC. Es por esto que se pensó en una interfaz gráfica web. Los contenidos generados y enviados al *host* son código *HTML* puro, con tabla de estilos *CSS*. Esto no sólo cumple con los estándares actuales de la industria, sino que permite su visualización en cualquier navegador web, sin importar el sistema operativo, aún en un PC con pocos recursos.

Los contenidos son generados por un compilador de scripts *PHP* versión 5. Esta versión permite el desarrollo orientado a objetos. El formato de página es tomado en su totalidad de la tabla de estilos, lo que permite independizar el código del formato. No sólo incluye los tipos de fuente y colores, sino también la posición de las distintas secciones en la página.

La GUI cumple varias funcionalidades, en un principio muestra información sobre el estado del sistema, ya sea estadísticas sobre el uso de recursos, como las mediciones que se están realizando. Para ello la GUI utiliza un paquete que permite insertar comandos sobre la línea de comandos de linux. Esto permite que desde la interfaz sea posible reiniciar el servicio, o inclusive reiniciar el servidor donde éste corre, y verificar si el servicio está operativo.

Una de las principales funciones de la interfaz gráfica, es su interacción con la base de datos *MySQL*. Como vimos en capítulos anteriores, tanto la configuración como los datos, se encuentran en la base de datos. Para esta interacción se utiliza el paquete *php5-mysql*, que permite realizar la conexión con la base de datos, y realizarle las consultas correspondientes, para luego trabajar en *PHP* con arrays de datos.

Otra funcionalidad que presenta es la de comunicarse con el proceso en Java mediante *SNMP*. Este protocolo es implementado para realizar peticiones instantáneas sobre el proceso, y permite mostrar estadísticas en tiempo real y pedir informes. Para la implementación de este protocolo se utiliza el paquete *php5-snmp*.

8.3. Base de Datos

Como fue adelantado, implementamos la base de datos sobre un motor *MySQL*. De la misma manera que con el resto de las herramientas utilizadas, buscamos un sistema de gestión de base de datos que sea Software Libre. Los dos motores más importantes en este sentido son *MySQL* y *PostgreSQL*. Elegimos el primero sobre el segundo por su velocidad en las operaciones y su bajo consumo de recursos, además de que las ventajas que ofrecía *PostgreSQL* sobre *MySQL* no eran utilizadas en nuestra aplicación.

Instalamos, por lo tanto, un servidor *MySQL* en la misma máquina que corre la aplicación principal en Java, aunque perfectamente se puede instalar en una máquina distinta, que se encuentre en la misma red (la aplicación en Java también soporta esta arquitectura).

MySQL es una comunidad muy extendida, y con el soporte de *Sun*. Esto facilita la implementación, debido a la enorme cantidad de información disponible. En particular provee herramientas para hacer más fácil las tareas de mantenimiento y puesta en marcha de la base de datos, como ser herramientas de diseño de la base de datos, herramientas de backup, monitoreo de recursos, etc.

Para facilitar la instalación se crearon una serie de instaladores, entre los que se incluye uno para la puesta en marcha de la base de datos. El instalador se basa en un *script* escrito en *bash*, que se encarga de realizar las tareas de inicialización de la base de datos: crea el usuario *astuser*, crea la base de datos *astdb*, y luego crea todas las tablas descriptas en la sección 7.5.

8.4. Proxy SMPP-HTTP

El modo de conexión con el SMS-Gateway se implementó como una aplicación independiente del AST ya que, por disposiciones de seguridad de ANTEL, ambos deben correr en máquinas distintas. Esta disposición esta determinada básicamente porque el Proxy debe tener una conexión a internet para poder enviar y recibir SMS. El SMS-Gateway tiene una interfaz HTTP sobre la cual, por medio de un protocolo propietario de ANTEL([21]), puede enviar y recibir SMS, además de mensajes de gestión.

Dados estos requerimientos, se implementó un Proxy SMPP-HTTP el cual, en una interfaz acepta conexiones SMPP y en otra se encuentra un simple servidor HTTP que escucha los *requests* que envía el SMS-Gateway. Para la interfaz SMPP se utilizó el paquete SMPP de *Logica* ([16]) mencionado en la sección 8.1.2, el mismo permite generar un proceso que simula un SMS Center, el cual por medio de una interfaz SMPP acepta conexiones y permite enviar y recibir SMS. Los SMS que son recibidos por esta interfaz son reenviados hacia la *url* del SMS-Gateway (a través de la interfaz conectada a internet y con el protocolo propietario de ANTEL) con las funcionalidades provistas por la librería *java.net*.

El servidor HTTP simplemente escucha los *requests* http provenientes desde el SMS-Gateway (los cuales son básicamente envíos de SMS) y se reenvían vía la interfaz SMPP.

Capítulo 9

Descripción del Hardware

En esta sección en particular nos concentramos en lo que refiere al hardware necesario para que funcione el sistema, y a los servicios de software que deben ser instalados en el hardware previamente a poner a funcionar nuestro software.

Uno de los objetivos principales del proyecto es que el producto final sea funcional por sí mismo, esto significa que no sólo debemos entregar el software en un CD, e instrucciones de cómo instalarlo, sino que se debe entregar un sistema completo con su hardware funcionando.

9.1. Estructura general del sistema

Como vimos en capítulos anteriores, el sistema está compuesto por un PC corriendo el software. Este PC está conectado a través de una red IP a distintos elementos de la red del operador, y localmente a teléfonos móviles celulares.

Nos encontramos entonces con algunas restricciones básicas; los móviles deben contar con alguna interfaz para conectarse al PC, y el PC debe contar con las interfaces para los móviles y una placa de red.

El testeado del sistema fue llevado a cabo en el LATTIA (Laboratorio de Tecnología de Telecomunicaciones de ANTEL), donde instalamos el PC con los móviles los primeros días de julio de 2008. Nos adjudicaron un block (figura 9.1) y una dirección IP, pero, como ya mencionamos, sin salida a internet. Por lo tanto las fallas en el sistema no podían ser detectados remotamente, y cada cambio en la versión del software debía ser realizado en el lugar. Esto enlentenció el proceso de depuración, y recién tuvimos una versión estable en agosto.

9.2. Características necesarias de los móviles

Los móviles son en general móviles estándar, de venta masiva, no es necesario el uso de móviles especiales para el correcto funcionamiento del sistema. Los móviles deben ser GSM, y operar en la banda del operador que se desea medir. Además deben contar con capacidad para enviar y recibir mensajes, e interfaz USB o RS-232.

Estos móviles permiten ser gestionados utilizando su puerto USB o RS-232, y deberían implementar las recomendaciones *ETSI TS 127 007*, y *ETSI TS 127 005* ([13], [14]). En la práctica encontramos que no todos los móviles respetan la totalidad de las



Figura 9.1: LATTIA

mismas, y que además, hay huecos en la recomendación que permiten distintas implementaciones de la misma funcionalidad, lo que implica que existan aún más diferencias en las respuestas de los distintos móviles. Incluso dentro de un mismo fabricante se pueden encontrar diferencias grandes en las distintas versiones de móviles y de firmware.

Esto impide que la conexión del móvil sea *plug-and-play*. Debe ser testeado previamente, y la información de respuesta debe ser cargada en un driver que utiliza el software para comunicarse con el móvil. Este driver es sencillamente un archivo escrito en XML que representa algunas de las respuestas donde se encontró falta de estandarización entre los distintos fabricantes. El producto final ya incorpora los drivers de una serie de móviles que pudieron ser probados.

Una vez que se tiene el driver del móvil cargado, el software lo reconoce y lo instala automáticamente.

Se realizaron pruebas con la mayor cantidad de móviles posibles, considerando la falta de recursos. A continuación detallamos los equipos probados, y algunas particularidades sobre su implementación:

- Motorola V3
 - Alimentación mediante USB. No soporta condiciones de transmisión continua (situación de stress normal para nuestras pruebas)
- Motorola L6
 - No fue probado en condiciones de stress
 - Presenta características muy similares a las del V3
- Huawei E220
 - Probado en condiciones de stress (recibiendo mensajes cada 20 o 30 segundos), posibles problemas de disipación de calor, lo que produce comportamiento inestable luego de funcionamiento prolongado.
 - Genera 3 interfaces a nivel de sistema operativo (2 para la gestión del móvil y 1 para acceder a la memoria interna que contiene el driver para windows). Se deben eliminar 2, para que no haya problemas de reconocimiento por el driver de Java (*RxTx*).

- Nokia 3220
 - Implementación en modo texto con muchas diferencias respecto a los otros fabricantes. Por ejemplo, no incluye un modo en el cual muestre el contenido del mensaje al llegar, sino que muestra el registro en memoria, y luego hay que leerlo.
 - No fue probado en condiciones de stress, sólo fue probado para asegurar compatibilidad.
- Nokia 5070b
 - Características similares al otro modelo del mismo fabricante.
- Telular SX5e
 - No fue probado en condiciones de stress, sólo fue probado para asegurar compatibilidad. El modelo obtenido no operaba en la banda de ANCEL.
- Sony Ericsson t290
 - No fue probado en condiciones de stress
 - Sólo permite recibir mensajes, no enviar.
- Ericsson BVF221m
 - Probado en condiciones de stress con buenos resultados.
 - Sólo implementa modo hexadecimal.

El sistema quedó finalmente funcionando con dos terminales móviles: un Huawei E220 y un Ericsson BVF221m (figura 9.2), y las conexiones con el SMS-C y el SMS-Gateway.

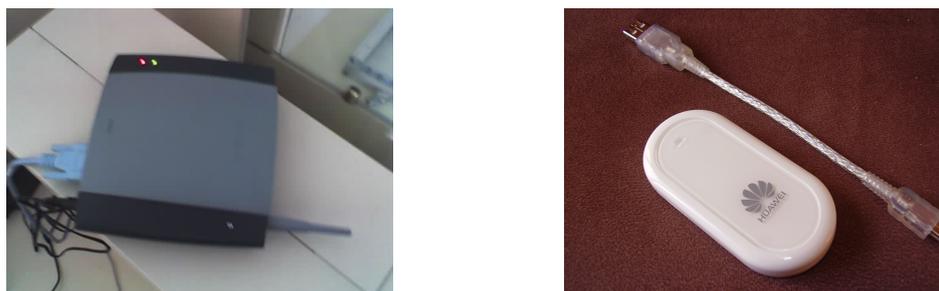


Figura 9.2: Terminales móviles utilizados en la versión final

9.3. Requerimientos del PC y servicios necesarios

El software a pesar de estar escrito en Java, el cual es multiplataforma, debe ser ejecutado en Linux. Esto es debido a que los drivers para los puertos seriales y USB son sólo compatibles con Linux. Los requerimientos básicos de hardware son entonces una PC, a la cual se le pueda instalar alguna versión de Linux. Esta PC debe contar con los puertos necesarios para la interconexión con los móviles, y con una interfaz

Fabricante	Modelo	Interfaz	Envía	Recibe	PSE	PSR
Motorola	V3	USB	Sí	Sí	-	Insatisfactoria
Motorola	L6	USB	Sí	Sí	-	-
Huawei	E220	USB	Sí	Sí	Satisfactoria	Insatisfactoria
Nokia	3220	USB	Sí	Sí	-	-
Nokia	5070b	USB	Sí	Sí	-	-
Telular	SX5e	RS-232	Sí	Sí	-	-
Sony Ericsson	t290	USB	No	Sí	-	-
Ericsson	BVF221m	RS-232	Sí	Sí	Satisfactoria	Satisfactoria

PSE: Prueba de Stress al Enviar

PSR: Prueba de Stress al Recibir

Cuadro 9.1: Móviles testeados

Ethernet para conectarse a la red de gestión y al SMS-Gateway. Los requerimientos en cuanto a la capacidad del PC no son muy exigentes; nuestra implementación cuenta con un procesador Pentium III de 500Mhz, 128Mb de memoria RAM, y un disco duro de 10Gb, pero hemos realizado una variedad de pruebas sobre un AMD de 166 Mhz, que fue la implementación original, sin problemas. Fue necesario cambiar el PC porque ocurrió una falla con el hardware irreparable.

Nuestra implementación utiliza una variante de la distribución *Ubuntu* para optimizar el uso del hardware, pero en principio se puede utilizar cualquier distribución basada en *Debian* con una versión del kernel mayor a *2.6.20*, sin cambiar los paquetes que describimos a continuación. Estos paquetes son (se considera que con cada paquete instalado se instalan todas las dependencias):

- Maquina Virtual - *Java v1.5* o posterior
- Base de datos - *MySQL v5* o posterior
- Servidor WEB - *lighttp* (o *apache2*)
- Servidor de páginas dinámicas - *php5* y los paquetes mencionados en la sección de implementación de GUI
- Compilador *L^AT_EX*

Parte IV

Análisis de resultados y Conclusiones

Capítulo 10

Análisis de resultados

En este capítulo se presentan algunos de los resultados obtenidos en el período de pruebas, y se analizan casos particulares. El escenario de prueba incluye la conexión a la red GSM de ANCEL mediante dos móviles y una sesión SMPP con el SMS-C. Luego de realizadas estas pruebas se incluyó una segunda conexión con el SMS-Gateway como fuera detallado en la sección 3.1, pero debido a retrasos en la operativa de ANCEL, no ha sido posible obtener una cantidad significativa de datos como para incluirlos en este informe.

Comenzamos analizando los resultados de calidad de servicio (simplemente los retardos), y luego, agregando conocimiento sobre el comportamiento de los usuarios, analizamos resultados en la calidad de experiencia. Finalmente, presentamos algunos comentarios sobre una vulnerabilidad del servicio encontrada accidentalmente.

10.1. QoS

Describimos primero los resultados obtenidos en situaciones de actividad regular de la red, y luego analizamos situaciones de falla encontradas en el período de testeo.

10.1.1. Valores de funcionamiento normal

En condiciones de funcionamiento normal, encontramos que los mensajes entre móviles tienen un retardo¹ del orden de 6,8s. Este valor depende del móvil utilizado para realizar las mediciones, encontramos que de acuerdo a la implementación puede diferir hasta en más de un segundo el resultado.

Para el caso de los mensajes que son enviados directamente al SMS-C mediante el protocolo SMPP, los retardos son menores. En los mensajes enviados desde el SMS-C hacia un móvil, encontramos un retardo promedio de 5,0s, y 3,8s para los mensajes con origen un móvil y destino el SMS-C.

La varianza en todos los casos es del orden de medio segundo, este valor puede estar afectado en gran manera por los buffers de la RBS, y depender del grado de utilización de los canales de control, no creemos que sea un parámetro que represente en forma general el comportamiento del servicio.

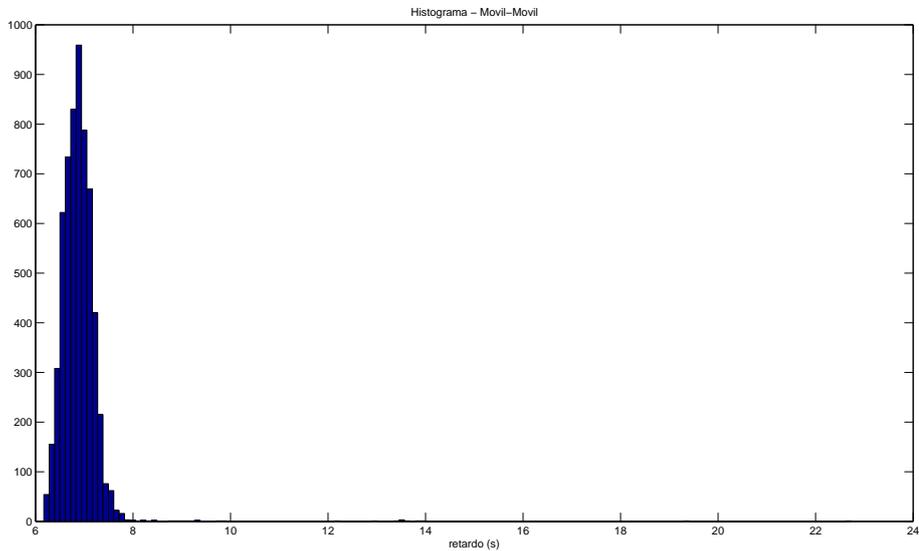
En general no hemos tenido pérdida de mensajes, inclusive en condiciones extremas como veremos al final de éste capítulo.

¹Retardo entre el instante en el cual el usuario envía el mensaje y éste es recibido por el destinatario, según fue definido en 5.1.4

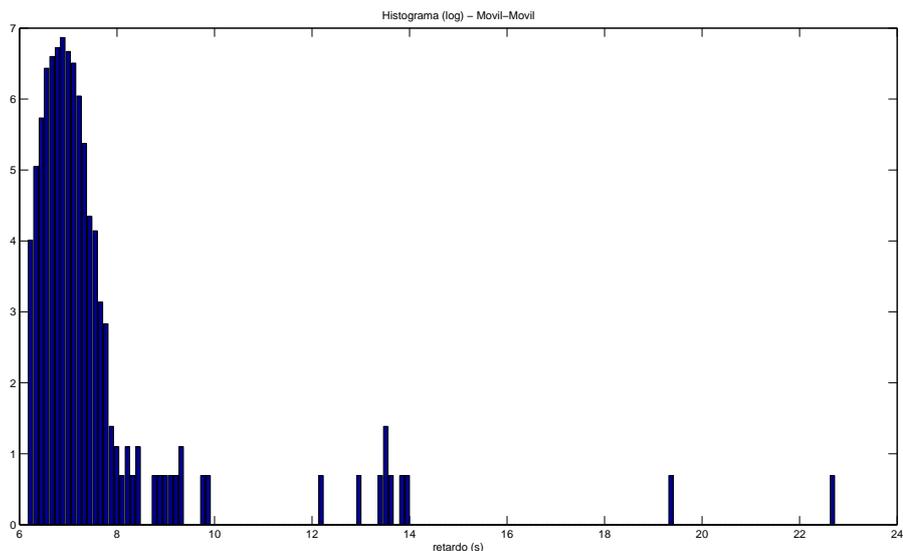
10.1.2. Dispersión de los retardos

De las observaciones realizadas para los mensajes entre móviles surge que para los retardos se distinguen dos grupos bien diferenciados. En efecto, la mayoría de los mensajes tienen un retardo del orden de los $7s^1$, mientras que otro grupo menor de mensajes tienen retardos superiores a los $12s$ (en nuestras observaciones, la cantidad de mensajes que cae dentro de esta categoría es del orden de 0.2%)

Se aprecia claramente lo observado en el párrafo anterior en la figura 10.1, donde está representado un histograma de los retardos de los mensajes correspondientes a un período de tiempo de una semana, en escala lineal y logarítmica para poder observar el porcentaje de mensajes del grupo menor.



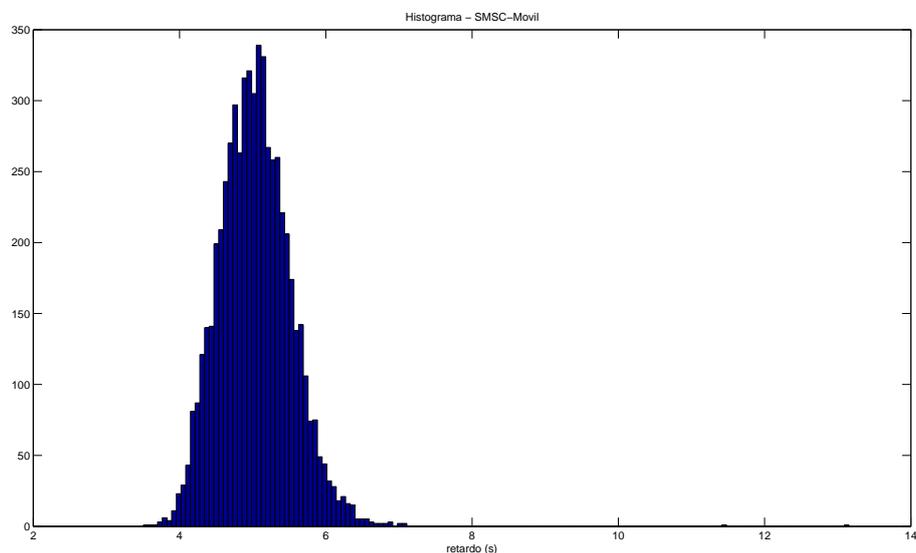
(a) Escala lineal



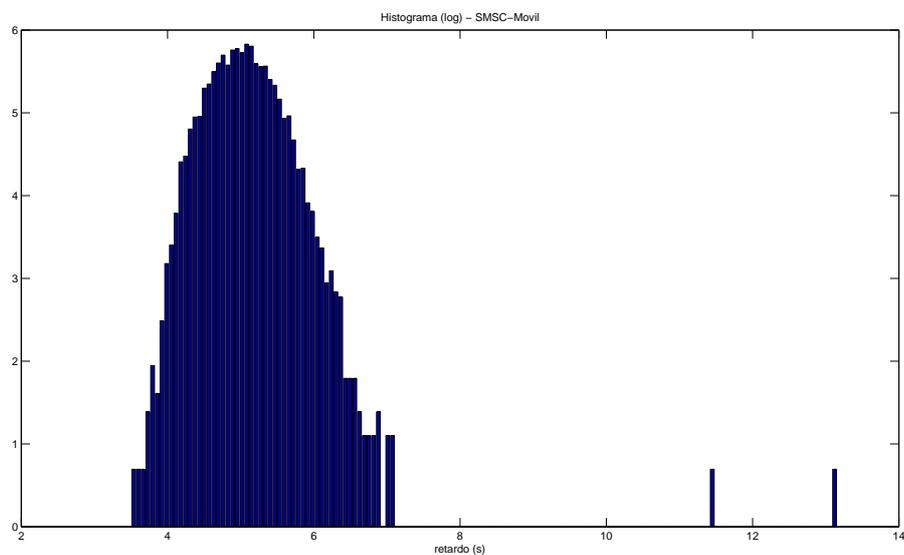
(b) Escala logarítmica

Figura 10.1: Histogramas de retardos de mensajes para el modo Móvil - Móvil

Este comportamiento no se observa en forma tan evidente para los mensajes que son enviados desde o hacia el SMS-C, como se puede ver en la figura 10.2.



(a) Escala lineal



(b) Escala logarítmica

Figura 10.2: Histogramas de retardos de mensajes para el modo SMS-C - Móvil

Dada nuestra condición de observadores externos a la red no podemos saber con exactitud la razón por la cual se produce este fenómeno. Sin embargo, una posible explicación es que algunos de los mensajes sean enrutados de distinta manera, ya sea por equipos distintos o mediante procedimientos distintos.

10.1.3. Comportamientos periódicos

Observando los retardos de los mensajes en períodos de tiempo prolongados, percibimos algunos patrones que se repiten.

En particular, a continuación presentamos los resultados del procesamiento de los datos correspondientes a los modos móvil-móvil y SMS-C-Móvil.

Para eliminar los efectos de las variaciones rápidas de los retardos, y poder observar el patrón de comportamiento a mayor escala, filtramos los datos con un filtro pasa-bajos. En la figura 10.3 se observa la señal original correspondiente al modo SMS-C-Móvil, y en la figura 10.4 la misma señal filtrada. Además, en la figura 10.5 mostramos la señal filtrada correspondiente al modo Móvil-Móvil.

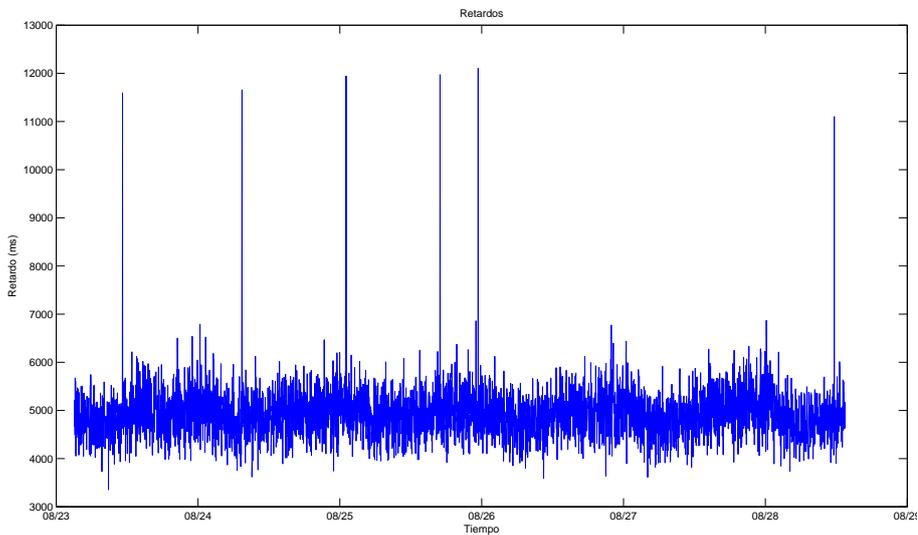


Figura 10.3: Señal original correspondiente al modo SMSC-Móvil

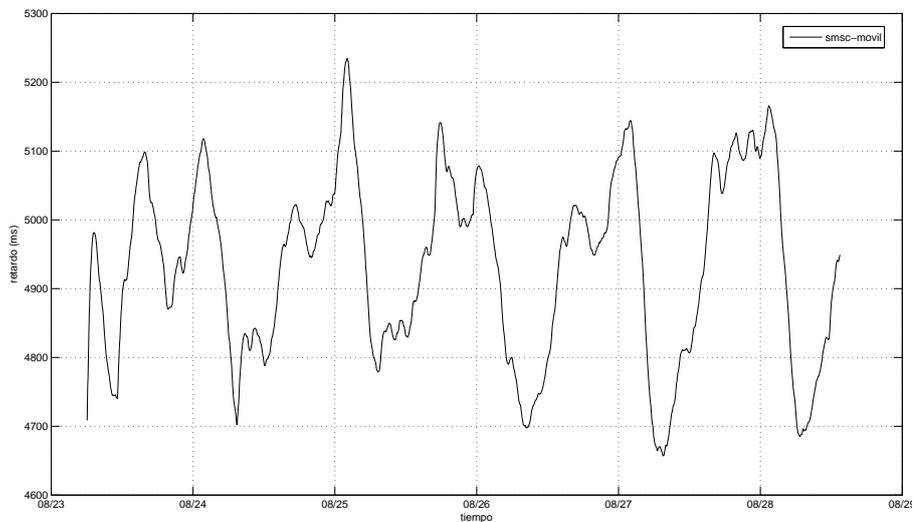


Figura 10.4: Señal filtrada correspondiente al modo SMSC-Móvil

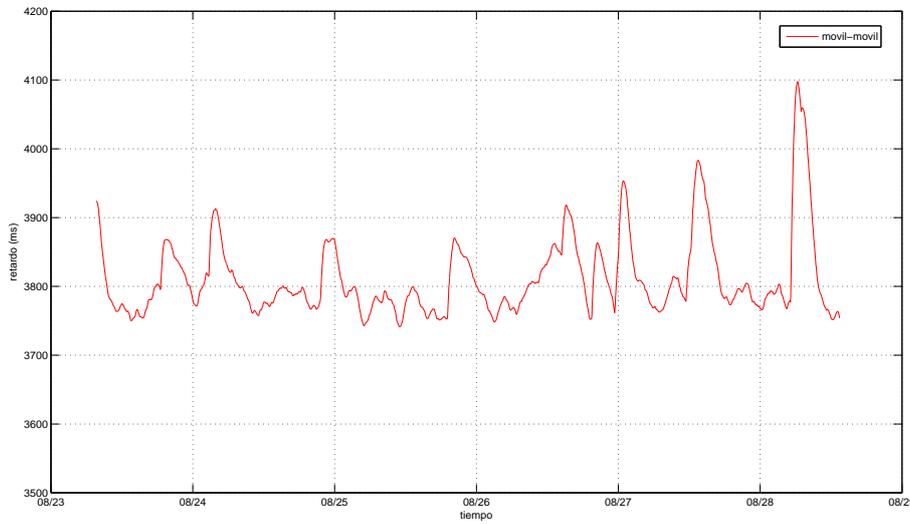


Figura 10.5: Señal filtrada correspondiente al modo Móvil-Móvil

En la figura 10.4 se puede distinguir un patrón que se repite diariamente. Este comportamiento no es atribuible a la performance de la radiobase servidora, ya que en la figura 10.5 no se observa el mismo comportamiento. Por lo tanto, se deduce que el originador de este patrón se encuentra en el *core* de la red.

Para verificar esta periodicidad realizamos un análisis frecuencial de la señal. Si la señal presenta un comportamiento repetitivo diario, deberíamos encontrar un pico en la frecuencia correspondiente. Esto es, $f_p = \frac{1}{86400s} \approx 1,15 \times 10^{-5} Hz$. En la figura 10.6 se puede ver el módulo de la transformada rápida de Fourier (FFT) de la señal, y se confirma la hipótesis observando el pico cercano a f_p .

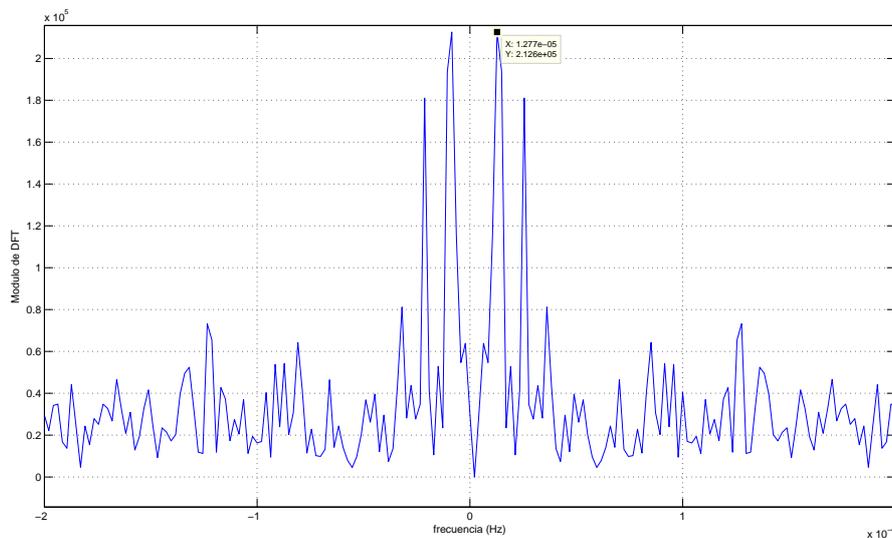


Figura 10.6: Módulo de la FFT de la señal correspondiente al modo SMSC-Móvil

10.1.4. Fallas o eventos

En el período de observación encontramos dos eventos atribuibles a fallas en el servicio de mensajería corta. Si bien podría pensarse que estas fallas afectaron solamente la zona desde la cual se realizaron las mediciones (falla de la RBS, transmisión, etc.), existen ciertos comportamientos que demuestran que fueron eventos que afectaron a toda la red. No tenemos hasta el momento una verificación de parte de ANCEL, que respalde esta conclusión. Dado que ambos eventos tuvieron comportamientos similares, analizaremos a continuación uno de ellos.

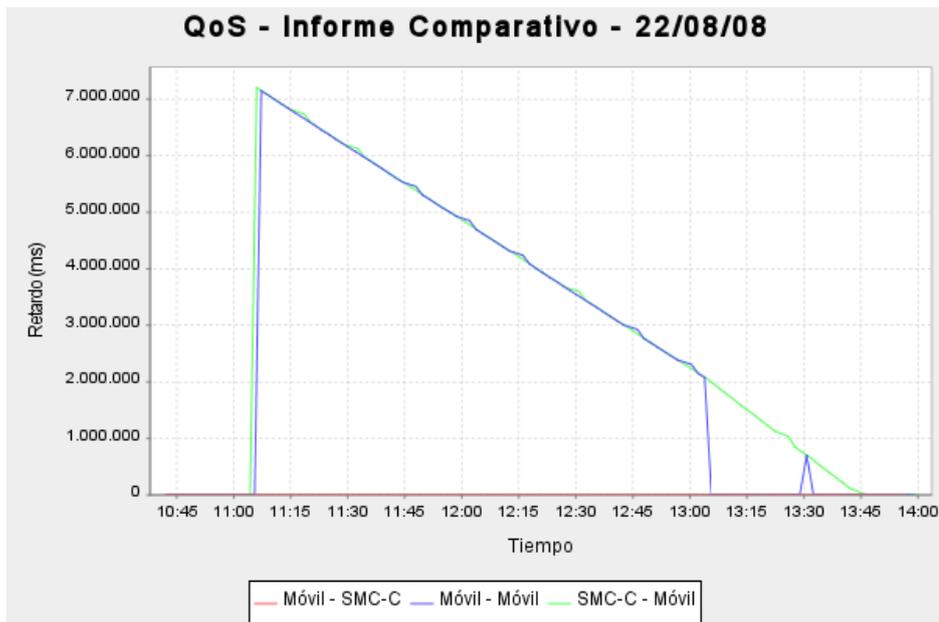


Figura 10.7: Retardos durante una falla

En primer lugar, si miramos la gráfica de la figura 10.7, podemos percibir que los mensajes enviados desde un móvil con destino SMS-C, llegaban sin problema en todo momento. Esto nos da la pauta de que el problema no está dado por la RBS ni es local, pues si así lo fuera estos mensajes no llegarían con retardos normales.

En segundo lugar, se observa que el problema se solucionó a las 13:05 aproximadamente. Esto se justifica de dos maneras: por un lado, es el momento en el cual los mensajes entre móviles empezaron a tener retardos normales, y por otro lado, corresponde al momento en el cual llegó el primer mensaje enviado durante el período de falla. Efectivamente, el retardo de este mensaje es de 7200s, que corresponde a 2 horas, y fue enviado a las 11:05.

Si bien la falla es solucionada a las 13:05, los mensajes enviados desde el SMS-C luego de esta hora siguen teniendo un retardo mayor al normal. Este fenómeno corresponde al encolamiento de los mensajes. El tiempo de entrega de los mensajes no es nulo, y una vez que la falla es solucionada, el centro de mensajería comienza a despachar los mensajes. Los nuevos mensajes a enviar son colocados al final de la cola (cola FIFO), y por lo tanto deberán esperar a que los mensajes que llegaron antes sean despachados. Sin embargo encontramos un comportamiento distinto para los mensajes entre móviles, a saber, a partir del momento de solución de la falla los mensajes comenzaron a llegar con retardos normales. Nuevamente, este comportamiento podría ser atribuido a un

método distinto de enrutamiento, o a la presencia de otro agente enrutador que enruta los mensajes no guardados.

La pendiente de la recta está relacionada con la velocidad con la cual el centro de mensajería despacha los mensajes que están en la cola.

Por último, cerca de las 13:30 uno de los mensajes entre móviles sufrió un retardo igual a los mensajes provenientes del SMS-C, lo cual parece indicar que ese mensaje fue encolado en el SMS-C con el resto de los mensajes.

En ambos casos, el sistema generó alarmas en forma rápida, alertando del problema.

10.2. QoE

En esta sección se muestran resultados de las experiencias de distintos tipos de usuarios respecto de la calidad del servicio ofrecida para ciertos intervalos de medición. Para esto, simulamos 800 usuarios y definimos, en forma arbitraria, los modelos² para los distintos tipos de usuarios. Estos 800 usuarios están compuestos por dos perfiles distintos. El primero representa un usuario que envía aproximadamente 3 mensajes diarios, para los cuales en general no espera respuesta; el modelo de este usuario está compuesto por dos estados. El segundo tipo, representa a usuarios con mayor actividad, que en media inician 10 conversaciones diarias, las cuales pueden tener una duración de uno o más mensajes; está representado por un modelo de tres estados.

El objetivo de esta sección es mostrar cómo variaciones en el desempeño de la red (y sus posibles interpretaciones a partir de los valores de QoS) son realmente percibidos por los usuarios finales del servicio. Al presentar la información de esta manera, permitimos a los agentes de decisión visualizar el impacto real que tiene la performance de la red sobre los usuarios, de una forma mucho más directa y efectiva que la QoS.

Esta forma de presentar el desempeño de la red respecto a cómo lo percibe el usuario, permite tomar acciones teniendo en cuenta cuáles fueron los tipos de usuarios más afectados. Por ejemplo, en determinadas condiciones los usuarios que utilizan más los servicios tipo “chat” pueden verse más afectados que otros o, más aún, pueden ser los únicos que detectaron una degradación del servicio y por ende puede ser beneficioso compensarles por esta condición.

Período de funcionamiento habitual

En primer lugar, realizamos pruebas sobre un intervalo de tiempo donde el desempeño de la red es el habitual. En la figura 10.8 podemos observar el porcentaje de usuarios del primer tipo que experimentó cada nivel de calidad de experiencia en función del tiempo (no se gráfica la curva correspondiente a QoE correspondiente al nivel *excelente* para observar mejor los otros picos). En particular, solamente un 0,25 % de los usuarios (1 en 400 de los simulados) tuvieron una experiencia *muy buena*, y el resto experimentó una calidad *excelente*.

Por otro lado, en la figura 10.9 se puede observar una representación similar para los usuarios del segundo tipo, durante el mismo período de tiempo. Se observa que estos usuarios percibieron en mayor medida la degradación del servicio que los usuarios anteriores. En efecto, el porcentaje de usuarios con QoE menor a 5 es un orden superior, hay usuarios que tuvieron calidad de experiencia de nivel 3, y los eventos están presentes con más frecuencia.

²Modelos que incluyen los umbrales de QoE

Aún en condiciones normales, donde los parámetros de calidad de servicio se encuentran debajo de los umbrales, encontramos que existe cierto grado de afectación en la calidad percibida por algunos tipos de usuarios, pero no para otros.

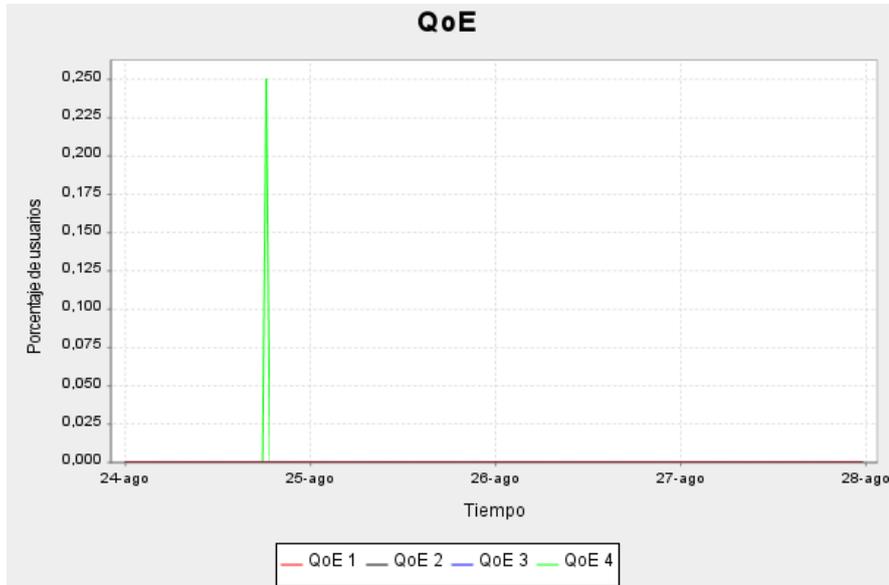


Figura 10.8: QoE para usuarios del primer tipo en un período de funcionamiento normal

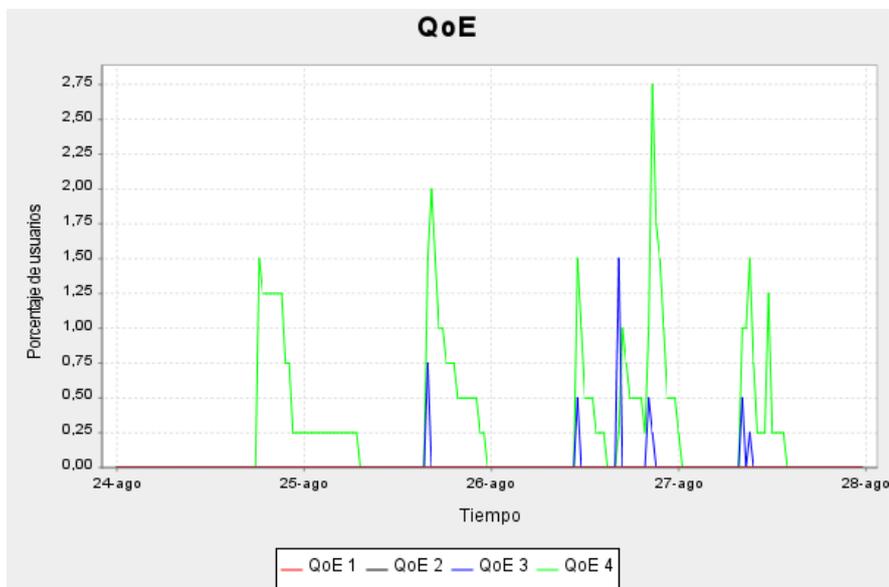


Figura 10.9: QoE para usuarios del segundo tipo en un período de funcionamiento normal

Período de funcionamiento con anomalías

En segundo lugar, tomamos un intervalo de medición en el cual el funcionamiento de la red fue normal, salvo en un período de 2 horas en el que se presentó una anomalía en el desempeño, generando retardos del orden de 2000 segundos.

Nuevamente, simulamos 400 usuarios de cada tipo. Los resultados son los que se encuentran en las figuras 10.10 y 10.11 para el primer y segundo tipo respectivamente.

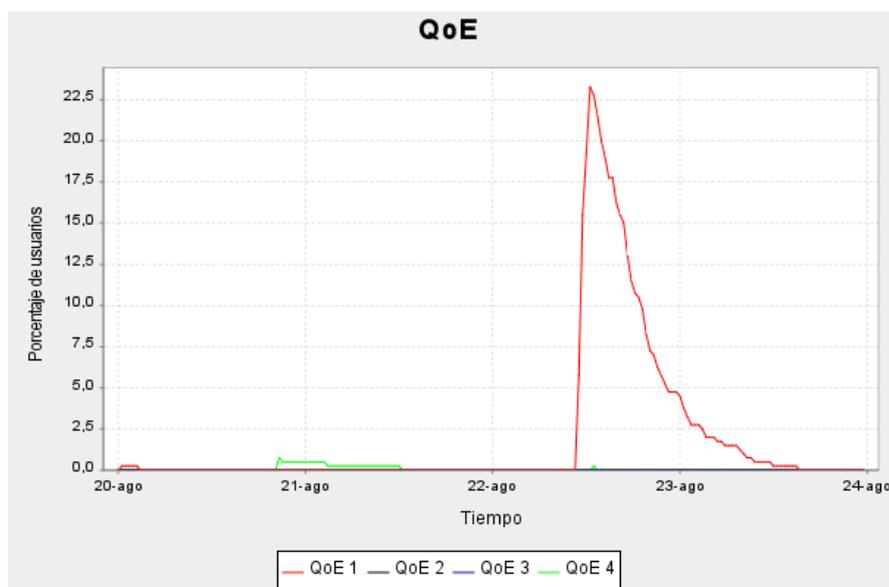


Figura 10.10: QoE para usuarios del primer tipo en un período con fallas

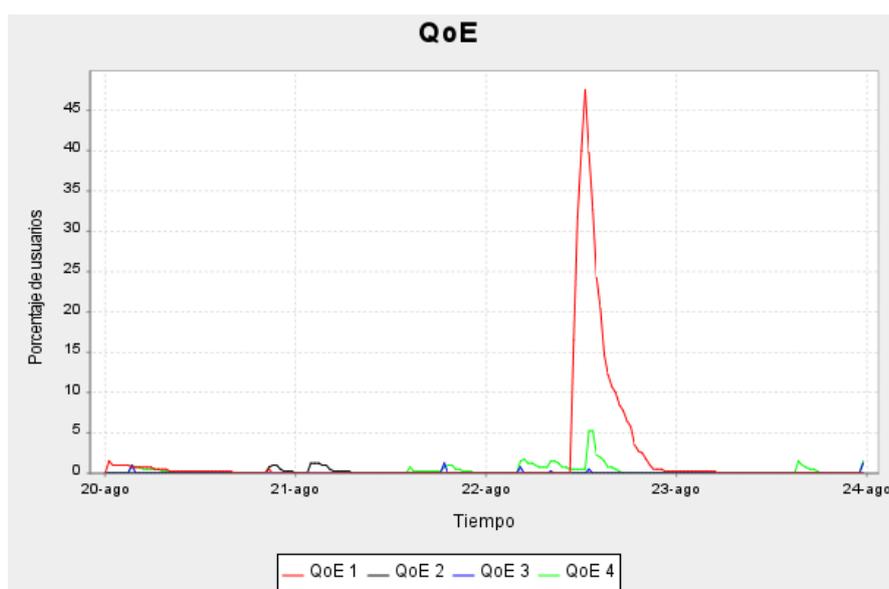


Figura 10.11: QoE para usuarios del segundo tipo en un período con fallas

En estas gráficas podemos ver la diferencia en el impacto que tuvo un problema en la red sobre dos tipos de usuarios distintos. Se visualiza que para cierto tipo de usuarios (naturalmente el usuario con tiempo medio entre disparos de mensajes mayor), el impacto es considerablemente menor. Quizás este tipo de conclusión pueda parecer trivial, pero complementos de este tipo pueden facilitar sensiblemente la toma de decisiones.

En particular, la mitad de los usuarios del segundo tipo tuvo una calidad de experiencia *mala* en el momento de la falla, y se observan además otros momentos de calidad mala y regular. Por otro lado, los usuarios del primer tipo que percibieron la falla son del orden del 20 %, y en general tuvieron buenas experiencias en el resto del período.

Remarcamos que en las gráficas de QoS (los retardos) solamente se percibe el problema importante (el del 22 de agosto), pero pasan desapercibidas los otros eventos, y además no permiten evaluar cómo afectó el problema a los distintos tipos de usuario.

10.3. Vulnerabilidades o comportamientos indeseados

Hemos encontrado, en forma accidental, un posible punto de vulnerabilidad del servicio de mensajería corta de ANCEL. Durante el período de pruebas y depuración, encontramos varios problemas con los móviles Huawei (móviles cedidos por ANCEL), éstos, como explicamos en el capítulo 9, dejaban de comportarse en forma estable luego de unos días de uso. El problema principal era que los móviles continuaban contestando los comandos de forma normal, pero dejaban de enviar la señal correspondiente a la recepción de un mensaje al AST. Esto generó que los mensajes no fueran leídos ni borrados. Luego de unos minutos la memoria del móvil se llenaba y éste empezaba a rechazar todos los mensajes siguientes. A medida que el móvil rechazaba los mensajes, el SMS-C los almacenaba. Muchas veces este problema sucedía un fin de semana, de modo que nos era imposible detener el envío de mensajes, y resetear el móvil para que siguiera recibiendo los mismos.

Llegamos a encontrar hasta 6000 mensajes almacenados en el SMS-C, algunos con más de 72 horas de antigüedad. Los mensajes nunca fueron descartados. Para un operador que cobra los mensajes en el *delivery* y no en el *submit*, no parecería ser una política muy segura. Sería tan fácil como apagar un móvil y poner otro a enviar mensajes indefinidamente para saturar el almacenamiento del SMS-C, sin ningún costo.

Capítulo 11

Conclusiones

En este capítulo presentamos las conclusiones generales del proyecto, evaluando aspectos positivos y negativos tanto del transcurso como de los resultados. Describimos también algunas líneas de trabajo a futuro que fueron surgiendo a lo largo del proceso.

Conclusiones de los resultados

Desde un punto de vista funcional, es decir, observando el resultado material, podemos decir que se logró con éxito cubrir la totalidad de los objetivos planteados al inicio del proyecto, teniendo en cuenta las limitaciones y restricciones.

Se diseñó e implementó un sistema autónomo eficaz y estable que permite la medición de QoS y QoE *end-to-end* para el servicio de mensajería corta. El sistema fue diseñado desde cero, es decir, no es la continuación o mejora de otro sistema implementado o diseñado anteriormente, y fue realizado con costos materiales despreciables y dentro del tiempo estipulado inicialmente.

La solución está desarrollada de forma tal que facilita la viabilidad de su incorporación en la plataforma de monitoreo de una red en producción, y el sistema fue testeado con buenos resultados durante un tiempo prolongado en las instalaciones de ANTEL, asegurando las propiedades de estabilidad buscadas.

Conclusiones del proceso

Por otro lado, desde un punto de vista más relacionado con el proceso y no tanto con los resultados, podemos destacar varios aspectos.

En primer lugar se realizó un estudio profundo del tema, que comprende, entre otras cosas, la estructura de las redes GSM; los protocolos y las particularidades del servicio de mensajería corta; los sistemas de monitoreo, operación y mantenimiento; estado del arte en cuanto a estudios de calidad. Este estudio permitió realizar un análisis del problema, para luego idear y proponer una solución original, que luego fue implementada.

En segundo lugar, se estudiaron las herramientas probabilísticas para modelar comportamientos de usuarios. En particular, se analizaron los modelos más comúnmente utilizados, concluyendo que no se adaptaban al servicio de mensajería. Luego se propuso un modelo nuevo, original, que luego de ser implementado resultó ajustarse de manera satisfactoria a lo deseado.

Por último, se adquirió conocimiento y práctica en la planificación de los distintos aspectos que comprenden el proyecto, como ser la administración de tiempos, alcance y especificación, costos, calidad, riesgos y recursos humanos.

Trabajo a futuro

Respecto al trabajo a futuro, identificamos algunas posibles mejoras al sistema, que creemos son la continuación natural del trabajo realizado.

En primer lugar, se puede generalizar el sistema para otros tipos de servicio (como por ejemplo MMS, GPRS, etc). El diseño de software realizado permite introducir este complemento de manera razonable.

Debido a las restricciones y falta de información sobre el comportamiento de los usuarios en la red GSM, no pudimos generar parámetros que se ajusten a los usuarios de la red de ANCEL. Si bien se generaron parámetros que parecen ser lógicos para realizar las pruebas, restaría hacer un estudio estadístico de los datos de tráfico de ANCEL, para estimar los parámetros del modelo que mejor ajustan a esos datos. Esto incluye estudiar cuáles son los mejores estimadores para cada parámetro.

Relacionado a este último aspecto, se encuentra el estudio de la calidad de experiencia. Los umbrales utilizados en las pruebas para los distintos perfiles de usuarios, también fueron elegidos de manera arbitraria, utilizando el sentido común, pero merece un estudio más profundo.

Por otro lado, el sistema tal como está planteado aquí es sensible a problemas de acceso (por ejemplo, congestión en el canal de control de la RBS, o fallas zonales que perjudiquen el área donde se realizan las medidas). Una posible solución a este problema es distribuir varios puntos de medida en la extensión geográfica de cobertura del operador. Luego, con estos sub-sistemas distribuidos, se puede generar un algoritmo que detecte si las fallas son generales o zonales, e identifique el lugar en este último caso.

Bibliografía

- [1] G. Folland. *Real Analysis, Modern Techniques and Their Applications*. Wiley Interscience, 1999.
- [2] M. Wschebor. Notas para el curso de introducción a los procesos estocásticos. Centro de Matemática, UdelaR, 2004. <http://www.cmat.edu.uy/~wschebor/Archivos/notas2.pdf>.
- [3] V. Petrov y E. Mordecki. *Teoría de Probabilidades*. Editorial URSS, 2002.
- [4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., N. Y., 1991.
- [5] Patricia Kisbye. Generación de eventos en procesos de poisson. Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, 2008.
- [6] ITU-T. Recommendation e.800: Terms and definitions related to quality of service and network performance including dependability. Technical report, ITU-T, 1994.
- [7] Claudia Ximena Bustamante. Calidad en los servicios de telecomunicaciones. <http://www.nap.com.co/>, Febrero 2007. Comisión de Regulación de Telecomunicaciones, República de Colombia.
- [8] MS(C) Mónica Andrea Rico Martínez. Protocolo de especificaciones de gestión de calidad en redes de telefonía móvil en colombia. <http://www.informaticahabana.com/>.
- [9] Directorate of Technical Research & Standards. Quality of service (qos) indicators for mobile services. <http://www.ncc.gov.ng/>. Nigerian Communications Commission.
- [10] Nokia. Quality of experience (qoe) of mobile services: Can it be measured and improved?, 2004.
- [11] Architecture & Transport Working Group. Triple-play services quality of experience (qoe) requirements. Technical report, DSL Forum, December 2006.
- [12] T. Jarvi. <http://www.rxtx.org>, Mayo 2008.
- [13] ETSI. Ts 127 005 v3.1.0 digital cellular telecommunications system (phase 2+) (gsm); universal mobile telecommunications system (umts); use of data terminal equipment - data circuit terminating; equipment (dte - dce) interface for cell broadcast service (cbs). Technical report, ETSI, 2001. 3G TS 27.005 version 3.1.0 Release 1999.

-
- [14] ETSI. Ts 127 007 v3.10.0 (2001-12) digital cellular telecommunications system (phase 2+) (gsm); universal mobile telecommunications system (umts); at command set for user equipment (ue). Technical report, ETSI, 2001. 3GPP TS 27.007 version 3.10.0 Release 1999.
- [15] ETSI. Gsm 03.38 alphabets and language-specific information. Technical report, ETSI, 1996. Version 5.3.0, Reference: TS/SMG-040338QR2.
- [16] Logica. <http://opensmpp.logica.com/>, Junio 2008.
- [17] David Gilbert. <http://www.jfree.org/jfreechart/>, 2008.
- [18] Network Working Group. Rfc 2578, structure of management information version 2 (smiv2). Technical report, IETF, April 1999.
- [19] Network Working Group. Rfc 1213, management information base for network management of tcp/ip-based internets:mib-ii. Technical report, IETF, March 1991.
- [20] International Engineer Consortium. <http://gic1.cs.drexel.edu/people/sevy/snmp/>, 2008.
- [21] ANCEL. Proyecto ancel anexo 4 /mensajería b2b, 2004.
- [22] Wikipedia SNMP. <http://en.wikipedia.org/wiki/Gsm>, 2008.
- [23] Javier Pereira. Curso de introducción a la telefonía celular - presentación tecnología gsm, 2007.
- [24] Kuross Amri Tom Ceglarek. http://services.eng.uts.edu.au/userpages/kumbes/public_html/ra/sms/, 2008. University of Technology Sydney.
- [25] Wikipedia SNMP. <http://en.wikipedia.org/wiki/Snmp>, 2008.
- [26] Instituto de Ingeniería Eléctrica Desarrollo de Software. Material del curso, 2008. Facultad de Ingeniería - UdelaR.
- [27] Sitio oficial de php. <http://www.php.net/>, 2008.
- [28] Pjer M. Vuckovic and Nevena S. Stefanovic. Quality of experience of mobile services. *14th Telecommunications forum TELFOR 2006*, 2006.
- [29] Dr. David Soldani. Qoe and qos monitoring and data analyses. Special Course on Networking Technology for Ph.D students at TKK, 2006.
- [30] Peter Reichl. From ‘quality-of-service’ and ‘quality-of-design’ to ‘quality-of-experience’: A holistic view on future interactive telecommunication services. *Telecommunications Research Center Vienna*, 2007.
- [31] International Engineer Consortium. http://www.iec.org/online/tutorials/wire_sms/, 2008.
- [32] P. Griffiths G. Peersman, S. Cvetkovic and H. Spear. Global system for mobile communications short message service. *IEEE Personal Communications IEEE Wireless Communications*, 7:15–23, 2000.

-
- [33] María Valentina Vega. Cadenas de markov de tiempo continuo y aplicaciones. *Trabajo Monográfico, Centro de Matemática, Udelar*, 2004. Orientador: Ricardo Fraiman.
- [34] William Feller. *An Introduction to Probability Theory and Its Applications, Volume 1*. Wiley, January 1968.
- [35] Charles M. Grinstead and Laurie J. Snell. *Introduction to Probability*. American Mathematical Society, July 1997.
- [36] Luc Devroye. *Non-Uniform Random Variate Generation*. Springer, April 1986.
- [37] F. Robledo. Material del curso *Introducción a la investigación de operaciones*, 2007.
- [38] J. Virtamo. Queueing theory / probability theory. <http://www.netlab.tkk.fi/opetus/s38143/luennot/english.shtml>.
- [39] GSM Association. Permanent reference document ir 41 identification of qos aspects of popular services (gsm & 3g). Technical report, GSM Association, Abril 2002. Version 3.1.0.
- [40] GSM Association. Permanent reference document ir 42 definition of qos parameters and their computation. Technical report, GSM Association, Abril 2007. Version 3.3.
- [41] GSM Association. Permanent reference document ir 43 typical procedures for qos measurement equipment. Technical report, GSM Association, Agosto 2002. Version 3.0.0.
- [42] ETSI. Eg 202 057-2 speech processing, transmission and quality aspects (stq); user related qos parameter definitions and measurements; part 2: voice telephony, group 3 fax, modem data services and sms. Technical report, ETSI, Octubre 2005. version 1.2.1.
- [43] Bruce Eckel. *Thinking in Java*. Prentice Hall PTR, Upper Saddle River, 1998.
- [44] William E. Skimin and Integrated Management Systems. A process-based model for project planning and management.
- [45] Malaysian technical standards forum BHD. Quality of service for voice, short messaging service and packet-switched traffic dor public celular services. www.mtsfb.org.my/.

Parte V

Apéndices

Apéndice A

Introducción a la tecnología GSM

En este capítulo mostramos básicamente cómo funciona una red GSM. El objetivo es entender el funcionamiento de la misma para poder comprender cómo funciona el servicio de mensajería instantánea, y dónde podrían aparecer problemas.

GSM se refiere a Global System for Mobile communications¹, y es un estándar para comunicaciones móviles cuya especificación comienza en Europa en el año 1982. Si bien ya tiene algunos años, hoy día es el estándar de comunicaciones más utilizado del mundo, con más de 3 mil millones de usuarios en más de 200 países. Como vemos a continuación, GSM especifica detalladamente las interfaces y las distintas funciones de la red, pero no especifica requerimientos de hardware. Esto permite que distintos fabricantes puedan proveer sus soluciones para los distintos elementos de la red.

A.1. Arquitectura de una red GSM

En una red GSM existen varios elementos cumpliendo distintas funciones. A continuación explicamos brevemente las funciones de cada uno de los elementos, y profundizamos en aquellos que presentan mayor importancia debido a su relación con el servicio que estamos considerando.

La topología de la red se divide en 4 sub-sistemas ([23]):

- MS - Mobile Station - Equipo móvil utilizado por el usuario para comunicarse con la red
- BSS - Base Station Subsystem - Grupo de elementos de la red que realiza las funciones relacionadas con el enlace de radio de una red GSM
- NSS o SS - Network Switching System o Switching System - Elementos de la red que cumplen funciones de conmutación, facturación, comunicación con otras redes, etc
- NMS - Network Management Subsystem - Realiza el monitoreo y control de la red

¹originalmente Groupe Spécial Mobile ([22])

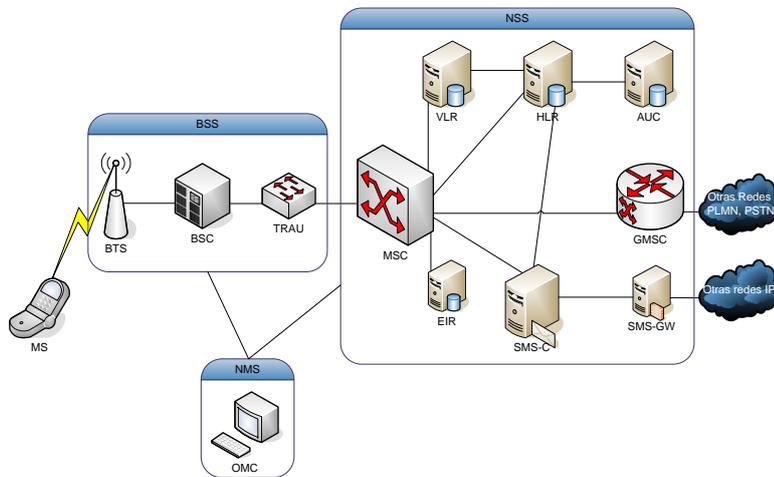


Figura A.1: Arquitectura de la red GSM

A.1.1. MS

El MS es el equipo utilizado por el usuario para acceder a la red. Consiste básicamente en 2 partes:

- SIM - Subscriber Identity Module
- ME - Mobile Equipment

La tarjeta SIM almacena información del usuario, y el ME debe tener una SIM para poder hacer uso de la red (salvo para llamadas de emergencia). En particular la SIM guarda el IMSI, número que identifica al usuario en la red, claves de autenticación y algoritmos de seguridad.

El ME es un equipo que debe realizar las funciones de transmisión, modulación, filtrado, y conversión D/A. Más adelante vemos algunas particularidades de estas funciones. La norma cataloga las distintas funciones como básicas, opcionales, y adicionales. Algunas de las funciones básicas deben ser implementadas por todos los móviles obligatoriamente, otras depende del fabricante. En el caso de las funciones opcionales y adicionales, la implementación es potestad del fabricante. El servicio de mensajería corta debe ser soportado por todos los móviles, dado que es una funcionalidad obligatoria según la norma. La incorporación de una interfaz DTE/DCE² es una funcionalidad básica, pero no obligatoria, por lo que no necesariamente la encontramos en todos los móviles, pero sí en gran parte de ellos. Más adelante, en la sección 9.2, vemos en profundidad qué móviles disponen de esta funcionalidad, y hasta qué nivel la implementan.

A.1.2. BSS

El sistema de estación base es responsable por todas las funciones relacionadas con el radio del sistema, esto incluye:

²Data Terminal Equipment/Data Circuit-Terminating Equipment

- Enlace de radio entre el móvil y la red
- Handover

No profundizamos en esta área, dado que como vemos más adelante, no está dentro del alcance del proyecto detectar problemas en esta parte de la red.

Básicamente se compone por 3 elementos:

- BTS o RBS - Base Transceiver Station o Radio Base Station - Controla la interfaz de radio con el móvil, comprende los equipos de radio y las antenas
- BSC - Base Station Controller - Controla las funciones de radio de la red; asignación de TS (*Time Slots*), handover, etc, para varias BTSs
- TRAU - Transcoder Rate Adapter Unit - Nodo que adapta la tasa de transmisión de los canales de voz. Este nodo sólo afecta al tráfico de voz, dado que el tráfico de señalización (incluyendo los SMSs) no son modificados

A.1.3. NSS o SS

El sistema de conmutación de la red comprende a todos los equipos de *core* de la misma. Existen muchos elementos en esta área, mostramos aquí los más importantes:

- MSC - Mobile Service Switching Center - Este es el conmutador central de la red. Es por este equipo que se conmutan todas las llamadas de voz de la red, y donde se concentran los vínculos de señalización con el BSS
- HLR - Home Location Register - Base de datos que contiene la información sobre los usuarios
- VLR - Visitor Location Register - Es la base de datos local de MSC. Cuando un móvil se asocia a la red, se copia la información de HLR al VLR
- GMSC - Gateway MSC - Es el equipo que interconecta la red GSM con otras redes (PLNM, PSTN, etc)
- SMS-C - Centro de mensajería instantánea - Nodo que se encarga de procesar y enrutar SMSs
- SMS-Gateway - Este equipo se encarga de la interacción del servicio de mensajería corta entre la red GSM y otras redes

Estas funciones no siempre están separadas, muchas veces un solo equipo cumple más de una función. Pero es importante destacar que todas estas funciones tienen interfaces estándar, y por lo tanto cada una podría llegar a ser implementada por distintos fabricantes.

A.1.4. NMS

Esta área se encarga del monitoreo de la red, y controla ciertos parámetros de la misma. Se comunica con el MSC y los BSC por medio de una red X.25 o IP.

A.2. Protocolos e interfaces

La interacción y comunicación entre los distintos elementos de la red, se da mediante distintos protocolos de señalización que vemos a continuación. Las interfaces son denominadas de la siguiente manera:

- Interfaz Um
- Interfaz Abis
- Interfaz A
- Interfaces de la NSS

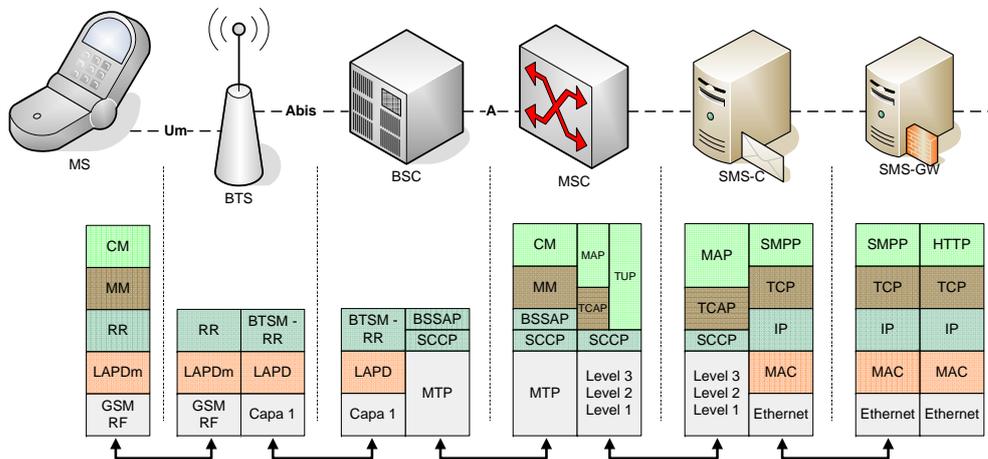


Figura A.2: Stack de protocolos de señalización para el envío de mensajes ([24])

A.2.1. Interfaz Um

Corresponde a la interfaz entre el móvil y la BTS. Especifica el tipo de modulación (GMSK), separación entre tramas, cantidad de TS por trama, etc.

Además se especifica el protocolo de señalización utilizado. Se hace un agrupamiento por canales lógicos. Cada canal lógico cumple distintas funciones:

- BCH - Broadcast Control Channel - Se encarga del sincronismo, da información de la red a todos los móviles, etc.
- CCH - Common Control Channel - Se encarga de llamar a los móviles para abrir un canal dedicado.
- DCCH - Dedicated Control Channel - Este canal se abre entre un terminal y la red, y se utiliza para establecer llamadas, hacer handover, y enviar SMSs.

Normalmente se utiliza un número fijo de canales físicos para transportar estos canales lógicos. Es posible que en condiciones donde hay picos de tráfico o si no está bien dimensionado el canal de señalización, el servicio de mensajería corta se vea afectado.

Este tipo de afectación es local a la BTS dado que se trata de los canales disponibles en el aire.

En particular para los SMSs se utiliza un stack de protocolos como muestra la figura A.2. La capa 2 según modelo OSI (capa de enlace de datos) está basada sobre LAPD (Link Access Protocol in the Dedicated Channel) proveniente de ISDN, y sobre ésta se soportan las capas de CM (Connection Management), MM (Mobility Management) y RM (Resource Management).

A.2.2. Interfaz Abis

Esta interfaz interconecta la BTS con el BSC. Para la señalización se utiliza también LAPD como protocolo de capa 2 y preserva las capas superiores. La función de la BTS es simplemente de switcheo hacia la BSC.

A.2.3. Interfaz A

La interfaz A conecta al BSC con el MSC. El BSC utiliza protocolos basados en SS7, en particular utiliza MTP (Message Transfer Part) como protocolo de capa 2 y SCCP (Signalling Connection Control Part) como protocolo de capa 3. Las capas de CM y MM permanecen intactas desde el móvil hasta el MSC.

A.2.4. Interfaces de la NSS

El MSC tiene implementado el stack de protocolos para comunicarse, por un lado con el BSS, y por otro lado con el NSS. Los protocolos del NSS están basados en SS7, utilizando MAP-TCAP en las capas superiores. Sobre estos protocolos el MSC se comunica con el SMS-C para pasarle los SMSs.

A.3. Proceso de envío de SMS

En lo que sigue, ejemplificamos el proceso de envío de mensajes de texto entre dos terminales móviles dentro de un mismo operador. Se hace referencia a los distintos elementos definidos previamente en esta sección.

1. El móvil origen transfiere el mensaje al MSC
2. El MSC interroga al VLR para verificar que el mensaje debe ser transferido³
3. El MSC envía el mensaje al SMS-C
4. El SMS-C interroga al HLR sobre cuál es el MSC que sirve al móvil destino
5. El SMS-C envía el mensaje a dicho MSC
6. El MSC busca la información del destinatario en el VLR
7. El MSC transfiere el mensaje al móvil destino
8. El MSC avisa al SMS-C el resultado del envío

³Consulta si ese móvil tiene el servicio habilitado.

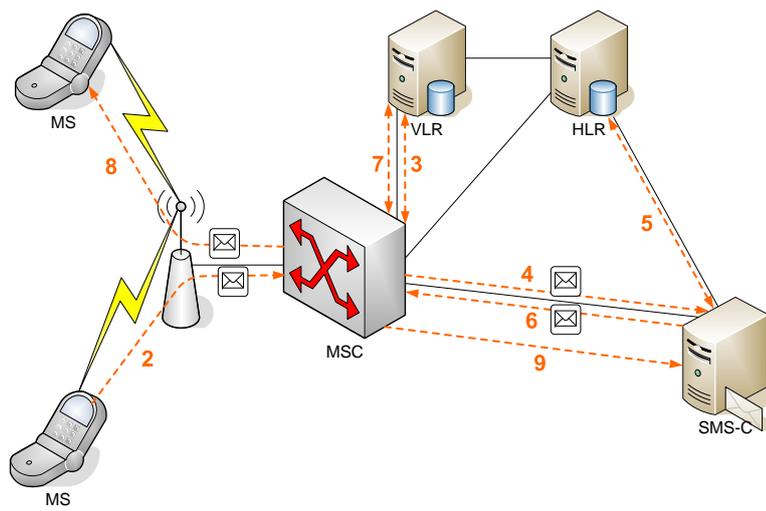


Figura A.3: Ejemplo de flujo de un mensaje de texto

Apéndice B

Manual del usuario

Leyendo este manual el operador será capaz de configurar, gestionar y obtener reportes y estadísticas provistos por el AST a través de su interfaz gráfica. La GUI consta de cuatro secciones principales: inicio, configuración, estadísticas y ayuda, que describimos a continuación. Además explicamos concretamente en la sección que corresponde las tareas más habituales que el operador debe realizar, como por ejemplo, cómo configurar un móvil, una conexión SMPP, etc.

¿Cómo acceder?

Para acceder a la interfaz gráfica es necesario saber en qué máquina se encuentra el servidor web.

Abrir un navegador y dirigirse a

`http://nombre o dirección del servidor web/GUI/`

Se abrirá entonces una pantalla como la que se puede ver en la figura B.1



Application service Tester

[Inicio](#) | [Configuración](#) | [Estadísticas](#) | [Ayuda](#)

Estado del Sistema
Estado del AST

Estado del AST

Aquí se puede ver el estado general del AST.

Parámetro	Estado
En ejecución	NO
Modos Activos	3
Celulares Conectados	0
Celulares en Uso	0
Gateways Conectados	0
Gateways en Uso	0

[Acerca De](#) | [Política de privacidad](#) | [Contactenos](#) | ©2008 AST - Sofia Depons, Marcelo Fiori, Julio Fitipaldó, Sergio Martínez

Figura B.1: Página inicial

B.1. Inicio

Aquí se pueden ver tanto el estado del servidor en que se ejecuta el AST como el estado del AST mismo en tiempo real. Los parámetros a los que se puede acceder se listan a continuación.

B.1.1. Estado del sistema

- *Memoria total*: Indica la capacidad de memoria total del PC
- *Memoria en uso*: Indica la memoria que está siendo empleada actualmente

B.1.2. Estado del AST

- *En ejecución*: Indica si el AST está siendo ejecutado o no
- *Modos activos*: Un modo se refiere a una asociación origen-destino. Por lo tanto la cantidad de modos activos indica la cantidad de combinaciones emisor-receptor que se están empleando
- *Celulares (Gateways) conectados*: Cantidad de terminales móviles (gateways) conectados físicamente al PC
- *Celulares (Gateways) en uso*: Cantidad de celulares (gateways) que el AST está empleando para enviar y/o recibir mensajes

B.2. Configuración

En esta sección se pueden ver y editar todos los parámetros que necesita el AST para funcionar. Están agrupados de acuerdo al área que afectan

B.2.1. General

- *TimeOut*: Es el tiempo máximo que el programa espera alguna respuesta de un terminal móvil. Pasado ese tiempo la consulta expira.
- *TActualización*: Indica cada cuánto tiempo la aplicación debe chequear si hubo alguna modificación en la configuración del AST.
- *TGrilla*: Es el período con el que se muestrea la red.

B.2.2. Móviles y Conexiones SMPP

Aquí se listan todos los móviles y conexiones SMPP que alguna vez fueron conectados al AST y sus características. Un ejemplo de estas tablas puede verse en la figura B.2.

Los campos en la tabla son los siguientes:

- *ID*: Este numero es el identificador del móvil o de la conexión SMPP. Por móvil se entiende el conjunto de un terminal y una SIM determinada.

Móviles

Lista de móviles

Aquí se listan todos los móviles que alguna vez fueron conectados al AST.

Para poder emplear un móvil por primera vez éste debe conectarse y luego se deben completar algunos datos, para ello pinche en editar. Para obtener más información acerca de los campos de la tabla pinche sobre ellos.

Editar	ID	Número	IMSI	IMEI	Conectado	Activo	En uso
	1	59898390757	748010030019314	358954001277249			
	2	59898390756	748010030019313	358954001023569			
	30	59898390757	748010030019314	358269003121679			
	31	59898390757	748010030019314	350379020772244			

Lista de conexiones SMPP

Aquí se listan todas las conexiones SMPP que tiene el AST.

Explicacion

Editar	ID	Número	IP	Puerto	Conectado	Activo	En uso
	29	59890001018	10.35.33.249	9201			

Figura B.2: Tabla de móviles

- *Número*: Se refiere al MSISDN (Número de Suscripción de ISDN del Móvil) en una red GSM. Su longitud máxima es de 15 dígitos. Se compone de un símbolo de “+”, el código de país, el código del proveedor de servicio y finalmente el número “de teléfono” del abonado. Por ejemplo: +598 99 717 660
- *IMSI*: Es el acrónimo de International Mobile Subscriber Identity (Identidad Internacional del Suscriptor Móvil). Es un código de identificación único para cada usuario de telefonía móvil, integrado en la tarjeta SIM. Se conforma por tres dígitos correspondientes al país, dos dígitos para codificar al operador de la red y finalmente diez dígitos que comprenden la identificación del usuario en la red del operador.
- *IMEI*: Es el acrónimo de International Mobile Equipment Identity (Identidad Internacional de Equipo Móvil). Es un código grabado en los teléfonos móviles GSM, que identifica al equipo unívocamente.
- *IP*: Dirección de red del SMS-Gateway
- *Puerto*: Puerto TCP del SMS-Gateway (por defecto 9201)
- *Conectado*: Indica si el móvil se encuentra físicamente conectado al sistema, o, en el caso del SMS-Gateway, si tiene la sesión activa.
- *Activo*: Indica que el móvil o la conexión ya están configurados en el sistema.
- *En uso*: Indica si el móvil o la conexión están siendo empleados por el sistema.

¿Cómo agregar un nuevo móvil?

Para poder emplear un móvil por primera vez éste debe conectarse. Luego de haber estado conectado físicamente al AST, el móvil aparece en la lista de la figura B.2. Es necesario agregar algunos datos, para ello debe seleccionar *Editar* y se abrirá una ventana como se puede ver en la figura B.3. Los campos a ingresar son el *número* y el operador que brinda servicio al terminal. Además, dado que no todos los móviles tienen la funcionalidad de enviar y recibir de esta manera es necesario seleccionar cuáles son las capacidades del terminal. Finalmente seleccionar *Activar* para dar de alta al móvil. Cuando los datos estén completos pulse Actualizar y verá el móvil ingresado a la tabla de móviles.

Actualizar datos de móvil

Marca:

huawei

Modelo:

E220

IMSI:

748010030019314

IMEI:

358954001277249

Debes completar los siguientes parámetros asociados al móvil. Si deseas dar este móvil de alta en el sistema debes seleccionar la opción Activar.

Número:

59898390757

Service Provider:

ANCEL

El móvil es capaz de:

Envía:



Recibe:



Activar:



Actualizar

Figura B.3: Editar datos de un terminal móvil

¿Cómo agregar una nueva conexión SMPP?

Para agregar una nueva conexión SMPP pulse *Agregar una nueva conexión* y complete los siguientes datos: IP, puerto, número y service provider y, al igual que para los móviles, debe seleccionarse si es capaz de recibir y/o enviar. Finalmente seleccionar *Activar* para dar de alta la conexión. Cuando los datos estén completos pulse Actualizar y verá la conexión ingresada a la tabla.

B.2.3. Modos

Un modo determina la interfaz de origen y destino de cada mensaje. Puede haber más de un modo activo funcionando a la vez. En la figura B.4 se puede ver una lista donde se encuentran los modos que alguna vez fueron configurados en el sistema. En esta lista se pueden ver todos los parámetros de cada modo configurado. Los campos de la tabla son los siguientes:

- *Eliminar*: Pulsando aquí se elimina el modo
- *A/D*: Pulsando aquí se activa o desactiva el modo
- *ID origen (destino)*: Identificador del dispositivo de origen (destino), ya sea una conexión SMPP o un móvil
- *Número origen (destino)*: Corresponde al MSISDN del dispositivo de origen (destino)
- *Descripción*: Comentario que identifique al modo
- *Activo*: Indica si el AST lo está empleando, es decir, si se están cursando mensajes para ese modo
- *Monitoreado*: Indica si se está monitoreando la calidad del modo

Lista de modos

En la siguiente lista se encuentran todos los modos que alguna vez fueron configurados en el sistema.

Eliminar	A/D	ID Origen	Número Origen	ID Destino	Número Destino	Descripción	Activo	Monitorear
	Activar	2	59898390756	1	59898390757	movil2 - movil1		
	Activar	29	59890001018	1	59898390757			
	Activar	1	59898390757	2	59898390756	movil1 - movil2		
	Activar	29	59890001018	2	59898390756	smsc - movil2		
	Activar	1	59898390757	29	59890001018	movil1 - smsc		
	Desactivar	2	59898390756	29	59890001018	movil2 - smsc		
	Activar	2	59898390756	30	59898390757	huawei motorola		
	Activar	29	59890001018	30	59898390757	smsc motorola		
	Desactivar	2	59898390756	31	59898390757	Hua-Ericc		
	Desactivar	29	59890001018	31	59898390757	SMSC-Ericc		

 [Agregar un modo nuevo](#)

Figura B.4: Tabla de modos

¿Cómo crear un modo nuevo?

Para crear un modo nuevo debe pulsar *Agregar un modo nuevo*. En la ventana que se abre a continuación se deben seleccionar el origen y destino del modo, de una lista de móviles. Los móviles son elegibles dependiendo de su capacidad para enviar y recibir indicadas al configurar el móvil y/o la conexión SMPP. Además debe ingresarse la descripción del modo. Finalmente al pulsar *Crear modo* se retorna a la lista de modos, donde aparece el recientemente creado.

B.2.4. Modelado y QoE

En esta sección pueden ver los tipos de usuarios creados y crear nuevos. A partir de estos usuarios se realiza el monitoreo del servicio. En la pantalla inicial de esta subsección se puede ver una lista de usuarios que son los activos en el momento. De estos usuarios sólo se pueden ver algunos atributos. Para verlos todos pinche en *Ver Atributos*:

- *Tipo de usuario*: comentario que describe al tipo de usuario
- *Cantidad de usuarios corriendo*: cantidad de usuarios virtuales con los mismos parámetros que están siendo simulados en simultáneo
- *Modo asociado*: descripción del modo en que trabaja el tipo de usuario
- *Tabla de transición de estados*: en cada espacio de la tabla está indicada la probabilidad de pasar del estado de la fila al estado de la columna.¹
- *Tabla que define el comportamiento correspondiente a cada estado*². En particular, la distribución que modela el tiempo entre mensajes se indica con un valor entero (1 indica distribución uniforme y 2 indica distribución exponencial). Para cada distribución se indica el parámetro característico de la misma (por ejemplo en el caso de la exponencial, el valor de λ). Por último, se listan los umbrales de QoE en milisegundos.

Al pinchar en *Editar* solamente podrá modificar la cantidad de usuarios de ese tipo.

¿Cómo agregar nuevos tipos de usuario?

Al pulsar sobre *Agregar un tipo de usuario nuevo* sobre el final de la pantalla el primer dato a completar es la cantidad de estados del usuario. Esta cantidad de estados refiere al modelo de usuario empleado por el AST. Tal cual fue la intención del capítulo 4, el AST contempla el hecho de que el usuario establezca una conversación a través de mensajes de texto. De esta manera la cantidad de estados significa la cantidad de mensajes que el usuario envía a lo largo de la conversación³. Al indicar la cantidad de estados del tipo de usuario y pulsar *Crear usuario* se pasa a una pantalla como la que se ve en la figura B.5.

Allí se deben ingresar los datos listados anteriormente, en función del tipo de usuario deseado.

¹En la subsección siguiente se brinda más información a este respecto

²En la subsección siguiente se brinda más información a este respecto

³Por más información referirse a la sección 4.

Crear nuevo Usuario

En el siguiente formulario, debes completar los parámetros asociados al tipo de usuario que quieres crear.

Descripción:

Cantidad de usuarios corriendo:

Modo al que se asocia el usuario:
 movil2 - smsc ▾

Matriz de transición de estados, en cada espacio de la tabla debe llenarse la probabilidad de pasar del estado de la fila al estado de la columna.

<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Valores de estados... Explicar mejor

Estado	Distribucion	Parametro de dist.	Umbral 1	Umbral 2	Umbral 3	Umbral 4
1	Uniforme ▾	<input type="text"/>				
2	Uniforme ▾	<input type="text"/>				
3	Uniforme ▾	<input type="text"/>				

Figura B.5: Ingresar datos de usuarios

B.2.5. SNMP

Aquí se pueden ver los datos relacionados al agente SNMP.

- *IP Agente* Dirección de red de la máquina donde corre el agente SNMP
- *IP Servidor* Dirección de red de la máquina donde corre el agente SNMP
- *Red autorizada* Prefijo de red autorizada para realizar consultas SNMP
- *Máscara de red autorizada* Máscara para determinar las direcciones autorizadas.
- *Comunidad* Equivalente a la clave

Para modificarlos se debe pinchar en la lupa

B.2.6. Umbrales

En esta sección se definen los umbrales para los cuales se dispara una alarma. Si alguno de los valores en las medidas supera los umbrales se disparará una alarma.

Umbrales de QoS

Aquí aparece una tabla con los modos cuya calidad se está monitoreando. A su vez se muestran dos umbrales para cada modo. Si el tiempo de entrega de un SMS supera

el umbral 1 se disparará una alarma, si superara el umbral 2 se disparará una alarma de mayor gravedad

¿Cómo seleccionar los modos a monitorear y sus umbrales?

Al final de la tabla mencionada recién se brinda la posibilidad de seleccionar los modos a monitorear y los umbrales asociados a ese modo. Al pinchar aquí se puede ver una tabla similar a la de los modos, con la salvedad de que para cada modo hay dos campos nuevos: umbral 1 y umbral 2. Al seleccionar *Editar monitoreo* se setean estos valores y se selecciona *Monitorear*. Finalmente pulse *Actualizar*.

Umbrales de QoE

Aquí se puede ver el umbral de alarma establecido para la calidad percibida por el usuario. Por otra parte se ven los ponderadores que indican qué tanto pesa una experiencia mala respecto a una buena. A partir de estos valores se calcula un valor para la QoE que si supera el umbral disparará una alarma.

B.3. Estadísticas

B.3.1. Ver valores actuales

Se visualizan los valores de desempeño de la red en tiempo real.

B.3.2. Generar reportes

En esta sección se pueden pedir informes respecto a la QoS y QoE elaborados por el sistema. Para poder obtener estos reportes es imprescindible que el AST se encuentre en ejecución.

Informe estándar

El informe estándar brinda información de QoS para un determinado modo. Se muestran gráficas independientes para el día en que se realiza el pedido, la semana, el mes y el año. También se muestran las gráficas en escala logarítmica para poder apreciar, si hubiere, grandes rangos de valores. Para pedir un informe pulse en *Pedir informe estándar*. A continuación seleccione el modo, cuyo reporte desea obtener y presione *Crear reporte*. Se abrirá una ventana con un enlace al reporte en formato pdf.

Informe personalizado

El informe personalizado permite realizar reportes comparativos entre modos y a su vez seleccionar el rango de fechas a considerar. Presenta además, al igual que el informe estándar, las gráficas en escala logarítmica. Para solicitar el reporte presione *Pedir informe personalizado*. En la pantalla que aparece a continuación se pueden seleccionar los modos deseados y la fecha y hora de inicio y fin del reporte. De la misma manera que en el informe estándar al presionar *Crear reporte* se abrirá una ventana con un enlace al reporte en formato pdf.

Informe de QoE

El informe de QoE permite observar cómo los distintos tipos de usuarios perciben el desempeño de la red. Para solicitar el reporte presione *Pedir informe QoE*. A continuación complete la cantidad de usuarios que desea simular de cada tipo, y el rango de fechas a considerar. Al presionar *Crear reporte* se enviará la consulta, y se devolverá el enlace al reporte en formato pdf.

Apéndice C

Gestión del proyecto

En este capítulo describimos parte de la planificación original del proyecto, incluyendo recursos como tiempo y costos, y parte del análisis de riesgos; luego comparamos dicha planificación con el desarrollo real que tuvo el proyecto, analizando los desvíos.

C.1. Riesgos

Dentro del análisis de riesgos, al comenzar la planificación, identificamos los siguientes (entre otros):

1. Que no se consigan MS con las características supuestas para la implementación del sistema
2. Las funcionalidades proporcionadas por el SMS-Gateway no cumplen en su totalidad los requerimientos supuestos
3. Mala estimación de tiempo de las tareas y asignación dentro del grupo

Los ítem 1 y 2 se calificaron con probabilidad moderada e impacto medio, y el ítem 3 como poco probable, pero con alto impacto:

	Probabilidad		
Impacto	Poco probable	Moderada	Muy probable
Ninguno			
Bajo			
Medio		1,2	
Alto	3		
Extremo			

Como fue enunciado en el capítulo 9, durante el transcurso del proyecto, sufrimos la ocurrencia del riesgo enumerado en primer lugar. A saber, los fabricantes de terminales cumplen las especificaciones a medias, y debido a los terminales móviles tuvimos grandes problemas para mantener el sistema funcionando durante períodos de tiempo prolongados.

Con mucho trabajo extra, que se ve reflejado en las secciones a continuación, solucionamos dichos inconvenientes.

Por otro lado, un riesgo no previsto correctamente fue la dependencia de un actor externo como lo es ANTEL. A pesar de que se pudieron lograr todos los objetivos, se presentaron problemas de discordancia de plazos entre el grupo de proyecto y el organismo estatal. Esto generó desviaciones importantes en la planificación original.

C.2. Tiempo

En la siguiente tabla vemos una comparación entre los plazos y comienzos de trabajo planificados y los reales. En general vemos que muchas de las tareas ocupan una porción bastante más grande de tiempo que lo planificado. Esto no implica necesariamente un error en el cálculo de la dedicación horaria, como vemos en el siguiente capítulo no hubo diferencias tan grandes entre la dedicación planificada y la real. La mayor parte del trabajo en las tareas fue realizado dentro de los plazos. Sin embargo, debido a dependencias con otros trabajos no planificadas o la incidencia de factores externos, algunas tareas se vieron interrumpidas, o postergado su inicio.

Tarea	Comienzo Planificado	Comienzo Real	Duración planificada	Duración real
1. Diseño UML	13/09/07	09/09/07		
1.1. Kernel, Base de Datos e Interfaces	13/09/07	09/09/07	50 días	45 días
1.2. Alarmas y estadísticas	08/11/07	23/02/08	45 días	30 días
1.3. GUI	22/12/07	18/03/08	32 días	10 días
2. Hardware y S.O.	13/09/07	09/09/08		
2.1. Elegir tipo de hardware y S.O.	13/09/07	09/09/07	9 días	10 días
2.2 Implementar Hardware e instalar S.O.	24/05/08	05/03/08	20 días	83 días
3.Lenguaje de Programación	22/09/07			
3.1. Elección	22/09/07	22/09/07	5 días	3 días
3.2. Aprendizaje	27/09/07	22/09/07	138 días	125 días
4. Modelado de usuarios	30/09/07	25/01/08	47 días	63 días
5. GUI	10/02/08	18/03/08	79 días	70 días
6. Interfaz con MS y SMS-GW	16/12/07	11/10/07	60 días	300 días
7. Alarmas y estadísticas	17/02/08	01/03/08	62 días	75 días
8. Algoritmo de usuario	13/01/08	25/03/08	62 días	30 días
9. Base de datos	08/11/07	07/11/07	44 días	224 días
10. Kernel	06/01/08	11/10/07	79 días	104 días
11. Testeo	29/04/08	01/07/08	92 días	65 días
12. Recopilación y armado de documentación	25/05/08	27/05/08	47 días	121 días

C.3. Costos

Hay dos fuentes de costos evaluables para este proyecto, en un primer lugar los que derivan del costo de los materiales utilizados, y por otro lado las horas hombre invertidas en el mismo. Debido a que en este proyecto no se puede asociar un costo en dinero a las horas trabajadas, hacemos un análisis independiente para ambas fuentes.

C.3.1. Costo en horas hombre

En la siguiente tabla, comparamos las horas hombre planificadas e invertidas realmente, para cada una de las tareas.

Tarea	Costo Planificado (horas hombre)	Costo Real (horas hombre)
1. Diseño UML		
1.1. Kernel, Base de Datos e Interfaces	250	200
1.2. Alarmas y estadísticas	80	100
1.3. GUI	20	20
2. Hardware y S.O.		
2.1. Elegir tipo de hardware y S.O.	10	12
2.2 Implementar Hardware e instalar S.O.	25	50
3. Lenguaje de Programación		
3.1. Elección	5	10
3.2. Aprendizaje	220	300
4. Modelado de usuarios	150	250
5. GUI	180	200
6. Interfaz con MS y SMS-GW	400	650
7. Alarmas y estadísticas	150	160
8. Algoritmo de usuario	200	250
9. Base de datos	80	100
10. Kernel	130	150
11. Testeo	250	400
12. Recopilación y armado de documentación	450	700
Total	2600	3552

Por otro lado, algunas de las tareas que tuvimos que realizar no estaban dentro de lo planificado, porque fueron requerimientos nuevos que surgieron en el transcurso del proyecto (por políticas de ANTEL por ejemplo). La más importante fue la implementación del Proxy SMPP-HTTP, que significó unas 40 horas hombre, debido a que se pudo reutilizar trabajo realizado para el paquete *mensajería*.

C.3.2. Costo en materiales

El costo de materiales no varía sobre las proyecciones iniciales. No se considera aquí el costo del envío de los mensajes a través de la red del operador, dado que en nuestro caso es el mismo operador el que esta interesado en las pruebas y presenta un costo prácticamente nulo para cursar los mensajes de prueba.

Equipo	Costo en U\$S
PC	100
2 MS	500
SIMs	Costo despreciable

C.4. Evaluación de la gestión

En líneas generales se respetó la planificación realizada originalmente, sobretudo en la primera mitad. Luego se intercambiaron algunas tareas, principalmente se adelantaron algunas tareas de implementación de los módulos ya diseñados, y se atrasó el diseño completo del resto de los módulos.

En cuanto a las horas de trabajo, las diferencias más grandes entre los recursos planificados y los utilizados se encuentran en la implementación de las interfaces con los terminales móviles y equipos SMPP, y en la documentación final. La primera de las actividades mencionadas, se extendió en tiempo y recursos por dos razones principales: las demoras burocráticas para la obtención de materiales por parte de ANTEL, y los problemas de implementación por incumplimientos de los estándares por parte de los fabricantes de equipos móviles. Por otro lado, la elaboración de la documentación fue subestimada, además de que no se hizo paralelamente al desarrollo del proyecto (por lo menos no consistentemente) como fue estipulado al inicio.

En cualquier caso, los desvíos, tanto en plazos como en horas de trabajo, fueron absorbidos por los *buffers* de tiempo incluidos en la planificación.

Apéndice D

Contenido del CD

En el CD se encontrará la siguiente información ubicada en la raíz del mismo:

- El archivo *documentación.pdf* contiene la documentación completa.
- El archivo *licencia.txt* contiene la especificación de la licencia del software.
- El archivo *diagrama.png* contiene el diagrama de clases completo del sistema.
- Directorio */AST* - Contiene todos los archivos de código y compilados que componen la aplicación en Java. Dentro del mismo se encuentra un script *bash* que al ser ejecutado inicia el programa. Referimos al lector al capítulo 9 donde se especifican los requerimientos necesarios para el correcto funcionamiento.
- Directorio */Proxy* - En este directorio se encuentran todos los archivos que componen la aplicación en Java Proxy-SMPP-HTTP.
- Directorio */GUI* - En este directorio se encuentran todos los archivos *php*, *html*, e imágenes que componen la interfaz gráfica. Éstos deben ser instalados en el servidor web
- Directorio */BD* - Este directorio incluye un único archivo que corresponde a una imagen completa de la base de datos que permite una vez instalada la aplicación de base de datos, recuperar todas las configuraciones necesarias para el correcto funcionamiento.

Glosario

AD-SMS-MO	Access Delay SMS Mobile Originated, 34
ANTEL	Administración Nacional de Telecomunicaciones, 8, 9, 12, 13, 67, 83, 107, 110
AST	Application Service Tester, 10–13, 36, 38, 45, 55, 64, 66, 82, 111
AT	Grupo de comandos para modems, proveniente de la palabra atención, 8, 12, 50
BSC	Base Station Controller - Controla las funciones de radio de la red; asignación de TS, handover, etc, para varias BTSs, 93, 95
BSS	Base Station Subsystem - Grupo de elementos de la red GSM que realiza las funciones relacionadas con el radio de una red GSM, 91–93, 95
CD	Compact Disc, 50, 67, 111
CMTC	Cadena de Markov de Tiempo Continuo, 27, 28, 30
ETSI	European Telecommunications Standards Institute, 34, 67
GSM	Global System for Mobile communications, 4, 7, 11, 12, 34, 35, 45, 50, 52, 67, 73, 83, 84
GUI	Graphic User Interface - Interfaz gráfica, 36, 49, 55, 65, 111
HTTP	Hypertext Transfer Protocol, 13, 110
IMEI	International Mobile Equipment Identity - Este código identifica al ME unívocamente a nivel mundial, 37, 51, 57, 99
IMSI	International Mobile Subscriber Identity - Código de identificación único para cada usuario de telefonía móvil, 37, 51, 57, 92, 99
IP	Internet Protocol, 11, 12, 37, 50, 51, 55, 57, 67, 93, 100, 103

ITU-T	Telecommunication Standardization Sector, 33
KPI	Key Performance Indicators - Indicadores Clave de Desempeño, miden el nivel del desempeño, 34–36
L^AT_EX	Lamport TeX - Language utilizado para la creación de documentos de texto, basado en T _E X. T _E X proviene del griego, significa Arte, Técnica, y fue creado por Donald Knuth, y adaptado luego por Lamport creando L ^A T _E X, 70
LATTIA	Laboratorio de Tecnología de Telecomunicaciones de ANTEL, 67
ME	Mobile Equipment - equipo móvil, 92
MOS	Mean Opinion Score, 42
MS	Mobile Station - Conjunto del equipo móvil y SIM, 13, 51, 91
MSC	Mobile Service Switching Center - Conmutador central de la red GSM, 93, 95
MySQL	Relational database management system, 70
NMS	Network Managment System - Realiza el monitoreo y control de la red GSM, 91, 93
NSS	Network Switching System - Elementos de la red GSM que cumplen funciones de conmutación, facturación, comunicación con otras redes, etc, 5, 91, 93–95
PC	Personal Computer, 8, 13, 63, 65, 67, 69
PDU	Protocol Data Unit, 35
QoE	Quality Of Experience - Calidad de Experiencia, 6, 7, 9, 41–45, 53, 55, 62, 64, 79, 83
QoS	Quality Of Service - Calidad de Servicio, 6, 7, 33, 34, 36, 38, 39, 42, 55, 62, 64, 73, 79, 83
RBS	Radio Base Station, 73, 77, 78, 84, 93
RS-232	Recomended Standard 232, 67, 69
SIM	Subscriber Identity Module - Tarjeta que almacena información del usuario. También conocida como “chip”, 12, 60, 91, 92

SMPP	Short Message Peer to Peer Protocol - Protocolo estándar de telecomunicaciones pensado para el intercambio de mensajes SMS, 12, 13, 50, 57, 73, 97, 98, 100, 101, 110
SMS	Short Message Service - En este documento se hace referencia por esta sigla tanto a los mensajes como al servicio, 3, 6, 7, 21, 26, 30, 33, 34, 36, 39, 42, 45, 51, 52, 55, 61, 64, 66, 69
SMS-C	SMS Center - Centro de Mensajería, elemento de la red GSM cuya función es la de enviar/recibir mensajes SMS, 12, 13, 34, 36-38, 69, 73-76, 78, 79, 82, 93, 95
SMS-GW	SMS Gateway - Este equipo se encarga de la interacción del servicio de mensajería instantánea entre la red GSM y redes externas, 12, 13, 36, 73, 93, 107
SNMP	Simple Network Management Protocol - El Protocolo Simple de Administración de Red es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red, 4, 15, 38, 49, 53, 55, 59, 64
SS7	Signaling System 7, 95
URSEC	Unidad Reguladora de Servicios de Comunicación, 39
USB	Universal Serial Bus, 51, 67-69
WEB	Red, haciendo referencia a una red de datos como Internet, 5, 15, 70, 97