



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Implementación de una  
interfaz gráfica de manejo de  
un controlador de sesión de  
frontera para realizar  
llamadas de voz sobre IP**

**INFORME DE ACTIVIDADES PROFESIONALES**

Que para obtener el título de  
**Ingeniero Eléctrico Electrónico**

**P R E S E N T A**

Alejandro Paris Hernández Suárez

**ASESORA DE INFORME**

M. E. Jacquelyn Martínez Alavez



Ciudad Universitaria, Cd. Mx., 2019

# ÍNDICE

<b>Objetivo</b> .....	1
<b>Marco teórico</b> .....	1
<b>Antecedentes</b> .....	1
<b>Introducción</b> .....	2
Telefonía .....	2
FDM .....	2
TDM .....	3
Redes e Internet.....	4
Modelo OSI .....	5
Modelo TCP/IP .....	6
Voz sobre IP (VoIP).....	7
Ventajas de la voz sobre IP .....	9
Desventajas de la voz sobre IP .....	9
Factores que afectan la voz y métricas de calidad para la voz .....	10
Protocolos .....	12
<b>Definición del problema y contexto de participación profesional</b> .....	14
Session Border Controller (SBC) .....	14
<b>Análisis y metodología empleada</b> .....	15
Instalación de Sistema Operativo (GBLinux) .....	16
Configuración de Red.....	17
Instalación de aplicación (iServer) .....	17
Configuración de VoIP.....	17
Integración con interfaz gráfica (RSM).....	17
Llamada de voz sobre IP usando <i>Softphone X-Lite</i> .....	17
<b>Participación Profesional</b> .....	18
Instalación y actualización de SBC a 9.2.8 .....	18
Requerimientos de software .....	19
Requerimientos de hardware.....	19
Descarga de imagen de sistema operativo base .....	20
Creación de imagen de arranque .....	20
Instalación de versión 8.1.1.0 .....	21
Configuración de interfaces usando <i>YaST</i> .....	24
Actualización a GBLinux 8.3 y 9.2 .....	27

Instalación de 8.1.1-6-sb y saltos requeridos para llegar a 9.2.0-10 .....	29
Instalación de paquetería de medios y aplicación del SBC ( <i>iServer</i> 9.2.8.0) .....	33
Integración con RSM ( <i>Real-time Session Manager</i> ) .....	38
Integración de SBC con RSM.....	39
Configuración de Interfaces .....	41
Configuración de Interfaces a través de FCE ( <i>Firewall Control Entity</i> ).....	43
Acceso a RSM y FCE .....	43
Dispositivos de Ruta de Medios ( <i>Media Routing Device</i> ) .....	47
Vnet de medios .....	48
Agregar Rutas de Medios .....	48
Creación de Fuente de Recursos de Medios ( <i>Media Resource Pool</i> ) .....	49
Creación de Fuentes de Servicios de Medios ( <i>Media Service Pools</i> ).....	51
Creación de Reinos ( <i>Realms</i> ) y Puntos Finales ( <i>Endpoints</i> ).....	52
<i>Realms</i> .....	52
Configuración de <i>Endpoints</i> .....	54
<b>Resultados Obtenidos</b> .....	<b>56</b>
Registro y llamada telefónica.....	56
Registro de <i>Endpoints</i> .....	56
Llamada Telefónica.....	59
Resultados .....	61
<b>Conclusiones</b> .....	<b>64</b>
<b>Anexo: Lista de comandos</b> .....	<b>66</b>
<b>Glosario</b> .....	<b>68</b>
<b>Bibliografía</b> .....	<b>72</b>

## **Objetivo.**

Realizar el análisis, procedimiento y establecimiento de una llamada de voz sobre IP mediante la integración de la interfaz gráfica de manejo de un *Realtime Session Manager* (RSM) y mediante la instalación y configuración de un controlador de sesión de frontera, (*Session Border Controller* o SBC).

De manera personal, la realización de este proyecto tiene la finalidad de reforzar mis conocimientos en la instalación del controlador de sesión de frontera y su integración con una interfaz gráfica, para aplicarlo a un proyecto que me permita superarme en el ámbito profesional al utilizar conceptos de ingeniería y actualizarme en las áreas de Linux y redes.

## **Marco Teórico.**

El rápido desarrollo del internet y la convergencia de comunicaciones sobre la red ha creado un campo con muchas oportunidades de crecimiento en las aplicaciones integradas, y, al mismo tiempo, de mucha competitividad en la industria de las telecomunicaciones; este crecimiento ha generado la necesidad de integrar más ingenieros con conocimientos en las áreas de diseño de redes, integración y soporte de servidores.

## **Antecedentes.**

El área de soporte es fundamental en las empresas que brindan servicios de telecomunicaciones ya que dicha área se encuentra en contacto directo con los clientes, se encarga de resolver los problemas técnicos y dar solución a las dudas y necesidades que pudieran surgir durante el uso de los servicios provistos por la empresa.

Así mismo, una de las actividades importantes del Ingeniero de soporte es su participación durante la instalación de servidores, en este caso en específico, del SBC.

El presente documento tiene como fin describir detalladamente el proceso de la instalación de un SBC, así como su integración con una interfaz gráfica RSM.

Para llevar a cabo lo antes mencionado, se requiere contar con los conocimientos teóricos que se mencionan en la introducción.

## **Introducción.**

### **Telefonía.**

La primera línea telefónica que se construyó consistía en un cable de cobre que podía comunicar punto a punto un par de dispositivos electrónicos (teléfonos). Estos dispositivos requerían una conexión directa por un cable que funcionaba como canal de comunicación. Poco tiempo después fue evidente la utilidad del teléfono como sistema de comunicación, entonces se empezó a construir en serie e implementarse a mayor escala usando cada vez más dispositivos.

Debido a la poca practicidad de conectar cada una de las terminales telefónicas entre ellas se instauró el uso de centrales telefónicas. De esta manera los costos excesivos del cableado fueron disminuidos y se estableció una conexión la cual después se conocería como tipo malla (Szymanczyk, 2013).

Gran parte del éxito y razón por el que el servicio telefónico se fue extendiendo rápidamente fue debido a la comunicación personalizada y sencilla que este sistema proporcionaba (Figueiras, 2002).

El teléfono tuvo tanto éxito que, en los Estados Unidos, se pasó de 230 teléfonos instalados a mediados de 1877 a 30 mil en marzo de 1880. Éstos fueron soportados por 138 centrales telefónicas. (Figueiras, 2002)

### **FDM.**

Debido al éxito del teléfono más y más usuarios se unieron a la red telefónica. Pero entonces empezó a surgir un problema de infraestructura. No era práctico enviar las señales de voz en una línea por cada conversación. Se debía buscar la forma de mandar varias conversaciones por un solo canal. Entonces se desarrolló la tecnología FDM (*Frequency-division Multiplexing*) o Multiplexación por división en frecuencia. El teléfono transmite únicamente un ancho de banda de entre 300 y 3,400 [Hz], este ancho de banda se puede enviar directamente a unos hilos de cobre. Si se enviara una segunda señal de voz con las mismas frecuencias, ésta

terminaría mezclándose y confundiéndose con la primera. Pero, para un segundo teléfono, a estas frecuencias se les puede sumar 4 [kHz] más, dando como resultado 4,300 y 7,400 [Hz] en un espectro de frecuencia que va de 60 a 108 [kHz], por lo que esta nueva señal se puede transmitir con la primera sin que una interfiera con la otra. Este proceso se puede repetir con una tercera o cuarta o hasta 12 señales de voz para lograr mandar varias conversaciones por un mismo canal (Huidobro, 2011).

Más adelante, la invención de los transistores, creación de microprocesadores y mejora de los sistemas digitales fueron claves en el desarrollo de la comunicación electrónica, así como en la creación de computadoras y circuitos digitales.

## **TDM.**

Antes de 1960 la transmisión por FDM de señales analógicas era predominante en las telecomunicaciones. Pero entonces llegó la tecnología TDM (*Time-Division Multiplexing*) o multiplexación por división del tiempo. En contraste a FDM, la multiplexación por división del tiempo se basa en varias señales separadas en el dominio del tiempo en vez del dominio de la frecuencia como en FDM. Otra diferencia fundamental entre las dos técnicas es que TDM es una tecnología digital a diferencia de FDM que es analógica. En el caso de TDM, varias señales son muestreadas y convertidas a cadenas de bits por medio de un convertidor analógico digital. El proceso usado para convertir las señales analógicas en digitales es PCM (*Pulse-Code Modulation*) o Modulación por impulsos codificados. La multiplexación por división del tiempo toma señales convertidas a impulsos digitales y estas cadenas son agrupadas en diferentes tiempos para cada señal de voz. Esto permite tener una cadena de bits con diferentes señales que puede ser enviada por un mismo canal. Cuando esta cadena es recibida, las señales son reordenadas basándose en las divisiones de tiempo y, con un convertidor digital analógico, reconstruida analógicamente para ser escuchada. (Hioki, 1998).

## **Redes e Internet.**

A inicio de los años setenta, el departamento de defensa de los Estados Unidos, y empresas como IBM y UNIVAC comenzaron a desarrollar técnicas y tecnología que permitiera la comunicación entre computadoras. (Figuerias, 2002).

Las primeras redes consistían en computadoras conectadas físicamente a través de cables coaxiales que transmitían información en forma de ceros y unos. De la misma manera que las redes telefónicas, con forme se fueron añadiendo más computadoras a estas redes, se encontraban nuevas dificultades que propiciaron el desarrollo de nuevos protocolos de comunicación y nuevos dispositivos de redes (*Hubs, Switches, Routers*).

Debido a que eran muchos los competidores trabajando en esta área, cada grupo de desarrollo usaba su propia tecnología, lenguaje y sistemas de comunicación. Esto hizo que buena parte de los trabajos se orientaran a la creación y uso de normas y protocolos para estandarizar y facilitar la comunicación entre las empresas competidoras (Figuerias, 2002).

Durante la década de los sesenta el departamento de defensa estadounidense (DoD) se vio en la necesidad de crear una red muy segura. Dos de las características importantes de esta red eran: que soportara la destrucción de algunos de sus nodos, por lo que la información podría viajar por otro camino y que fuera una red difícil de interceptar puesto que la información no va toda completa, una parte iría por un camino y la otra parte por otro, lo que significa que la información tendría que ser enviada y manejada en paquetes. Esto llevó a la creación del sistema ARPANET que se llegó a estandarizar en el ejército estadounidense. Llegó un momento en que el departamento de defensa decidió ceder esta red a las universidades. La cual tomó el nombre de DARPA y a principios de los 90 se pensó en darle un uso comercial y se llamó Internet (*Huidobro, 2011*).

El Internet está compuesto por la interconexión de diferentes redes. Estas redes están compuestas por líneas que conectan varias computadoras y servidores a través de *routers* y *switches*.

El número de usuarios conectados ha crecido de manera dramática al considerar que alrededor de 1985 había unos cuantos usuarios conectados, pero en el año 2005 se llegó a más de 350 millones de usuarios conectados a través de 60 mil redes unidas por 150 países. (Horak ,2007).

En el año 2018, según la página *Internet Live Stats (s.f)* el número de usuarios conectados al internet es de 4,023,102,397 y este número sigue creciendo. Esto nos puede dar una idea de cómo el Internet se ha vuelto el mayor sistema de comunicación en el planeta.

### **Modelo OSI.**

El modelo OSI consiste en un grupo de 7 capas o niveles los cuales tienen la función de modelar los sistemas de comunicaciones, describiendo y fragmentando los sistemas para un mejor análisis. Cada capa o nivel tiene una funcionalidad específica que van desde las características físicas de las conexiones hasta la interfaz con el usuario.

Nivel 1. Físico: Proporciona las características físicas (eléctricas y mecánicas) que deben tener los sistemas de comunicación para lograr una correcta transmisión de bits por un canal.

Nivel 2. Enlace: Este nivel tiene como finalidad la organización y envío de información en forma de tramas (o *frames*). Estas tramas incluyen bits dedicados a la detección de errores. De la misma manera tener un control del flujo de tramas enviadas.

Nivel 3. Red. Este nivel se encarga de proveer direcciones de red a cada uno de los dispositivos conectados a ésta. También se encarga del enrutamiento de los



paquetes y de determinar la mejor ruta para el intercambio de éstos. Para lograr lo anterior, esta capa también debe tener conocimiento de la topología de la red.

Nivel 4. Transporte. Esta capa del modelo OSI tiene la función de garantizar la entrega en secuencia correcta de los paquetes y como detectar errores en la información enviada. Su misión básicamente es optimizar el servicio presentado por el nivel de red y corregir posibles deficiencias.

Nivel 5. Sesión. Este nivel tiene el objetivo de lograr la comunicación correcta y proporcionar los servicios precisos y necesarios para lograr una conexión entre dos entidades y lograr interacciones ordenadas de transferencia de datos.

Nivel 6. Presentación. Este nivel tiene como finalidad el proporcionar los procedimientos precisos para representar de manera adecuada la información enviada y que será usada después por los dispositivos de salida del usuario (pantallas o impresoras, por ejemplo). A este nivel pertenece el código ASCII y HTML.

Nivel 7. Aplicación. Este es el último nivel de este modelo dado que es el nivel por el cual los procesos de aplicación acceden al entorno OSI. Funciona como interfaz con el usuario y proporciona los procedimientos que permiten a los usuarios ejecutar los comandos relativos a sus propias aplicaciones. Esta interfaz le permite al usuario poder tener una interacción directa con la información transmitida.

*(Huidobro, 2011).*

### **Modelo TCP/IP.**

El modelo TCP/IP es el modelo en el que se basa el Internet por lo cual tiene una gran importancia y relevancia. TCP/IP no es un protocolo único sino un modelo en capas en el cual cada capa está formada por un grupo de protocolos. Cada capa depende de la capa inmediata inferior para lograr un correcto funcionamiento. A

pesar de tener similitudes con el modelo de interconexión de sistemas abiertos de la Organización Internacional de Estandarización (OSI) tiene claras diferencias.

-El nivel inferior de este modelo lo ocupa la capa **Física**, la cual agrupa las características físicas y de hardware necesarias para lograr la transmisión correcta de bits.

-El nivel **Internet** corresponde a la interconexión de redes. En este nivel se encuentra el protocolo IP cuya principal función es proveer direcciones de red, conectividad de un extremo a otro en las redes y descripción de la topología de la red.

-El siguiente nivel es **Control de la Transmisión**, de aquí toma el nombre el protocolo TCP. Este protocolo está encargado de la entrega correcta y garantizada de los paquetes de Internet.

-El último nivel en el modelo TCP/IP es la capa de **Aplicación**. En este nivel se da el desarrollo de las aplicaciones con las que el usuario tiene contacto directo y sirve como interfaz con el usuario. (Huidobro, 2011).

### **Voz sobre IP (VoIP).**

El alcance del internet, el desarrollo tecnológico y la demanda de los usuarios ha dado pie a una nueva área en la comunicación: las aplicaciones en tiempo real. Estas aplicaciones reciben su nombre debido a que son sensibles al tiempo de respuesta de envío de los datos, éstas deben dar la sensación que la información está siendo recibida “en tiempo real”.

Algunas aplicaciones, como una página web, pueden tardar un par de segundos en cargar sin mayor conflicto para el usuario, pero en el caso de una aplicación de tiempo real, el tiempo que tarda en enviar y recibir la información afecta la experiencia y percepción del usuario. Una aplicación de la telefonía sobre Internet no puede tener los mismos tiempos de *delay* (retraso) que se le permiten a una página web sin alterar o afectar de manera drástica la comunicación. Este tipo de aplicaciones son bidireccionales e interactivas. La experiencia depende

de la respuesta de otro usuario, o de un dispositivo al otro lado de la red; este tipo de aplicaciones necesitan ser rápidas para crear una sensación de “realidad virtual”. (Dudley, Joshi, 2004).

Hasta antes de la concepción de las aplicaciones en tiempo real, todas las redes IP eran construidas con base en el “mejor esfuerzo”. Esto implica que estas redes tenían pérdidas de paquetes, cierto retraso en la comunicación y no había ningún tipo de jerarquía en los paquetes para determinar qué tan urgente era la transmisión de ciertos datos. Este paradigma era práctico y útil para las aplicaciones que no son tan sensibles al tiempo, pero estos sistemas basados en el “mejor esfuerzo” no eran suficientes para las aplicaciones de tiempo real. Esto llevó al desarrollo de redes convergentes. La idea de la convergencia de una red es integrar diferentes servicios en un mismo canal de comunicación (email, https, VoIP, etc.) Una red convergente se puede definir como la fusión del tráfico de dos o más redes en una sola donde diferentes tecnologías y protocolos interactúan para transmitir exitosamente información de diferentes servicios (aplicaciones de tiempo real y aplicaciones no sensibles al tiempo). (Dudley, Joshi, 2004).

La voz sobre IP se desenvuelve en una infraestructura totalmente diferente a la usada por los circuitos de conmutación en la telefonía clásica. La telefonía VoIP utiliza la infraestructura existente del internet y de redes ya construidas. También utiliza técnicas de compresión más avanzadas al igual que procesadores digitales de señales y códecs de voz.

A continuación, se explica de manera básica la forma en que la voz sobre IP funciona:

Este tipo de telefonía comienza con la conversión de la voz como señal analógica a información digital. Uno de los algoritmos usados en el muestreo y compresión de la voz es G.723.1, usando este algoritmo se consiguen 160 muestras, las cuales representan 125  $\mu$ s. La voz es convertida en información binaria e introducida en un paquete IP. Cada uno de los paquetes es enviado por la red de *routers* y

*switches*, tomando diferentes rutas dependiendo de cuál sea la más rápida en el momento. Al llegar al destino los paquetes son colocados en una fila y re-ordenados para ser decodificados y ser escuchados por el destinatario. Existen principalmente dos modos de conexión de telefonía de voz sobre IP entre los usuarios. De *Softphone* a *Softphone* la cual se lleva totalmente a través del Internet, o de *Softphone* a la red pública de conmutación de circuitos o *Public Switched Telephone Network (PSTN)*, en este escenario la llamada se hace por una gran porción a través del internet y después un *gateway* se encarga de conectar la llamada al servicio telefónico local. (Horak, 2007).

### **Ventajas de la voz sobre IP.**

Algunas de las ventajas que existen al utilizar voz sobre IP son:

- Las llamadas VoIP son más baratas debido a que usan la infraestructura existente del internet y no es necesario hacer un gasto extra para su implementación, por ejemplo, en gastos de cableado.
- Debido a la convergencia, ahora las llamadas VoIP pueden transmitir imágenes, datos o video (interacción de diferentes servicios).

### **Desventajas y soluciones a las desventajas.**

Sin embargo, también existen algunas desventajas que se deben considerar:

- Existen factores para generar retraso en el envío de voz sobre IP, ya que debido a la idea en la que se basó la construcción del Internet, unos paquetes pueden tomar un camino y los siguientes otros dependiendo de la congestión existente en la red, lo que puede causar *delays* en su llegada y que necesiten ser re-ordenados.
- Si los paquetes llegan con errores o no llegan a tiempo (siendo descartados) pueden generar huecos en la señal de voz.

Como se ve en los puntos anteriores las principales desventajas de las llamadas VoIP son los procesos involucrados y que la información debe ser enviada por una

red que fue construida con la filosofía del “mejor esfuerzo”. La solución a estos problemas ha surgido con el desarrollo tecnológico en la electrónica (microcontroladores, procesadores digitales de señales), programación y la convergencia de redes. Ahora es posible enviar mayor cantidad de datos por una red con mayor velocidad y menores tiempos de *delay*, la introducción de la fibra óptica ha sido de gran ayuda en cuanto a esto respecta. En el caso de los paquetes que no llegan a tiempo o llegan con errores, el desarrollo de procesadores de señales ha ayudado para la generación de algoritmos que pueden llenar esos huecos y lograr que esa pérdida sea casi imperceptible, uno de estos mecanismos es la Conciliación de Paquetes Perdidos o *Packet Loss Concealment* (PLC). Esta técnica consiste en el procesamiento de la señal para remover los “*clicks*” creados por la falta de audio y rellenar los huecos en la señal. Esta técnica abarca desde señales muy simples que requieren muy poco procesamiento a métodos muy complejos que pueden restaurar la calidad de voz casi a la señal original (Dudley, Joshi, 2004).

### **Factores que afectan la voz y métricas de calidad para la voz.**

Los factores que afectan la calidad de la voz sobre IP se pueden clasificar en 2 grupos: intrínsecas o no controlables y controlables.

Se consideran como factores controlables cuando el ingeniero a cargo de la red puede modificar los parámetros de estos factores para variar la afectación en la calidad, por otro lado, se consideran como factores no controlables aquellos cuyos parámetros están limitados y/o definidos por el equipo y la red. Existen cuatro factores que afectan en mayor medida a la calidad de la voz sobre IP: el códec de voz usado, *delay*, pérdida de paquetes y el eco.

-El códec (elemento que transforma la señal de voz a una cadena de bits y viceversa) usado tendrá una gran influencia en la calidad de la llamada, debido a la

calidad propia del códec como la respuesta de éste a otros factores como la pérdida de paquetes, ruido de fondo o *transcoding*.

-El *delay* de extremo a extremo de una señal toma en cuenta el tiempo del sonido a entrar al transmisor desde un lado de la llamada, ser codificado a una señal digital, atravesar la red y ser convertido en analógico en el otro extremo.

-La pérdida de paquetes en una red IP es bastante frecuente, pero para una llamada de VoIP esto representa un gran problema debido a la gran cantidad de información que se puede perder y que es imposible reenviar debido a la naturaleza de la información en tiempo real.

-El eco (reflexión de una señal eléctrica en un canal de comunicación que regresa al emisor de la señal) en una llamada afecta la comunicación exitosa en la llamada. Puede interferir al momento de que alguna de las dos personas se encuentra hablando, dificultando que el interlocutor termine su mensaje (Dudley, Joshi, 2004).

Para controlar estos factores y tener un estándar de calidad en lo que respecta a la voz sobre IP se han creado algunas métricas. Por ejemplo, *Mean Opinion Score* (MOS), un estimador objetivo de MOS llamada *Perceptual Evaluation of Speech Quality* (PESQ) y una métrica computada llamada *Transmission Rating* (R). El MOS es principalmente una medida subjetiva basada en pruebas aplicadas a diferentes usuarios y en su experiencia (QoE) con la prueba de voz. El parámetro PESQ funciona a través de algoritmos y puede dar una respuesta rápida y repetible de la distorsión en una señal. Sin embargo, esta medida no refleja la calidad de la voz.

El índice de transmisión (R) es una métrica objetiva que indica la calidad en general de la voz en una llamada. Utiliza 15 parámetros diferentes (incluyendo los 4 principales mencionados anteriormente) para predecir y dar un valor a la calidad de la llamada.

Estos parámetros son utilizados para medir la calidad de la llamada y la eficiencia de los datos enviados para la llamada de voz sobre IP (Dudley, Joshi, 2004).

## **Protocolos.**

El internet está basado principalmente en los protocolos *Transmission Control Protocol* (TCP) e *Internet Protocol* (IP), los cuales pueden ser identificados como componentes principales de dos de las diferentes capas del modelo TCP/IP. El protocolo TCP pertenece a la capa 4: Transporte, mientras que el protocolo IP pertenece a la capa 3: Red (Horak 2007).

De la misma forma, existe una gran variedad de protocolos usados en el modelo TCP/IP, pero los principales en la telefonía VoIP son los siguientes:

**IP (*Internet Protocol*).** El protocolo IP está definido en el artículo RFC (*Request for Comment*) 791 y pertenece a la capa 3 (Red) del modelo TCP/IP. Está encargado de proveer direcciones IP a los diferentes dispositivos conectados a la red. De la misma forma se encarga del envío de paquetes a través de la red. Es un protocolo que puede ser considerado “sin conexión” dado que los paquetes son enviados sin tener conocimiento real de la existencia del destino. De la misma manera se puede categorizar como un protocolo de datagramas dado que cada paquete no tiene “conocimiento” de que pertenece a un grupo mayor de paquetes. Se le considera como un protocolo de *mejor esfuerzo* dado que no tiene mecanismos de detección de errores ni garantiza la entrega ni la secuencia en que se entregan los paquetes (Horak, 2007).

**TCP (*Transport Control Protocol*).** El protocolo TCP pertenece a la capa 4 del modelo TCP/IP y está definido en RFC 793. Su principal función es garantizar una transmisión confiable de datos por la red solucionando las carencias de la capa inmediata inferior (Protocolo IP). Este protocolo se encarga de proveer segmentación y re-ensamblado de paquetes. También otorga una secuencia a los paquetes enviados y detección de errores. El protocolo TCP provee un mecanismo para que el destino confirme que el paquete fue recibido. En caso de que no se confirme la recepción o exista un error, el paquete vuelve a ser enviado

**UDP (*User Datagram Protocol*).** Como se define en RFC 768, este protocolo pertenece a la capa 4 (transporte), es *host-to-host* (dispositivo a dispositivo) y es un protocolo mucho más sencillo en comparación con el TCP. El protocolo UDP no ofrece confirmación de entrega del paquete o que el receptor esté disponible, al igual que el protocolo IP. Como TCP, el protocolo UDP se basa en IP para el direccionamiento y envío de información. Pero, a diferencia de TCP, el protocolo UDP no provee secuencia de paquetes, control de error o mecanismos de control de flujo. Esto le da una ventaja de ligereza y velocidad por lo que lo convierte en una mejor opción frente al TCP en funciones donde la velocidad de transmisión y recepción es muy sensible. Las aplicaciones que usan UDP asumen la responsabilidad de todos los problemas de confiabilidad, como la pérdida de datos, la integridad de la información, el retraso de paquetes, la secuencia de los paquetes y la pérdida de conectividad. Este protocolo es ampliamente usado en las aplicaciones de voz sobre IP y las aplicaciones de *streaming* (transmisión por secuencias) donde técnicas de compresión son diseñadas para disminuir los problemas antes mencionados en una red de alto tráfico de datos. El protocolo UDP también funciona bien en transacciones de corta duración (Horak, 2007).

**RTP (*Real Time Transport Protocol*).** Este protocolo es usado para el envío de información (medios) en tiempo real, por ejemplo, audio y video.

**H.323.** Es una recomendación por parte del ITU-T que define la señalización necesaria para establecer sesiones para servicios de tiempo real.

**SIP (*Session Initiation Protocol*).** La función de este protocolo consiste en el establecimiento de sesiones multimedia. Parte de la información que incluye es la localización del usuario, dirección y disponibilidad. Los mensajes usados por este protocolo fueron creados con una sintaxis similar a la HTTP.

**SDP (*Session Description Protocol*).** Este protocolo se encuentra dentro de los mensajes SIP y contiene información necesaria para establecer una correcta comunicación entre dos puntos como las direcciones IP por donde se envían los



paquetes de voz y/o video, el tipo de códec o la forma en que serán interpretados los tonos DTMF.

## **Definición del problema y contexto de participación profesional.**

### **Session Border Controller (SBC).**

El rápido crecimiento de las funciones multimedia (voz, mensajería, video, etc), ofrece nuevas oportunidades de funcionamiento y desarrollo de redes para los proveedores de servicios y empresas. Estos flujos multimedia están conectados con diferentes puntos y existen interconexiones con diferentes proveedores de servicios. El SBC (*Session Border Controller*) entra en juego en esta parte al estar diseñado para proteger flujos multimedia (tanto como de señalización como intercambio multimedia). Es colocado en los bordes o fronteras de las redes para controlar y separar estos flujos IPs y dar privacidad a las redes privadas de las públicas. El SBC es principalmente usado en las redes de voz sobre IP, SIP y H.323. Para el protocolo SIP tiene una función de servidor *Proxy* (Intermediario) y para H.323 una función de *Gatekeeper* (portero o control de acceso). Este servidor tiene un sistema operativo Linux personalizado y adaptado llamado *GBLinux* el cual está basado en la distribución SUSE.

Dado que el sistema operativo se encuentra y funciona de intermediario entre el hardware y el usuario, el servidor tiene una aplicación (*iServer*) específica de este servidor que controla y se encarga de las funciones antes mencionadas que corresponden al *Session Border Controller*. Esta aplicación se llama *iServer*. El software *iServer* tiene la función de controlar y configurar las funciones de intercomunicación, configuración SIP y H.323, ajustes globales, modificaciones de mensajes, códecs soportados, seguridad, etc.

## SBC



Imagen 1. Hardware de SBC Q10.

En este proyecto muestro los pasos requeridos para actualizar el sistema operativo usado para controlar el servidor (*Linux*) y el software utilizado por la aplicación del *Session Border Controller (iServer)*. La actualización del software tiene un papel importante debido a la corrección de errores de programación (*Bugs*), la implementación de nuevas funciones y el manejo de licencia. Una vez instalado y configurado correctamente es posible realizar una llamada telefónica mediante VoIP.

### **Análisis y metodología empleada.**

El proceso de actualización de software y configuración para manejo de VoIP lleva los siguientes pasos:

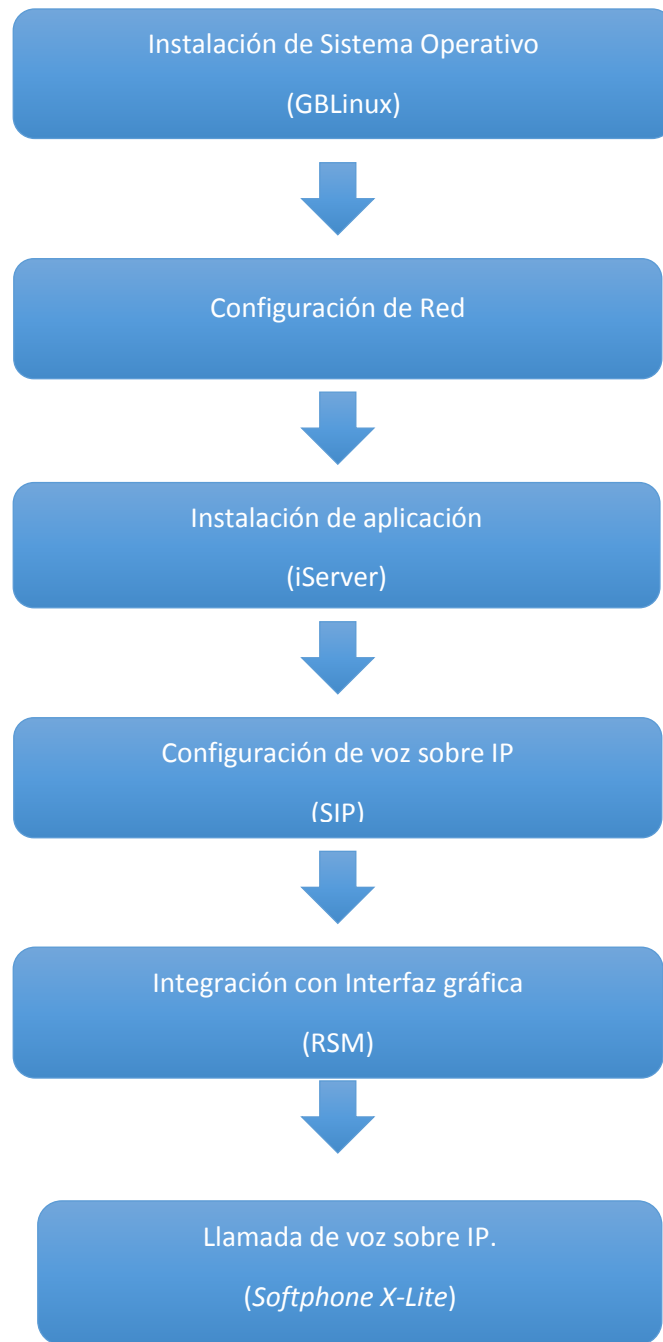


Imagen 2. Pasos de metodología empleada

**Instalación de Sistema Operativo (GBLinux).** Los sistemas operativos basados en Linux han demostrado ser de mucha flexibilidad y, dado que consumen pocos recursos del equipo, desempeña las tareas de manera eficaz y robusta. En el caso

del SBC se utilizará la versión base GBLinux 8.1.0-1sb y se aplicarán las actualizaciones necesarias hasta llegar a 9.2.0-10.

**Configuración de red.** Una vez instalado el sistema operativo se procederá a hacer la configuración de las interfaces de red. La configuración se hará usando subredes ya existentes en el laboratorio y el servidor se colocará en la frontera de dos redes.

**Instalación de aplicación (*iServer*).** Se procederá a instalar la aplicación *iServer*. Esta aplicación es la que se encargará de gestionar y controlar el manejo del SBC. Se instalará la versión 9.2.8 ya que ésta es la última versión.

**Configuración de VoIP.** La configuración de voz sobre IP se realizará utilizando el protocolo SIP (*Session Initiation Protocol*) ya que este protocolo es el de más importancia y mayor uso en la industria de voz sobre IP.

**Integración con interfaz gráfica (RSM).** Para completar la configuración del servidor se debe integrar con la interfaz gráfica (RSM) para crear las entidades lógicas por las que el SBC se conectará a otras redes. De la misma manera la interfaz gráfica ofrece una forma más amigable al usuario de provisionar el servidor.

**Llamada de voz sobre IP usando *Softphone X-Lite*.** Una vez completada la instalación y configuración del servidor se procederá a hacer una prueba para confirmar su correcto funcionamiento. Esta prueba se realizará usando el *Softphone X-Lite*. Decidí usar esta aplicación por su practicidad y por ser software libre. Para el análisis del intercambio de mensajes necesarios para establecer la llamada se utilizará la aplicación *Wireshark*.

## Participación profesional

En el puesto de ingeniero en soporte técnico (*Technical Support Engineer*) soy responsable de actividades como:

- Soporte a servidores.
- Servicios utilizados para tráfico de voz sobre IP (VoIP).
- Manejo de tonos DTMF (*Dual-Tone Multi-Frequency*)
- Problemas de hardware.
- Detección de posibles problemas de software.

También realizo la asesoría en la instalación, manejo y configuración de servidores (SBC) y la realización de pruebas de laboratorio para replicar escenarios.

Para llevar a cabo las actividades anteriores es indispensable contar con conocimientos en telefonía, redes, línea de comandos (*Linux*), dispositivos de redes, jerarquía TCP/IP y protocolos de voz sobre IP como SIP. Entre las funciones que desempeño también se encuentra el fungir como enlace con el cliente, con representantes de otras compañías, con equipos de diseño y producción a nivel global por lo que en este puesto se requiere un dominio conversacional y escrito del idioma inglés. El apoyo se brinda de manera remota usando llamadas telefónicas o correos electrónicos. En el día a día debo participar en los procesos de evaluación, simulación, reproducción de problemas, e investigación para dar una solución viable utilizando todos los medios disponibles.

## Instalación y actualización de SBC a 9.2.8

A continuación, se presenta el proceso seguido para hacer la instalación de la versión 9.2.8 de *iServer* en un SBC Q10. Este proceso empieza desde la instalación a través de una memoria USB con la imagen del sistema operativo base (*GBLinux 8.1.1.0*), se procede a la actualización del sistema operativo a 8.3 y 9.2, instalación

del *patch* correspondiente y configuración básica del servidor para lograr comunicación VoIP.

**Requerimientos de software.** Este procedimiento considera que se realizará una instalación “desde cero”, lo que significa que no tendrá ningún software previamente instalado, por lo que no hay prerequisites de Software. Sin embargo, el procedimiento a seguir para la instalación contempla los requisitos de software que cada versión del sistema operativo y parche de la aplicación.

En el caso de que se tenga un sistema operativo Linux ya instalado se puede saber qué tipo de servidor es y qué versión de *GBLinux* tiene instalado con los siguientes comandos:

```
# cat /etc/GbLinux-release ... (1)
```

Se mostrará la lista de versiones de sistemas operativos que se han instalado previamente.

De la misma manera se requerirá una licencia para poder operar el SBC. Esta licencia consiste en un archivo formato.xml que se obtiene directamente con la empresa.

**Requerimientos de hardware.** La instalación se puede realizar en servidores tipo *Annapolis*, Q10, Q20 y Q21. En este caso realicé la instalación en un Q10 que se muestra en la imagen 3.



Imagen 3. Servidor Q10.

En caso de que no se conociera el tipo de hardware que se tiene se puede usar el siguiente comando:

# **hwinfo --bios | grep Product** ... (2)

En la tabla 1 se muestran diferentes salidas a este comando. En amarillo se observa la salida para el Q10.

Product ID	Machine Type
FWA-3210	QUANTI <sup>™</sup> X Q10 is the 1U (1 rack unit)
S2600CO	QUANTI <sup>™</sup> X Q20 is the 2U (2 rack unit)
	QUANTI <sup>™</sup> X Q20T is the 2U (2 rack unit)
	QUANTI <sup>™</sup> X Q21 is the 2U (2 rack unit)

Tabla 1. Servidores compatibles con 9.1.

**Descarga de imagen de sistema operativo base.** La empresa cuenta con un servidor FTP (imagen 4) en el cual se localizan las imágenes de formato .iso de los sistemas operativos para ser instalados. Me conecté a través del software de *SFTP* (*Secure File Transfer Protocol*) *Filezilla* y descargué la versión *8.1.1.0* de *GBLinux*.

Descargué los paquetes que contienen el sistema operativo y *iServer* para *8.3*, *9.1* y *9.2* que serán utilizados más adelante.

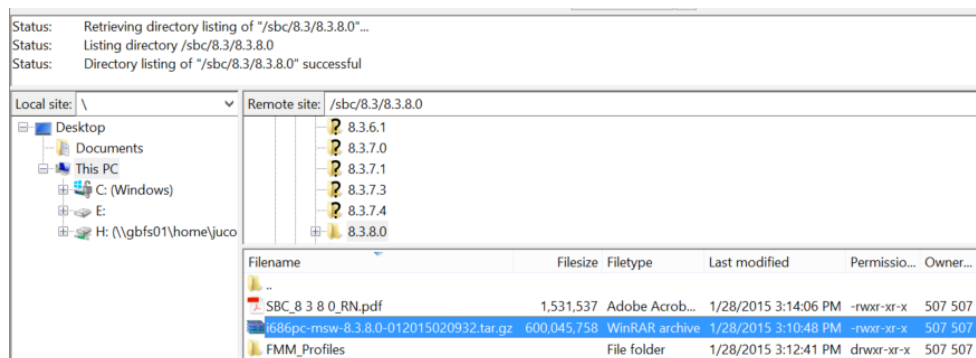


Imagen 4. Servidor FTP con sistema operativo base.

**Creación de imagen de arranque.** Una vez descargada la imagen formato .iso, procedí a instalarla en una memoria USB y darle formato *bootable* (de arranque), para esto utilicé el software libre *Rufus* versión 2.18.1213 (imagen 5):

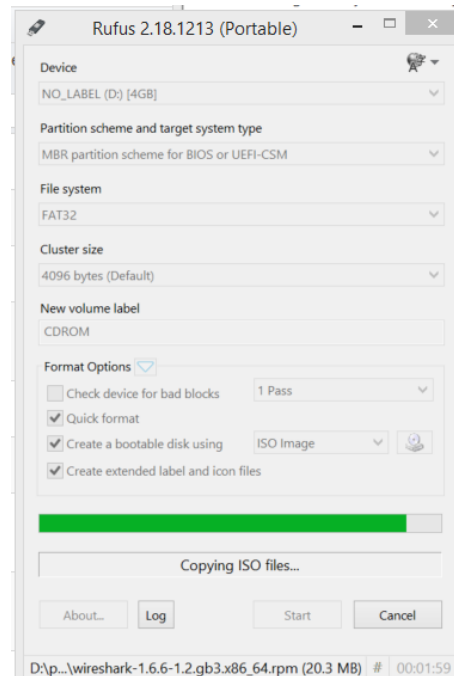


Imagen 5. *Rufus* 2.18.12.16 usado para crear la imagen booteable.

**Instalación de versión 8.1.1.0.** Para este punto conecté un teclado y monitor al SBC para poder manipular el servidor desde la consola.

Accedí al **BIOS** del servidor para habilitar la memoria USB como forma de arranque en vez del disco duro. Para acceder es necesario presionar F2 en el teclado al momento en que el servidor está encendiendo. Una vez ya en el menú **BIOS**, procedí a seleccionar *BOOT OPTION PRIORITIES* usando F4 (imagen 6).



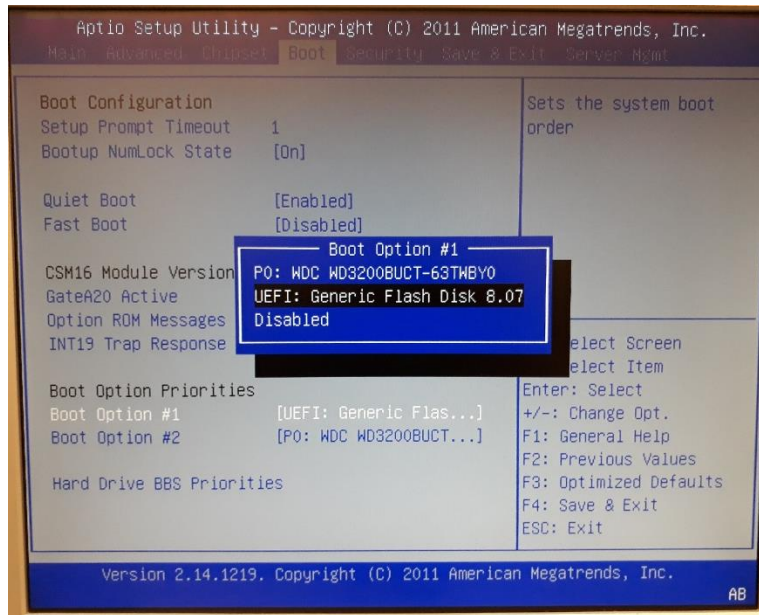


Imagen 6. Selección de memoria USB en Boot Option Priorities.

Seleccioné la imagen USB, en este caso *UEFI: Generic Flash Disk 8.07*.

En la opción *HARD DRIVE BBS PRIORITIES* seleccioné nuevamente la memoria USB como primera opción.

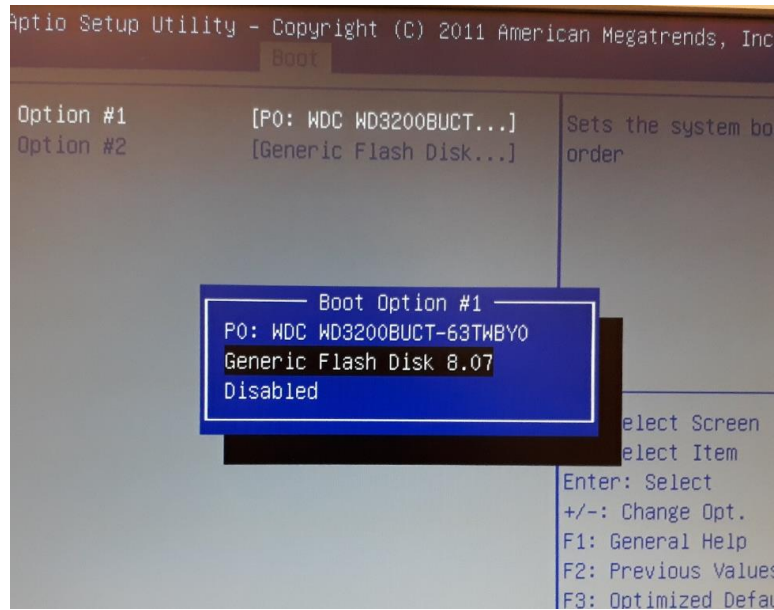


Imagen 7. Selección de USB en *HARD DRIVE BBS PRIORITIES*.

Una vez habiendo habilitado estas opciones procedí a guardar los cambios, insertar la memoria en servidor y reiniciar la unidad.

Al reiniciar la unidad, ya con la memoria USB conectada, el servidor empezó la instalación automáticamente (imagen 8).

```
SYSLINUX 4.07 EDD 2013-07-25 Copyright (C) 1994-2013 H. Peter Anvin et al
Welcome to the GENBAND SBC INSTALLER

The system will automatically start the installation.

For other options, type a command below and press <return>

Available boot options:

sbc_install      - Install Genband SBC software onto system
sbc_install_serial - Install Genband SBC software onto system (serial port)
harddisk         - Local boot from hard disk
shell            - Boot into diskless linux shell

Genband, Inc
boot:
-
```

Imagen 8. Inicio de instalación.

Durante esta primera parte de la instalación no fue requerida acción alguna. En la pantalla se mostraron las diferentes fases de instalación y, al finalizar el proceso, se presentó la siguiente pantalla (imagen 9) indicando que la instalación del sistema operativo base había sido completada. En este momento removí la memoria USB y reinicié nuevamente el servidor.

```
finished network

Installation complete sbc 8.1.1-20 Wed Feb 27 13:05:59 EST 2013 sandybridge1u
Setting up swap space version 1, size = 6250492 KiB
no label, UUID=dc0bc3ac-deba-4ab7-b7a8-ae8ad3d3c7e9

sbc install complete

To reboot from hard disk, remove bootable USB stick, and hit control-alt-delete
```

Imagen 9. Pantalla mostrando la instalación completa.

Al iniciar el servidor entró a un ambiente Linux (*GBLinux*). Confirmé la versión del sistema operativo usando el comando (1). (imagen 10).

```
msx-02:~ # cat /etc/GbLinux-release
GENBAND GbLinux 8.1.0-1-sb (x86_64)
VERSION = 8.1.0-1-sb
PATCHLEVEL = 0

gblinux 8.1.0-1-sb build date: 2013/02/27
```

Imagen 10. Salida del comando mostrando la versión del sistema operativo.

**Configuración de interfaces usando YaST.** Una vez que confirmé que el SBC tenía el sistema operativo base instalado continué con la configuración de las interfaces de red a través de YaST. Para acceder simplemente tecleé *yast* y presioné *enter*:

**# yast** ...(3)

Se mostró un menú con varias opciones de configuración. Para configurar las interfaces de red seleccioné *Network Devices* y en seguida *Network Settings*, como se muestra en la imagen 11:

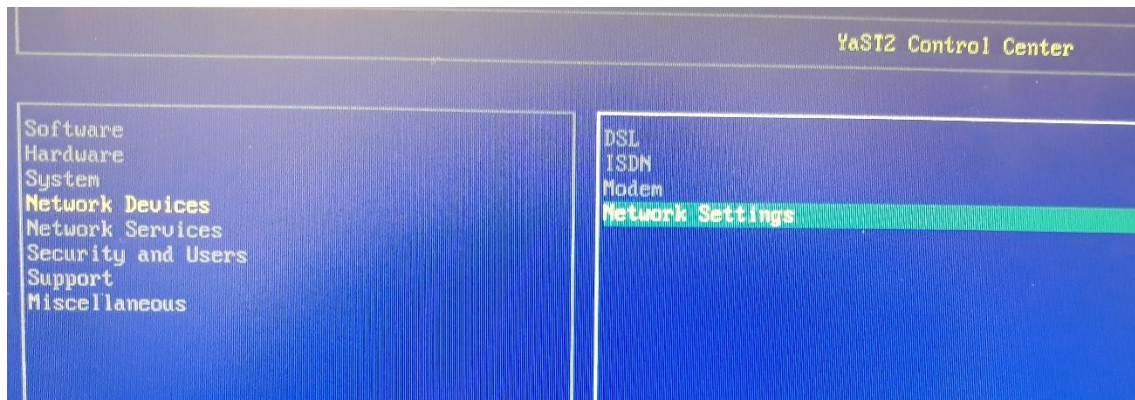


Imagen 11. Menu de *yast* para la configuración de red.

En la pestaña *Hostname/DNS*, en el campo *Hostname* tecleé el nombre del equipo: *msx-02* (imagen 12).

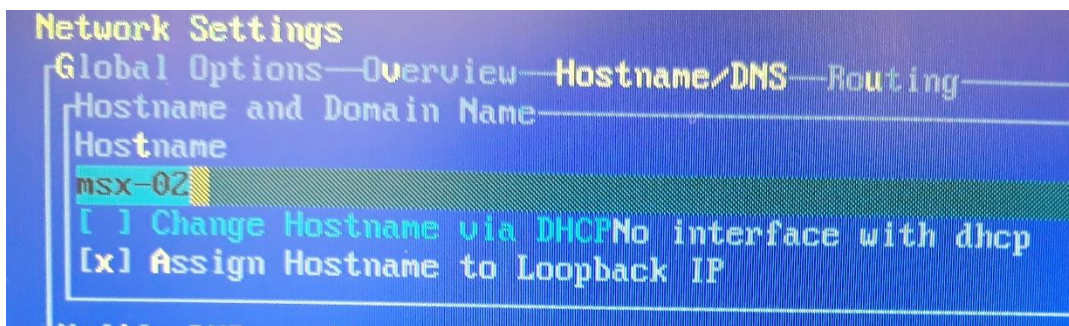


Imagen 12. Se define el nombre del servidor.

Procedí a configurar las interfaces de red del SBC. Para esto, usando la tecla TAB me dirigí a la pestaña *Overview*, y con las flechas seleccioné la interfaz a configurar. En este caso configuré *Eth0 (MGMT)*, *Eth2 (Public Access RSA)* y *Eth3 (Private Access RSA)* como se muestra en la imagen 13. Las direcciones IP y máscaras a utilizar dependen de la configuración de la red en que se encuentre el dispositivo.

Network Settings

Global Options — Overview — Hostname/DNS — Routing

Name	IP Address
82574L Gigabit Network Connection	172.24.29.152
82574L Gigabit Network Connection	Not configured
82574L Gigabit Network Connection	172.24.29.186
82574L Gigabit Network Connection	172.24.29.170
82574L Gigabit Network Connection	Not configured
82574L Gigabit Network Connection	Not configured
82574L Gigabit Network Connection	Not configured
82574L Gigabit Network Connection	Not configured

Imagen 13. IPs de las interfaces a usar.

Es importante mencionar que la interfaz Eth1 se usa cuando la unidad está en modo de *High availability* (conexión entre 2 servidores para proveer redundancia, esto es, si un servidor falla, el par entra a suplir las funciones del primero), y su función es comunicar ambos servidores. En este caso que configuré un solo servidor, no la usé.

Una vez configuradas las interfaces definí el *Default Gateway* (Puerta de Enlace) a utilizar. Para esto me dirigí a la pestaña *Routing* e introduje la IP del *Default Gateway* (imagen 14).

```

Default IPv4 Gateway
172.24.29.145
Default IPv6 Gateway

```

Imagen 14. Default Gateway de SBC.

Finalmente salí del submenú *Network Settings* y en *Set Clock and Time Zone* y seleccioné la hora y región en que estoy ubicado (imagen 15).

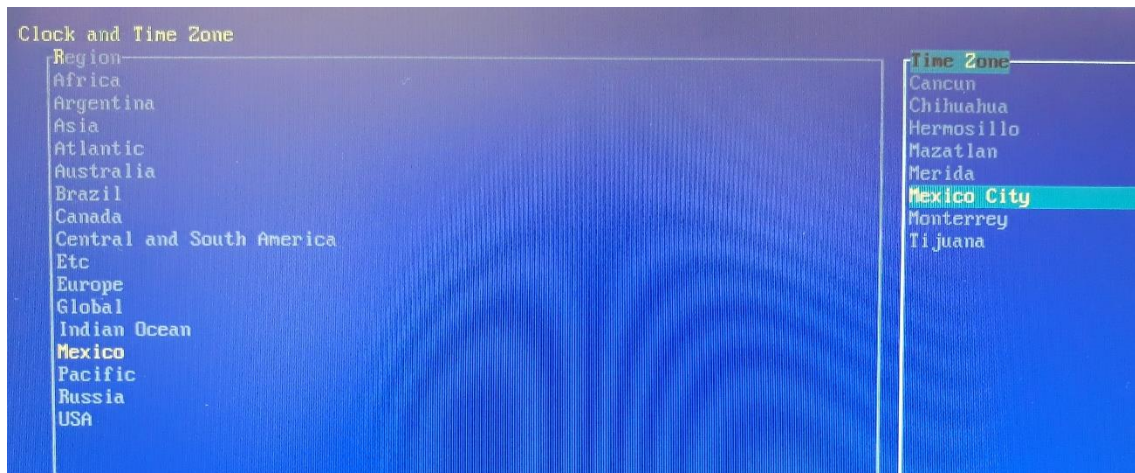


Imagen 15. Definición de hora y huso horario.

Guardé los cambios, salí de YaST y reinicié el servidor. Volví a entrar a YaST para confirmar que todo estuviera bien configurado. Después de configurar las interfaces de red procedí a hacer el resto de la instalación a través de una conexión SSH.

### Actualización a **GBLinux 8.3 y 9.2.**

Para llegar de *8.1.0-1-sb* a *9.2.0-10* es necesario pasar por algunas actualizaciones del sistema operativo (*master updates*). Al revisar *Release notes* (Notas con información de la versión del software) de la versión 9.2.8 se observa la siguiente información en los prerrequisitos (Tabla 2):

9.2.8.0 Upgrade Prerequisites (8.3)			
Platform	SBC Application	Operating System	Media Processor
Annapolis	8.3.25.0	gblinux-master-update-8.3.0-75	hk-8.3-36_sb.x86_64.rpm
Q10	8.3.25.0	gblinux-master-update-8.3.0-75	hk-8.3-36_sb.x86_64.rpm
Q20	8.3.25.0	gblinux-master-update-8.3.0-75	hk-8.3-36_sb.x86_64.rpm

Tabla 2. Pre-requisitos de sistema operativo para instalación de 9.2.8.

Se puede observar que se requiere tener al menos una versión 8.3.0-75 en el sistema operativo. Debido a esto realicé una serie de actualizaciones (“saltos”) del

sistema operativo base (*8.1.0-1-sb*) para poder llegar hasta la versión deseada. Estos saltos serán descritos a continuación:

Como se mostró anteriormente, el SBC se encontraba en una versión base *8.1.0-1-sb*. Al observar los *Release Notes* de la versión *8.3.6.0* (tabla 3), versión donde se encuentra la actualización del sistema operativo a *8.3.0-75*, se aprecia que se requiere un salto más en nuestro proceso. Se debe instalar la actualización *gblinux-master-update-8.1.1-6-sb*.

8.3.6.0 Prerequisites			
QUANTiX Platform	SBC Application	Operating System	Media Processor
Annapolis	8.1.2.0	<i>gblinux-master-update-8.1.1-3</i>	hk-8.0-44.x86_64.rpm
Annapolis	8.2.1.0	<i>gblinux-master-update-8.2.0-17</i>	hk-8.2-15_sb.rpm
S3 ATCA	The S3 ATCA platform is not supported in release 8.3. GENBAND recommends that S3 ATCA customers continue to run on release 8.1.1		
Q10/Q20	Release 8.1.1.0 or higher	<b><i>gblinux-master-update-8.1.1-6-sb</i></b>	hk-8.0-52sb.x86_64.rpm

Tabla 3. Pre-requisitos para upgrade a 8.3.6.

El camino de actualizaciones se explica en la imagen 16.

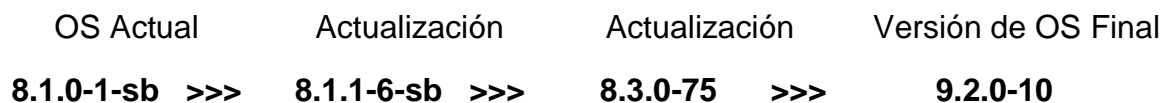


Imagen 16. Saltos entre actualizaciones para llegar a la versión deseada.

## Instalación de 8.1.1-6-sb y saltos requeridos para llegar a 9.2.0-10

Descargué el archivo de instalación 8.1.1-6-sb del servidor FTP mencionado anteriormente y creé un directorio (imagen 17) para cargar el archivo en el SBC (La carga la realicé también con un servidor FTP).

# **mkdir <nombre del directorio a crear>** ... (4)

```
msx-02:/home/genband/builds # mkdir 8.1.1.0-6-sb
```

Imagen 17. Directorio creado en el servidor.

Descomprimí el archivo en este directorio usando el siguiente comando:

# **tar -xzf <archivo comprimido>** ... (5)

Y una vez que se tuvieron los archivos descomprimidos usé el comando (5) para mostrar el contenido de ese directorio (imagen 18):

# **ls -lh** ... (6)

```
msx-02:/home/genband/builds/8.1.1.0-6-sb # ls -lh
total 479M
-rwxr-xr-x 1 root root 55M Nov 2 2012 gblinux-master-update-8.1.1-6-sb
-rw-r--r-- 1 root root 112M Nov 2 2012 hk-8.0-52sb.x86_64.rpm
-rw-r--r-- 1 root root 204M Feb 8 12:06 i686pc-msw-8.1.1.0-sb-110212025536.tar.gz
-rw-r--r-- 1 root root 9.5M Nov 2 2012 install.tar
-rw-r--r-- 1 root root 550K Nov 2 2012 iserveradm-install.tar
-rw-r--r-- 1 root root 100M Nov 2 2012 iserverinstall.tar
```

Imagen 18. Contenido del archivo comprimido.

En este directorio se encuentran, aparte del *master update 8.1.1-6-sb*, los archivos necesarios para instalar el software requerido para la tarjeta de medios y para la aplicación del SBC, pero, como mencioné anteriormente, realicé solamente la instalación de las actualizaciones de software y la instalación de medios y la aplicación la hice al final del proceso.



Para instalar la actualización *master update 8.1.1-6-sb* corré el siguiente comando:

```
# ./<script a ejecutar> ... (7)
```

```
msx-02:/home/genband/builds/8.1.1.0-6-sb # ./gblinux-master-update-8.1.1-6-sb
GBLinux updater
```

Imagen 19. Ejecución de master update 8.1.1-6-sb.

Al terminar la instalación de la actualización confirmé que el servidor se encontrara en la versión instalada usando el comando (1). Se puede observar que realizó una instalación más (*gblinux 8.1.0-14-sb*) para después llegar a la *8.1.1-6-sb*.

```
msx-02:/home/genband/builds # cat /etc/GbLinux-release
GENBAND GBLinux 9.2.6-2 (x86_64)
VERSION = 9.2.6-2
PATCHLEVEL = 0

gblinux 8.1.0-1-sb build date: 2013/02/27

gblinux 8.1.0-14-sb build date: 2012/11/01

gblinux 8.1.1-6-sb build date: 2012/11/01
```

Imagen 20. Lista de actualizaciones del sistema operativo hasta la versión 8.1.1-6-sb.

Ya teniendo *8.1.1-6.ssb* instalado, se puede proceder a instalar *8.3.6.0*.

En este caso seguí con el mismo proceso: descargué la versión *8.3.6.0* por servidor FTP, creé un directorio en el SBC donde descomprimí los archivos (imagen 21).

```
msx-02:/home/genband/builds/8.3.6.0 # ls -lh
total 1.2G
drwxr-xr-x 3 root root 4.0K Feb  8 12:40 audiocode
-rwxrwxrwx 1 root root 479M Feb  8 12:25 gblinux-master-update-8.2.2-9
-rwxrwxrwx 1 root root 297M Feb  8 12:24 gblinux-master-update-8.3.0-31
-rwxrwxrwx 1 root root 120M Feb  8 12:14 hk-8.3-18_sb.x86_64.rpm
drwxr-xr-x 2 root root 4.0K Feb  8 12:42 install
-rwxrwxrwx 1 root root 9.5M Feb  8 12:12 install.tar
-rwxrwxrwx 1 root root 720K Feb  8 12:12 iserveradm-install.tar
-rwxrwxrwx 1 root root 223M Feb  8 12:17 iserverinstall.tar
-rwxr-xr-x 1 root root 612 Oct  3 2014 setup
```

Imagen 21. Contenido de paquete 8.3.6.0

En este archivo se encontraban dos actualizaciones para el sistema operativo. Procedí a instalar *gblinux-master-update-8.2.2-9* y después *gblinux-master-update-8.3.0-31*. Confirmé la versión instalada en el servidor usando el comando (1) nuevamente.

```
msx-02:/home/genband/builds # cat /etc/GbLinux-release
GENBAND GbLinux 9.2.6-2 (x86_64)
VERSION = 9.2.6-2
PATCHLEVEL = 0

gblinux 8.1.0-1-sb build date: 2013/02/27
gblinux 8.1.0-14-sb build date: 2012/11/01
gblinux 8.1.1-6-sb build date: 2012/11/01
gblinux 8.1.2-3-sb build date: 2014/01/13
gblinux 8.2.0-18-sb build date: 2014/01/13
gblinux 8.2.2-9-sb build date: 2014/01/13
gblinux 8.3.0-31 build date: 2014/09/15
```

Imagen 22. Lista de actualizaciones del sistema operativo hasta 8.3.0-31.

Ahora el SBC tenía una versión de software compatible para poder instalar 9.2.8.0. Descargué el archivo de instalación para 9.2.8.0. De la misma manera creé un directorio en el SBC, descomprimí este archivo y revisé su contenido con el comando (5). Este archivo contenía un subdirectorio llamado *patch\_9.2.8.0*. Al

acceder a este directorio encontré los archivos de instalación necesarios (imagen 23).

```
msx-02:/home/genband/builds/9.2.8 # ls -lh
total 2.5G
-rw-r--r-- 1 root root 2.5G Feb 8 16:51 i686pc-msw-9.2.8.0-040517045713.tar.gz
drwxr-xr-x 4 506 users 4.0K Feb 8 17:41 patch_9.2.8.0
msx-02:/home/genband/builds/9.2.8 # cd patch_9.2.8.0/
msx-02:/home/genband/builds/9.2.8/patch_9.2.8.0 # ls -lh
total 2.8G
drwxr-xr-x 3 root root 4.0K Feb 8 17:41 audiocode
-rwxr-xr-x 1 506 users 2.2G Feb 17 2017 gblinux-master-update-9.2.6-2
-rw-r--r-- 1 506 users 125M Apr 5 2017 hk-9.2.7-1_sb.x86_64.rpm
-rw-r--r-- 1 506 users 32M Apr 5 2017 imedia-q21-9.2.7-1.x86_64.rpm
drwxr-xr-x 2 root root 4.0K Feb 8 17:41 install
-rw-r--r-- 1 506 users 49M Apr 5 2017 install.tar
-rw-r--r-- 1 506 users 810K Apr 5 2017 iserveradm-install.tar
-rw-r--r-- 1 506 users 383M Apr 5 2017 iserverinstall.tar
-rwxr-xr-x 1 root root 344 Dec 9 2016 setup
msx-02:/home/genband/builds/9.2.8/patch_9.2.8.0 # █
```

Imagen 23. Contenido de directorio patch\_9.2.8.0

Instalé la actualización del sistema operativo usando el comando (7). En la imagen 24 se observan las versiones que la actualización *gblinux-master-update-9.2.6-2* hizo para llegar finalmente a 9.2.6-2, versión de sistema operativo deseada para la instalación de *iServer 9.2.8.0*.

```
gblinux 8.3.0-31 build date: 2014/09/15
gblinux 8.3.0-75 build date: 2017/03/21
gblinux 9.1.0-30 build date: 2016/01/14
gblinux 9.2.0-1 build date: 2016/04/08
gblinux 9.2.0-10 build date: 2016/05/02
gblinux 9.2.1-3 build date: 2017/02/14
gblinux 9.2.2-3 build date: 2017/02/14
gblinux 9.2.3-2 build date: 2017/02/14
gblinux 9.2.4-2 build date: 2017/02/14
gblinux 9.2.5-1 build date: 2017/02/14
gblinux 9.2.6-2 build date: 2017/02/14
```

Imagen 24. Actualizaciones del sistema operativo hasta llegar a 9.2.6-2. Versión deseada.

## Instalación de paquetería de medios y aplicación del SBC (*iServer 9.2.8.0*)

Una vez que el servidor se encontraba con la versión del sistema operativo requerida procedí a instalar la paquetería de medios y la aplicación *iServer*, la aplicación que maneja las funciones del SBC.

Para instalar la paquetería de medios usé el siguiente comando:

```
# rpm -ivh <paquetería de medios> ... (8)
```

```
msx-02:/home/genband/builds/9.2.8/patch_9.2.8.0 # rpm -ivh hk-9.2.7-1_sb.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
 1:hk-9.2.7-1_sb ##### [100%]
Cavium nic2e (cn6335 Pass 2.2) card is detected
```

Imagen 25. Instalación de paquetería de medios.

Comprobé que la paquetería de medios fuera exitosa usando el comando (9):

```
# rpm -q hk ... (9)
```

```
msx-02:/home/genband/builds/9.2.8/patch_9.2.8.0 # rpm -q hk
hk-9.2.7-1_sb.x86_64
```

Imagen 26. Confirmación de correcta instalación de paquetería de medios.

Ahora, finalmente, procedí a instalar la versión 9.2.8.0 del SBC.

En el directorio */home/genband/builds/9.2.8/patch\_9.2.8.0* se puede observar el archivo *install.tar*.

```
msx-02:/home/genband/builds/9.2.8/patch_9.2.8.0 # ls -lh
total 2.8G
drwxr-xr-x 3 root root 4.0K Feb 8 17:41 audiocode
-rwxr-xr-x 1 506 users 2.2G Feb 17 2017 gblinux-master-update-9.2.6-2
-rw-r--r-- 1 506 users 125M Apr 5 2017 hk-9.2.7-1_sb.x86_64.rpm
-rw-r--r-- 1 506 users 32M Apr 5 2017 imedia-q21-9.2.7-1.x86_64.rpm
drwxr-xr-x 2 root root 4.0K Feb 8 17:41 install
-rw-r--r-- 1 506 users 49M Apr 5 2017 install.tar
-rw-r--r-- 1 506 users 810K Apr 5 2017 iserveradm-install.tar
-rw-r--r-- 1 506 users 383M Apr 5 2017 iserverinstall.tar
-rwxr-xr-x 1 root root 344 Dec 9 2016 setup
```

Imagen 27. Contenido de directorio patch\_9.2.8.0.

Usando el comando 5 descomprimí ese archivo. Dentro de este archivo hay un directorio llamado *install*. Dentro de ese directorio se encuentra la rutina (*script*) necesario para la instalación. En la imagen 28 se observa este *script* al final de la captura de pantalla.

```
msx-02:/home/genband/builds/9.2.8/patch_9.2.8.0/install # ls -lh
total 41M
-rwxr-xr-x 1 root root 20K Apr 5 2017 CommonFuncs.pm
-rwxr-xr-x 1 root root 22K Apr 5 2017 CommonFunctions.sh
-rwxr-xr-x 1 root root 1.4K Apr 5 2017 CreateSecureSessionWithoutPasswd.sh
-rwxr-xr-x 1 root root 1.7K Dec 9 2016 InsertDynEps.sh
-rwxr-xr-x 1 root root 5.6K Dec 9 2016 LICENSE
-rwxr-xr-x 1 root root 7.7K Dec 9 2016 Setup.sh
```

Imagen 28. Contenido de directorio install.

Ejecuté este script usando el comando (7) se mostró la siguiente pantalla (imagen 29):

```
GENBAND Installer v0.2
Copyright (c)1998-2018, GENBAND Inc.

This program is used to install, uninstall or upgrade
various licensed GENBAND software products.

sandybridge1u

GENBAND Setup Utility. v0.5

1.      Install a version of software.
2.      Uninstall a version of software.
3.      Upgrade a version of software.

q.      Exit this utility.

Select choice by number or type q to quit: 1
```

Imagen 29. Menú de instalación de *iServer*.

Seleccioné la opción 1 *Install a versión of software*. A continuación, se mostraron en la pantalla las condiciones de uso del software y licencia. Al final del texto se preguntaba si se aceptaban estos términos a lo que inserté Y (Yes) (imagen 30) para proseguir.

```
Do you agree to the above license terms? [y/n]:
```

Imagen 30. Mensaje de aceptación de términos.

A continuación, el sistema solicitó crear una contraseña para la base de datos del sistema (imagen 31). Es importante mencionar que esta contraseña es diferente a la que se usa para acceder al sistema operativo.

```
Enter a new password for the iServer database user and press <Enter>:  
Retype the password you entered for the iServer database user and press <Enter>
```

Imagen 31. Creación de contraseña.

Una vez que definí la contraseña, requerí especificar qué tipo de sistema se estaba instalando: *stand alone* o *cluster*. Como mencioné en la sección de la definición de las IPs de las interfaces, esta instalación se está haciendo en *stand alone*, por lo que seleccioné la opción *a*.

```
iServer Peering Configuration:  
-----  
  
Is this system standalone or part of a cluster? [stand alone]:  
    a) => stand alone  
    b) => cluster  
Valid values are displayed above  
Select your choice: a
```

Imagen 32. Selección de tipo de sistema.

En el siguiente punto se debe definir la IP de manejo del servidor. El SBC soporta el uso del protocolo IPv6. Si éste se requiriera, se puede usar una IP versión 6 dando *Enter* en la sección de IPv4 y así llegando a la sección IPv6. Dado que yo no requería usar una IP versión 6 proseguí a insertar una IP versión 4 y saltar el paso de introducir una IP versión 6 dando *Enter*.

```
Enter the management IP address []: 172.24.29.152
```

Imagen 33. Definición de IP de manejo.

Inmediatamente apareció una pantalla mostrando un resumen de la información que había sido introducida. Confirmé la información y los guardé dando *Enter*.

```
-----  
Attribute Name                               Value  
-----  
'server-type'                               'disabled'  
'mswname'                                    'msx-02'  
'mgmt-ip'                                    '172.24.29.152'  
'fwname'                                     'MS'  
'internalifs'                               'all'  
-----  
Do you want to commit the changes(y/n)? [y]: y
```

Imagen 34. Resumen de parámetros definidos durante la instalación.

La instalación continuó por un breve momento guardando la información ingresada y finalmente se mostró un mensaje confirmando que la instalación se había logrado exitosamente.

```
Iserver Admin package Installation Complete !!  
Hit [CR] to continue...
```

Imagen 35. Instalación completada.

Regresé al menú de la imagen 29 y presioné *q* para salir.

Para completar la instalación reinicié el servidor. Una vez que volvió el servidor confirmé la versión de *iServer* instalada usando el siguiente comando.

# ***gis -v*** ...(10)

```
msx-02:~ # gis -v  
GENBAND GIS Directory Server v9.2.8.0, 04-05-2017  
Copyright (c) 1998-2017 GENBAND Inc.
```

Imagen 36. Versión de *iServer* instalada.



Y para hacer funcional el SBC instalé la licencia. Para esto primero transferí la licencia al SBC usando un cliente SFTP como hice para cargar los archivos de instalación.

Agregué la licencia al directorio: `/opt/nextone/bin`

```
msx-02:/opt/nextone/bin # ls -lhr | grep GB*  
-rw-r--r-- 1 root root 1.2K Nov 17 02:41 GB_Mexico_Lab3_SBC1_92_iServer_timeless.xml
```

Imagen 37. Licencia colocada en directorio `/opt/nextone/bin`.

Instalé la licencia usando el siguiente comando:

```
# nxconfig.pl -L ... (11)
```

## **Integración con RSM (*Real-time Session Manager*).**

El RSM (*Realtime Session Manager*) es un servidor que tiene la finalidad de administrar los SBCs. La forma de administración se hace a través de una interfaz visual (*GUI*) a la que se puede acceder a través de un navegador.

Existen dos versiones de este servidor: La versión completa con todas las herramientas de administración y reportes con una caja dedicada (RSM) y el RSM-*Lite*, este último es una versión más ligera, con las funciones básicas y que puede ser instalada en el mismo hardware del SBC.

En el laboratorio que se encuentra en la ciudad de México se tiene un RSM con su servidor dedicado, por lo que procederé a explicar cómo realicé la integración del SBC recién instalado con el RSM.

# 1. Integración de SBC con RSM

Primero accedí a la interfaz gráfica a través de la IP del RSM.

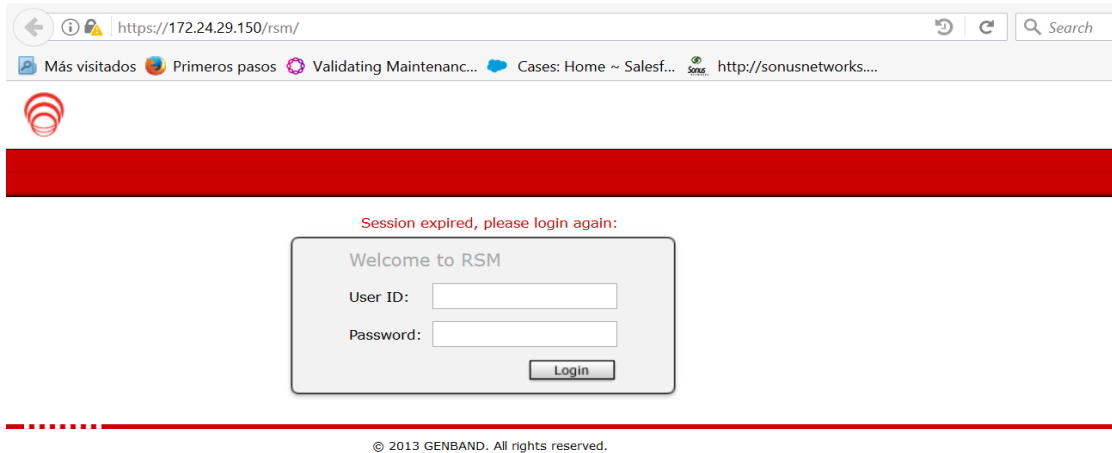


Imagen 38. Acceso a RSM.

Una vez dentro de la interfaz procedí a dar *click* en la pestaña *Devices* (Dispositivos). Esta pestaña muestra el estado de la conexión con el servidor y sincronización de los servidores con el RSM.

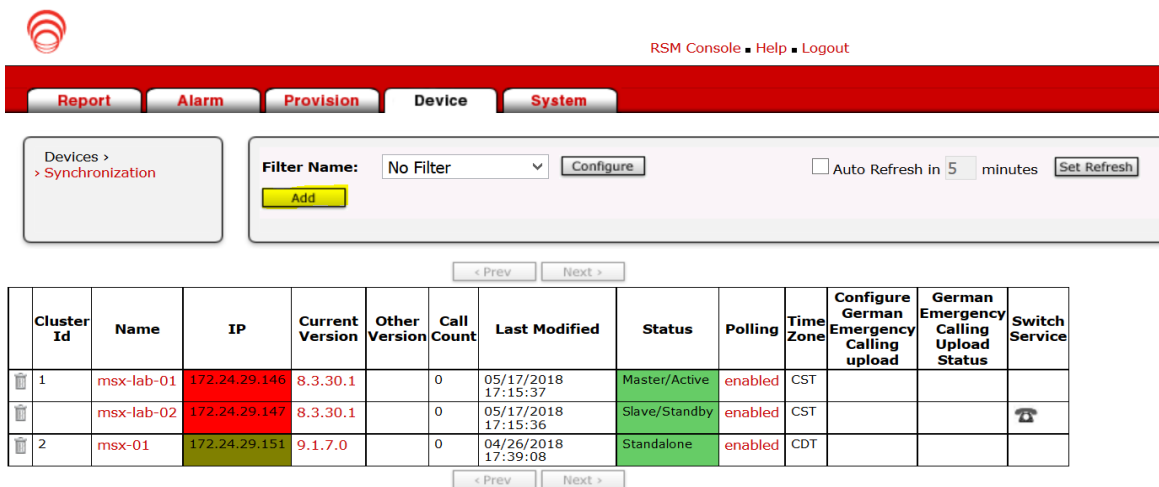


Imagen 39. Tabla que muestra los servidores ya agregados al RSM.

Procedí a dar *click* en *Add* para agregar el SBC. En la siguiente ventana agregué la información del SBC necesaria para integrarlo: nombre, IP, contraseña de manejo, contraseña de base de datos, habilité la opción de *SNMP Polling* para poder obtener información del comportamiento del SBC a través de SNMP, y las demás opciones las dejé como estaban predefinidas.

Finalicé dando *click* en *Submit*.

The screenshot shows the 'Add Device' configuration page. The navigation bar at the top is red with white text for the tabs: Report, Alarm, Provision, Device, and System. The main content area is divided into two panels. The left panel shows a breadcrumb trail: 'Devices > > Synchronization'. The right panel is titled 'Add Device' and contains the following fields:

- Cluster Id: --new-- (dropdown)
- Name: msx-02 (text)
- IP Address: 172.24.29.152 (text)
- CLI Code: (text)
- Description: (text)
- Msx Password: (password)
- SWM Password: (password)
- Autosync Master: NONE (dropdown)
- SNMP Version: v2c (dropdown)
- SNMP Retries: 0 (text)
- SNMP Timeout: 20 (text)
- SNMP Polling : enabled (dropdown)
- Device Polling : Disabled (dropdown)

Below the form is the 'SNMP V2 Configuration' section with the following field:

- SNMP Read Community: public (text)

At the bottom of the form are two buttons: 'Submit' and 'Back'.

Imagen 40. Ventana en que se agregó el SBC msx.02.

Después de agregar el SBC volví a la pantalla anterior. Bajo la columna de *Status* se podía observar que el SBC ya era reconocido por el RSM, pero estaba completando la sincronización, por lo que aparecía en color amarillo. Después de uno o dos minutos el color cambió a verde indicando que la unidad se había agregado exitosamente.

Cluster Id	Name	IP	Current Version	Other Version	Call Count	Last Modified	Status	Polling	Time Zone	Configure German Emergency Calling upload	German Emergency Calling Upload Status	Switch Service
1	msx-lab-01	172.24.29.146	8.3.30.1		0	05/17/2018 17:15:37	Master/Active	enabled	CST			
	msx-lab-02	172.24.29.147	8.3.30.1		0	05/17/2018 17:15:36	Slave/Standby	enabled	CST			☎
2	msx-01	172.24.29.151	9.1.7.0		0	04/26/2018 17:39:08	Standalone	enabled	CDT			
3	msx-02	172.24.29.152	9.2.8.0		0	05/28/2018 20:14:05	Standalone	enabled	CDT			

Imagen 41. El SBC msx-02 se agregó correctamente. Ya aparece en la tabla.

## 2. Configuración de Interfaces.

Una vez que agregué el SBC al RSM procedí a configurar las interfaces y dispositivos virtuales necesarios para establecer comunicación VoIP a través del servidor.

### a. Creación de Vnets.

Una *Vnet* (*Virtual Network*) se puede definir como la combinación de una interfaz física, una dirección IP de *Gateway* (*Router*) y una *VLAN* (*Virtual Local Area Network*) opcional. Una *Vnet* puede ser de medios o de señalización.

Primero agregué las *Vnets* **pub\_sig\_vn** y **pri\_sig\_vn**:

```
# cli vnet add <vnetname> ipver 4
```

...(11)

```
msx-02:~ # cli vnet add pub_sig_vn ipver4
```

```
msx-02:~ # cli vnet add pri_sig_vn ipver4
```

Imagen 42. Creación de *vnets*.

Después procedí a asignar una interfaz física a las *Vnets* recién creadas (los pasos descritos a continuación los apliqué de la misma manera para *pri\_sig\_vn* utilizando la interfaz eth3 y el *gateway* 172.24.29.161):

```
# cli vnet edit <vnetname> ifname <interface name> ... (12)
```

```
msx-02:~ # cli vnet edit pub_sig_vn ifname eth2
```

Imagen 43. Asignación de interfaz física a *vnets* ya creadas.

Y finalmente asigné el *gateway* correspondiente:

```
# cli vnet edit <vnetname> primary-gateway <primary gateway IP> ... (13)
```

```
msx-02:~ # cli vnet edit pub_sig_vn primary-gateway 172.24.9.177
```

Imagen 44. Asignación de *Gateway* a *vnets* ya creadas.

Usé el comando 14 para ver un resumen de las *Vnets* creadas y su configuración:

```
# cli vnet list
```

...(14)

```
msx-02:~ # cli vnet list
Vnet                pri_sig_vn
Main Interface      eth3
FwZone              def_sig_zone
VLAN Id             None
Routing Table Id    2
Primary Gateway     172.24.29.161
Secondary Gateway   0.0.0.0
IP Version          IPv4
Arp Monitoring      disabled
Gateway Monitoring  None

Vnet                pub_sig_vn
Main Interface      eth2
FwZone              def_sig_zone
VLAN Id             None
Routing Table Id    1
Primary Gateway     172.24.29.177
Secondary Gateway   0.0.0.0
IP Version          IPv4
Arp Monitoring      disabled
Gateway Monitoring  None

2 VNET entries
```

Imagen 45. Resumen de características de *vnets*.

### 3. Configuración de interfaces a través de FCE (*Firewall Control Entity*).

FCE significa *Firewall Control Entity*. Es una sección en el *RSM* cuya función es configurar la señalización y medios de las interfaces. Para configurarla seguí los siguientes pasos.

#### Acceso a RSM y FCE.

Para acceder a la interfaz gráfica del *RSM* primero escribí la IP del servidor en un navegador, con lo que obtuve la pantalla de acceso.

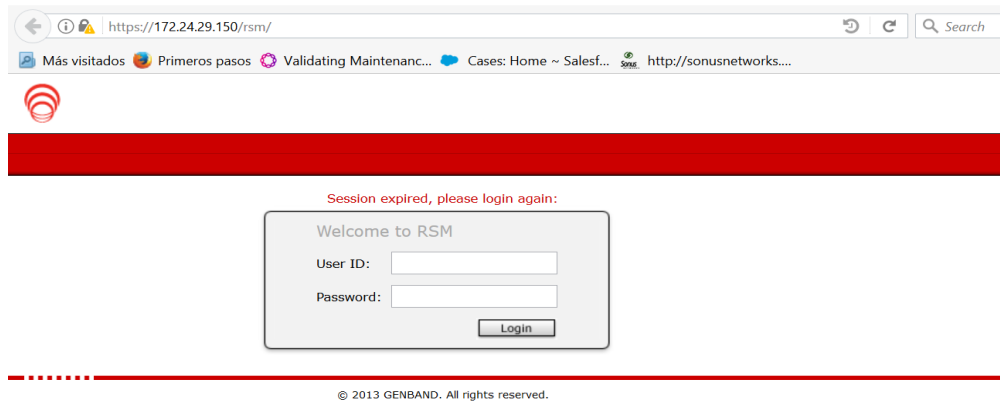


Imagen 46. Acceso a RSM.

Para acceder utilicé las contraseñas de manejo del servidor (las mismas se usan para acceder a través de una conexión *SSH*).

Una vez dentro di *click* en la opción *RSM Console*. Una nueva ventana apareció donde mostraba el *SBC* que anteriormente ya había agregado.

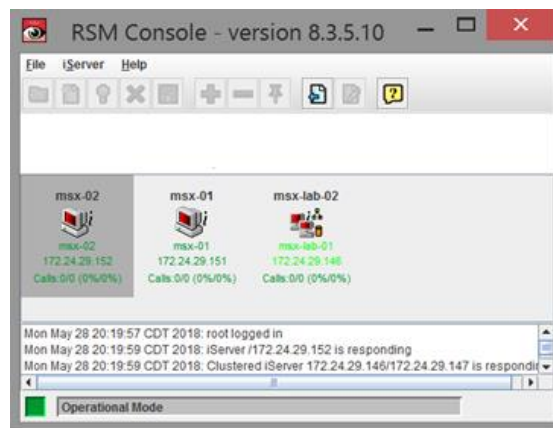


Imagen 47. Ventana de *RSM Console*.

Seleccioné el *SBC msx-02*. Otra ventana se abrió. Esta nueva ventana tiene la función de poder manejar y configurar el *SBC* de una manera gráfica y visualmente más amigable que por consola. En el menú *iServer* seleccioné *Configure*.

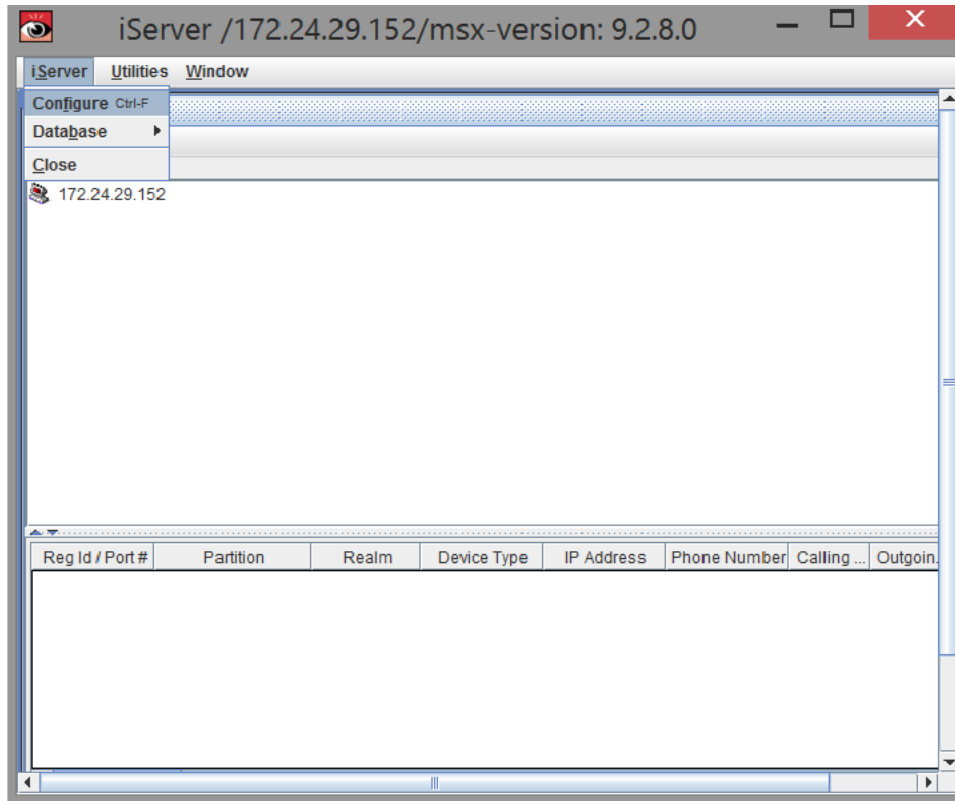


Imagen 48. Ventana de configuración de SBC msx-02.



Y finalmente seleccioné la pestaña *FCE*.

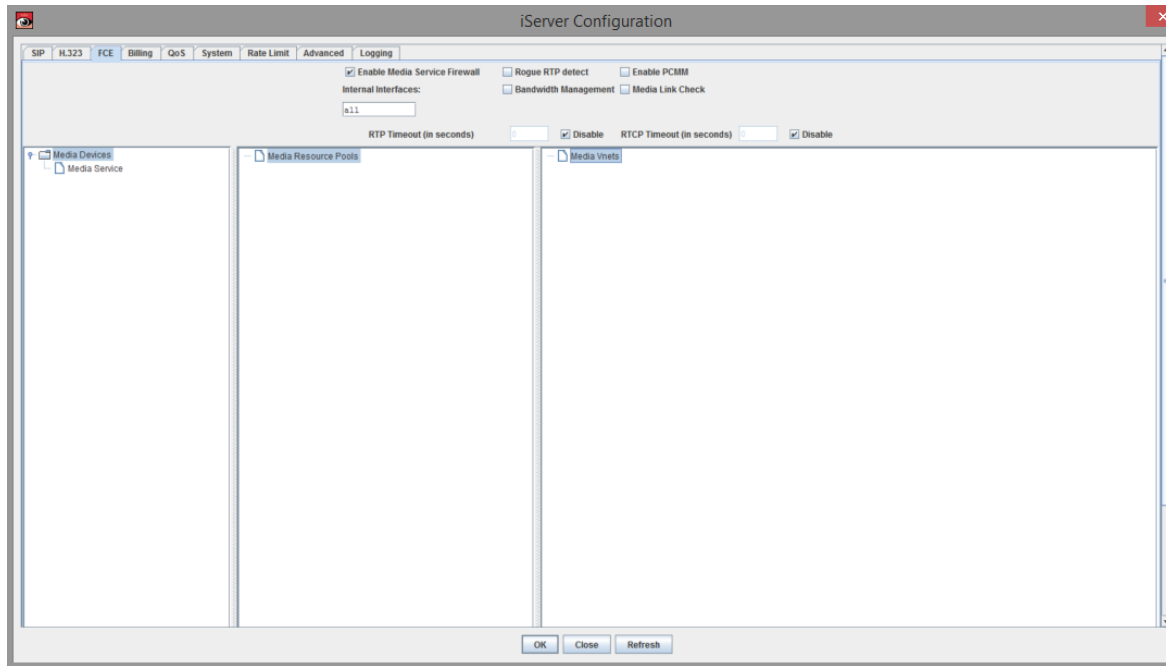


Imagen 49. Vista de pestaña FCE.

**Nota:** La configuración que se presenta a continuación utiliza valores específicos de la red del laboratorio. Las IPs a utilizar varían dependiendo de cada red.

Hice la configuración de las interfaces usando los siguientes pasos que serán descritos a continuación:

- Agregar un dispositivo de ruta de medios (*Media Routing Device*).
- Agregar *Vnet* de medios.
- Agregar las rutas de medios.
- Crear fuente de recursos de medios (*Media Resource Pool*).
- Crear fuente de rutas de medios (*Media Routing Pool*).

## Dispositivo de Ruta de Medios (*Media Routing Device*).

Agregué un dispositivo de ruta de medios dando *click* secundario en *Media Devices* (sección izquierda) y después *Add Device*. En la nueva ventana sólo llené el campo *Name* con el nombre *hk* y seleccioné *mfc* (esto se debe al tipo de hardware de las interfaces físicas) de las opciones. Los demás campos permanecieron con los valores preestablecidos incluida la dirección IP y máscara predeterminada.

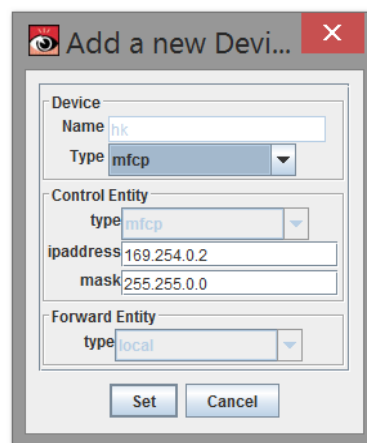


Imagen 50. Ventana para agregar dispositivo.

Una vez hecho esto apareció el dispositivo que creé en los 3 paneles de la ventana *FCE*

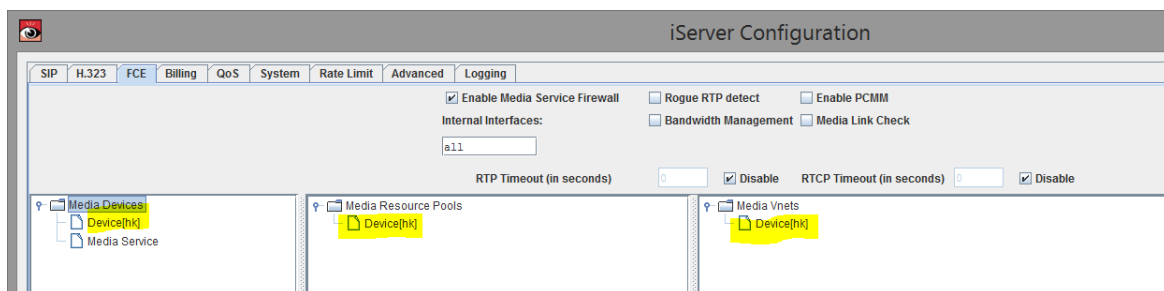
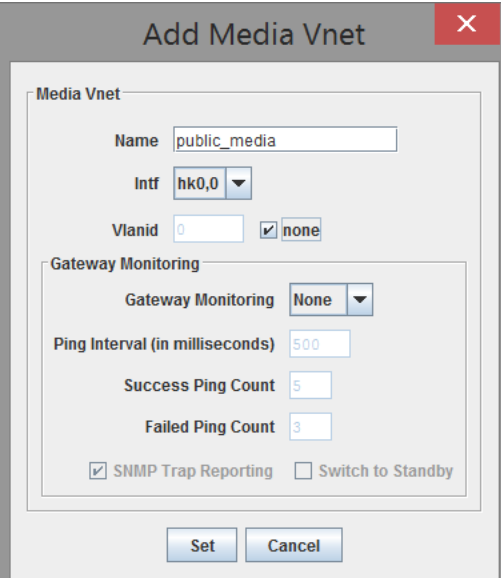


Imagen 51. Dispositivo creado.

## Vnet de medios.

Para agregar la *Vnet* de medios di *click* secundario en *Media Vnets* (sección de la derecha) y después *Add*.

En la sección del nombre utilicé las *Vnets* antes definidas por línea de comando (*pub\_sig\_vn* y *pri\_sig\_vn*). Seleccioné la interfaz *hk0,0* para la *Vnet* pública y *hk0,1* para la privada. No *Vlanid* fue requerida para esta configuración. Los demás campos los dejé con los valores preestablecidos.



The image shows a dialog box titled "Add Media Vnet" with a close button (X) in the top right corner. The dialog is divided into two main sections: "Media Vnet" and "Gateway Monitoring".

**Media Vnet section:**

- Name: public\_media
- Intf: hk0,0
- Vlanid: 0, with a checked checkbox labeled "none".

**Gateway Monitoring section:**

- Gateway Monitoring: None
- Ping Interval (in milliseconds): 500
- Success Ping Count: 5
- Failed Ping Count: 3
- SNMP Trap Reporting: checked
- Switch to Standby: unchecked

At the bottom of the dialog are two buttons: "Set" and "Cancel".

Imagen 52. Ventana para agregar *Vnet* de medios.

## Agregar Rutas de Medios.

Una vez definida la *Media Vnet* procedí a crear las rutas de medios. Para esto, en las *Vnet media* recién creadas di *click* secundario y seleccioné *Add route*.

En esta ruta se debe definir la dirección IP de un *router* o *Gateway* capaz de alcanzar las IPs con las que el *SBC* se comunicará. En estos casos utilicé las IPs 172.24.29.113 y 172.24.29.121.

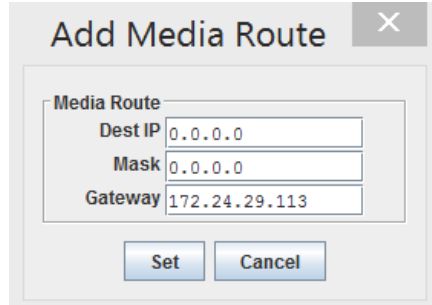


Imagen 53. Ventana para agregar ruta de medios.

### **Creación de Fuente de Recursos de Medios (*Media Resource Pool*).**

Esta Fuente (*Pool*) define la interfaz virtual de media que será asociada con las interfaces físicas.

Para agregar este *Pool* di *click* secundario en *Devie[hk]* bajo la sección *Media Resource Pools* (Sección central).

En el campo *ID* usé el valor 1 para identificar a la Fuente de medios pública, para la Fuente de medios privados usé el valor 2 y utilicé los nombres *public\_media* y *private\_media*.

En la sección de *vnet* seleccioné entre las *vnets* anteriormente creada dependiendo si esta Fuente sería pública o privada. El campo de dirección IP y máscara dejé los valores predeterminados.

Los valores de *Low Port* y *High Port* determinan los puertos que el *SBC* utilizará para establecer los flujos de media y transacción de mensajes SIP. El rango de los puertos va de 11,000 a 65,535. Utilicé los valores 11,001 y 065,000.

Imagen 54. Ventana para agregar fuente de recursos de medios.

Antes de terminar de agregar la fuente seleccioné la opción *DTMFGEN*. Esta opción se utiliza para establecer que el sistema soportará tonos DTMF.

Imagen 55. Selección de DTMFGEN.

Una vez creadas las fuentes (*Pool*) pública y privada, di *click* secundario sobre estos y seleccioné *Add LinkGroup*.

Las ligas de grupos (*Link Groups*) son utilizados para conectar el *SBC* con los *Gateways* de *transcoding*. Cada una de estas ligas de grupo están asociados con Fuentes de Recursos de Medios específicos.

En este caso utilicé los números correspondientes a las Fuentes de Medios creadas anteriormente (1 pública y 2 privada).

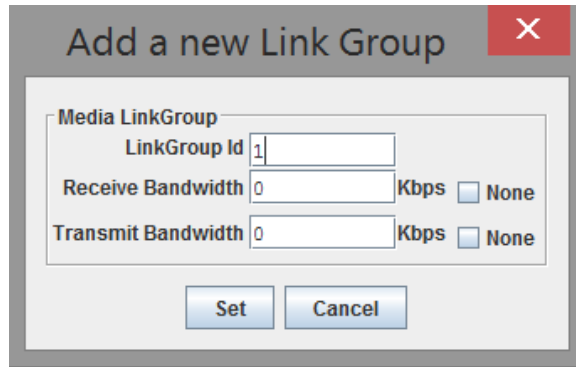


Imagen 56. Ventana para agregar ligas de grupos.

Los demás campos los dejé con los valores predeterminados.

### Creación de Fuentes de Servicios de Medios (*Media Service Pools*).

Finalmente creé las Fuentes de Servicios de Medios. Para hacer esto di *click* secundario en *Media Service* que se encuentra en la sección izquierda y seleccioné *Add Pool*. Para el lado público utilicé el identificador *ID 1* y para el lado privado fue *ID 2*. Nombré los *Pools* y en el menú *Sub Dev* escogí *HK* y guardé los cambios. El campo *Sub Pool* lo dejé con el valor que tenía predeterminado.

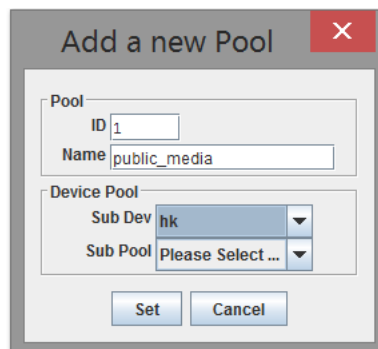


Imagen 57. Ventana para agregar fuentes de servicios de medios.

La configuración quedó de la siguiente manera. Guardé los cambios dando *click* en *OK*.

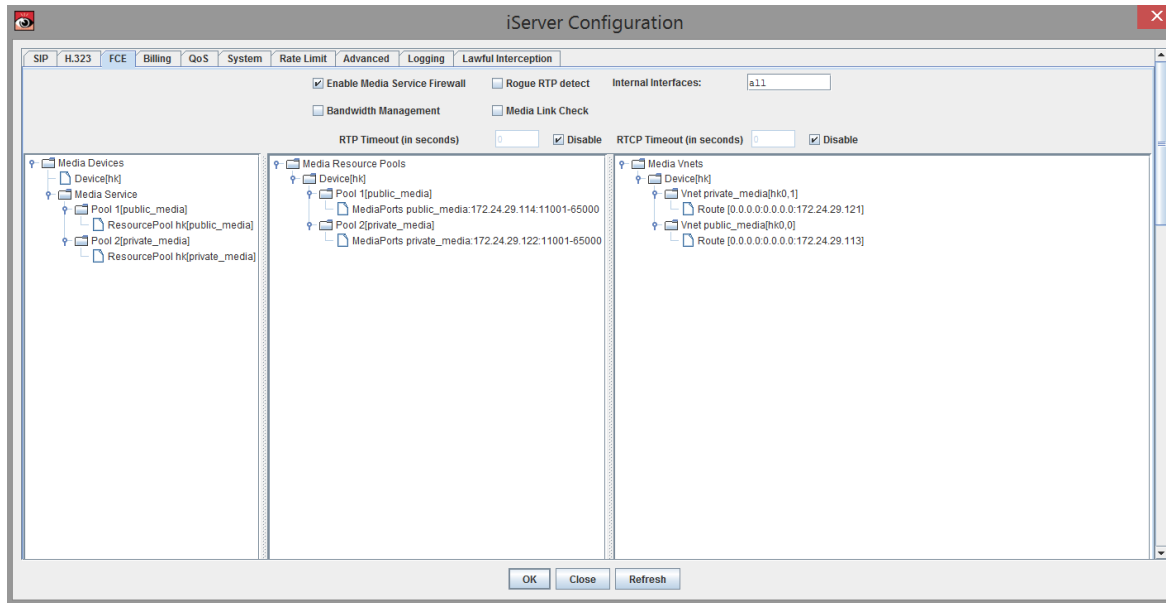


Imagen 58. Ventana de la pestaña *FCE* mostrando la configuración completa.

## Creación de Reinos (*Realms*) y Puntos Finales (*Endpoints*).

### ***Realms.***

Un *realm* (Reino) es un servicio lógico que funciona como punto de entrada para que dispositivos externos se conecten al SBC.

Los *realms* los configuré usando la siguiente información: Para el público usé el nombre *public-access*, IP 172.24.29.187, máscara 255.255.255.255.240 y lo configuré en la *VNET pub\_sig\_vn*. Para el *realm* privado usé *private-access*, IP 172.24.29.171, máscara 255.255.255.240, en la *VNET pri\_sig\_vn*.

Después ingresé los siguientes comandos (estos pasos los seguí de la misma manera para el *realm* privado).

Utilicé el comando 15 para crear y nombrar un *realm* nuevo:

**# cli realm add <realm-name>** ...(15)

```
msx-02:~ # cli realm add public-access
```

Imagen 59. Creación de *realm*.

Agregué la IP y máscara editando el *realm* recién creado (comando 16):

**# cli realm edit <realm-name> rsa <IP-address> mask <rsa-subnet-mask>**  
...(16)

```
msx-02:~ # cli realm edit public-access rsa 172.24.29.187 mask 255.255.255.240
```

Imagen 60. Se agrega la dirección IP y máscara en el *realm* recién creado.

La *VNET* la asigné usando el comando 17.

**# cli realm edit <realm-name> vnet <vnet-name>** ...(17)

```
msx-02:/ # cli realm edit public-access vnet pub_sig_vn
```

Imagen 61. Asignación de *vnet* a *realm*.

Asigné una fuente de medios (*Media Pool*) con el comando 18 (Utilicé el ID o identificador utilizado en la imagen 56).

**# cli realm edit <realm-name> medpool <media-routing-pool-ID>** ...(18)

```
msx-02:~ # cli realm edit public-access medpool 1
```

Imagen 62. Asignación de identificador al *realm*.

Finalmente, usé los comandos 19 y 20 para definir que el *realm* pueda enviar medios interna y externamente.



**# cli realm edit <realm-name> emr alwayson** ... (19)

**# cli realm edit <realm-name> imr alwayson** ... (20)

Después de terminar de editar los *realms* confirmé que los cambios hubiesen sido aplicados correctamente usando el comando 21.

**# cli realm brief** ... (21)

```
msx-02:/ # cli realm brief
Realm          public-access / 2
Rsa            172.24.29.187
Mask           255.255.255.240
VNET           pub_sig_vn
Adm Status     ENABLED
Oper Status    UP

Realm          private-access / 1
Rsa            172.24.29.171
Mask           255.255.255.240
VNET           pri_sig_vn
Adm Status     ENABLED
Oper Status    UP

Found 2 realm entries
```

Imagen 63. Resumen de las características de los dos *realms* creados.

### Configuración de *endpoints*.

Finalmente llegué al proceso de definir los *endpoints*. Un *endpoint* se puede definir como conexión final, puede ser un teléfono, otro SBC, un proveedor de servicios, etc.

Los *endpoints* son entidades creadas bajo los *realms* (en este caso son *private-access* y *public-access*). A continuación, describo los pasos que seguí para la configuración de éstos.

Creé *Endpoint1* para *public-realm* con número telefónico 963 y *Endpoint2* para *private-access* con número 852.

Los comandos presentados a continuación los utilicé para ambos *endpoints*.

Agregué el *endpoint* usando el comando 22. El parámetro *Uport* define el puerto que se usará en el *endpoint*. Se puede utilizar un valor numérico desde 0 al 256. En este caso usé el 0 ya que estos dispositivos serán únicos.

**# cli iedge add <Endpoint-Name> <UPort>** ... (22)

```
msx-02:~ # cli iedge add Endpoint1 0
```

Imagen 64. Creación de *endpoint*.

Procedí a definir el número telefónico de éste usando el comando 23.

**# cli iedge phones <Endpoint-Name> <UPort> <Phone Number>** ... (23)

```
msx-02:~ # cli iedge phones Endpoint1 0 963
```

Imagen 65. Definición de número telefónico para el *endpoint*.

Agregué *Endpoint1* al *realm* público.

**# cli iedge edit <Endpoint-Name> <UPort> realm <realm-name>** ... (24)

```
msx-02:~ # cli iedge edit Endpoint1 0 realm public-access
Dynamic Blacklisting Policy configuration (inherit-from-realm) will be applied
ublic-access) combination
```

Imagen 66. Asignación del *endpoint* al *realm*.

Finalmente habilité la opción SIP (la forma de señalización a utilizar) en el *endpoint*.

**# cli iedge edit <Endpoint-Name> <UPort> sip enable** ... (25)

```
msx-02:~ # cli iedge edit Endpoint1 0 sip enable
```

Imagen 67. Protocolo SIP es habilitado en el *Endpoint*.

Con la creación de los *Endpoints* terminé la configuración del SBC.

## RESULTADOS OBTENIDOS.

### Registro y llamada telefónica.

#### Registro de *Endpoints*.

Una vez terminada la configuración y creación de entidades necesarias para tener comunicación *VoIP* procedí a hacer pruebas. Para esto utilicé el *softphone X-Lite* para establecer una llamada. Agregué los números telefónicos e IPs de los *realms* correspondientes al lado privado y público en diferentes computadoras. A continuación, se muestra una imagen antes de registrar el teléfono en el *softphone*:

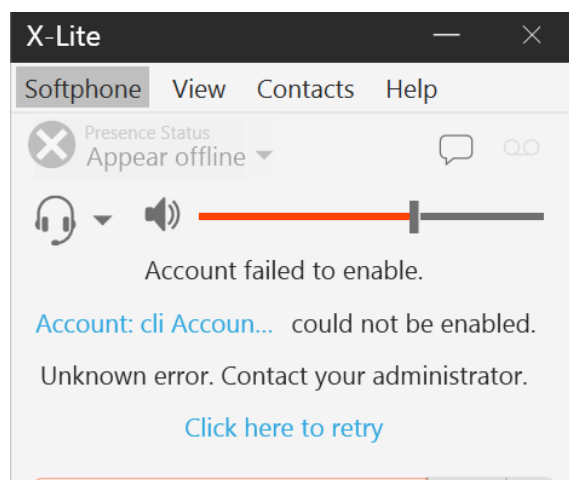


Imagen 68. X-Lite sin número telefónico registrado.

Para configurar el *softphone* accedí a las opciones *Softphone > Account Settings* y llegué al menú que se presenta a continuación:

SIP Account

Account Voicemail Topology Presence Transport Advanced

Account name: kli Account 1

Protocol: SIP

Allow this account for

- Call
- IM / Presence

User Details

\* User ID: 852

\* Domain: 172.24.29.171

Password:

Display name:

Authorization name:

Domain Proxy

- Register with domain and receive calls

Send outbound via:

- Domain
- Proxy Address:

OK Cancel

Imagen 69. Configuración de X-Lite.

Una vez que agregué la información del teléfono y la IP del *Realm* correspondiente di *click* en *OK* y el teléfono se registró en el SBC, esto se observó cuando el icono cambió a una *palomita* verde y el número telefónico aparece en la parte superior:

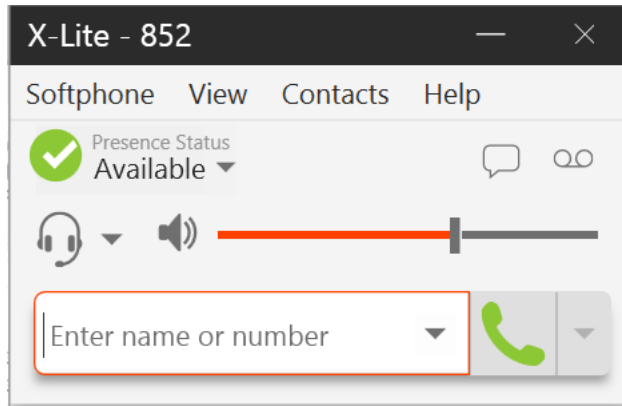


Imagen 70. Teléfono registrado correctamente en X-Lite.

De la misma manera comprobé que los *endpoints* se registraron de manera exitosa en el SBC al correr el comando 26 y buscar los *endpoints* creados:

**# cli iedge list | less**

...(26)

```

Registration ID      Endpoint1
Port                0
Network Phone       963
Phone               963
IpAddr              172.24.18.69/0
Realm               public-access/2
Type                iphone

Registration ID      Endpoint2
Port                0
Network Phone       852
Phone               852
IpAddr              172.24.18.88/0
Realm               private-access/1
Type                iphone

```

Imagen 71. Las IPs de las computadoras ya aparecen en la información de los *endpoints*. Esto significa que se registraron correctamente.

El *Endpoint1* se registró en la IP 172.24.18.69 mientras que el *Endpoint2* se registró en la IP 172.24.18.88. Estas IPs corresponden a la IP de las computadoras en las que se registró cada *endpoint*.

## Llamada Telefónica.

Una vez registrados los dos *endpoints* procedí a establecer una llamada y correr un trazado en el SBC para poder capturar los paquetes enviados. Para correr el trazado utilicé el comando 27:

```
# tshark -i any -w Call.pcap -P port 5060 ... (27)
```

i: Interfaz

w: escribir/crear archive

P: puerto

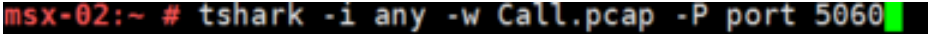
A terminal window screenshot showing a command prompt. The prompt is 'msx-02:~ #'. The command entered is 'tshark -i any -w Call.pcap -P port 5060'. The cursor is at the end of the command, indicated by a green vertical bar.

Imagen 72. Comando para realizar captura de paquetes usando *tshark*.

Al correr este trazado se creó un archivo .pcap el cual abrí con el programa *WireShark*, (software utilizado para el análisis de captura de paquetes).

Generé un diagrama de escalera en el cual se puede observar el intercambio de mensajes de manera exitosa para establecer la llamada:

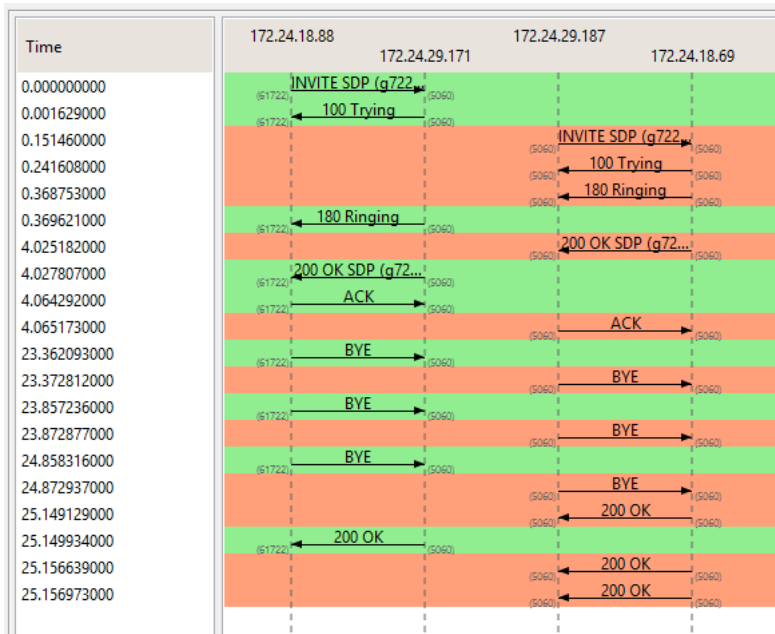


Imagen 73. Intercambio de mensajes SIP capturados durante la llamada.

## Resultados:

La ejecución de este proyecto involucró varios pasos, los cuales tuve que desarrollar de diferentes maneras:

Empecé este informe laboral con una introducción histórica y teórica del desarrollo de las telecomunicaciones, redes sobre IP y los protocolos usados en la comunicación VoIP. En esta área introductoria tuve que realizar una consulta bibliográfica de la información necesaria para poder presentar el contexto en el que se desarrolla mi informe laboral.

En la sección de los pre-requisitos necesarios para la instalación de la imagen del *Session Border Controller*, describí la creación de la imagen de arranque, requerimientos de *hardware* y *software*. Para describir estos pasos utilicé conceptos de redes, así como el uso de Linux.

Una vez que instalé y configuré correctamente el SBC procedí a realizar una llamada telefónica. Al analizar el diagrama confirmé que las IPs que aparecen a los extremos (172.24.18.88 y 172.24.18.69) corresponden a las IPs de las computadoras desde las que se estableció la llamada. También observé que las IPs internas (172.24.29.171 y 172.24.29.187) corresponden a las IPs de los *Realms* configurados en el SBC.

El diagrama de escalera se puede descomponer de la siguiente manera:

- **Tiempo 0.0000.** La llamada empieza cuando el *Endpoint2* manda un mensaje INVITE al SBC (el SBC puede ser identificado como en el trazado como el espacio que existe entre las IPs 172.24.29.171 y 172.24.29.187). En este mensaje SIP puede observar que contiene el protocolo SDP. En cada línea *a* (*Media Attribute*) se encuentran los códecs que soporta el *Endpoint2*.



```

Session Initiation Protocol (INVITE)
  Request-Line: INVITE sip:963@172.24.29.171 SIP/2.0
  Message Header
  Message Body
    Session Description Protocol
      Session Description Protocol Version (v): 0
      Owner/Creator, Session Id (o): - 13181766746987525 1 IN IP4 172.24.18.88
      Session Name (s): X-Lite release 5.3.3 stamp 92990
      Connection Information (c): IN IP4 172.24.18.88
      Time Description, active time (t): 0 0
      Media Description, name and address (m): audio 60838 RTP/AVP 9 8 120 0 84 101
      Media Attribute (a): rtpmap:120 opus/48000/2
      Media Attribute (a): fmtp:120 useinbandfec=1; usedtx=1; maxaveragebitrate=64000
      Media Attribute (a): rtpmap:84 speex/16000
      Media Attribute (a): rtpmap:101 telephone-event/8000
      Media Attribute (a): fmtp:101 0-15
      Media Attribute (a): sendrecv

```

Imagen 74. Lista de códecs en el INVITE de *Endpoint2*.

- **Tiempo 0.0016** El SBC mensaje es re-enviado al *Endpoint1*. Éste responde con un mensaje provisional (*100 Trying*) y luego genera un tono (*180 Ringing*). Este mensaje es re-enviado del SBC (**0.3696**)
- En 4.0251 *Endpoint1* responde con un mensaje 200 OK aceptando la llamada. Al aceptar la llamada envía la selección de códecs que se usarán a partir de lo que fue ofrecido en el mensaje INVITE en **0.0000**.

```

Session Initiation Protocol (200)
  Status-Line: SIP/2.0 200 OK
  Message Header
  Message Body
    Session Description Protocol
      Session Description Protocol Version (v): 0
      Owner/Creator, Session Id (o): - 750523142 3 IN IP4 172.24.18.69
      Session Name (s): X-Lite release 5.3.3 stamp 92990
      Connection Information (c): IN IP4 172.24.18.69
      Time Description, active time (t): 0 0
      Media Description, name and address (m): audio 54002 RTP/AVP 9 8 120 0 84 101
      Media Attribute (a): rtpmap:120 opus/48000/2
      Media Attribute (a): fmtp:120 useinbandfec=1; usedtx=1; maxaveragebitrate=64000
      Media Attribute (a): rtpmap:84 speex/16000
      Media Attribute (a): rtpmap:101 telephone-event/8000
      Media Attribute (a): fmtp:101 0-15
      Media Attribute (a): sendrecv

```

Imagen 75. Contenido de protocolo SDP.

- Cuando este mensaje llega al *Endpoint2* **4.0278** y éste responde con un mensaje de *Acknowledge (ACK)* la llamada se establece.
- Una vez establecida la llamada los códecs negociados permitieron la transmisión de voz.

- La llamada es terminada por el *Endpoint2* en **23.3620** cuando envía un mensaje *BYE*. Este mensaje pasa por el SBC y es enviado al *Endpoint1*. Este último responde con 200 *OK*.

## Conclusiones.

El proceso y procedimientos que realicé en este proyecto permitieron una correcta instalación del SBC y su correcta integración con la interfaz gráfica RSM para generar un sistema de comunicación de voz sobre IP.

Verifiqué los requerimientos básicos de *hardware* y de *software*. Al seguir el proceso de “saltos” de versión de sistema operativo actualicé a la versión deseada (9.2.8) y así configuré adecuadamente los servicios de red.

La integración con la interfaz gráfica RSM se llevó de manera exitosa al configurar las entidades físicas y lógicas para poder realizar las llamadas de voz sobre IP.

Los resultados obtenidos en la llamada VoIP realizada muestran el intercambio de mensajes y señalización necesaria para iniciar la comunicación entre dos dispositivos telefónicos. De la misma manera se pudo observar la negociación de códec a utilizar para, finalmente, establecer una llamada de VoIP. Este intercambio de mensajes SIP es la base de la mayoría de las llamadas que se realizan actualmente en la telefonía.

Una vez que el servidor tenía la versión de Sistema Operativo deseada y la aplicación del *Session Border Controller* estaba instalada proseguí a realizar la configuración de red, interfaces lógicas, integración con la interfaz gráfica RSM y creación de los dispositivos lógicos para poder realizar llamadas VoIP. En esta sección utilicé varios temas de redes, máscaras de red.

Finalmente, al realizar la llamada para comprobar el correcto funcionamiento del sistema, utilicé conceptos de SIP y SDP para poder analizar los resultados obtenidos.

El aprendizaje y experiencia que obtuve en la Facultad de Ingeniería me ha sido de utilidad en el tiempo que he estado laborando para esta empresa. La forma racional de analizar y abordar problemas definitivamente ha contribuido positivamente en mi desarrollo profesional. Y combinado con los conocimientos que he adquirido

trabajando, tales como un mayor conocimiento en redes, Linux y voz sobre IP, me han ayudado a tener una posición de guiar y capacitar nuevos ingenieros.

## Anexo:

Lista de comandos usados:

1. Versión de sistema operativo instalado.

**# cat /etc/GbLinux-release**

2. Información sobre tipo de caja.

**# hwinfo --bios | grep Product**

3. Acceso a YaST.

**# yast**

4. Comando de Linux para crear directorios:

**# mkdir <nombre del directorio a crear>**

5. Comando de Linux para descomprimir archivo.

**# tar -xzvf <archivo comprimido>**

6. Listar elementos en la ubicación actual.

**# ls -lh**

7. Comando para ejecución de *scripts*.

**# ./<script a ejecutar>**

8. Comando de distribución *Red Hat* para la instalación de paqueterías.

**# rpm -ivh <paquetería de medios>**

9. Comando de distribución de *Red Hat* para comprobar que la paquetería haya sido instalada correctamente.

**# rpm -q hk**

10. Comando de aplicación de SBC para revisar versión de *Software*.

**# gis -v**

11. Comando usado para instalar licencia.

**# nxconfig.pl -L**

11. Comando de aplicación de SBC para crear *VNETs*.

**# cli vnet add <vnetname> ipver 4**

12. Comando de aplicación de SBC para asignar interfaces de red.

**# cli vnet edit <vnetname> ifname <interface name>**

13. Comando de aplicación de SBC para asignación de *Gateways*.

**# cli vnet edit <vnetname> primary-gateway <primary gateway IP>**

14. Comando de aplicación de SBC para mostrar lista de *VNETs*.

**# cli vnet list**

15. Comando de aplicación de SBC para crear *Realms*.

**# cli realm add <realm-name>**

16. Comando de aplicación de SBC para agregar IP y máscara de red a *Realms*.

**# cli realm edit <realm-name> rsa <IP-address> mask <rsa-subnet-mask>**

17. Comando de aplicación de SBC para asignar *VNET* a *Realm*.

**# cli realm edit <realm-name> vnet <vnet-name>**

18. Comando de aplicación de SBC para agregar *Media Pools* a *Realms*.

**# cli realm edit <realm-name> medpool <media-routing-pool-ID>**

19. Comando de aplicación de SBC para habilitar envío de medios externamente.

**# cli realm edit <realm-name> emr alwayson**

20. Comando de aplicación de SBC para habilitar envío de medios internamente.

**# cli realm edit <realm-name> imr alwayson**

21. Comando de aplicación de SBC para ver un breve resumen de los *Realms* existentes.

**# cli realm brief**

22. Comando de aplicación de SBC para crear un *Endpoint*.

**# cli iedge add <Endpoint-Name> <UPort>**

23. Comando de aplicación de SBC para definir número telefónico del *Endpoint*.

**# cli iedge phones <Endpoint-Name> <UPort> <Phone Number>**

24. Comando de aplicación de SBC para agregar *Realm* a *Endpoint*.

**# cli iedge edit <Endpoint-Name> <UPort> realm <realm-name>**

25. Comando de aplicación de SBC para habilitar protocolo SIP en *Endpoint*.

**# cli iedge edit <Endpoint-Name> <UPort> sip enable**

## **Glosario.**

**CLI:** Interfaz de Línea de Comandos. (*Command Line Interface*).

**Códec:** Codificador – Decodificador. Es un circuito integrado que transforma las señales de voz analógicas en una cadena de bits y después transforma las señales digitales a una señal analógica de nuevo. (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 726)

**DTMF:** Dual Tone Multi-Frequency. Procedimiento de marcación telefónica mediante la selección de dos frecuencias para cada cifra, de entre un grupo de 16 combinaciones posibles. (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 89)

**Datagrama:** En las redes de conmutación de paquetes es una forma de encaminamiento. En ésta un paquete se dirige hacia su destino final, independientemente del resto, por los tramos de menor carga y retardo sin que previamente se haya establecido un círculo virtual o real. (Huidobro Moya, J. M. 2011. P. 388).

**Default Gateway:** Puerta de Enlace. En un host IP (Anfitrión IP), es la dirección IP que tiene el *router* al cual el host manda los paquetes de red cuando estos paquetes tienen una dirección de destino diferente a la red en que se encuentra el host (Odom, W. 2013. P. 847).

**Eco:** En comunicaciones electrónicas, eco es la reflexión de una señal eléctrica en un canal de comunicación. (Horak, Ray. 2007. P. 241, 615).

**FDM:** (Frequency-Division Multiplexing) Este tipo de multiplexación utiliza el dominio de la frecuencia para mandar múltiples señales simultáneamente a través del ancho de banda disponible en un solo canal. (Hioki, Warren. 1998. P. 602).

**G.723.1:** Codificador *Dual-rate speech* y algoritmo de compresión estándar con 5.3 y 6.3 kbps para comunicaciones multimedia. (Horak, Ray. 2007. P. 241, 615).

**GBLinux:** Sistema Operativo usado por el SBC de la empresa basado en SuSE.

**GUI:** Interfaz Gráfica para usuario (*Graphic User Interface*).

**H323:** Es una recomendación por parte del ITU-T que define la señalización necesaria para establecer sesiones para servicios de tiempo real. (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 182)

**HK:** *Hot Knife*. Interfaces del Controlador de Sesión de Frontera usadas para la señalización y medios.

**HTTP:** Hypertext Transfer Protocol. (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 834)

**Hub:** Un dispositivo LAN que provee un punto de conexión centralizado para un cableado LAN, repitiendo cualquier señal que reciba en los otros puertos, por lo tanto, creando un bus lógico. (Odom, W. 2007. P. 879)

**IP:** Internet Protocol. Es el protocolo de la capa de red del modelo TCP/IP. Provee estándares y servicios de rutas y direcciones lógicas en una red. (Odom, W. 2007. P. 879)

**ITU-T:** ITU (International Telecommunication Union) Standardization Sector. Sector de Normalización de las Telecomunicaciones de la UIT. Las Comisiones de Estudio del Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T) reúnen a expertos de todo el mundo para elaborar normas internacionales conocidas como Recomendaciones UIT, que actúan como elementos definitorios de la infraestructura mundial de las Tecnologías de la Información y la Comunicación (TIC). (Odom, W. 2007. P. 882)

**LAN:** Local Area Network – Red de área local. (Odom, W. 2007. P. 882)

**MOS:** Mean Opinion Score - Puntuación de opinión media. Una métrica de calidad subjetiva que consiste en el promedio de un grupo de calificaciones de cinco puntos y con la escala 1=malo, 2=pobre, 3=aceptable, 4=bueno y 5=excelente. (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 52)

**PCM:** (Pulse Code Modulation). Modulación por impulsos codificados. Es una técnica - muestreo a 8,000 veces por segundo y codificación de muestras con 8



bits- para transmitir de forma digital señales analógicas. Normalmente la voz, sobre un flujo digital de 64 kbit/s. (Huidobro Moya, J. M. 2011. P. 397)

**PESQ:** Perceptual Evaluation of Speech Quality – Método objetivo para estimar la calidad de los canales de los teléfonos de banda corta y los códecs de voz. Estandarizado por la ITU Rec. P.862. (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 52)

**Proxy (Servidor Proxy):** Es una entidad intermediaria que actúa como un servidor y como cliente con el propósito de hacer una petición en nombre de los licentes. Un servidor proxy principalmente funge la función de enrutador, esto es, que las peticiones sean enviadas a una entidad más cercana al usuario objetivo. También sirven para reforzar las políticas y autorización de usuarios. Por ejemplo, un proxy puede interpretar, y si es necesario, modificar partes de un mensaje antes de enviarlo.

**PSTN:** (Public Switched Telephone Network). Red pública de conmutación de circuitos. Es la red telefónica básica empleada en todos los países para establecer comunicaciones vocales. (Huidobro Moya, J. M. 2011. P. 402)

**Red:** Una colección de computadoras, impresoras, routers, switches y otros dispositivos que se pueden comunicar entre ellos sobre algún medio de transmisión. (Odom, W. 2007. P. 4)

**Router:** Dispositivo de capa 3 del modelo OSI. cuya función consiste en enviar paquetes entre diferentes redes, direccionamiento de dispositivos en una red y la comunicación entre estos dispositivos. (Odom, W. 2007. P. 898)

**RTP:** Real Time Protocol. Protocolo usado para el envío de información (media) en tiempo real, por ejemplo, audio y video. (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 162)

**SBC:** Session Border Controller. Es un dispositivo encargado de regular y proteger los flujos de comunicación multimedia IP. Se coloca en los bordes o fronteras de las redes para controlar y separar las sesiones multimedia públicas de las privadas. (IETF. RFC 5853. 2017. P. 3)

**SIP:** Session Initiation Protocol. Protocolo usado para el establecimiento de sesiones multimedia usando mensajes basados en la sintaxis HTTP. (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 768)

**Softphone:** Software que funciona como teléfono y típicamente usa el protocolo SIP para hacer llamadas de voz sobre IP.

(<https://help.ubuntu.com/community/SoftPhone>)

**Switch:** Un dispositivo de red que filtra, envía y maneja tramas de Ethernet basadas en la dirección de destino de cada trama. (Odom, W. 2007. P. 905)

**TCP:** Transport Control Protocol. Un protocolo que se encuentra en la capa de transporte del modelo TCP/IP que provee transmisión de datos de manera confiable. Definido por IETF RFC 793 (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 753)

**TDM:** (Time Division Multiplexing). Técnica de multiplexación por división en el tiempo, que permite intercalar los datos procedentes de varios usuarios en un único canal, vía serie. (Huidobro Moya, J. M. 2011. P. 406)

**Trama (*frame*):** equivalente del bloque en protocolo de enlace, particularmente en DIC. (Huidobro Moya, J. M. 2011. P. 408)

**Transcoding:** Se le llama *transcoding* a la conversión de un códec a otro.

**Trazado:** Se llama trazado a la captura de paquetes de red por un *Software*.

**UDP:** User Datagram Protocol. Un protocolo que se encuentra en la capa de transporte del modelo TCP/IP. Es un protocolo sencillo que intercambia datagramas sin confirmar o asegurar la entrega de la información. Definido por IETF RFC 768 (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 754)

**VoIP:** Voice over IP. El transporte de voz dentro de paquetes IP sobre una red IP. (Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca. 2004. P. 773).

**WireShark:** *Software* de análisis de protocolos de red. (<https://www.wireshark.org/>)

**YaST:** *YaST* es la herramienta de instalación y configuración de openSUSE y la distribución SUSE Linux Enterprise. (<https://es.opensuse.org/Portal:YaST>)

## **Bibliografía y mesografía.**

- Dudley, S. M., Joshi, S., Thorpe, L. A. & Mendonca, T. A. (Ed.). (2004). *Essentials of Real Time Networking. How Real-Time Disrupts the Best-Effort Paradigm*. Nortel Networks
- Figueiras, A. R. (2002). *Una panorámica de las telecomunicaciones*. (1ª ed.) Madrid: Pearson Educación, S. A.
- Huidobro Moya, J. M.I. y Conesa Pastor, Rafael (2006) *Sistemas de Telefonía*. (5ª ed.). Thomson.
- Hioki, Warren (1998). *Telecommunications*. (3ª ed). Columbus: Prentice-Hall.
- Internet Live Stats. (s.f). Recuperado el 9 de enero de 2017, de <http://www.internetlivestats.com/internet-users/>
- Huidobro Moya, J. M. (2011) *Telecomunicaciones* (1ª ed.). Madrid: Ra-Ma Editorial.
- Horak, Ray. (2007) *Telecommunications and Data Communications Handbook* (1a ed). New Jersey: John Wiley & Sons, Inc.
- El UIT-T en breve. (2017). Recuperado el 9 de enero de 2017, de <http://www.itu.int/es/ITU-T/about/Pages/default.aspx>
- [IETF](https://tools.ietf.org/html/rfc5853). RFC 5853 (2017) Recuperado el 20 de febrero de 2017, de <https://tools.ietf.org/html/rfc5853>
- Portal YaST de Open SUSE Reuperado 5 de marzo 2018 de <https://es.opensuse.org/Portal:YaST>

- Odom, W (2007). *CCENT/CCNA ICND1 Official Exam Certification Guide* (2<sup>nd</sup> ed). Indianapolis: Cisco Press
- Odom, W (2013). *Cisco CCENT/CCNA ICND1 100-101 Official Cert Guide* (4<sup>nd</sup> ed). Indianapolis: Cisco Press
- Szymanczyk, O. (2013). *Historia de las telecomunicaciones mundiales*. (1<sup>a</sup> ed.). Buenos Aires: Dunken.