



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“ANÁLISIS E IMPLEMENTACIÓN EN EL LENGUAJE ABIERTO
PREPROCESADOR DE HIPERTEXTO (PHP) SOBRE LINUX DE UNA
CENTRAL DE CONMUTACIÓN (MSC) VIRTUAL PARA LA INTERACCIÓN A
NIVEL DE PROTOCOLO DE PARTE DE APLICACIONES MÓVILES (MAP)
CON EL REGISTRADOR DE UBICACIÓN DE ABONADO, (HLR)”

INFORME DE MATERIA DE GRADUACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

PRESENTADO POR:

Jefferson Gustavo Suárez Piloza

David Hernán Paucar Naranjo

GUAYAQUIL – ECUADOR

2015

AGRADECIMIENTOS

Al culminar con éxito nuestros estudios universitarios queremos agradecer primeramente a Dios por brindarnos salud y vida para haber podido asistir a todas nuestras clases designadas de la malla curricular, agradecemos a nuestros padres por ser los que nos impulsaron día a día a salir adelante reflejados en su ejemplo y el apoyo económico que nos brindaron en todos nuestros estudios. Agradezco a la ESPOL por su excelencia académica y a sus docentes por todos los conocimientos adquiridos jornada tras jornada.

DEDICATORIA

Este objetivo cumplido llevo mucho sacrificio y tiempo, lo dedico todo mi trabajo a mi madre Mercy Pilozo y a mi querida abuela Virginia García por todo su apoyo y sus palabras de aliento que me brindaba.

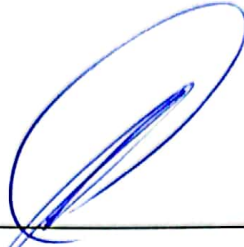
Le agradezco profundamente a mi querido padre Washington Suarez por su comprensión, dedicación y apoyo en mis estudios, además a mi Tía Mercedes Suárez que día tras día me brindó sus consejos para poder lograr esta meta.

Jefferson Suárez Pilozo

Dedico todo el esfuerzo de mi trabajo a mis padres Hernán Paucar y Silvia Naranjo, a mis hermanas Valeria y Daniela, a mi abuelito Luis Paucar, a mis tías y a mis primas, ya que todos ellos han sido un pilar fundamental durante toda mi carrera universitaria.

David Paucar Naranjo

TRIBUNAL DE SUSTENTACIÓN



Ing. José Miguel Menéndez S., Msc.

PROFESOR DE LA MATERIA DE GRADUACIÓN



Ing. Víctor Asanza, MG.

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este informe, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".

(Art. 12 del Reglamento de Graduación de la ESPOL)



JEFFERSON SUÁREZ PILOZO



DAVID PAUCAR NARANJO

RESUMEN

El presente informe consiste en realizar una simulación de la señalización del protocolo de Aplicaciones Móviles (MAP) para demostrar el cambio de localización de un abonado cuando cambia de cobertura, describir los parámetros que son enviados y recibidos en cada mensaje de señalización entre la Estación Móvil (MS), la MSC NEW, la MSC OLD y el HLR.

La programación de la señalización se la realizará en lenguaje abierto preprocesador de hipertexto y para la presentación se armará una topología de red donde intervendrán *switch*, *router*, laptop que simularán la interacción de los componentes mencionados en el primer párrafo.

Se observará el cambio de ubicación de Guayaquil a Quito o viceversa, los parámetros que posea como información dicho número telefónico se actualizarán siguiendo los mensajes de señalización MAP.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	II
DEDICATORIA	III
DECLARACIÓN EXPRESA.....	V
RESUMEN	VI
ÍNDICE GENERAL.....	VII
ABREVIATURAs.....	X
ÍNDICE DE FIGURAS.....	XII
INTRODUCCIÓN	XVI
CAPÍTULO 1	1
1. DESCRIPCIÓN GENERAL.....	1
1.1. Antecedentes SS7	1
1.2. Antecedes MAP (Parte de Aplicaciones Móviles)	2
1.3. Justificación de Protocolo MAP	3
1.4. Descripción del Proyecto	3
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos	4

1.5. Alcance	5
1.6. Limitaciones	5
CAPÍTULO 2.....	7
2. FUNDAMENTOS TEÓRICOS.....	7
2.1. Señalización SS7 – Protocolo MAP	7
2.2. Red Celular GSM (Global System for Mobile).....	12
2.3. Topología Celular – Operaciones de Protocolo MAP.....	13
2.4. Centro de Conmutación Móvil (MSC).....	19
2.5 Lenguaje de Programación PHP (Preprocesador de Hipertexto)...	20
2.6 Sistema Operativo Linux	22
CAPÍTULO 3.....	25
3. DESARROLLO DEL PROYECTO	25
3.1 Proceso de Ubicación del Abonado Móvil.....	25
3.2 Elaboración del Proyecto	28
3.2.1 Software	28
3.2.2 Hardware	36
3.3 Proceso de implementación	41
3.3.1 Arquitectura de la Red.....	41

3.3.2 Señalización	42
3.3.3. Código Fuente – PHP	46
3.3.4 Implementación	55
3.4 Aplicaciones	59
3.4.1 Aplicaciones Educativas	59
3.4.2 Aplicaciones Comerciales.....	60
CAPÍTULO 4.....	62
4. ANÁLISIS DE RESULTADOS	62
4.1 Resultados obtenidos.....	62
4.2 Simulaciones.....	63
4.3. Interpretación de Resultados.....	73
CONCLUSIONES	78
RECOMENDACIONES.....	81
BIBLIOGRAFÍA.....	83

ABREVIATURAS

AMPS	Sistema de Teléfono Móvil Avanzado
A3	Algoritmo de autenticación de celulares
A8	Algoritmo para generar clave de celulares
BS	Estación Base
CC	Código del País
CDMA	Multiplexación por División de Código
CLIP	Identificador de Llamadas
CLIR	Identificación Privada
GPS	Sistema de Posicionamiento Global
GSM	Sistema Global Móvil
HTML	Lenguaje de Marcas de Hipertexto
HLR	Registro de Localización Local
ID	Identificación del usuario
IMSI	Identidad del Abonado Internacional a un móvil
ISUP	Protocolo de Circuitos Conmutados
KC	Llave Cifrada
KI	Autenticación individual de un suscriptor
LAI	Identificador de Localización de Área

MAP	Parte de Aplicación Móvil
MNC	Código del País
MSC	Centro de Conmutación Móvil
MSIN	Serie del chipe del abonado asignado por el Operador
MSS	Estado de la Estación Móvil
NAMPS	Servicio de Teléfono Móvil Avanzado
NDC	Indicativo Nacional de Destino
NRI	Identificador de MSC
PHP	Preprocesador de Hipertexto
SS	Estado del Suscriptor
SS7	Sistema de Señalización SS7
TDMA	Multiplexación por División de Tiempo
TMSI	Identificador del Suscriptor Temporal Móvil
VLR	Registro de Localización del Visitante

ÍNDICE DE FIGURAS

Figura 2.1 Esquema SS7	8
Figura 2.2 Red Celular GSM.....	12
Figura 2.3 Esquem de red celular procesando protocolo MAP.....	13
Figura 2.4 Formato IMSI	15
Figura 2.5 Formato TMSI.....	16
Figura 2.6 Formato LAI.....	17
Figura 2.7 Proceso de Autenticación	18
Figura 2.8 Centro de Conmutación Móvil.....	20
Figura 2.9 Proceso de Ejecución-Código PHP	21
Figura 2.10 Interfaz Unity (Ubuntu 14.04).....	24
Figura 3.1 Registro de ubicación por primera vez.....	26
Figura 3.2 Registro de ubicación dentro de la misma MSC	27
Figura 3.3 Registro de ubicación cambio de MSC.....	27
Figura 3.4 Registro de ubicación por tiempo establecido	28

Figura 3.5 Instalación Ubuntu	30
Figura 3.6 Ingreso de nombre de equipo en Ubuntu.....	31
Figura 3.7 Comando para inicializar instalación XAMPP	32
Figura 3.8 Comando para ejecutar el archivo de instalacion XAMPP	32
Figura 3.9 Proceso de Instalación XAMPP	33
Figura 3.10 Instalación de XAMPP junto MySQL, PHP Y PERL.....	33
Figura 3.11 Designación de URL para ingreso de XAMPP.....	34
Figura 3.12 Activación de las aplicaciones contenidas en XAMPP.....	34
Figura 3.13 Instalación de Netbeans en Ubuntu	36
Figura 3.14 Laptop para simulación MSC 1	37
Figura 3.15 Laptop para simulación MSC 2	37
Figura 3.16 Laptop para simulación HLR.....	38
Figura 3.17 Laptop para simulación de MS	39
Figura 3.18 Router red inalámbrica – Cobertura Gye	39
Figura 3.19 Router red inalámbrica – Cobertura Quito	40
Figura 3.20 Switch para conexión MSC1 y MSC2	41

Figura 3.21 Topología de la Red usada en MAP	42
Figura 3.22 Señalización MAP realizada durante el informe	43
Figura 3.23 Programación mensaje Location_update_request.....	47
Figura 3.24 Código para envío de mensajes	48
Figura 3.25 Código función map_send_identity_request_ack	49
Figura 3.26 Código función authentication_parameter_request_ack.....	50
Figura 3.27 Código función authentication_reques.....	50
Figura 3.28 Código función map_update_location.....	51
Figura 3.29 Código función map_cancel_location_request	52
Figura 3.30 Código función map_insert_data_reques_ack.....	53
Figura 3.31 Código de función location_update_request_ack	54
Figura 4.1 Selección de número celular	64
Figura 4.2 Información del abonado seleccionado.....	65
Figura 4.3 Interfaz de Señalización MAP	66
Figura 4.4 Interfaz de información de la función de señalizacion.....	67
Figura 4.5 Interfaz función map_send_identity_request	68

Figura 4.6 Interfaz función authentication_parameter_request	68
Figura 4.7 Interfaz función authentication_parameters_request_ack	69
Figura 4.8 Interfaz función authentication_request	70
Figura 4.9 Interfaz función authentication_request_ack.....	70
Figura 4.10 Interfaz función map_cancel_location_request.....	71
Figura 4.11 Interfaz función map_cancel_location_request_ack	71
Figura 4.12 Interfaz función insert_suscriber_data_request	72
Figura 4.13 Interfaz móvil – Cambio de Localización	73
Figura 4.14 Interfaz de la ubicación del móvil ubicado en Riobamba	76
Figura 4.15 Interfaz de la ubicación del móvil ubicado en Guayaquil	77

INTRODUCCIÓN

Cada día que pasa el mundo sigue evolucionando tecnológicamente aprovechando los distintos recursos que nos ofrecen día a día ofreciendo soluciones y mayores ganancias en la producción de una empresa.

Los sistemas de señalización son un punto muy importante en lo que corresponde a la telefonía tanto fija como móvil, ya que permiten el tráfico de voz y datos, para ello fue necesario diseñar un método de intercambio de información entre ellos, por lo cual los organismo de estandarización comenzaron a desarrollar sus protocolos con respecto a la Señalización de Canal común (SS7) obteniendo una correcta transferencia de datos.

El rango de cobertura en una central telefónica es muy limitado, por lo cual se utilizan los sistemas de señalización siendo necesarios para el intercambio de información entre centrales, permitiendo al usuario permanecer conectado a una red sin interrupciones.

Con el fin de comprender el procesamiento de centrales al momento de cambiar de estación base se realizará una simulación de los mensajes que se envían entre las distintas partes de una red telefónica, se analizará el

protocolo Parte de Aplicación Móvil (MAP) correspondiente a la señalización SS7, utilizaremos el lenguaje abierto Preprocesador de Hipertexto (PHP) y el sistema operativo LINUX para realizar la simulación.

CAPÍTULO 1

1. DESCRIPCIÓN GENERAL

1.1. Antecedentes SS7

El SS7 es un protocolo que ha sido desarrollado por las principales compañías telefónicas y el Sector de Normalización de las Telecomunicaciones (UIT-T) desde 1975, el primer protocolo de señalización por canal común fue definido como sistema de señalización N ° 6 (SS6) en 1977. El Sistema de señalización N ° 7 se define como un estándar internacional por la UIT-T en sus recomendaciones de la serie Q.7XX de 1980.

SS7 fue diseñado para reemplazar SS6, que poseía una señal de 28 bits restringida que tenía algunas limitaciones en su diseño y era susceptible a los sistemas digitales. SS7 ha sustituido al SS6, SS5 (Sistema de

Señalización N° 5), con la excepción de R1 y R2 las cuales aún se utilizan en numerosos países.

El SS5 y normas anteriores, utilizaban señalización dentro de banda es decir un intercambio de información de control de llamadas dentro del mismo canal, la cual es usada durante una llamada telefónica, pero ocasionaba una gran cantidad de problemas de seguridad cuando los usuarios descubrían que se podían simular estos tonos en su terminal y controlar la red de los operadores sin alguna restricción.

SS6 y SS7 usan protocolos de señalización fuera de banda, el cual mantiene el canal de voz separado de los canales de señalización para evitar la posibilidad de que los tonos producidos sean introducidos por los usuarios.

1.2. Antecedes MAP (Parte de Aplicaciones Móviles)

MAP es un protocolo de SS7 que proporciona una capa de aplicación para los distintos nodos en GSM (Sistema Global para Comunicaciones Móviles) y UMTS (Sistema Universal de Telecomunicaciones Móviles) redes centrales móviles y redes de núcleo GPRS (Servicio General de Paquetes Vía Radio) para comunicarse unos con otros a fin de proporcionar servicios a los usuarios de teléfonos móviles.

MAP es el protocolo de capa de aplicación utilizado para acceder principalmente a bases de datos centralizadas como al HLR (Registro de Localización Local), VLR(Registro de Localización del Visitante), MSC

(Centro de Conmutación Móvil), EIR (Equipo de Registro de Identidad) y la red de línea fija.

1.3. Justificación de Protocolo MAP

El desarrollo de este tipo de sistemas de señalización con respecto a un protocolo en este caso MAP nos permite conocer el funcionamiento y operación del protocolo, así como también los aspectos técnicos que intervienen durante la operación de señalización, nos invita a conocer los mensajes que utiliza MAP a través de una red móvil para intercambiar información.

MAP interviene directamente con los elementos que constituyen una red GSM ya que accede a la información de un abonado, logrando así tener un registro de aspectos importantes como: su localización, central en donde se encuentra, su cobertura, servicios suplementarios

1.4. Descripción del Proyecto

El proyecto busca en si realizar algo didáctico para implementar una MSC que emule el comportamiento de la señalización SS7 / MAP, y así poder describir los principales mensajes de señalización del protocolo MAP en el momento en que se realice un cambio de MSC o se encienda por primera vez el celular para localizarlo y registrarlo en el HLR (que significa las siglas Registro de Localización Local) con toda la información necesaria.

El software libre es más común y se ha vuelto casi necesario en algunas empresas debido a la facilidad que nos brinda al momento de acceder a todos los servicios que nos brinda, así como también la optimización de recursos económicos, por lo cual para la simulación de esta aplicación SS7 utilizaremos LINUX y bajo el lenguaje abierto Preprocesador de Hipertexto (PHP).

1.4.1 Objetivo General

Este informe de materia integradora tiene como objetivo principal: analizar e implementar en lenguaje abierto Preprocesador de Hipertexto (PHP) sobre Linux de una Central de Conmutación Móvil (MSC) virtual para la interacción a nivel de protocolo de Parte de Aplicaciones Móviles (MAP) con el registrador de Ubicación de Abonado, HLR.

1.4.2 Objetivos Específicos

1. Describir el funcionamiento y operación del protocolo MAP.
2. Implementar una central de conmutación móvil virtual que emule el comportamiento a nivel de señalización MAP.
3. Describir los principales mensajes de señalización de protocolo MAP usado en las diferentes funciones de una MSC en una red móvil.

1.5. Alcance

El desarrollo de la investigación del Protocolo MAP (Parte de Aplicaciones Móviles) tiene como alcance, analizar y comprender todos los procesos realizados en las distintas capas de señalización de una red móvil pero centrada sobre todo en la parte aplicaciones y siendo más específicos, en los diferentes mensajes de interacción que se producen entre las centrales telefónicas, sus estaciones bases junto con sus respectiva base de datos de los visitantes en dicha celda de cobertura.

Se realizará para la demostración de todo el contenido investigado una interfaz amigable para el usuario para de esta manera lo pueda entender y comprender sin necesidad de tener conocimiento técnicos en esta área. Para lo cual se usara como lenguaje de programación, el lenguaje abierto PHP, que será mencionado posteriormente con más detalles. Realizando una simulación gráfica o virtual de todo el ambiente a nivel de Protocolo MAP.

1.6. Limitaciones

Debido a que el protocolo de aplicaciones móviles es un campo muy extenso por todos los diferentes servicios que se dan en la telefonía móvil, vale mencionar entre ellos los más conocidos: gestionar enrutamiento de llamadas, dar servicios de mensajes cortos e incluso servicios de datos por paquetes que se ven reflejados para todos aquellos servicios GPRS (Servicio general de paquetes vía radio) para celulares y otro servicios suplementarios.

Sin embargo nuestro informe de esta materia de graduación está orientado a un campo específico, lo centraremos en las actualizaciones que se realizan cada vez que un usuario móvil cambia de ubicación e ingresa a una nueva cobertura de una diferente estación base a la que se encontraba.

El análisis del protocolo MAP, estará limitada a la comunicación que envía la MSC hacia el HRL y viceversa. Los campos y mensajes que recibe con sus respectivas respuestas, e incluso las medidas que toma la parte inteligente de la red celular cuando se produce los fallos o caídas del sistema.

CAPÍTULO 2

2. FUNDAMENTOS TEÓRICOS

2.1. Señalización SS7 – Protocolo MAP

Niveles de la arquitectura de SS7: MAP

La red SS7 está diseñada para intercambiar mensajes en el área de las Telecomunicaciones facilitando este tipo de funciones y mantener la red que proporciona la misma.

Esta señalización fue creada para poder sustituir a versiones de señalizaciones anteriores como son: El sistema de señalización No.5 (SS5), sistema de señalización No.6 (SS6) y R2. SS7 se diferencia de las anteriores versiones porque posee para su seguridad un canal de señalización separado de los canales de datos del usuario.

La señalización SS7 está compuesta por cuatro capas, las cuales se observan a continuación: [1]

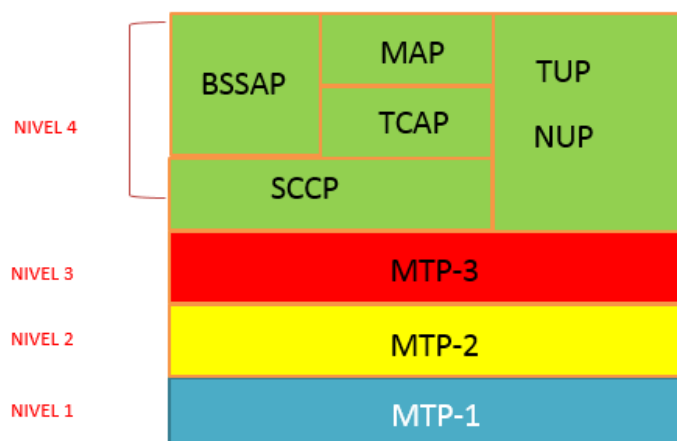


Figura 2.1 Esquema SS7

Capa Física Nivel 1

Esta capa define las características físicas de los enlaces de señalización de la red SS7. Posee canales de transmisión digital que trabajan a 64 kbits/s y que transportan los datos SS7 entre dos puntos de la red.

Parte de Transferencia de Mensajes Nivel 2

MTP2 ofrece una transferencia fiable de los mensajes de señalización, así como también una correcta funcionalidad de la capa de enlace para lo cual incorpora funciones útiles en su capa como lo son el control de flujo, comprobación de errores. El protocolo MTP2 es específico para enlaces de banda estrecha (56 o 64 kbps).

Parte de Transferencia de Mensajes Nivel 3

MTP3 se encarga de la parte de transmisión de mensajes de un nodo a otro ya que el encamina sus mensajes salientes a su destino del usuario previamente definido, es decir este protocolo tiene capacidades de enrutamiento, control de la congestión, nodo de direccionamiento y enrutamiento alternativo.

Conexión de señalización Parte de Control (SCCP)

SCCP tiene la capacidad para hacer frente a aplicaciones dentro de un punto de señalización. El MTP sólo puede recibir y enviar mensajes desde un nodo en su conjunto, que no se ocupa de las aplicaciones de software dentro de un nodo. Proporciona medios flexibles de enrutamiento y proporciona mecanismos para transferir datos a través de la red SS7.

SCCP transfiere consultas y respuestas entre el VLR y HLR en sus bases de datos. Dichas transferencias se llevan a cabo por un número de razones. La razón principal es actualizar HLR del abonado con el área de servicio actual VLR para que las llamadas entrantes puedan ser entregadas.

Parte usuario de la RDSI (TUP ISUP NUP)

ISUP define sus mensajes y el protocolo que va a utilizar en la conexión y desconexión de llamadas de voz y datos a través de la red pública, y para gestionar su red troncal. ISUP también tiene soporte inherente para servicios

suplementarios, tales como llamada automática, llamando a la línea identificación, y así sucesivamente.

Capacidades de transacción Parte de Aplicación (TCAP)

TCAP define los mensajes y el protocolo utilizado para la comunicación entre aplicaciones en los nodos. Brinda el soporte necesario para el protocolo MAP el mismo transporte los mensajes servicio que hoy en día ya no es muy utilizado debido a la existencia del servicio de datos y redes sociales. Dado que los mensajes TCAP deben ser entregados a las aplicaciones individuales dentro de los nodos que abordan, que utilizan la SCCP para el transporte.

Subsistema de Estaciones Base Parte de Aplicación (BSSAP)

Esta capa se la utiliza en conjunto con la red GSM la cual interviene en el proceso de autenticación y asignación de TMSI (Identificación del Abonado Temporal) entre la BSC (Subsistema de Estación Base) y MSC.

La parte de aplicación móvil (MAP)

El protocolo MAP es una extensión de la señalización SS7, proporciona una capa de aplicación para los distintos nodos en las redes celulares. MAP es transportado y encapsulado con los protocolos MTP, SCCP y TCAP. TCAP sobre los protocolos SCCP y MTP los cuales se encargan del transporte confiable y la verificación de errores [2]

MAP está asociado directamente con el protocolo TCAP para transferir toda su información en tiempo real, además dirige las entregas en una base, conecta comandos y respuestas dentro de una comunicación.

Tiene una capa de aplicación sobre la cual se construye los servicios que intervienen en una red GSM tiene dos versiones las cuales son incompatibles entre sí las cuales son: GSM-MAP y ANSI-41 MAP, la diferencia entre ellas radica en que la GSM-MAP soporta únicamente GSM mientras que ANSI-41 MAP soporta AMPS(Sistema de teléfono Móvil Avanzado), GSM, NAMPS(Servicio de teléfono móvil avanzado),D-AMPS/TDMA,CDMA (Multiplexación por división de código), así como también GSM es la versión internacional y ANSI-41 es la versión que usa en Norteamérica.

El protocolo define operaciones entre los componentes de una red GSM como lo son el MSC, HLR, VLR para permitir la actualización de la ubicación del abonado, su autenticación, el enrutamiento de llamadas y mensajes hacia y desde los móviles así como también transfiere la información acerca de los abonados.

2.2. Red Celular GSM (Global System for Mobile)

Una red celular consiste en un esquema de equipos de telecomunicaciones que cubren determinadas áreas incluyendo un conjunto de celdas para dar cobertura al mayor espacio físico posible. Constituidas por radio bases que poseen un transceptor haciendo a su vez las funciones de emisor y receptor para las comunicaciones a largas distancias.

Las celdas en las topologías de redes celulares son agrupadas de forma circular y hexagonal, para poder lograr la reutilización de frecuencias y disminuir las interferencias de cocanal y lograr aumentar la capacidad en troncalización.

Una topología de una red celular GSM como se observa en la Figura 2.2 consta de muchas componentes, pero podemos mencionar entre las más operativas a la MSC, BS, HLR, VLR, MS, AuC, EIR. [3]

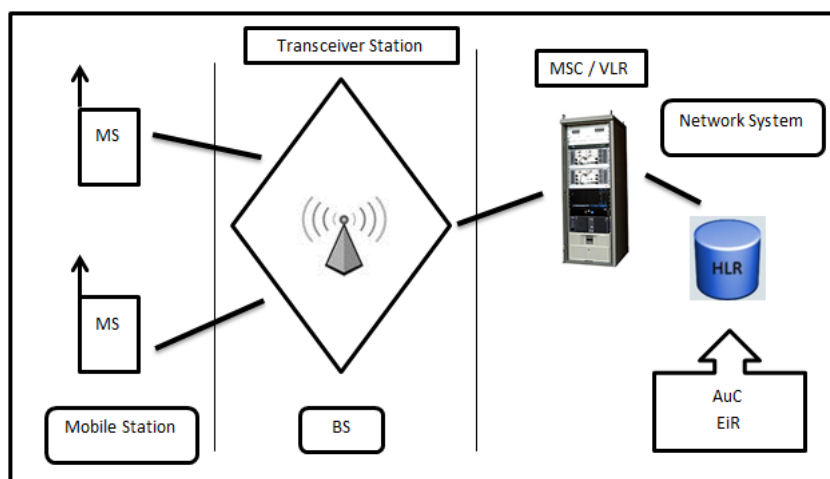


Figura 2.2 Red Celular GSM

2.3. Topología Celular – Operaciones de Protocolo MAP

La red celular tiene como prioridad cubrir el mayor número de área para ofrecer una excelente cobertura a todos sus usuarios móviles. Para que todo este esquema genere las mejores condiciones de comunicación entre móviles mencionaremos las partes que hacen posible este proceso de transmisión y recepción, pero centrándonos en los conocimientos en el escenario de nuestro protocolo de estudio llamado MAP, que corresponde a las aplicaciones móviles [4]

Realizaremos una breve descripción de cada una de ellas como se observa en la Figura 2.3:

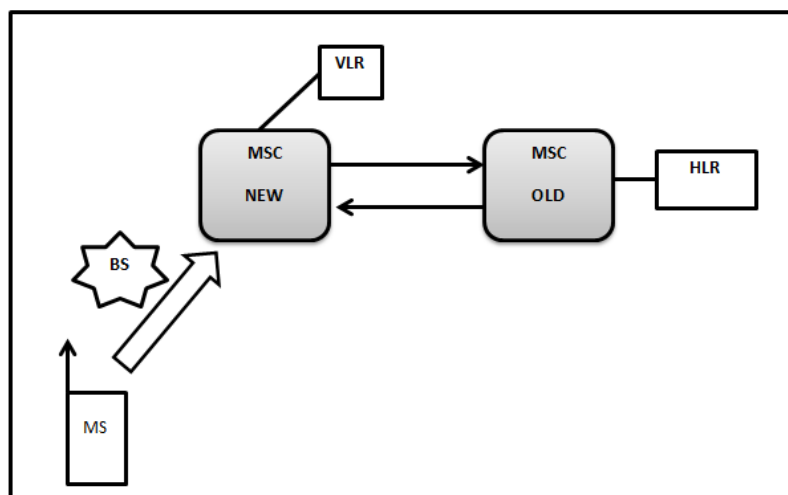


Figura 2.3 Esquema de red celular procesando protocolo MAP

VLR (Registro de Localización del Visitante).- Es el registro local de visitantes, forma parte de una red móvil que contiene una base de datos que almacena los datos de los móviles disponibles dentro de un área de cobertura de su respectiva MSC. La VLR generalmente se encarga de disminuir el número de consultas que haga la MSC a la HLR sobre información del nuevo móvil al momento de ingresar a su área de cobertura. La VLR contiene el IMSI, el TMSI, el área de localización donde el móvil se ha registrado, ciertos servicios complementarios.

BS (Estación Base).- Formadas por los receptores y transmisores que se encuentran en una celda, permiten monitorear la conexión por las diversas antenas instaladas en las torres operativas, se menciona que pueden existir más de una estación base por área pero no en una misma celda.

HLR (Registro de Localización Local).- Es considerada como la principal base de datos que reúne toda la información de un usuario móvil. La principal información que almacenan el Registro de Localización Local es la ubicación de cada móvil, para de esta forma poder enrutar las respectivas llamadas realizadas por cada una de ellas. En la base de datos de dicha HLR se almacena el IMSI de cada suscriptor, el número de VLR conectada, las restricciones que posee el móvil referentes a sus servicios contratados [5]

Parámetros necesarios para la señalización en una red GSM

IMSI

Cada suscriptor es asignado con un único número que es conocido como IMSI. IMSI es la identidad absoluta de un suscriptor que se encuentra en una red GSM, y es almacenado en la tarjeta SIM.

El IMSI es especificado en GSM 03.03, por 3GPP en TS 23.003, y el UIT en E.212. El formato del parámetro IMSI se lo puede observar en la siguiente figura.

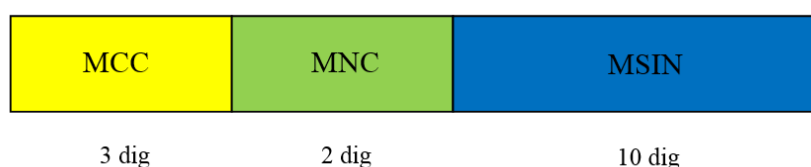


Figura 2.4 Formato IMSI

IMSI=MCC+MNC+MSIN: 15 dígitos decimales

MCC: 3 dígitos decimales (Código del país).

MNC: 2 dígitos decimales (Código de área).

MSIN: 10 dígitos decimales (Número de identificación del abonado asignado por el operador- Serie del Chip).

TMSI

El TMSI es un alias que utiliza el VLR que se encarga de proteger la confidencialidad del suscriptor. Es un dato temporal que se utiliza para

sustituir el IMSI, el número de veces que el IMSI hace su difusión para el aire debido a que intrusos podrían usar este IMSI para ingresar a una red GSM.

TMSI es usado habitualmente en el proceso de Actualización de ubicación, y es asignada a cada una de las MS. La trama se puede observar a continuación [6]

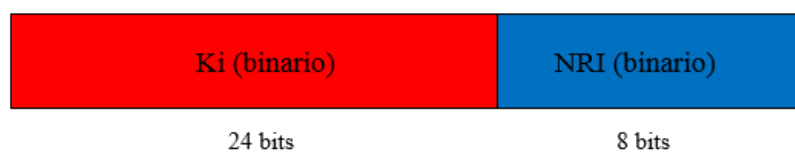


Figura 2.5 Formato TMSI

$TMSI = KI + NRI$

KI: 24 bits binarios (Llave de Autenticación Individual del Subscriptor).

NRI: 8 bits binarios (Identificación de la MSC en donde se encuentra).

LAI

El LAI es utilizado en redes móviles GSM. Cada área de ubicación de una red móvil tiene su propio identificador único LAI. Este identificador único internacionalmente se utiliza para la actualización de la posición de los abonados.

La estación móvil reconoce el LAI y lo almacena en la SIM. Si existe un cambio de zona de localización da lugar a la solicitud de actualización de la ubicación. El formato de la trama es el siguiente.

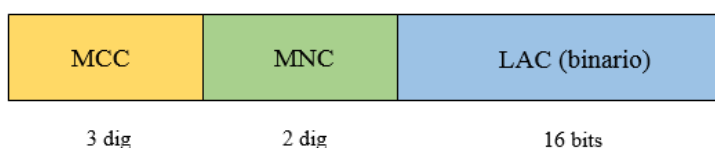


Figura 2.6 Formato LAI

LAI=MCC+MNC+LAC: 10 dígitos decimales

MCC: 3 dígitos decimales (Código del país).

MNC: 2 dígitos decimales (Código de área).

LAC: 16 bits binarios o (LAC<FFFF o LAC<65536) (Código de área local)

(LAC Formato: HEX).

TRIPLETAS DE AUTENTIFICACIÓN

Cuando se solicita algún tipo de servicio se realizan algunas tareas de seguridad para proteger la información que se transmite. Para llevar este proceso a cabo se define la siguiente tripleta [7]

- Un número aleatorio (RAND).
- La respuesta firmada (SR).
- La clave de cifrado (Kc).

Esta tripleta se corresponde con cada abonado al servicio GSM y es almacenada en el HLR. La tripleta se genera en el AuC (base de datos del HLR), según el proceso que se muestra en la Figura 2.7

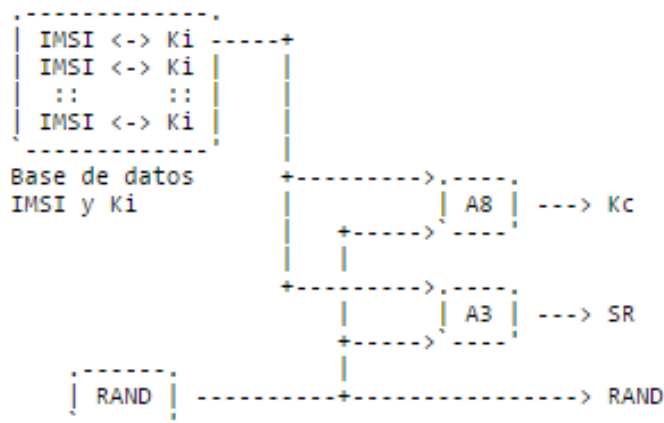


Figura 2.7 Proceso de Autenticación

El IMSI almacenado en la base de datos del AuC se mira la Ki, o lo que es lo mismo, la clave secreta del usuario, que es la misma que se almacena en la tarjeta SIM. Esta clave (Ki) es usada como entrada, junto con un número aleatorio (RAND), tanto para generar la clave de cifrado (Kc) como la respuesta firmada (SR).

Para obtener Kc, se usa el algoritmo denominado A8. SR se obtiene con los mismos datos y un algoritmo al que se le denomina A3.

2.4. Centro de Conmutación Móvil (MSC)

El Centro de Conmutación de Servicios Móviles se encarga de realizar todas las conmutaciones posibles de las estaciones móviles comprendidas en cada MSC correspondiente debido a su situación geográfica y alcance de cobertura. Es considerada como el cerebro de una red móvil, asocia y sincroniza las llamadas, su liberación y en el enrutamiento de las mismas. Debido a la evolución de la tecnología y los nuevos equipos ingresando al campo de las telecomunicaciones trayendo equipos más sofisticados, con mejoras en sus funciones e incluso nuevos servicios, las MSC en la actualidad también se encargan del enrutamientos de mensajes (SMS), llamadas en conferencias, fax, ciertos servicios que usan facturación e interconectar incluso dicha MSC con otras redes como se presenta en un PSTN. Todas estas comunicaciones deben pasar por una MSC o en ciertos servicios por algunas de ellas por dicha funcionalidad es considerada el mecanismo central para la comunicación de los usuarios.

La MSC interviene en procesos de autenticación de usuarios móviles al momento de su activación, al momento de realizar una actualización de su nueva conexión a una diferente MSC y mencionamos también la intervención en el proceso de HAND-OVER (traspaso de conexión de un móvil a otra estación base por su insuficiencia de enlace o cobertura).



Figura 2.8 Centro de Conmutación Móvil

2.5 Lenguaje de Programación PHP (Preprocesador de Hipertexto)

PHP es un lenguaje de programación de código abierto, en la actualidad es muy usado tanto en persona que empiezan en el campo de la programación y programadores ya profesionales que desean darles más detalles a sus trabajos. El lenguaje PHP tiene una particularidad especial: se basa en una programación desde el servidor, significa que su código se ejecuta en el servidor y puede ser incrustado en HTML. Al momento de recibir el cliente el ejecutable del código en páginas web, podemos hasta observar la aplicación en lenguaje HTML, mas no en el propio PHP que originalmente fue creado dicho código añadiendo seguridad a nuestra implementación; definiéndose este sistema como un código embebido.

El beneficio de este lenguaje desde el servidor consiste en que el usuario pueda observar incluso varias páginas HTML integrándolas a sus propios

procesos, más no como procesos separados. Puede ser usado en cualquier sistema operativo sin necesidad de realizar algún cambio en código [8]

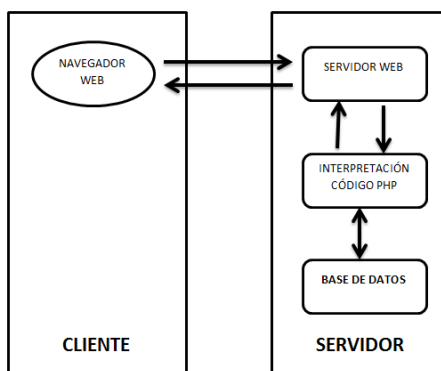


Figura 2.9 Proceso de ejecución de código PHP

Como se muestra en la imagen, el proceso empieza en el navegador usado por el usuario, haciendo una petición de página al servidor de internet, en el servidor se procesan las páginas en PHP dándonos como resultado de una compilación de lenguaje un código HTML, que podrá ser visible de la misma manera en nuestro navegador, una pequeña desventaja de esta presentación apunta hacia las aplicaciones realizadas, debido a que ciertos browser no admiten dichas configuraciones o presentaciones requiriendo instalaciones de complementos externos.

Este lenguaje es muy similar a el lenguaje C o C++, donde se pueden usar librerías e incluso agregarle extensiones como gráficas, XML, pdf y las demás utilizadas en el ambiente de programación. Esto quiere decir que

personas con previos conocimientos de lenguaje C podrán desenvolverse y aprender muy rápido a utilizar PHP. Se puede ejecutar en servidores web como Apache, Roxen, entre otros.

Como se describió anteriormente es un lenguaje OPEN SOURCE que no requiere comprar ningún tipo de licencia, ni realizar aportes económicos a ninguna compañía comercial por usarlo. PHP nos ayuda a interactuar con reconocidas bases de datos, cabe mencionar entre ellas las más utilizadas y nombradas como MySQL, Oracle, MS SQL, etc.

Requerimientos para implementación de nuestro proyecto para usar PHP

En nuestro proyecto usaremos el servidor Web Apache, el software de programación PHP y la base de datos MySQL. Se deben sincronizar y configurar las tres aplicaciones para que trabajen conjuntamente, para usuarios principiantes es recomendable usar una aplicación extra llamada EasyPHP. Esta aplicación realiza automáticamente la instalación conjunta evitando problemas de configuración.

2.6 Sistema Operativo Linux

Ubuntu es un sistema operativo que tiene como base Linux y que se distribuye como software libre. Fue creado con el objetivo de ayudar al usuario ya que provee algunas herramientas de fácil uso lo que ayudara a adquirir una mejor experiencia en el mismo. Tiene como característica

principal una variedad de software que a contrario de otros sistemas operativos son distribuidos bajo una licencia libre o de código abierto.

Ubuntu soporta dos diseños de hardware en computadoras personales y servidores: 32-bit y 64-bit. Es capaz de actualizar todas las aplicaciones instaladas en la máquina a través de repositorios. El sistema operativo se ha traducido a más de 130 idiomas diferentes y cada uno de sus usuarios puede ayudar a su mejora a través de Internet. Posee una gran variedad de aplicaciones para entretenimiento, desarrollo y aplicaciones que pueden ser útiles para nuestro diario vivir, así como también nos permita configurar de manera rápida el sistema.

El sistema operativo de Linux también se puede instalar en máquinas virtuales, pero en nuestro proyecto no se recomienda el uso de máquinas virtuales debido a las tarjetas de red que necesitamos en la topología, ya que en la misma computadora asignamos una dirección IP a la red conectada por puerto Ethernet y otra dirección IP al conectarse a la red inalámbrica.

La interfaz que usa Ubuntu se compone de tres elementos: una pantalla al inicio que despliega acceso a aplicaciones y medios que son los más importantes, una barra en la parte superior para indicar los sistemas y menús disponibles, un lanzador de aplicaciones en la parte izquierda. La interfaz que viene por defecto es Unity (ver Figura 2.10) que utiliza aplicaciones de GNOME [9]

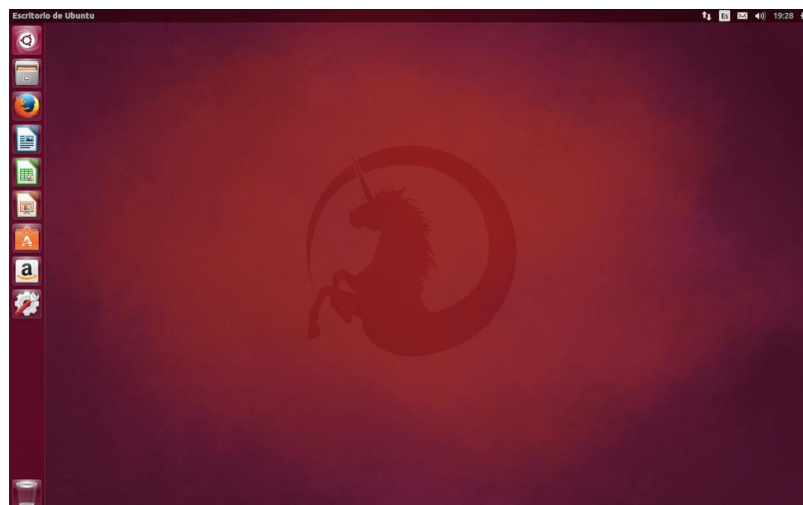


Figura 2.10 Interfaz Unity (Ubuntu 14.04)

CAPÍTULO 3

3. DESARROLLO DEL PROYECTO

3.1 Proceso de Ubicación del Abonado Móvil

Una red GSM tiene un mecanismo capaz de determinar la ubicación de una MS ya sea cuando este en reposo o en constante movimiento para lo cual el celular realiza peticiones constantemente de la actualización de su ubicación que puede ocurrir cuando:

- La MS se registra por primera vez en la red celular: Puede ocurrir cuando es un nuevo usuario o se encuentra apagado el celular y lo vuelve a iniciar entonces inicia un proceso de autenticación LOCATION UPDATE REQUEST que lleva como parámetros el TMSI y

el LAI los cuales mediante el proceso descrito en la Figura 3.1 logra el registro de su ubicación.

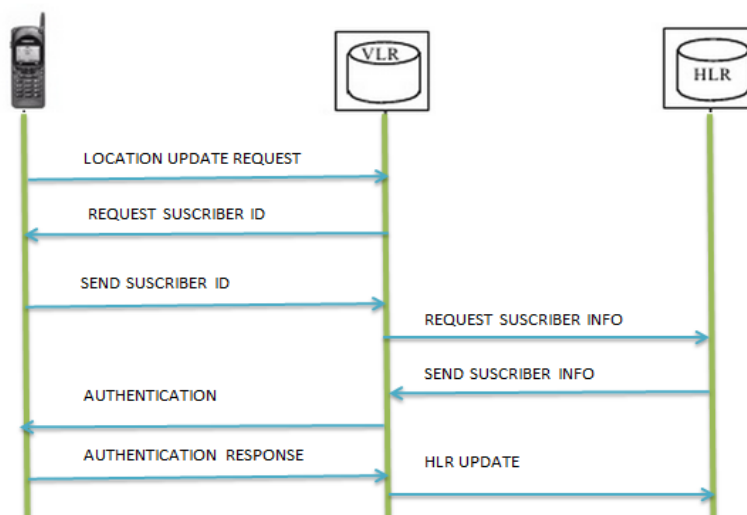


Figura 3.1. Registro de ubicación por primera vez.

- La MS se mueve dentro de la misma área (misma MSC): Esto ocurre cuando el abonado ha estado en movimiento pero dentro de una misma MSC al momento que ocurre la actualización este informa que no ha ocurrido algún cambio en el registro de su ubicación como lo podemos apreciar en la Figura 3.2

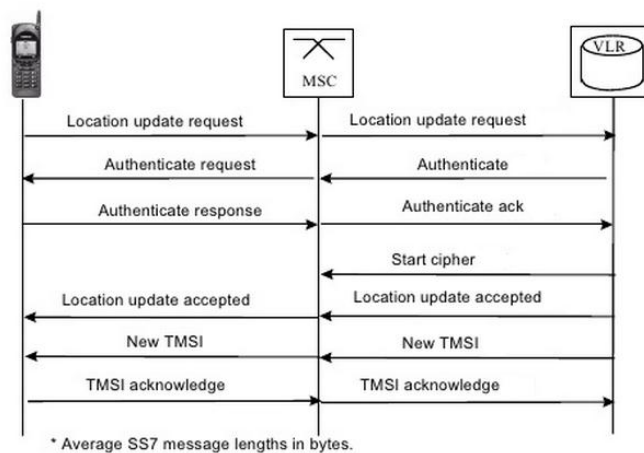


Figura 3.2 Registro de ubicación dentro de la misma MSC

- La MS se mueve a una nueva área (cambio de MSC): Esto ocurre cuando el usuario ha entrado en una nueva área por lo cual el MS identifica que se ha cambiado a un nuevo MSC por lo cual al momento de la actualización realiza el proceso descrito en la Figura 3.3 para asignarle un nuevo TMSI con el cual se encuentra registrado en la nueva MSC.

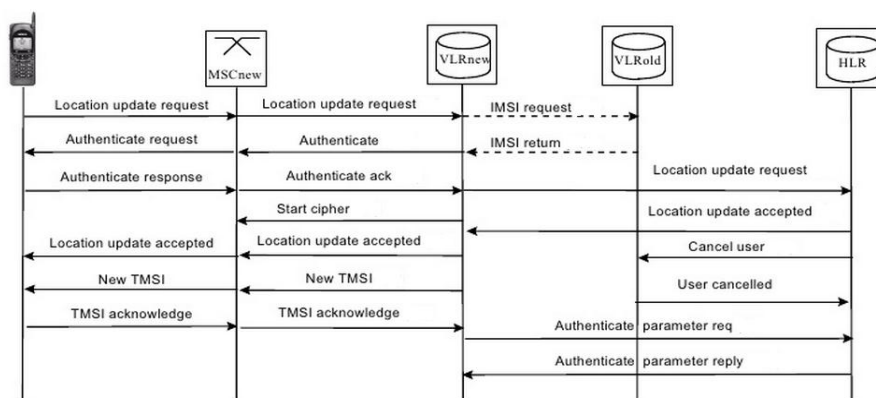


Figura 3.3 Registro de ubicación cambio de MSC

- La MS es actualizada cada cierto tiempo: Se lo realiza para hallar la ubicación del abonado automáticamente para no perder la información acerca de su ubicación por lo cual el proceso descrito en la Figura 3.4 se lo realiza durante intervalos de tiempo. [10]

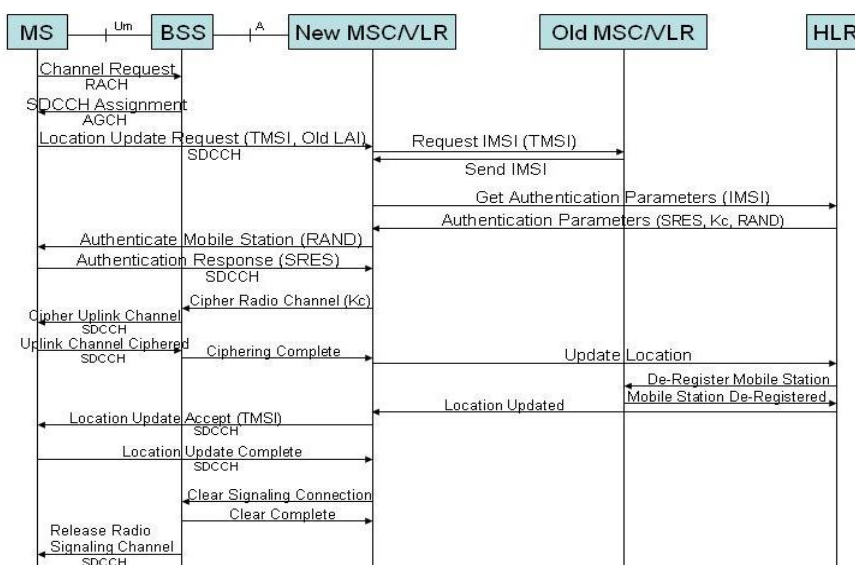


Figura 3.4 Registro de ubicación cada cierto tiempo

3.2 Elaboración del Proyecto

3.2.1 Software

Para la ejecución total del proyecto hemos usado distinto software que serán mencionados a continuación, pero donde nuestro objetivo principal fue trabajar en un sistema operativo de código abierto, por lo cual usamos Ubuntu. Netbeans para la programación y también XAMPP para la base de datos usada para la información de nuestros abonados móviles.

Detallaremos la versión que usamos de cada software y una breve descripción de los requerimientos mínimos que deben poseer las computadoras para poder ser instalados correctamente y no presenten alguna incompatibilidad al momento de ser usados.

INSTALACIÓN DE UBUNTU

Para la programación de todo el proyecto se instaló un sistema operativo de Linux, específicamente Ubuntu 14.02. LTS. Este software es de versión libre, se lo puede instalar sin necesidad de tener gastos económicos en la compra de licencia para su uso. Cabe destacar que su manejo es mucho más complicado, debido que la mayoría o casi todas las instalaciones se las debe realizar por comando en el terminal del sistema operativo. Las especificaciones mínimas para poder instalar este sistema operativo en una pc de escritorio o laptop son:

- Procesador x86 a 700 MHz
- Memoria RAM de 512 Mb
- Espacio en Disco Duro como mínimo de 5 GB (swap incluida)
- Tarjeta gráfica y monitor capaz de soportar una resolución de 1024 x 768
- Lector de DVD o puerto USB
- Conexión a internet

Instalación:

- Descargar la imagen iso de 32 o 64 bits de acuerdo a la compatibilidad de nuestra computadora, en la siguiente dirección web [11]
- Luego de ejecutar el archivo descargado escogemos nuestro idioma y damos clic en instalar Ubuntu.



Figura 3.5 Instalación Ubuntu

- Luego Ubuntu mismo se encargará de comprobar todos los requerimientos necesarios para su instalación. Procedemos a dar clic en Continuar. El siguiente paso es muy importante, ya que decidirás si quieres instalar Ubuntu junto a tu sistema operativo actual o si deseas borrar tu sistema operativo y dejar en tu computadora instalado solo Ubuntu.
- El proceso de instalación realizará las particiones necesarias del disco duro para continuar con la instalación. Se realizarán como mínimo 4

particiones en el disco duro.

- Escogemos instalar ahora, luego nos permitirá escoger el sistema horario, dejaremos que se proceda a instalar el sistema operativa y finalmente nos pedirá ingresar nombre de usuario, contraseñas de usuarios y darle un nombre al equipo.

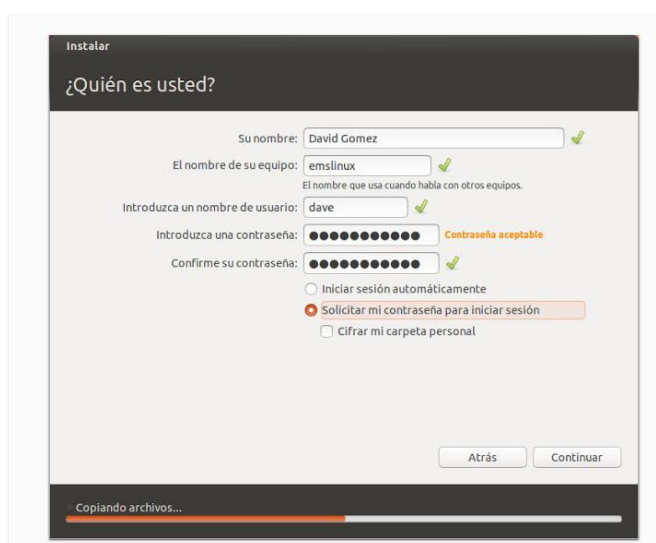


Figura 3.6 Ingreso de nombre de equipo en Ubuntu

INSTALACIÓN DE XAMPP

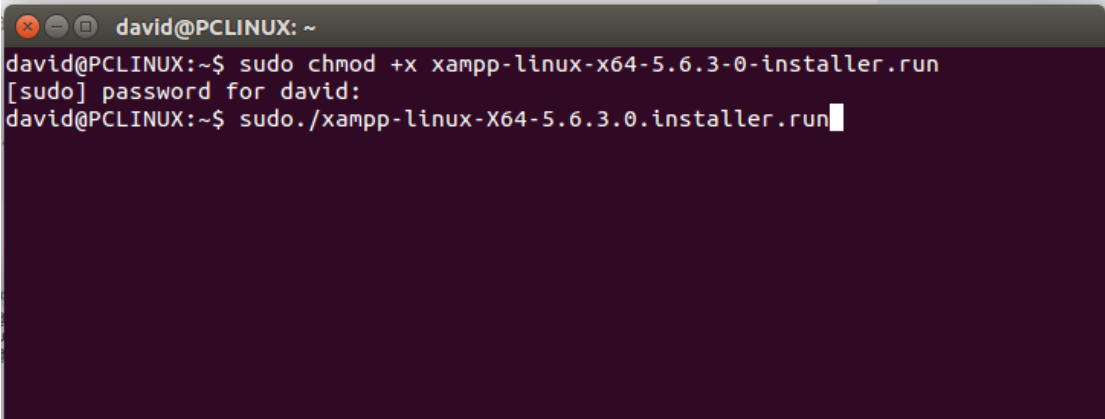
Por el uso de base de datos en nuestro proyecto escogimos esta herramienta que es una distribución de apache que integra MYSQL, PHP Y PERL. Esta herramienta en Linux cada vez que la queramos usar debemos levantarla antes de usarla por medio de comando en el terminal de Ubuntu [12]

Instalación:

- Una vez obtenido el archivo de instalación que se puede obtener

gratuitamente de la página oficial de Xampp utilizamos el siguiente comando en el terminal para iniciar la instalación.

```
sudo chmod +x xampp-linux-x64-5.6.3.0-installer.run
```

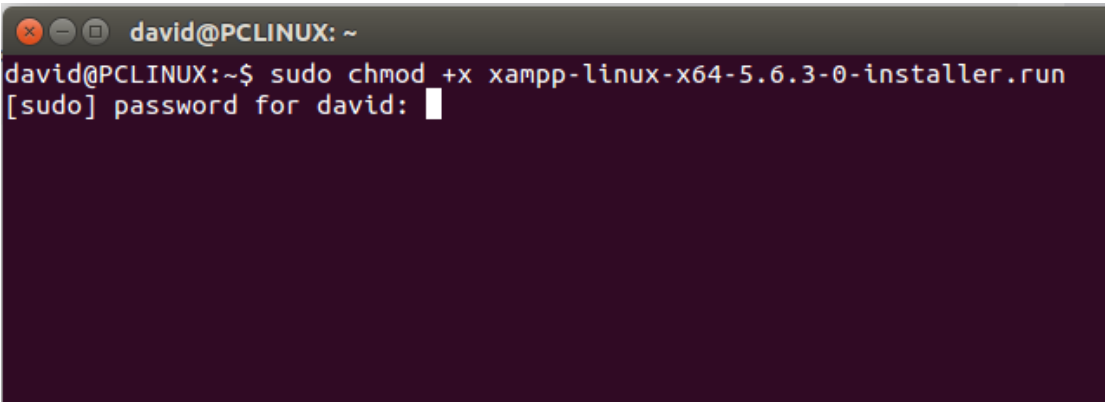


```
david@PCLINUX: ~  
david@PCLINUX:~$ sudo chmod +x xampp-linux-x64-5.6.3.0-installer.run  
[sudo] password for david:  
david@PCLINUX:~$ sudo ./xampp-linux-x64-5.6.3.0.installer.run
```

Figura 3.7 Comando para iniciar instalacion XAMPP en ubuntu

- Ejecutar el archivo de instalación de Xampp.

```
sudo chmod +x xampp-linux-x64-5.6.3.0-installer.run
```



```
david@PCLINUX: ~  
david@PCLINUX:~$ sudo chmod +x xampp-linux-x64-5.6.3.0-installer.run  
[sudo] password for david:
```

Figura 3.8 Comando para ejecutar el archivo de instalación XAMPP

- Al momento de ejecutar el comando nos aparecerá una pantalla de inicio de instalación y presionamos en Next.

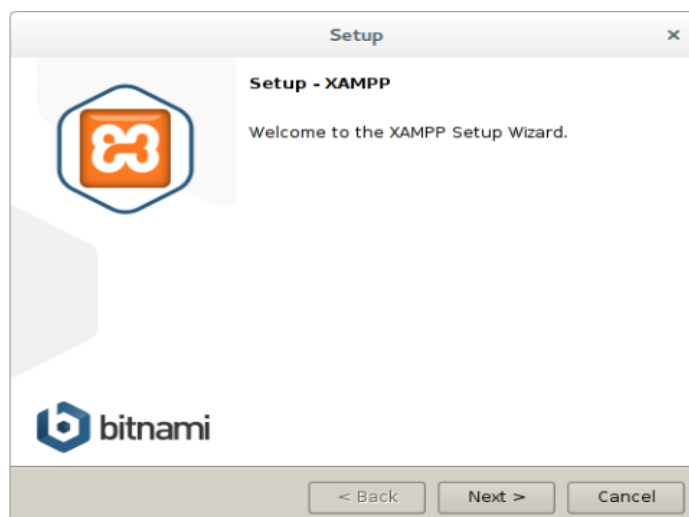


Figura 3.9 Proceso de instalación XAMPP

- Presionamos Next en las siguientes ventanas hasta que inicie la instalación del programa.



Figura 3.10 Instalación de XAMP junto MySQL, PHP y PERL

- Una vez finalizada la instalación se abrirá una página en el navegador si no se abre automáticamente debe dirigirse a la siguiente dirección en el navegador <http://localhost/xampp>.



Figura 3.11 Designación de URL para ingreso de XAMPP

- Simultáneo con la anterior ventana se abrirá una nueva ventana en la cual debemos verificar en la pestaña Manage Servers que todos los servicios de Xampp estén habilitados, caso contrario deberíamos activarlos con el botón Start.

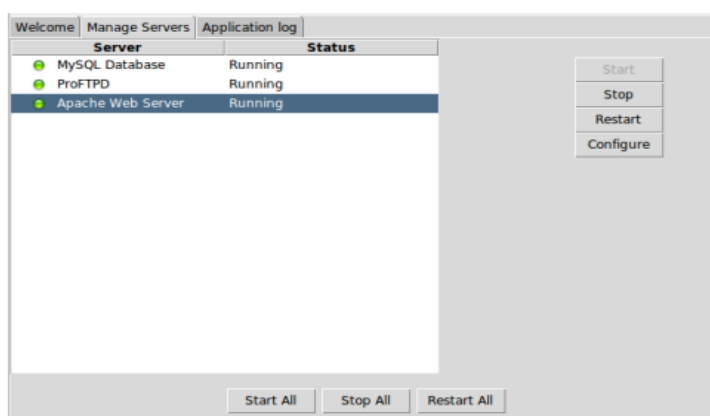


Figura 3.12 Activación de las aplicaciones contenidas en XAMPP

INSTALACIÓN DE NETBEANS 8.02

Programa que se usa para la programación de la señalización de MAP en lenguaje de hipertexto (PHP), es un ID fácil de instalar, maneja y ejecuta lenguajes de programación tales como HTML, PHP, C++ y sobre todo aplicaciones WEB con JAVA. Es un software libre, sin restricción de uso que puede ser instalado en sistemas operativos de Linux, Microsoft y otros. Permite una programación por módulos mucho más esquematizadas que cualquier otro software de programación, su código será mucho más fácil de leer y entender.

Instalación:

- Descargar el archivo de instalación de Netbeans para Ubuntu desde la página web oficial, estaremos muy atento en la carpeta donde se guardará [13]
- Se realiza la instalación ingresando comandos en la terminal de Ubuntu.
- Como en nuestro caso la imagen del instalador la guardamos en descargas colocamos en el terminal: **\$cd Descargas.**
- Le damos permiso para su ejecución: **\$sudo chmod +x netbeans-7.4-php-linux.sh**
- Colocaremos la contraseña, realizamos la instalación del sistema operativo: **sudo sh netbeans-7.4-php-linux.sh**

- Esperamos que se instale completamente.

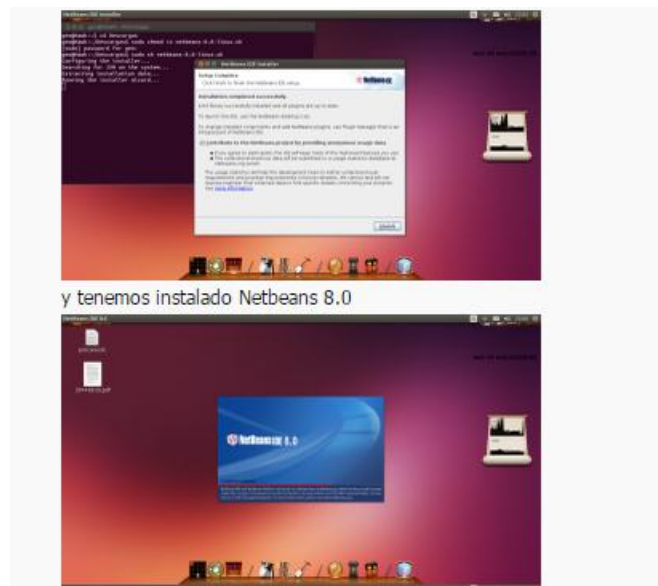


Figura 3.13 Instalación de Netbeans en ubuntu

3.2.2 Hardware

Para nuestra simulación física del protocolo MAP, hicimos en lo posible de utilizar equipos que den más realismo a nuestro proyecto de lo que sucede en las operadoras móviles con equipos muchos más robustos y caros por supuesto. Los equipos a continuación son los que realmente se usaron:

Laptop Hp pavilion dv4 → Simula la MSC 1 localizada en el Área de Guayaquil

- Procesador Intel Core 2 Duo T6600 de 2,2 GHz
- 4 Gb de Ram
- Disco Duro de 500 Gb

- Tarjeta de gráficos ATI Radeon HD 4530
- Wifi 802.11 b/g



Figura 3.14. Laptop para Simulación de MSC 1

Laptop Hp Pavilion g4-1284la → Simula la MSC 2 Localizada en el área de Quito

- Procesador AMD A10 Elite Quad-CORE
- Disco Duro de 750 Gb
- Ram de 8 Gb
- Puerto Ethernet
- Wifi 802.11 b/g



Figura 3.15 Laptop para simulación de MSC 2

Laptop Dell Inspiron 14-3421 Intel Core i5 → Simula la HLR (Registro de Localización Local)

- Procesador Core i5
- 4 Gb de Ram
- 750 Gb de Disco Duro
- Puerto Ethernet
- Wifi 802.11 g/b/n
- 3 puertos usb 2.0



Figura 3.16 Laptop para simulación de HLR

Laptop Gateway N522 Series → Simulo la Estación Móvil (Celular)

- AMD Dual-Core Procesor E1-2500
- 750 Gb de Disco Duro
- 6 Gb de Ram

- 1 puerto Ethernet
- Wifi 802.11 g/b/n
- 4 puertos Usb



Figura 3.17 Laptop para simulación de MS

Roter Belkin Modelo F9K1117V2 → Simula la Cobertura inalámbrica correspondiente a el área de Guayaquil

- Dual Banda Wirelles
- Opera entre 2.5 y 5 Ghz
- 4 puertos Ethernet
- Simple instalación
- Rangos de IP



Figura 3.18 Router para red inalámbrica - cobertura Gye

Router TPLink TL-WR741ND → Simula la Cobertura inalámbricas correspondiente a el área de Quito

- Facil Instalación
- 4 puertos Ethernet
- 1 puerto para la conexión POE o WAN



Figura 3.19 Router para red inalámbrica - cobertura Quito

Switch 3 com → Simula y ayuda en la interconexión de las diferentes áreas de cobertura de Quito y Guaayaqui

- 26 puertos Ethernet
- 2 Puertos Gigabit
- Fácil Instalación
- Puerto Consola



Figura 3.20 Switch para conexión de MSC1 y MSC2

3.3 Proceso de implementación

3.3.1 Arquitectura de la Red

La conexión de la red se la hizo usando una configuración estrella que consistía en usar un *switch* como punto central, donde saldrán las dos MSC (Guayaquil y Quito).

Como se observa en la Figura 3.21 se realizó una simulación de nuestra red en un software usado comúnmente para redes llamado PACKET TRACER y que es propietario de CISCO.

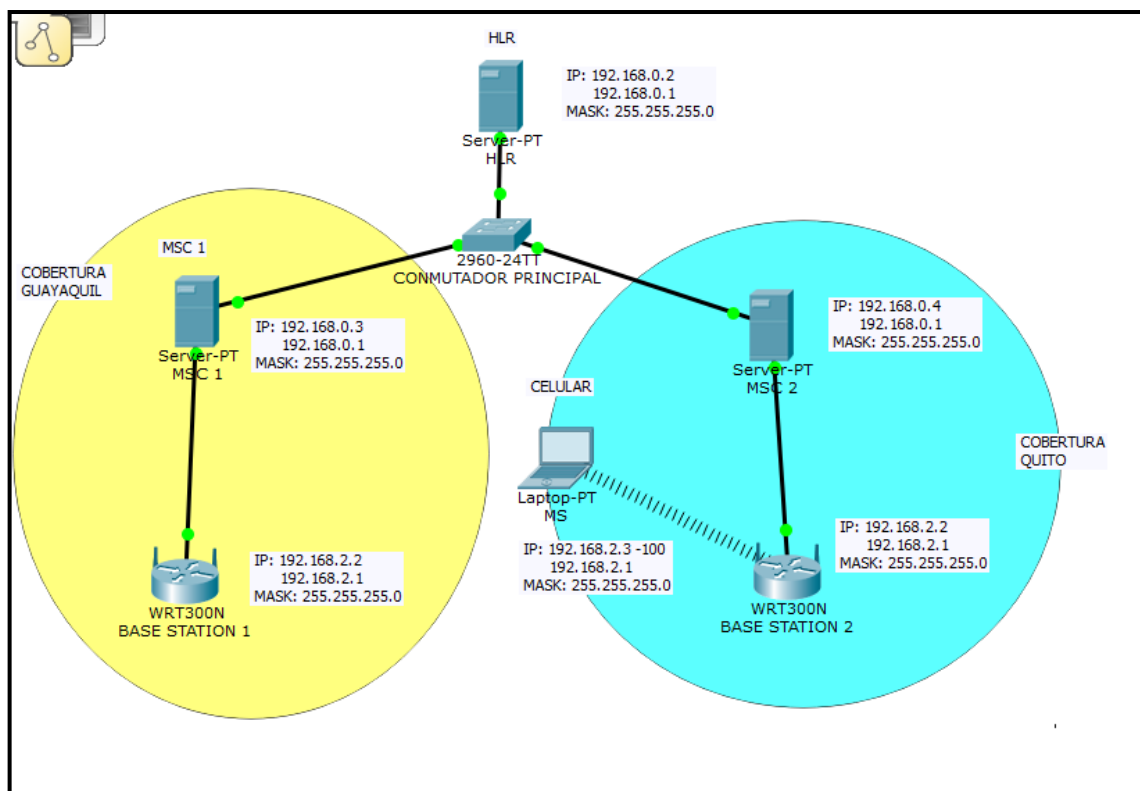


Figura 3.21 Topología de la red usada en MAP

3.3.2 Señalización

El proceso a ser analizado es cuando un usuario hace el cambio de MSC es decir su ubicación ha cambiado y se realizara la petición de localización para lo cual nos basaremos en el siguiente diagrama de la Figura 3.22

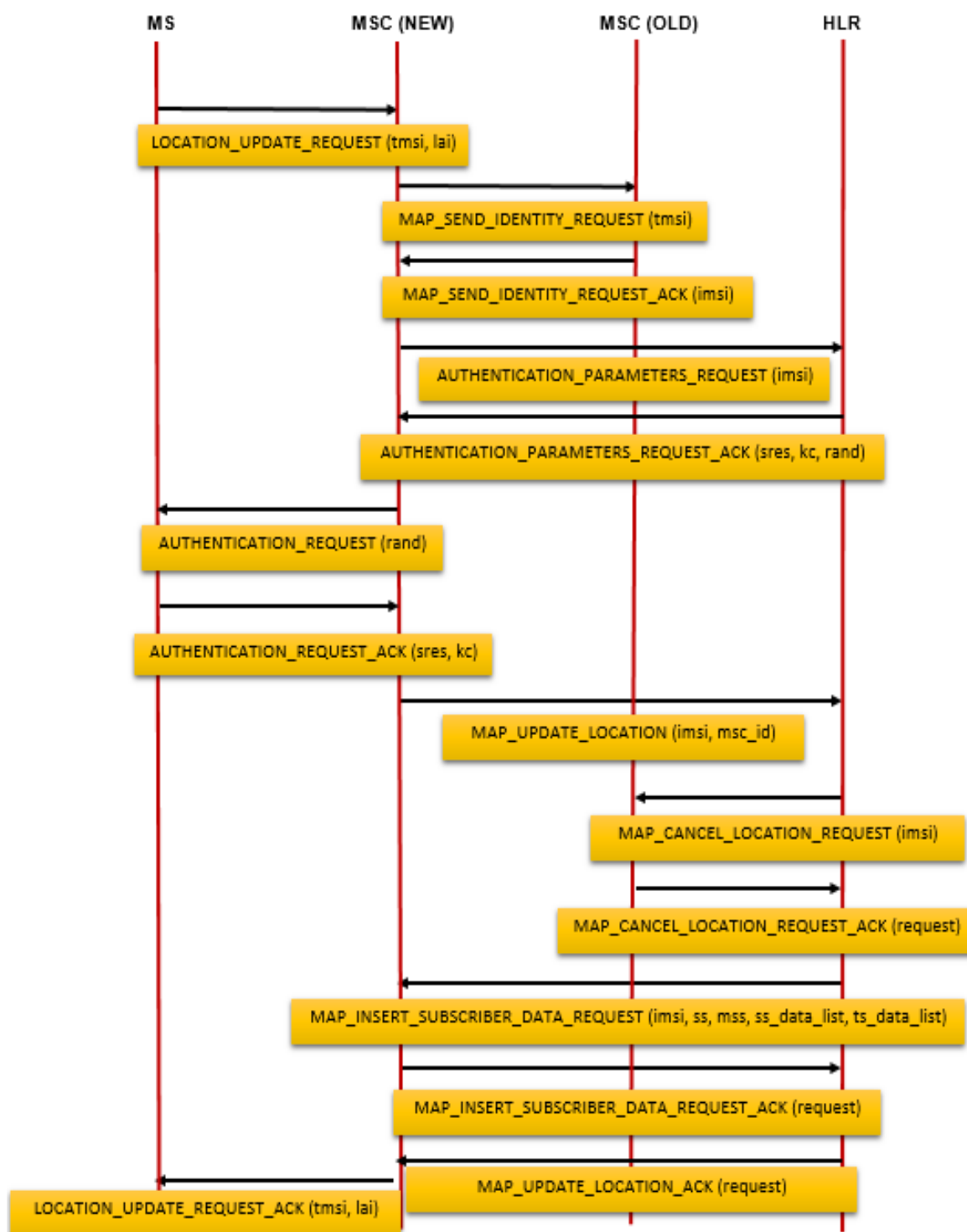


Figura 3.22 Señalización creada para el funcionamiento del protocolo MAP

Location_Update_Request (Tmsi,Lai)

Es el mensaje entre la MS y la MSC para realizar el registro de ubicación del abonado en la red telefónica para lo cual envía como parámetros el TMSI y el LAI [14]

MAP_Send_Identity_Request (Tmsi)

Mensaje entre la MSC nueva y la MSC vieja enviado el TMSI que se obtuvo anteriormente del mensaje Location_Update_Request.

MAP_Send_Identity_Request_Ack (Imsi)

Mensaje entre las MSC, el cual busca con el TMSI que tiene la MSC vieja en su base de datos (VLR) obteniendo el IMSI para enviarlo a la MSC nueva.

Authentication_Parameters_Request (Imsi)

Mensaje entre la MSC nueva y el HLR que es requerido al momento de registrar los servicios del abonado, enviado un IMSI el cual genera los parámetros necesarios para registrarlo.

Authentication_Parameters_Request_Ack (Sres,Kc,Rand)

Mensaje entre el HLR y la MSC nueva. Es la respuesta a la petición de la solicitud hecha anteriormente enviando el SRES, KC, RAND permitiendo que el MS se pueda registrar.

Authentication_Request (Rand)

Mensaje entre la MSC nueva y la MS. Envía el RAND que permite al MS autenticarse.

Authentication_Request_Ack (Sres, Kc)

Mensaje entre la MS y la MSC nueva el cual envía el SRES y KC que se obtuvieron a partir del RAND y se forma un Ki que es guardado en la Sim del abonado.

MAP_Update_Location (Imsi, Msc_id)

Mensaje entre la MSC nueva y el HLR el cual pide la actualización de los servicios correspondientes al IMSI y el MSC_ID enviado.

MAP_Cancel_Location_Request (Imsi)

Mensaje entre el HLR y la MSC vieja el cual toma el IMSI del abonado y lo busca en la MSC realizando un borrado de su información.

MAP_Cancel_Location_Request_Ack (Request)

Mensaje entre la MSC vieja y el HLR el cual confirma que se ha borrado correctamente la información del abonado de la base de datos (VLR).

MAP_InsertSubscriber_Data_Request(Imsi,Ss,Mss,Ss_data_list,Ts_data_list)

Mensaje entre el HLR y la MSC nueva el cual envía una serie de parámetros que son usados para los servicios suplementarios que son almacenados en la base de datos de la MSC.

MAP_Insert_Subscriber_Data_Request_Ack (Request)

Mensaje entre la MSC nueva y el HLR que indica que los servicios que se enviaron anteriormente han sido correctamente registrados en el IMSI proporcionado.

MAP_Update_Location_Ack (Request)

Mensaje entre el HLR y la MSC nueva mediante el cual nos indica que se ha registrado la localización del abonado.

Location_Update_Ack (Tmsi, Lai)

Mensaje entre la MSC nueva y la MS envía al celular los parámetros TMSI y LAI actualizados para ser utilizados en la red que se encuentran y tener la información de su ubicación [15]

3.3.3. Código Fuente – PHP

Se visualiza capturas de pantallas de los distintos mensajes de señalización MAP programados en PHP con el IDE de Netbeans para Ubuntu. Esta programación se realizó con lecturas de base de datos que contenían

información del abonado (cliente celular), como su TMSI, LAI, Número de Ms Conectado, número de teléfono celular, Servicios Suplementarios. El Esquema de Programación consiste en recibir cierto parámetro antes configurado o extraído de la base de datos y responder inmediatamente con un parámetro determinado, siguiendo la señalización. Sin embargo la programación que se observará solo será la que se configura en las MSC (cobertura Guayaquil y Cobertura Quito). Además de los respectivos códigos fuentes hechos en netbeans para seguir ordenadamente la señalización, también se ha implementado una librería de funciones propias para darle una mejor organización al código. Esta librería se encuentra en la carpeta útil del proyecto. A continuación, detallaremos la programación de los mensajes:

Mensaje Location_Update_Request

```

header('Access-Control-Allow-Origin: *');
include_once 'util/funciones.php';
include_once 'util/registrarse.php';
include_once 'util/enviar_mensaje.php';
include_once 'util/parametros.php';
include_once 'util/ss7.php';
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

//registrarse("location_update_request");
registrarse_recibido_nuevo("location_update_request", $_POST);

$SS7_ack=registrarseSS7("MS","MSC NEW","Location Update Request",$_POST);

$tmsi = $_POST["tmsi"];
$lai = $_POST["lai"];

$tmsiBinario = cadenahexadecimal2cadenabinaria($tmsi);
$nri = substr($tmsiBinario, 24, 8);

```

Figura 3.23 Programación mensaje Location_Update_Request

Cada mensaje de señalización como observamos en al Fig. 3.24 posee al inicio de código la invocación de nuestra librería de funciones que creamos al inicio. La MSC (NEW) recibe mediante un POST; un post es un procedimiento de programación que extrae de un formulario o base de datos cierta información, en nuestro caso recibe el TMSI y el LAI de la base datos. Luego la MSC (NEW) los guarda y realiza un envío de mensaje nuevo.

```

07
88
89 // $map_send_identity_request_ack = enviar_nuevo("msc_old(nri: $nri)", $ip_old, "map_send_identity_request", a
90 $SS7_ack=registrarSS7("MSC NEW", "MSC OLD", "MAP Send Identity Request", array('tmsi'=> $tmsi));
91
92 $map_send_identity_request_ack = enviar_nuevo("msc", $ip_old, "map_send_identity_request", array('tmsi'=> $tm
93
94 $imsi = substr($map_send_identity_request_ack[0], 0,15);
95
96 $SS7_ack=registrarSS7("MSC NEW", "HLR", "Authentication Parameters Request", array("imsi"=> $imsi));
97
98 $authentication_parameter_request_ack = enviar_nuevo("hlr", get_parametro_ip_hlr(), "authentication_parameter
99 $sres = $authentication_parameter_request_ack[0];
00 $kcc = $authentication_parameter_request_ack[1];
01 $rand = $authentication_parameter_request_ack[2];
02
03

```

Figura 3.24 Código para envío de mensajes

En la Fig. 3.24 para enviar el mensaje `map_send_identity_request` hacia la MSC (OLD), se usa la función `enviar_nuevo`. Esta función para enviar el mensaje a su destino lo hace por medio de IP, y el TMSI lo envía como un arreglo. Lo observamos en la implementación de esta función.

Mensaje `map_send_identity_request`

En el código `.php` `map_send_identity_request` observamos que primero se registra el TMSI con la función `registrar_recibido_nuevo` realizando un POST de este parámetro. Luego verifica en la base de datos de la VLR este

parámetro para enviar como respuesta a la MSC (NEW) el IMSI correspondiente. Esto se realiza mediante la función `enviar_respuesta`, que envía IMSI mediante un array (arreglo).

```

2
3 include_once 'util/funciones.php';
4 include_once 'util/registras.php';
5 include_once 'util/enviar_mensaje.php';
6 include_once 'util/parametros.php';
7 include_once 'util/ss7.php';
8
9 registrar_recibido_nuevo("map_send_identity_request", $_POST);
10 $tmsi= $_POST["tmsi"];
11 // $tmsiBinario= cadena hexadecimal a binario($tmsi);
12
13 $conexion = mysql_connect("localhost", "root");
14 if(!$conexion)
15 {
16     die("No se puede conectar a la base de datos");
17 }
18 mysql_select_db("vlr");
19 $resultado = mysql_query("select * from vlr where tmsi = '$tmsi'");
20 echo mysql_error();
21 $fila = mysql_fetch_assoc($resultado);
22
23 $imsi=$fila["imsi"];
24
25 $SS7_ack=registrarSS7("MSC_OLD","MSC_NEW","MAP Send Identity Request Ack",array('imsi' => $imsi));
26 enviar_respuesta(array('imsi' => $imsi));
27 ?>
28

```

Fig. 3.25 Código de la función `map_send_identity_request_ack`

Mensaje `authentication_parameters_request`

Este mensaje para llegar a la HLR, usa la función `enviar_nuevo`, envía mediante un arreglo el IMSI, para que esté parámetro llegue correctamente a su destino primero extrae la IP del HLR con la función creada en el código.

La Fig. 3.26 nos muestra la programación del mensaje `authentication_parameters_request_ack` donde consulta a la base de datos de la VLR, los parámetros del sres, kc y el rand para poder actualizarlos en la MSC (NEW).

```

94 $imsi = substr($map_send_identity_request_ack[0], 0,15);
95 $SS7_ack=registrarSS7("MSC NEW","HLR","Authentication Parameters Request",array("imsi"=> $imsi));
97
98 $authentication_parameter_request_ack = enviar_nuevo("hlr", get_parametro_ip_hlr(), "authentication_paramete
99 $sres = $authentication_parameter_request_ack[0];
100 $kc = $authentication_parameter_request_ack[1];
101 $rand = $authentication_parameter_request_ack[2];
102
103
104 //guardar y actualizar msc
105 $conexion = mysql_connect("localhost", "root");
106 if(!$conexion)
107 {
108     die("No se puede conectar a la base de datos");
109 }
110 mysql_select_db("vlr");
111 $sql= "insert into sesion(imsi,tmsi,rand,sres,kc) values ('$imsi','$tmsi','$rand','$sres','$kc)";
112 $result = mysql_query($sql);
113 mysql_close($conexion);
114
115 $SS7_ack=registrarSS7("MSC NEW","MS","Authentication Request",array('rand' => $rand));
117
118 enviar_respuesta(array('rand' => $rand));
119

```

Figura 3.26 Código función authentication_parametes_request_ack

Authentication_Request_Ack

Para programar este mensaje entre la MS y la MSC nueva debemos utilizar una función registrar que nos sirve para guardar en un archivo el mensaje y los parámetros a enviar como se lo muestra a continuación:

```

Source History
1 <?php
2
3 header('Access-Control-Allow-Origin: *');
4 include_once 'util/funciones.php';
5 include_once 'util/registrar.php';
6 include_once 'util/enviar_mensaje.php';
7 include_once 'util/parametros.php';
8 include_once 'util/ss7.php';
9
10
11 registrar_recibido_nuevo("authentication_request_ack", $_POST);
12
13 $SS7_ack=registrarSS7("MS","MSC NEW","Authentication Request Ack",$_POST);
14 $sres = $_POST["sres"];
15 $kc = $_POST["kc"];
16

```

Figura 3.27 Código mensaje authentication_request

RegistrarSS7 es el archivo donde se guardan todos los procesos, contiene el nombre del que envía y recibe, seguido del nombre del mensaje entre ellos y la palabra \$_POST que nos indica que vamos a enviar algunos parámetros, en este caso tenemos dos de ellos los cuales son \$res y \$kc y le pasamos el valor que tiene cada uno de ellos.

Map_Update_Location

El mensaje entre la MSC nueva y el HLR el cual será guardado dentro de un archivo de registro llamado SS7_ack utilizamos la función registrar SS7 que tiene como parámetros el nombre de quien le envía, nombre del que va a recibir el mensaje, nombre del mensaje, y los elementos del mensaje como se puede apreciar a continuación:

```

28 }
29
30 $fila = mysql_fetch_row($resultado);
31 mysql_close($conexion);
32
33 $imsi_anterior = $datos[0][0];
34 $tmsi_anterior = $datos[0][1];
35 $rand_anterior = $datos[0][2];
36 $sres_anterior = $datos[0][3];
37 $kc_anterior = $datos[0][4];
38
39 $conexion = mysql_connect("localhost", "root");
40 if(!$conexion)
41 {
42     die("No se puede conectar a la base de datos");
43 }
44 mysql_select_db("vlr");
45 mysql_query("delete from sesion") or die(mysql_error());
46
47
48 if($sres != $sres_anterior || $kc != $kc_anterior)
49 {
50     enviar_respuesta(array("ERROR"));
51     return;
52 }
53
54 $msc_id = get_parametro_msc_id();
55

```

Fig. 3.28 Código de mensaje map_update_location

Para el Mensaje Map_Update_Location enviamos como parámetros en un arreglo el imsi y el msc_id para lo cual utilizamos la función array que guarda en una de sus posiciones cada uno de los elementos que necesitamos.

Map_Cancel_Location_Request

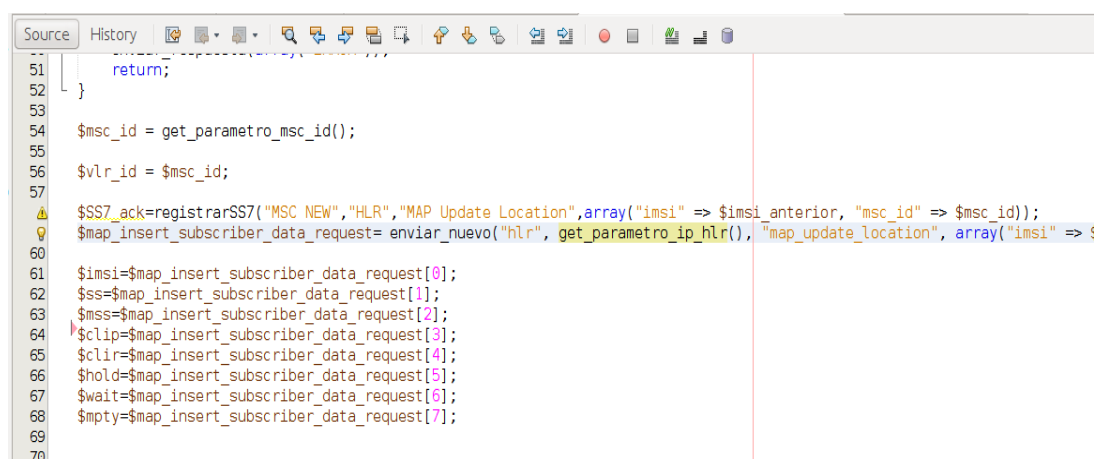
El mensaje entre la MSC vieja y el HLR va a ser creado dentro del registro SS7_ack el cual usa la función registrar SS7 que envía como parámetros del mensaje MSC vieja (nombre del que envía el mensaje), Map_Cancel_Location_Request (nombre del mensaje), HLR (Nombre del destinatario del mensaje), Request (elementos del mensaje), el request es simplemente un mensaje que envía un OK al instante que se realiza todo el proceso.

```
<?php
2
3 include_once 'util/funciones.php';
4 include_once 'util/registrar.php';
5 include_once 'util/enviar_mensaje.php';
6 include_once 'util/parametros.php';
7 include_once 'util/ss7.php';
8
9 registrar_recibido_nuevo("map_cancel_location_request", $_POST);
10 $imsi = $_POST["imsi"];
11
12 $conexion = mysql_connect("localhost", "root");
13 if(!$conexion)
14 {
15     die("No se puede conectar a la base de datos");
16 }
17 mysql_select_db("vlr");
18 mysql_query("delete from vlr where imsi = '$imsi'" ) or die(mysql_error());
19
20 $request=array('request'=>"OK");
21 enviar_respuesta($request);
22
23 ?>
```

Fig. 3.29 Código de mensaje map_cancel_location_request

Map_Insert_Suscriber_Data_Request_Ack

El mensaje entre la MSC nueva y el HLR va a ser creado dentro del registro SS7_ack el cual usa la función registrar SS7 que envía como parámetros del mensaje MSC nueva (nombre del que envía el mensaje), Map_Insert_Suscriber_Data_Request_Ack (nombre del mensaje), HLR (Nombre del destinatario del mensaje), Request (elementos del mensaje), el request es un mensaje en el cual actualiza todos los datos que se encuentran en la base de datos, en este caso de los servicios suplementarios que se encuentran en un arreglo.



```

51     return;
52 }
53
54 $msc_id = get_parametro_msc_id();
55
56 $vlr_id = $msc_id;
57
58 $SS7_ack=registrarSS7("MSC NEW","HLR","MAP Update Location",array("imsi" => $imsi_anterior, "msc_id" => $msc_id));
59 $map_insert_subscriber_data_request= enviar_nuevo("hlr", get_parametro_ip_hlr(), "map_update_location", array("imsi" => $
60
61 $imsi=$map_insert_subscriber_data_request[0];
62 $ss=$map_insert_subscriber_data_request[1];
63 $mss=$map_insert_subscriber_data_request[2];
64 $clip=$map_insert_subscriber_data_request[3];
65 $clir=$map_insert_subscriber_data_request[4];
66 $hold=$map_insert_subscriber_data_request[5];
67 $wait=$map_insert_subscriber_data_request[6];
68 $empty=$map_insert_subscriber_data_request[7];
69
70

```

Fig. 3.30 Código de mensaje map_insert_data_request_ack

Location_Update_Request_Ack

El mensaje entre la MSC nueva y la MS va a ser creado dentro del registro SS7_ack el cual usa la función registrar SS7 que envía como parámetros del mensaje MSC nueva (nombre del que envía el mensaje), Location_Update_Request_Ack (nombre del mensaje), MS (Nombre del

destinatario del mensaje), Request (elementos del mensaje), el request es simplemente un mensaje que envía un OK al instante que se realiza todo el proceso, en este caso al recibir el OK nos indicaría que ya se actualizado todos los nuevos datos del abonado tanto de su localización, así como también sus servicios [16]

```

85 //guardar y actualizar msc
86 $conexion = mysql_connect("localhost", "root");
87 if(!$conexion)
88 {
89     die("No se puede conectar a la base de datos");
90 }
91 mysql_select_db("vlr");
92 $sql= "insert into vlr(msc_id,vlr_id,imsi,tmsi,lai,suscriber_status,ms_status,clip,clir,hold,wait,mpty) values ('$msc_id
93 $result = mysql_query($sql);
94
95 /*
96 $request = "OK";
97 $map_update_ack = enviar_nuevo("hlr", get_parametro_ip_hlr(), "map_insert_suscriber_data_request_ack", array('request'=>
98 */
99 $conexion = mysql_connect("localhost", "root");
100 if(!$conexion)
101 {
102     die("No se puede conectar a la base de datos");
103 }
104 mysql_select_db("vlr");
105 $resultado = mysql_query("select * from vlr where imsi = '$imsi_anterior'");
106 echo mysql_error();
107 $fila = mysql_fetch_assoc($resultado);
108
109 $tmsi=$fila["tmsi"];
110 $lai=$fila["lai"];
111
112
113 $inform=array('tmsi' => $tmsi,'lai' => $lai);
114
115 $$SS7_ack=registrarSS7("MSC NEW","MS","Location Update Request Ack",$inform);
116
117 enviar_respuesta($inform);
118
119

```

Fig. 3.31 Código de mensaje location_update_request_ack

3.3.4 Implementación

Los procesos que se describen a continuación simulan el comportamiento de cada uno de las MSC al momento de que realiza el location_update_request así como también al momento de interactuar con la señalización descrita en la Figura 3.22

MSC

MS→MSC Leer parámetro “request” desde pedir_numeros(request)

Caso 0: request!=1: devolver vacío

Caso1: request =1

- Buscar en la base de datos VLR el msc_id
- enviar MSC→HLR (192.168.0.2) buscar_numeros(msc_id)

HLR→MSC Leer la lista msisdn_list de buscar_numeros_ack (msisdn_list)

MSC→MS (192.168.2.3)

Pedir_numero_ack (msisdn_list)

MSC1 (Nueva)

Lee tmsi y lai de:

(MS→MSC) location_update_request (tmsi, lai)

Convertir el tmsi a binario

Extrae NRI de tmsi

Buscar en la base de datos VLR el msc_id.

Compara MSC_id con NRI

Caso0: MSC_id==NRI; Responder a MS (192.168.2.3) el valor vacío

Caso1: MSC_id!=NRI;

- Convertir los 8 últimos bits de NRI a decimal
- Obtener IP (192.168.0.#; #: 8 últimos bits del NRI en formato decimal).

MSC1→MSC2

Enviar: map_send_identity_request (tmsi) a la ip 192.168.0.# #: 8 últimos bits del NRI en formato decimal.

Lee y almacena imsi de

(MSC2→MSC1) map_send_identity_request_ack (imsi)

MSC1→HLR

Enviar: authentication_parameters_request (imsi)

A la dirección 192.168.0.2

Lee y almacena sres, kc, rand de

(HLR→MSC1) authentication_parameters_request_ack (sres, kc, rand)

MSC1→MS

Authentication_request (rand)

Envía el rand obtenido anteriormente del HLR

Lee y almacena sres y kc de

(MS→MSC) Authentication_request_ack (sres,kc)

Nota: los sres, kc son del MS no del HLR

Comparar SRes y Kc del HLR con sres y kc del MS

Caso 0: [sres y kc] HLR \neq [sres y kc] MS

MSC: "Autenticación no completada"

MSC1→MS: disconnected()

Caso1: [sres y kc] HLR == [sres y kc] MS

MSC: "Autenticación completada"

(MSC1→HLR) map_update_location (imsi,msc_id)

Lee y actualiza su base de datos VLR de:

(HLR→MSC1) map_insert_subscriber_data_request (imsi,ss, mss,
ss_data_list, ts_data_list)

-Busca en su base de datos VLR el imsi y actualiza los parámetros ss, mss,
ss_data_list, ts_data_list.

MSC1→HLR

map_insert_subscriber_data_request_ack (request)

(donde request =1)

MSC1→MS

Buscar en la base de datos VLR el imsi.

Extraer tmsi con el imsi y extraer lai.

Data_location_update (tmsi, lai).

MSC: "Actualización exitosa".

MSC2 (Vieja)

MSC1→MSC2

Map_send_identity_request (tmsi)

Lee tmsi

Buscar y extraer en la base de datos VLR el imsi correspondiente al tmsi leído.

Transformar el tmsi a binario

Extraer el NRI del tmsi;

Convertir los 8 últimos bits del NRI en decimal

Obtener ip 192.168.0.#; #: 8 últimos bits del NRI en formato decimal

MSC2→MSC1 (192.168.0.#) #: 8 últimos bit del NRI en formato decimal del tmsi.

Map_send_identity_request_ack (imsi)

HLR→MSC2

Map_cancel_location_request (imsi)

Lee el imsi

Busca en su base de datos VLR los datos correspondientes al imsi leído.

Borra todos los datos correspondientes al imsi leído.

MSC2→HLR

Map_cancel_location_request_ack (request)

Caso 0: request!=1 MSC "Datos no Eliminados".

Caso 1: request=1 MSC "Datos Eliminados".

3.4 Aplicaciones

3.4.1 Aplicaciones Educativas

Este proyecto se usará para fines educativos, razón por la cual fue realizado a pequeña escala y con equipos físicos ya mencionados, que permita un ajuste lo más cercano a la realidad.

La simulación sirve para explicar muchos conceptos teóricos visto en materias universitarias como Comunicaciones Inalámbricas y Comunicaciones Digitales, explica el procedimiento que se realiza en la telefonía móvil, como sucede en nuestro proyecto, donde el sistema móvil se conecta a una red inalámbrica que nos ayudaría también a explicar

experimentalmente modelos de propagación de la señal Wifi en el medio inalámbrico que se transmite.

Podemos también mencionar la presencia de una pequeña red de dispositivos donde se explicaría el proceso de asignaciones de direcciones lógicas, IP, a cada uno de los componentes usados, explicando conceptos de subredes, comunicación entre todos ellos al momento de probar la interconexión como herramientas como el ping, conectar cables Ethernet revisando el estándar de colores del cable UTP, determinar si debemos usar cables directo o cruzados según la respectiva conexión entre equipos.

Finalmente con la simulación y su visualización general del funcionamiento del protocolo MAP se puede exponer conceptos de programación, en nuestro caso usando el lenguaje PHP, lenguaje html y ciertos comandos que se utilizan en la terminal de Ubuntu para la ejecución del proyecto, para levantar la base de datos y se sincronice perfectamente con netbeans, darle permisos a carpetas, entre otros.

3.4.2 Aplicaciones Comerciales

El proyecto puede ser comercializado a cualquier empresa telefónica que actualmente operan en el país para ser usado en posibles capacitaciones al personal técnico que realiza estas operaciones o para describir el respectivo proceso de localización del abonado junto con su autenticación de número telefónico en la central que opere.

Este proceso de comercialización también se puede usar en otros campos de telecomunicaciones, por su características de localización puede ser empleados en seguros de los autos de una concesionaria, ya que se comporta similarmente a lo que nosotros realizamos. Cada vez que el automóvil se mueva, el respectivo circuito de rastreo almacenará la ubicación del mismo, y podría comunicar al dueño su localización con coordenadas exactas mediante un SMS o un correo electrónico, adhiriendo otros equipos al proyecto como un GPS.

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

4.1 Resultados obtenidos

Al finalizar la implementación, se obtiene como resultado un servidor de Linux capaz de hacer funcionar aplicaciones del SS7. Para nuestro caso el protocolo MAP, el cual nos muestra paso a paso el intercambio de MSC para la localización de un abonado, cabe recalcar que para efectos de esta simulación cada MSC brinda una cobertura a través de una red Wifi y el celular está en la capacidad de conectarse a cualquiera de estas redes, o MSC.

El uso del lenguaje de programación de hipertexto PHP nos facilita la implementación de los mensajes, la interacción entre todos los elementos que intervienen en una red GSM, así como también que los mensajes entre

ellos se envíen en tiempo real lo que nos proporciona una estimación del tiempo que se demora en realizar el registro de ubicación.

Es necesaria la creación de una interfaz gráfica utilizando código HTML (Lenguaje de Marcas de Hipertexto), para poder visualizar de mejor manera todo el proceso de señalización incluyendo todos los parámetros que contienen cada uno de los mensajes, además de una interfaz para simular el celular, la cual nos brinda toda la información acerca de un abonado y teclas que son necesarios para el correcto funcionamiento

Observamos paso a paso todo el proceso de envío y recibido de respuesta que en un ambiente real se producen en tiempos imperceptibles por un abonado final.

4.2 Simulaciones

Para poder obtener una presentación más amigable al usuario y que se puede entender la señalización de nuestro protocolo de estudio MAP, fue diseñado en lenguaje HTML, como podemos observar en el Figura 4.1, el inicio de la simulación empieza con el encendido del móvil. Luego de ser encendido nos pedirá que escojamos un número de teléfono de los diez que se despliegan en ese instante [17]



Figura 4.1 Selección de número celular

Sin embargo, se debe mencionar que inicialmente estamos conectados a la red inalámbrica con cobertura de Quito o a la red inalámbrica con cobertura Guayaquil. De las dos MSC que usamos, cada una posee independientemente registrado cinco números de teléfono con sus respectivos parámetros en su respectiva base de datos. Luego que escogemos el número de teléfono, con el botón *i* (ubicado en la parte inferior derecha de la interfaz del móvil) podemos visualizar toda su información como lo muestra la Figura 4.2.

La información que contiene cada número de celular nos indica la operadora a la que pertenece, la MSC_ID que en ese instante se encuentra conectado, el TMSI, los algoritmos de autenticación del abonado, su respectivo LAI, la ubicación: UIO (Quito) o GYE (Guayaquil).



Figura 4.2 Información del abonado seleccionado

Como se mencionó anteriormente, cada MSC posee su base de datos llamada VLR, ambas MSC poseen números diferentes guardados. Si al momento de escoger el número celular, si éste se encuentra almacenado en la VLR de la MSC que estamos conectados no hará ninguna actualización en sus datos. Ya que la señalización MAP realiza actualizaciones de localización, siempre y cuando nos cambiemos de cobertura en las redes inalámbricas que dan los *router*. El proceso de señalización se visualiza presionando la cartita que tiene el celular en la parte inferior izquierda, como se muestra en la figura 4.3

SEÑALIZACION MAP

REGISTRADOR DE LOCALIZACION DE ABONADOS

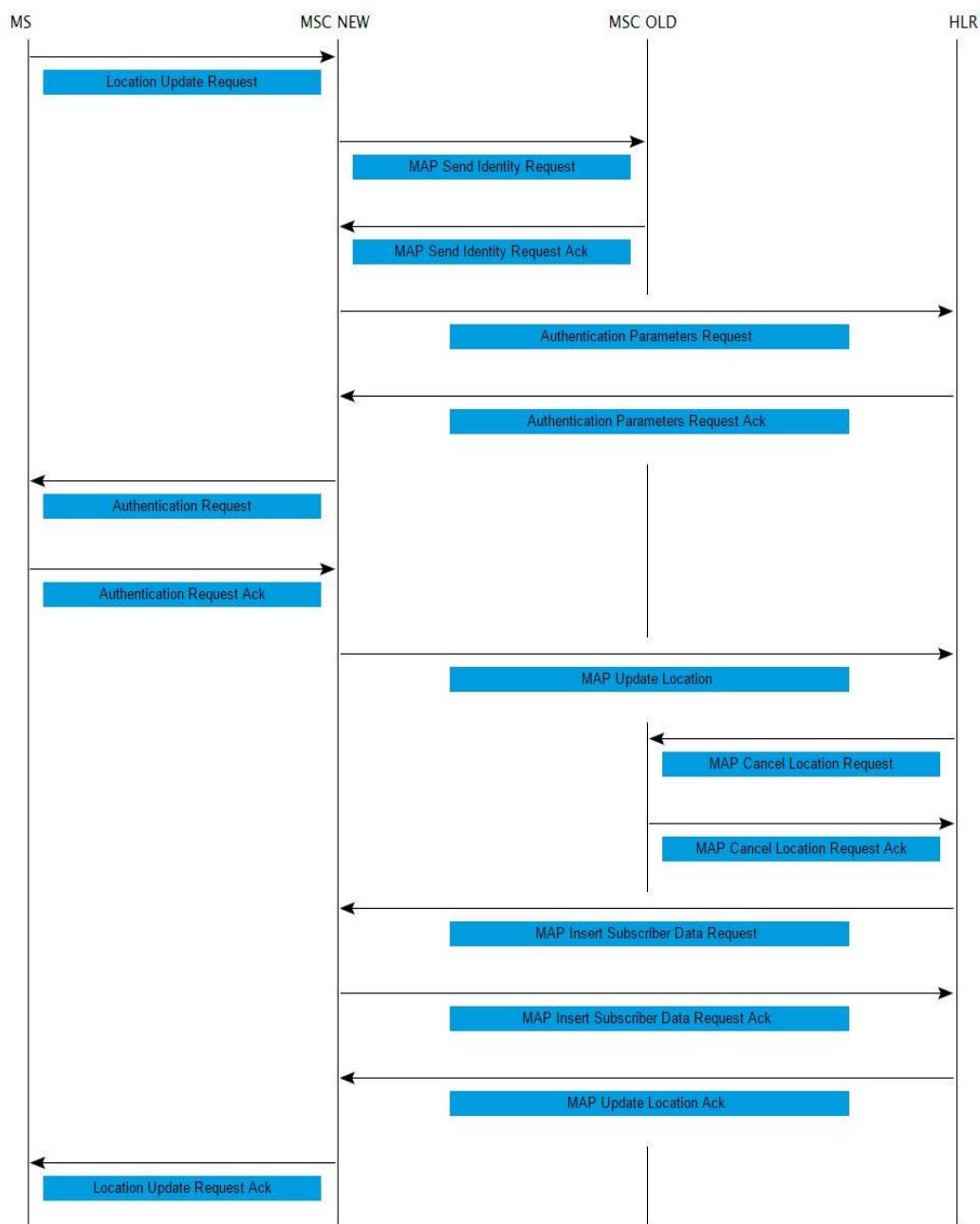


Figura 4.3 Interfaz realizada para la Señalización MAP

En la figura 4.3 observamos toda la interfaz de la señalización MAP, en la parte superior hay tres botones de color verde. El botón izquierdo verde realiza la señalización paso a paso, enviado uno por uno los mensajes. Si damos clic en los cuadros de color celeste de cada mensaje observaremos toda la información del abonado celular en dicho mensaje de señalización.

Lo visualizaremos tal como se muestra en la Figura. 4.4 siguiente:

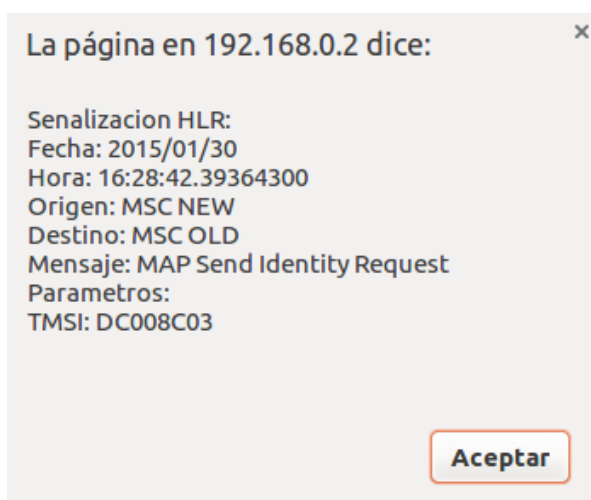


Figura 4.4 Interfaz de información de mensaje de señalización

Cada uno de los mensajes de señalización lleva información de fecha y hora en que se ejecuta la simulación, el origen y el destino del mensaje, el nombre del mensaje y sobre todo los parámetros que envía cada mensaje.

Por ejemplo, en la imagen anterior el Mensaje map_send_identity_request es enviado desde la MSC NEW a la MSC OLD, y lleva de parámetro un TMSI:DC008C03. Por consiguiente vemos la siguiente figura que detalla la respuesta del mensaje anterior.



Figura 4.5 Interfaz mensaje map_send_identity_request

Se observa que ahora la de origen es la MSC OLD y responde a la MSC NEW con el mensaje map_send_identity_request_ack con un IMSI: 740004008446873 de parámetro.



Figura 4.6 Interfaz mensaje authentication_parameters_request

En la figura 4.6 el origen del mensaje es la MSC NEW y su destino es el HLR, el mensaje enviado es el `authentication_parameter_request` y envía como parámetro el IMSI que recibió en el mensaje anterior por el HLR.



Figura. 4.7 Interfaz de mensaje `authentication_parameters_request_ack`

Ahora el HLR como origen enviará al MSC NEW el mensaje de `authentication_Parameter_request_ack` con los parámetro SRES: 3604515254, el KC: 2543604515 y el rand: 254.

Mencionamos que todos estos datos de SRES, KC y rand ya se encuentran almacenados en la base de datos de cada MSC respetivamente. Y esto se realiza para la autenticación respectiva del número que cogimos al inicio de la simulación.



Figura 4.8 Interfaz mensaje authentication_request

El origen en la Figura 4.7 es la MSC NEW y su destino es el MS. Realiza las respectivas negociaciones con el celular mediante el mensaje authentication_request con el parámetro de un Rand=254.

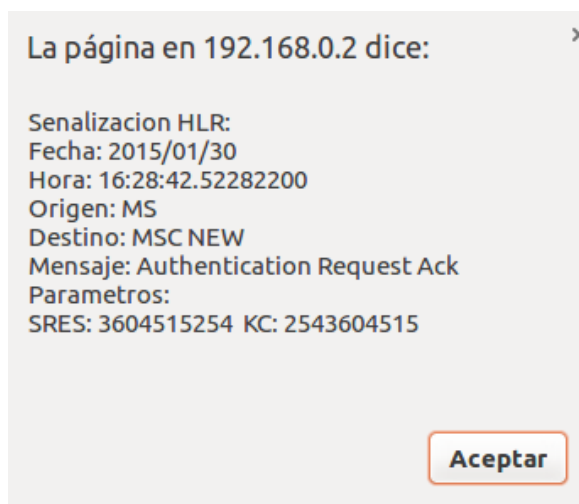


Figura. 4.9 Interfaz mensaje authentication_request_ack

Ahora responde el MS (celular) hacia la MSC NEW con el mensaje de señalización `authentication_request_ack` enviando los parámetros de un SRES: 3604515254 y un KC:2543604515.



Figura. 4.10 Interfaz mensaje `map_cancel_Location_Request`

En la figura 4.10 el HLR envía la petición a la MSC OLD mediante el mensaje `map_cancel_location_request` que borre todos los datos que posee el abonado con el IMSI 740004008446873 en su MSC.



Figura 4.11 Interfaz mensaje `map_cancelation_request_ack`

Por consiguiente, la MSC old procede a borrar los datos de este abonado en su base de datos y responde al HLR con OK como parámetro en el mensaje de map_cancel_location_reques_ack.



Figura 4.12 Interfaz mensaje map_insert_subscriber_data_request

En la Figura 4.12 el HLR envía hacia la MSC NEW todos los servicios suplementarios que posee dicho abonado móvil, en este caso con un IMSI: 740004008446873, sus servicios: SS:1, MSS:1, CLP:1, CLIR:1, HOLD:1, WAIT:1, MPTY:1 [18]

Finalmente se procederá a actualizar los datos del abonado en la nueva cobertura que ahora se encuentra. En la figura 4.13 se puede observar la localización inicial del abonado móvil y su localización final una vez que se realiza toda la señalización MAP.



Figura 4.13 Interfaz móvil – cambio de localización

En la figura 4.13 la interfaz del móvil del lado izquierda son los parámetros que tiene el móvil conectado a la MSC de Cobertura Quito y con un MSC_ID:00000100, mientras que el interfaz que tiene el móvil derecho presenta la información actualizada de los parámetros que posee ahora el móvil. Como podemos observar hay un cambio en la MSC, de Quito que estaba inicialmente ahora se encuentra en Guayaquil y el número de MSC_ID ahora es 00000011.

4.3. Interpretación de Resultados

Al encender el teléfono celular se tiene acceso a una cierta cantidad de números, de los cuales podremos escoger cualquiera, si es un usuario nuevo el deberá elegir el número con el que va a tener acceso a la red, una vez

conectado a la red el usuario tendrá acceso a todos los servicios que la operadora le provea.

Al trasladarse un abonado de un sitio a otro se realiza todo el proceso de localización, para actualizar su información y que pueda comunicarse sin ningún problema.

Al escoger el número 593983604515 con la red de Guayaquil se espera que, exista un cambio de localización, es decir cuando el teléfono se enganche con la red de Quito e inicie el proceso de localización.

Una vez que se inicia el proceso envía el mensaje Location Update Request desde el MS a la MSC nueva, que envía como parámetros el TMSI: DC008C03 y el LAI: 7400016110, que son los datos de un usuario de Movistar perteneciente al número antes mencionado.

La MSC nueva envía enseguida un mensaje a la MSC vieja con el TMSI: DC008C03 que le envió la MS en el mensaje anterior, con el TMSI la MSC vieja realiza la consulta en su base de datos VLR con el mensaje Map Send Identity Request y envía el IMSI: 740004008446873 hacia la MSC nueva y el mismo IMSI se envía al HLR a través del mensaje Authentication Parameter Request, el propósito es que al llegar al HLR este pueda tener acceso a los servicios suplementarios.

El HLR accede a su base de datos y devuelve a la MSC nueva las tripletas que necesita para identificarse SRES: 3604515254, KC: 2543604515, RAND: 254 y la MSC envía el RAND al teléfono celular para que registre, a través

del mensaje Authentication Request Ack para que con el RAND se generen el SRES y el KC, si los dos son iguales a los obtenidos anteriormente se permite continuar con el proceso, caso contrario si existe una autenticación falsa ya no continuará con el mismo.

El HLR envía el mensaje Map Cancel Location a la MSC vieja con el IMSI correspondiente al número para que realice el borrado de toda la información que tiene almacenada del abonado, si la operación fue realizada con éxito la MSC vieja envía al HLR el mensaje Map Cancel Location Ack que tiene como parámetros un Request: OK, lo cual indica que fue borrado correctamente.

Es necesario después de haber realizado el borrado de la información del abonado volver a registrarlo pero ahora en su nueva red, para ello el HLR envía a la MSC nueva el IMSI junto a los servicios suplementarios mediante el mensaje Map Insert Subscriber Data Request y la MSC nueva le envía una respuesta al HLR con el mensaje Map Insert Subscriber Data Request Ack para indicarnos que se ha recibido correctamente el mensaje.

El mensaje Map Update Location Ack envía como parámetros un Request: OK del HLR a la MSC nueva, para indicar que ya se localizó al abonado y simultáneamente la MSC nueva envía al celular los datos actualizados del abonado es decir el TMSI: DC008C04 y el LAI: 7400010540, aquí podemos ver que se realizó con éxito la señalización porque si hubo un cambio de MSC que se puede notar en el TMSI, ya que este contiene la MSC ID que son los últimos ocho bits de la trama o el último número del TMSI cuando

recién enviamos el mensaje estaba en 3 que estaba vinculada a la MSC de Guayaquil y una vez que sucedió el cambio paso a ser 4 que tiene relación con la MSC de Quito, además nos damos cuenta que su LAI cambió esto quiere decir que ya está conectada a otra área de ubicación de la misma operadora.

OTRAS APLICACIONES

Se utilizó un aplicativo móvil para android, que se lo descarga gratis de la tienda de Play Store, que se llama inViu OpenCellID el cual al conectarse a una red de datos o Wifi, obtiene información de la celda en donde se encuentra, así como también de su ubicación con la ayuda del GPS que poseen los teléfonos inteligentes, además nos muestra datos importantes del abonado como son su operadora, intensidad de la señal [19]



The screenshot shows the inViu OpenCellID application interface. At the top, there is a status bar with icons for location, Wi-Fi, cellular signal, and battery (73%), along with the time 03:43 PM. Below the status bar, the application title "inViu OpenCellID" is displayed. There are two buttons: "detener recogida" (stop recording) and "enviar células" (send cells). The main content area is divided into two sections: "actual célula" (current cell) and "posición actual" (current position). The "actual célula" section displays the following data: cell-ID: 554577, MNC: 00, MCC: 740, LAC: 30755, red: Movistar, and intensidad de la señal: -85 dBm. The "posición actual" section displays: latitud: -1,676577, longitud: -78,640935, and satélites: 12 de 16.

actual célula	
cell-ID:	554577
MNC:	00
MCC:	740
LAC:	30755
red:	Movistar
intensidad de la señal:	-85 dBm

posición actual	
latitud:	-1,676577
longitud:	-78,640935
satélites:	12 de 16

Figura 4.14 Interfaz de la ubicación del móvil ubicado en Riobamba

Observamos en la imagen anterior que muestra el número de célula de la red Movistar a la que en ese instante se encuentra conectado. Se visualizan las coordenadas de latitud y longitud de Riobamba.



Figura 4.15 Interfaz de la ubicación del móvil ubicada en Guayaquil

De igual manera se muestra el número de célula que se encuentra conectado el móvil a la red de CNT, las coordenadas de la latitud y longitud exacta de Guayaquil. Podemos concluir que en la aplicación se diferencia la MNC para CNT que es 02, y la MNC de MOVISTAR que es 00.

CONCLUSIONES

Luego de todo el análisis y las pruebas efectuadas en el tiempo que se realizó este informe, llegamos a concluir que:

1. El protocolo de Aplicaciones Móviles (MAP) posee, entre sus robustas características, un nivel de confiabilidad muy elevada, debido a que cuenta con un mecanismo de autenticación del abonado móvil al conectarse a una red, ayudándose de algoritmos como el A3 y el A8, números aleatorios que finalmente confirmarán la pertenencia del abonado a dicha operadora móvil y evitar problemas de clones de números telefónicos.
2. Se comprobó que entre los servicios que otorga la parte de aplicaciones móviles está la actualización de localización de un móvil, que se produce cada vez que un móvil se cambia de cobertura o se conecta a una MSC diferente, la MSC es la parte central en las redes inteligente móviles, realiza las respectivas peticiones a su componentes interconectados e incluso pedir información a otras MSC de su nuevo abonado móvil que se conectó. Para finalmente actualizar los datos del usuario, pedir además a la HLR todos los servicios que este posee habilitados y deshabilitados para continuar con su normal funcionamiento en la comunicación.

3. El protocolo MAP opera mediante mensaje que conforman una señalización, empezando con la autenticación del usuario, llevando un flujo ordenado de envío y recepción de parámetros para la interacción entre sí de equipos como: centro de conmutaciones móviles (MSC), Registro de Localización Local (HLR), Registro de Localización del visitante (VLR) y la estación móvil (MS).

4. Se comprobó que MAP conforma parte del Sistema de señalización SS7, por su trabajo e implementación en redes inteligentes, su funcionamiento, los servicios que brinda al usuario para una comunicación ordenada, gozando de mucha privacidad y otorgar al usuario final una óptima gestión de servicios suplementarios que pueda ofrecer a nivel de Parte de Aplicaciones Móviles.

5. El uso de lenguaje de programación abierto como el Preprocesador de Hipertexto (PHP) junto con el sistema operativo Ubuntu de Linux donde fue programado nuestro proyecto, nos ayuda a trabajar en software que reduzcan los gastos económicos al adquirir licencias muy costosas, estos software de código abierto no se depende de empresas específicas para el mantenimiento. Solo es necesario tener habilidad y mucho conocimiento para corregir uno mismo cualquier problema.

6. Desarrollamos una herramienta didáctica para que todo el proceso efectuado del proyecto sea entendible para cualquier persona que no domine el tema de señalización de protocolos.

RECOMENDACIONES

1. Se recomienda conocer los comandos necesarios para operar en el sistema operativo de Linux, ya que es de mucha importancia para poder otorgar los permisos necesarios para la instalación de programas o creación de archivos dentro de una carpeta.
2. Se debe tomar en cuenta que exista compatibilidad entre los programas a utilizar debido a que puede tener este inconveniente, por lo que afectaría al momento de ejecutar la aplicación.
3. Antes de iniciar la simulación debemos inicializar el terminal con las teclas Ctrl + Alt + T seguido del archivo `.log`, el cual es el que registra todos los movimientos en cada uno de los mensajes en tiempo real e imprime en el terminal, si no se ejecuta correctamente no se podrá continuar con todo el proceso de señalización.
4. Para realizar una correcta simulación es importante tener en cuenta que el servidor Xampp se encuentre correctamente levantado, es decir que todos sus componentes estén activados, debido a que puede existir un problema de conexión ya sea con las bases de datos o con la comunicación entre las computadoras.

5. Se debe realizar una actualización de la base de datos después que se ejecuta toda la señalización ya que los valores son afectados después de todo el procedimiento, es importante ya que si se vuelve a ejecutar sin actualizarlo la aplicación se puede caer debido a que no encontrará los datos anteriores.

BIBLIOGRAFÍA

- [1] Lee Dryburg & Jeff Hewett, "Signaling System No. 7(SS7/C7): Protocol Architecture and Services" , fecha de consulta noviembre de 2014
- [2] Marcombo Editores, Telecomunicaciones Móviles, <https://books.google.com.ec/books?isbn=8426711499>, fecha de consulta diciembre 2014
- [3] Telecom and Datacom Education, GSM COmunication Flow, <http://telecomedu.blogspot.com/2013/01/gsm-communication-flow.html>, fecha de consulta noviembre de 2014
- [4] Digital cellular telecommunication system, Mobile Aplication Part, European Telecommunications Standards Institute, fecha de consulta diciembre de 2014.
- [5] Bosse John, Signaling in Telecommunications Networks, http://books.google.com.ec/books?id=IJmNR1ygkRwC&pg=PA783&dq=MAP_FORWARD_NEW_TMSI&hl=es&sa=X&ei=XH1qVPz6HYmaNpPiglg&ved=0CCUQ6AEwAQ#v=onepage&q=MAP_FORWARD_NEW_TMSI&f=false, fecha de consulta enero de 2015
- [6] Global System for Mobile Communications, Addresses and Identifiers, http://www.tutorialspoint.com/gsm/gsm_addressing.htm, fecha de consulta febrero 2015

- [7] Falken, Gsm Segunda Parte, <http://www.set-ezine.org/index.php?num=13&art=9>, fecha de consulta febrero 2015
- [8] Heurtel Olivier, PHP y MySQL, fecha de consulta enero de 2015
- [9] Creative Commons Attribution-Share Alike, Primero Pasos con Ubuntu, <http://people.ubuntu.com/~chilicuil/pdf/ubuntu-manual-es.pdf>, fecha de consulta enero de 2015
- [10] Telecom Tigers, update location request procedure, http://telecomtigers.blogspot.com/2009/12/location-update-procedure.html#.VMqtB2iG_T4, fecha de consulta enero de 2015
- [11] Iglesias César, Lanzamiento Ubuntu 14.10 requisitos, <http://lignux.com/finalmente-lanzamiento-de-ubuntu-14-04-trusty-tahr/>, fecha de consulta enero 2015.
- [12] Ubuntu Zone, Instalar Xampp en Ubuntu, <http://www.ubuntu-guia.com/2013/10/instalar-xampp-ubuntu-1404.html>, fecha de consulta diciembre de 2014
- [13] Netbeans Org, Instrucciones de Instalación Netbeans IDE, https://netbeans.org/community/releases/61/install_es.html, fecha de consulta diciembre de 2014.
- [14] What-When-How, Sistema de Posicionamiento Global, <http://what-when-how.com/roaming-in-wireless-networks/scenarios-global-system-for-mobile-communication-gsm-part-1/>, fecha de consulta diciembre de 2014

- [15] Ivan Bernal, PH.D., Protocolos y Señalización, <http://bibing.us.es/proyectos/abreproy/11980/fichero/CAP%CDTULO+3++FUNDAMENTOS+GSM+Y+UMTS%252F3.3+GSM.pdf>, fecha de consulta enero de 2015
- [16] Moreno Jose, Soto Ignacio, Larrabeiti David, Universidad Carlos III Madrid, <http://www.it.uc3m.es/~jmoreno/articulos/protocolssenalizacion.pdf>, fecha de consulta diciembre de 2014
- [17] Desarrollo web, Aprendiendo HTML, <http://www.desarrolloweb.com/html/>, fecha de consulta diciembre de 2014
- [18] UIT –T Unión Internacional de Telecomunicaciones, Definición de Servicios Suplementarios, <file:///C:/Users/ADMIN/Downloads/T-REC-I.250-198811-!!!PDF-S.pdf>, fecha de consulta enero de 2015
- [19] Enaikoon, Aplicación para dispositivos móviles Inviu opencellid, <https://www.enaikoon.com/es/movil/plugins/opencellid/>, fecha de consulta enero de 2015