ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

"INTEGRACIÓN DE APLICACIONES DE VOIP CON EL USO DE GNU/LINUX EN UN SISTEMA EMBEBIDO PARA LA INTERCONEXIÓN DE DISPOSITIVOS MÓVILES"

TESIS DE GRADO

Previo a la obtención de título de:

INGENIERÍA EN SISTEMAS INFORMÁTICOS

Presentado por:

JOSÉ DARÍO VILLAGÓMEZ ZAMBRANO

2014

AGRADECIMIENTO

Mis agradecimientos a mis padres, hermanos y familiares que me hecho de mí una persona de carácter, con buenos valores de casa y sobretodo saber que cualquier cosa es posible si nos mantenemos unidos. A mi novia que ha sido esa parte fundamental de todo ser humano de sentir el cariño y amor, y de saber que siempre uno no estará solo.

A todas las personas, profesores, amigos, compañeros; que han hecho posible cumplir esta meta, de obtener un título de tercer nivel, con cualquier consejo, cualquier conocimiento impartido en el Alma Mater como es la Escuela Superior Politécnica de Chimborazo

DEDICATORIA

En primer lugar quisiera dedicar el esfuerzo y trabajo dedicado en esta investigación a mi abuelita Chayo que desde el cielo estará orgullosa y aunque físicamente no pudo acompañarme en este proceso hasta el último, espiritualmente sé que me está guiando.

FIRMAS RESPONSABLES Y NOTA

NOMBRE	**FIRMA**	**FECHA**
Ing. Iván Ménes Camejo **DECANO DE LA FACULTAD** **DE INFORMÁTICA Y ELECTRÓNICA**	……	……
Ing. Jorge Huilca Palacios **DIRECTOR DE ESCUELA DE** **INGENIERÍA EN SISTEMAS**	……	……
Ing. Washington Luna Encalada **DIRECTOR DE TESIS**	……	……
Ing. Paulina Vélez Núñez **MIEMBRO DE TRIBUNAL**	……	……
Ing. Eduardo Tenelanda **DIRECTOR (E) DEL CENTRO** **DE DOCUMENTACIÓN**	……	……
NOTA DE LA TESIS	……..	

RESPONSABILIDAD DEL AUTOR

Yo José Darío Villagómez Zambrano, soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la Tesis de Grado pertenece a la **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

…….

José Darío Villagómez Zambrano

ÍNDICE DE ABREVIATURAS

MGCP	Media Gateway Control Protocol
SIP	Session Initial Protocol
RTP	Real-Time Transport Protocol
SDP	Session Description Protocol
IP	Internet Protocol
VoIP	Voice over IP
GNU	General Public License
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
OSI	Open Systems Interconnection
ISP	Internet Service Provider
FXO	Foreign eXchange Office interface
FXS	Foreign eXchange Subscriber interface
PBX	Private Branch Exchange
PSTN	Public Switched telephone network
CRM	Customer Relationship Management
IETF	Internet Engineering Task Force
HTTP	Hypertext Transfer Protocol
SMTP	Simple Mail Transfer Protocol
UA	User Agents
UAC	User Agents Control

CONTENIDO

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

CAPÍTULO I

1. FORMULACIÓN GENERAL DEL PROYECTO DE TESIS

1.1 ANTECEDENTES

En la actualidad la gente posee dispositivos móviles que están cada vez más presentes en su vida cotidiana, la mayoría de estos aparatos tienen en común su conectividad a redes WiFi. El más utilizado y de mayor difusión entre estos dispositivos es el SmartPhone, por su portabilidad y su cobertura de señal.

En las empresas el uso de la telefonía IP es cada vez más frecuente así como la movilidad de los empleados entre oficinas y departamentos ya sea por colaboración o presentaciones, lo que en ciertos momentos no están disponibles para recibir llamadas potencialmente importantes.

La interconexión de los dispositivos móviles es más permanente al internet, aprovechando al máximo su portabilidad para llamadas, chat, consultas, email, oficina móvil, etc.; utilizando las redes WiFi para la interconexión de estos dispositivos, siendo de fácil utilización sus servicios, aprovechando las llamadas sobre VoIP para la comunicación de dichos dispositivos.

Actualmente existen pocos dispositivos de interconexión que tengan una dedicación continua a un servicio de llamadas sobre VoIP, ya que la mayoría de estos dispositivos solo cumplen con la tarea de dar conexión wireless. Además de existir software SIP como Cisco, 3Com, 3CX System; que son propietarios y de OpenSoure como Asterisk, Elastix, OpenSER, OpenSIPS.

Los protocolos existentes para las comunicaciones sobre VoIP son: H.323, Media Gateway Control Protocol (MGCP), Session Initiation Protocol (SIP), Real-time Transport Protocol (RTP), Session Description Protocol (SDP), Inter-Asterisk eXchange (IAX), Jingle XMPP VoIP extensions; siendo H.323 uno de los primeros protocolos de VoIP en el tráfico de larga distancia así como de área local.

La necesidad de crear una red WiFi con el uso de OpenSource en un sistema embebido que esté dedicado a la interconexión de dispositivos que soporten llamadas sobre VoIP viene de la necesidad e importancia

de la comunicación entre los usuarios que requieren estar comunicados constantemente a un precio bajo, para la pequeña, mediana y grandes empresas a parte de reducir gastos, su personal tendría acceso a una comunicación con llamadas sobre VoIP.

Actualmente en la ESPOCH y el resto del país se encuentra implementado muchos trabajos sobre centrales de VoIP utilizando diferentes tecnologías como Asterisk, Dundi o Elastix que puede ser la base para el desarrollo del presente trabajo de investigación pero no hay muchos trabajos relacionados en el desarrollo embebido para interconectar dispositivos móviles y comunicaciones con VoIP.

1.2 JUSTIFICACIÓN DEL PROYECTO DE TESIS

1.2.1 Justificación Teórica

La necesidad innata de los seres humanos por comunicarse hace que cada vez la contribución en Telecomunicaciones sea más visible, y es esta contribución que ayuda a que las personas logren comunicarse y por lo tanto estén más unidas.

El uso de la tecnología digital para lograr esta comunicación está en la cima, y la incesante necesidad de comunicarse por medio de la voz, hace preciso la búsqueda de nuevas alternativas que lleven a conseguir este fin.

Entre los beneficios los usuarios van a poder comunicarse mediante llamadas sobre VoIP a un costo cero dentro de la red, utilizando dispositivos móviles que soporten esta funcionalidad.

La utilización de software libre también es una de las ventajas ya que los costos son bajos y es una de las mayores tendencias en el desarrollo y el uso de software, puesto que la mayoría de los dispositivos de interconexión son de software propietario, elevando los costos de implementación en este tipo de red.

1.2.2 Justificación Aplicativa

Dado que una interconexión entre dispositivos de red, sistema embebido y los SmartPhone con la finalidad de realizar llamadas de voz sobre VoIP es factible y por las ventajas expuestas, se evidencia que la necesidad de un sistema embebido que conecte a tales dispositivos, es importante, ya que con esto se conseguirá accesibilidad entre los usuarios de la red WiFi.

En la Ilustración 1 se expone un posible ambiente de pruebas del proyecto de investigación experimental, en donde muestra como en una red en donde se encuentra conectado los dispositivos de uso diario. Mediante este ambiente de pruebas y a través del punto de acceso a red se conectan los dispositivos móviles para luego enlazarse con el dispositivo que tiene el sistema embebido para las llamadas sobre VoIP;

de esta manera se puede implementar una solución que mejore la accesibilidad y disponibilidad de la comunicación.

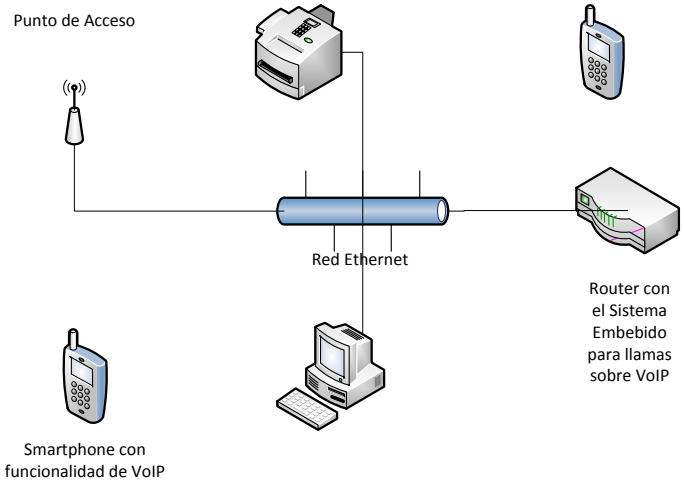


Ilustración 1. Ambiente de pruebas

1.3 OBJETIVOS

1.3.1 Objetivo General

Integrar las aplicaciones de VoIP con el uso de GNU/Linux en un sistema embebido para la interconexión de dispositivos móviles.

1.3.2 Objetivos Específicos

- Estudiar los estándares y protocolos que permiten implementar VoIP.

- Implementar los principales estándares y protocolos que permiten la implementación de VoIP.
- Determinar la aplicación de VoIP sobre la plataforma Gnu/Linux, para ser utilizado como un sistema embebido de llamadas de VoIP.
- Integrar la aplicación de VoIP en el sistema embebido.
- Asegurar la disponibilidad del servicio para la interconexión de dispositivos móviles.

1.4 HIPÓTESIS

La integración de aplicaciones de VoIP con el uso de GNU/Linux en un sistema embebido destinado a dispositivos móviles permitirá implementar una solución eficiente de comunicación.

1.4.1. TIPO DE HIPÓTESIS

Hipótesis que establece relación de causalidad. (Causa - efecto)

Variable Independiente: La integración de software libre de VoIP en un sistema embebido con el uso de dispositivos móviles

Variable Dependiente: Implementación de una solución eficiente de comunicación empresarial.

OPERACIONALIZACIÓN CONCEPTUAL

Variable	Tipo	Concepto
La integración de software libre de VoIP en un sistema embebido con el uso de dispositivos móviles	Variable Independiente	Análisis del funcionamiento del software libre especializado para ser utilizado como un sistema embebido de llamadas de VoIP.
Implementación de una solución eficiente de comunicación empresarial.	Variable Dependiente	Puesta en funcionamiento de la aplicación de VoIP en el sistema embebido.

Tabla 1. Operacionalización conceptual de variables

OPERACIONALIZACIÓN METODOLÓGICA

Variable	Indicadores	Técnica	Fuentes de verificación
La integración de software libre de VoIP en un sistema embebido con el uso de dispositivos móviles	• Tipos de dispositivos móviles • Software libre especializado en VoIP para el sistema embebido	• Observación • Pruebas	• Ambientes de prueba
Implementación de una solución eficiente de comunicación empresarial.	• Accesibilidad • Disponibilidad	• Observación • Pruebas	• Ambientes de prueba

Tabla 2. Operacionalización metodológica de variables

CAPITULO II

MARCO TEÓRICO

2.1. VOIP

Las señales digitales han predominado sobre las analógicas puesto que ofrecen mayores superioridades entre las que se pueden resaltar: Facilidad para multicanalizar las señales, factible señalización, generación de señales, baja razón señal-ruido y una encriptación eficientemente de la señal, la cual importa mucho en las comunicaciones militares y cualquier otra que requiera cumplir con niveles de seguridad (1). La red IP comenzó a desenvolverse exponencialmente con el aparecimiento del Internet. Nacieron los conceptos de nodos, servidores, enrutadores, repetidores, puentes, switches, gateways y demás elementos que conforman una red de paquetes conmutados para el intercambio de datos.

Poco a poco la información que se buscaba transferir empezó a ser más solicitante, al grado de aplicaciones populares como un Chat que no sólo comunica a dos usuarios por medio de mensajes escritos en tiempo real, sino que también les concedía la oportunidad de instaurar una conversación oral y visual con sólo una PC, micrófono, bocinas, cámara web y una conexión a Internet. Llegó el momento en el que por la red viajaban datos multimedia como videoconferencias a una tasa alta de transmisión y muestran un fuerte avance en las comunicaciones digitales. Es así como surgió la idea de implementar una red IP donde pudiera viajar la voz. Se ha preferido la red de paquetes conmutados sobre la red de circuitos conmutados puesto que la segunda exige un ancho de banda definido o fijo durante toda la transmisión punto a punto incluso cuando no se esté utilizando por completo este recurso, por ejemplo cuando ambas personas guardan silencio por instantes. Todo lo contrario ocurre en la red de paquetes conmutados, donde el ancho de banda es aprovechado al máximo.

Lo anterior se puede traducir en la diferencia de costos invertidos en cada red. Un objetivo de voz sobre IP es unificar las redes de voz y las de datos, de esta forma se adquieren muchos beneficios (1).

2.1.1. Estructura de la red VoIP

La estructura de la red de voz sobre IP es la misma estructura que se maneja en Internet, las aplicaciones, los medios de transporte, la organización del ruteo sobre la red, los modos de enlace y la transmisión de la señal por los medios físicos forman

parte del modelo OSI. La ventaja de la red VoIP es que no importa el tipo de aplicación mientras ésta pueda transformar su información en datos, segmentos, paquetes, tramas y finalmente bits. (2)

El protocolo que se utiliza para la capa de transporte es el RTP (Real-time Transfer Protocol) en segmentos de tipo UDP sobre paquetes IP. Se ha escogido éste sobre el TCP dado que, TCP es caracterizado por ser un protocolo donde se deben recibir señales de reconocimiento (acknowledge) por parte del receptor antes de enviar el siguiente segmento, es decir es un protocolo orientado a conexión que ofrece seguridad a la transmisión y recepción de los paquetes aunque introduce retardos en la comunicación (1).

El concepto de conmutador (central local, central de grupo, etc. Para una red tradicional de conmutación de circuitos) en VoIP es el Media Gateway Controller (MGC). Éste es un conjunto de productos, protocolos y aplicaciones capaces de permitir que cualquier dispositivo acceda a los servicios de Internet y de Telecomunicaciones sobre las redes IP. Este elemento es la pieza central en la red de telefonía IP, ya que es capaz de manejar inteligentemente las llamadas en la plataforma de servicio de los Proveedores de Servicio de Internet (ISP, Internet Service Provider). Por otro lado, sirven como plataformas de integración para aplicaciones e intercambio de servicios y son capaces de transportar tráfico de voz, datos y video de una manera más eficiente que los equipos existentes. (3)

El trabajo dentro del Media Gateway Controller es realizado por medio de hardware y software inteligentes; denominados por algunos autores como Softswitch y Gatekeeper (para el caso de redes H.323) (4).

En sí, estos tres elementos forman parte del mismo sistema, en otras palabras, el Gatekeeper es el hardware, el SoftSwitch es el software y ambos son controlados por el Media Gateway Controller. Todas las tareas se pueden dividir en cinco secciones:

Gateway Controller, Media Gateway, Signaling Gateway, Media Server y Feature Server. El Media Gateway Controller es eficiente gracias a su interacción con el Media Gateway y el Signaling Gateway. Las funciones principales son: control de llamada, protocolos de establecimiento de llamadas como H.323 y SIP, protocolos de control de media por ejemplo MGCP y H.248, control sobre la calidad y clase de servicio, conocimiento del enrutamiento, plan de numeración local, detalle de las llamadas para facturación, control de manejo del ancho de banda, crear un puente entre la señalización SS7 y VoIP, entre muchas otras más. Un ejemplo de elemento utilizado en redes VoIP es el puerto FXO que permiten conectar directamente una línea privada de una compañía a la PSTN, posibilitando a los terminales IP hacer llamadas a cualquier teléfono análogo (2).Con este “Gateway” se pueden realizar llamadas hacia y desde terminales telefónicos que no tienen acceso a internet. Los puertos FXS (Foreign Exchange Station) conectan su teléfono o fax convencional a la red VoIP. Se puede marcar hacia el exterior a través de un

Gateway a otras Gateways o Teléfonos IP. La Ilustración 2 muestra la estructura general de la red VoIP (5).

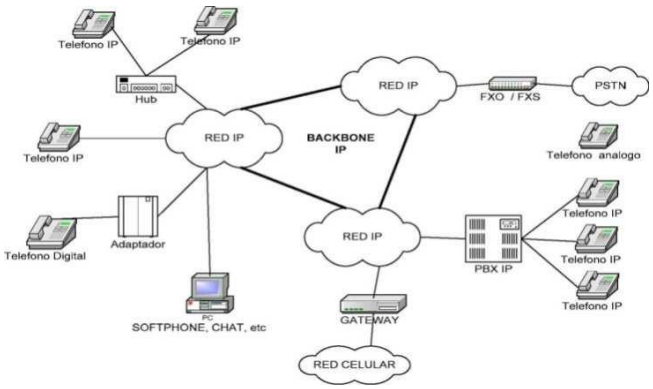


Ilustración 2. Estructura de VoIP

2.1.2. Codecs.

La red VoIP no sería posible sin que se realizara un proceso de compresión y descompresión de Voz, donde primero es codificada desde su estado análogo digital en paquetes IP, que pueden ser enviados a través de la red; finalmente se decodifican a su estado análogo original, es decir nuevamente a voz, en el terminal receptor. Para aplicaciones VoIP los más populares y/o utilizados son: G.711, G.723.1, y el G.729, Además de la ejecución de la conversión de analógico al digital, el CODEC comprime la secuencia de datos, y proporciona la cancelación del eco. La compresión de la forma de onda representada puede permitir el ahorro del ancho de banda. Esto es especialmente interesante en los enlaces de poca capacidad y permite tener un mayor número de conexiones de VoIP simultáneamente (1).

2.1.3. Ventajas de la VoIP.

Resulta fácil enumerar las siguientes ventajas:

- Ahorro en los costos de Administración. Todos los dispositivos telefónicos aprovechan el cableado Ethernet existente, con lo que se simplifica la instalación y mantenimiento del sistema telefónico.
- Permite la integración de aplicaciones propias de una institución o compañía, tales como los sistemas CRM (Customer Relationship Management) y de Centros de contacto (Contact Center).
- Se tienen acceso a servicios adicionales, tales como: mensajería unificada, administración de llamadas entrantes y salientes, control del flujo telefónico, etc...
- Mayor funcionalidad. Integración total con los sistemas PC actuales. Un ejemplo de esta integración es que se puede marcar el teléfono al que queremos llamar directamente desde Outlook.
- Escalable, las PBX convencionales tienen capacidades fijas que al ser sobrepasadas requieren el cambio completo del sistema, esto no es el caso con VoIP.
- Mejora la productividad. VoIP trata a la voz como si fuera cualquier otro tipo de dato, así los usuarios pueden adjuntar documentos a los mensajes de voz o participar en reuniones virtuales usando datos compartidos y video conferencias.

2.1.4. Calidad de la VoIP

Con la migración de la tecnología de conmutación de circuitos a la conmutación de paquetes IP en el proceso transmisión de voz, se introducen nuevas consideraciones, tales como pérdida de paquetes (lost), retardo de paquetes (delay) y el desplazamiento en el tiempo de los paquetes (jitter) (1).

Adicionalmente, problemas antiguos, tales como el eco, variación en el nivel de saturación y ruido de fondo, que eran problemas de los sistemas de conmutación de circuitos, también están presentes en las redes VoIP. La combinación de todos estos problemas que afectan la calidad de voz, se presentan como un gran reto para los planificadores y diseñadores de soluciones de comunicación VoIP.

2.1.5. Factores que influyen en la calidad de voz.

Debido a que la telefonía es un servicio orientado al cliente, los parámetros de medición de la calidad de la voz están basados en la apreciación que los usuarios tienen, sobre todo cuando se trata de una aplicación en tiempo real, como lo es una conversación telefónica. Por lo tanto la meta principal en la medición y evaluación de la calidad de voz en redes conmutadas por paquetes es el desarrollo de indicadores de la percepción que el usuario tiene de la calidad de voz (6), que sean confiables y creíbles, de tal forma que reflejen los efectos específicos de la conmutación de paquetes (3).

2.1.6. Consideración del usuario sobre la calidad de la Voz.

Cuando los usuarios hablan acerca de la calidad de voz, ellos tratan de describir generalmente su reacción a, o su insatisfacción con, uno de los dos siguientes atributos:

Calidad de Conexión. Determinada por lo que es escuchado sobre la conexión.

Usabilidad de la conexión. Determinada por lo que se experimenta en los intercambios de conversación sobre la conexión.

En términos simples uno de los métodos para poder medir la calidad de la voz se realiza a través de encuestas, en las cuales el usuario califica diferentes factores del servicio de voz que utiliza, lo que lo convierte en una estimación subjetiva de la calidad de la voz.

2.2. PROTOCOLOS UTILIZADOS EN VOIP

2.2.1. SIP

SIP son las siglas en inglés del Protocolo para Inicio de Sesión, siendo un estándar RFC 3261 desarrollado por el IETF. SIP es un protocolo de señalización para establecer las llamadas y conferencias en redes IP (7). El inicio de la sesión, cambio o término de la misma, son independientes del tipo de medio o aplicación que se estará usando en la llamada; una sesión puede incluir varios tipos de datos, incluyendo audio, video y muchos otros formatos. Es un protocolo de control

que se encuentra en la capa de aplicación del modelo OSI para crear, modificar y terminar sesiones con uno o más participantes. Las sesiones incluyen: llamadas telefónicas, transferencias de datos multimedia, y conferencias en tiempo real.

Las invitaciones SIP usadas para crear sesiones, llevan consigo la descripción de la sesión y esto permite a los participantes buscar la compatibilidad. SIP utiliza elementos llamados servidores proxy para ayudar a rutear las peticiones a los usuarios de una zona, autentificar y autorizar servicios para éstos, e implementar políticas para el ruteo de llamadas. SIP puede viajar sobre cualquier protocolo de transporte. Los usuarios son denominados user agent y éstos se pueden desplazar a través de la red y obtener diferentes denominaciones y mandar diversos tipos de datos (voz, texto, video). SIP ofrece la ventaja de invitar a los nuevos participantes a la sesión creando una nueva infraestructura en donde todos los user agent pueden registrarse, invitar a nuevas sesiones, modificar las características de la sesión, etc. A pesar de la movilidad del usuario, su identificador puede ser permanente sin importar la red en la que se encuentre (8). SIP posee las siguientes funciones principales:

- Determina los tipos de hosts que pretenden establecer una comunicación.
- Determina la disponibilidad de la persona que recibe la llamada para conectarse.
- Determina el tipo de datos y sus parámetros necesarios que se usarán durante la comunicación.

- Establece los parámetros de la sesión tanto en la persona que llama como en la que es llamada.
- Administra la sesión, en otras palabras, inicialización, transferencia, modificación y terminación de sesiones.

SIP es un protocolo el cual no trabaja de manera única sino que, lo hace en conjunto con otros protocolos de la IETF para crear una arquitectura multimedia más completa. Estos otros protocolos son: el RTP (Real-time Transport Protocol) para el envío de datos y revisar la calidad del servicio, RTSP (Real Time Streaming Protocol) para controlar el envío de datos multimedia, MEGACO (Media Gateway Control) para controlar las conmutaciones con la red PSTN y el SDP (Session Description Protocol), Protocolo de descripción de sesión para la descripción de las diferentes sesiones. SIP funciona tanto con IPv4 como IPv6. La intención de SIP es la comunicación entre dispositivos multimedia. SIP hace viable esta comunicación gracias a dos protocolos que son RTP/RTCP y SDP (4). El protocolo RTP se usa para transportar los datos de voz en tiempo real (igual que para el protocolo H.323, mientras que el protocolo SDP se usa para la negociación de las capacidades de los participantes, tipo de codificación, etc.). SIP fue diseñado de acuerdo al modelo de Internet. Es un protocolo de señalización extremo a extremo que implica que toda la lógica es almacenada en los dispositivos finales (salvo el ruteado de los mensajes SIP). El estado de la conexión es también almacenado en los dispositivos finales. El costo a pagar por esta capacidad de distribución y su gran escalabilidad es una sobrecarga en la cabecera de los mensajes producto de tener que mandar toda

la información entre los dispositivos finales. SIP es un protocolo de señalización a nivel de aplicación para establecimiento y gestión de sesiones con múltiples participantes. Se basa en mensajes de petición y respuesta y reutiliza muchos conceptos de estándares anteriores como HTTP y SMTP. SIP soporta funcionalidades para el establecimiento y finalización de las sesiones multimedia: localización, disponibilidad, utilización de recursos, y características de negociación. Para implementar estas funcionalidades, existen varios componentes distintos en SIP. Existen dos elementos fundamentales, los agentes de usuario (UA) y los servidores. User Agent (UA): consisten en dos partes distintas, el User Agent Client (UAC) y el User Agent Server (UAS). Un UAC es una entidad lógica que genera peticiones SIP y recibe respuestas a esas peticiones. Un UAS es una entidad lógica que genera respuestas a las peticiones SIP. Ambos se encuentran en todos los agentes de usuario, así permiten la comunicación entre diferentes agentes de usuario mediante comunicaciones de tipo cliente- servidor. Los servidores SIP pueden ser de tres tipos:

Proxy Server: retransmiten solicitudes y deciden a qué otro servidor deben remitir, alterando los campos de la solicitud en caso necesario. Es una entidad intermedia que actúa como cliente y servidor con el propósito de establecer llamadas entre los usuarios. Este servidor tiene una funcionalidad semejante a la de un Proxy HTTP que tiene una tarea de encaminar las peticiones que recibe de otras entidades más próximas al destinatario. Existen dos tipos de Proxy Servers: Statefull Proxy y Stateless Proxy.

- Statefull Proxy: mantienen el estado de las transacciones durante el procesamiento de las peticiones. Permite división de una petición en varias (forking[1]), con la finalidad de la localización en paralelo de la llamada y obtener la mejor respuesta para enviarla al usuario que realizó la llamada.
- Stateless Proxy: no mantienen el estado de las transacciones durante el procesamiento de las peticiones, únicamente reenvían mensajes.
- Register Server: es un servidor que acepta peticiones de registro de los usuarios y guarda la información de estas peticiones para suministrar un servicio de localización y traducción de direcciones en el dominio que controla.
- Redirect Server: es un servidor que genera respuestas de redirección a las peticiones que recibe. Este servidor reencamina las peticiones hacia el próximo servidor. La división de estos servidores es conceptual, cualquiera de ellos puede estar físicamente en una única máquina, la división de éstos puede ser por motivos de escalabilidad y rendimiento.

Mensajes y respuestas SIP.

SIP utiliza Métodos / Solicitudes y correspondientes Respuestas para establecer una sesión de llamada. SIP además es un protocolo textual que usa una semántica semejante a la del protocolo HTTP (7). Los UAC realizan las peticiones y los UAS

[1] Bifurcación. hace referencia a la creación de una copia de sí mismo por parte de un programa, que entonces actúa como un "proceso hijo" del proceso originario, ahora llamado "padre". Los procesos resultantes son idénticos, salvo que tienen distinto número de proceso.

retornan respuestas a las peticiones de los clientes. SIP define la comunicación a través de dos tipos de mensajes. Las solicitudes (métodos) y las respuestas (códigos de estado) emplean el formato de mensaje genérico establecido en el RFC 2822 , que consiste en una línea inicial seguida de uno o más campos de cabecera (headers), una línea vacía que indica el final de las cabeceras, y por último, el cuerpo del mensaje que es opcional.

Las peticiones SIP son caracterizadas por la línea inicial del mensaje, llamada Request-Line, que contiene el nombre del método, el identificador del destinatario de la petición (Request-URI) y la versión del protocolo SIP. Existen seis métodos básicos SIP (definidos en RFC 254) que describen las peticiones de los clientes:

INVITE: Este método indica que el usuario o servicio es invitado a participar en una sesión. Incluye una descripción de sesión y, para llamadas de full dúplex, la parte llamante indica el tipo de medio. Una respuesta con éxito a una invitación INVITE de dos partes (respuesta 200 OK) incluye el tipo de medios recibidos por la parte llamada. Con este simple método, los usuarios pueden reconocer las posibilidades del otro extremo y abrir una sesión de conversación con un número limitado de mensajes e idas y vueltas.

ACK: Estas respuestas corresponden a una petición INVITE. Representan la confirmación final por parte del sistema final y concluye la transacción indicada por el comando INVITE. Si la parte llamante incluye una descripción de la sesión, los parámetro en la petición INVITE se utilizan como los predeterminados.

OPTIONS: Este método permite consultar y reunir posibilidades de agentes de usuarios y servidores de red. Sin embargo, esta petición no se utiliza para establecer sesiones.

BYE: Este método se utiliza por las partes que llaman y son llamadas para liberar una llamada. Antes de liberar realmente la llamada, el agente de usuario envía esta petición al servidor indicando el deseo de terminar la sesión.

CANCEL: Esta petición permite que los agentes de usuario y servidores de red cancelen cualquier petición que este en progreso. Esto no afecta a las peticiones terminadas en las que las respuestas finales ya fueron recibidas.

REGISTER: Ese método se utiliza por los clientes para registrar información de localización con los servidores SIP. Sin embargo, existen otros métodos adicionales que pueden ser utilizados, publicados en otros RFCs como los métodos INFO, SUBSCRIBER, etc.

Respuestas (Códigos de estado) SIP.

Después de la recepción e interpretación de un mensaje de solicitud SIP, el receptor del mismo responde con un mensaje. Este mensaje, es similar al anterior, difiriendo en la línea inicial, llamada Status-Line, que contiene la versión de SIP, el código de la respuesta (Status–Code) y una pequeña descripción (Reason-Phrase). El código de la respuesta está compuesto por tres dígitos que permiten clasificar los diferentes tipos existentes. El primer dígito define la clase de la respuesta.

Una de las funciones de los servidores SIP es la localización de los usuarios y resolución de nombres. Normalmente, el agente de usuario no conoce la dirección IP del destinatario de la llamada, sino su e-mail. Las entidades SIP identifican a un usuario con las SIP URI (Uniform Resource Identifiers) definido en el RFC 2396. Una SIP URI tiene un formato similar al del email, consta de un usuario y un dominio delimitado por una @, como muestra los siguientes casos:

usuario@dominio, donde dominio es un nombre de dominio completo. usuario@equipo, donde equipo es el nombre de la máquina.

usuario@dirección_ip, donde dirección_ip es la dirección IP del dispositivo.

número_teléfono@gateway, donde el gateway permite acceder al número de teléfono a través de la red telefónica pública.

La solución de identificación de SIP, también puede ser basada en el DNS descrito en el RFC 3263, donde se describen los procedimientos DNS utilizados por los clientes para traducir una SIP URI en una dirección IP, puerta y protocolo de transporte utilizado, o por los servidores para retornar una respuesta al cliente en caso de que la petición falle

Clase de respuesta	Código de estado	Explicación
Informativa	100	Tratando
	180	Sonando
	181	La llamada esta siendo reenviada
	182	Puesta en cola
	183	Progreso de sesión
Success	200	OK
	300	Elección múltiple
	301	Movida permanente
	302	Movida temporalmente
	303	Véase otra
	305	Utilizar Proxy
	380	Servicio alternativo
Errores de solicitud	400	Petición defectuosa
	401	No autorizado
	402	Se requiere pago
	403	Prohibido

Ilustración 3. Códigos SIP

	404	No encontrado
	405	Método no permitido
	406	No aceptable
	407	Se requiere autenticación de proxy
	408	Se acaba tiempo de petición
	409	Conflicto
	410	Se ha marchado
	411	Se requiere longitud
	413	Entidad pedida demasiado larga
	414	URL pedido demasiado largo
	415	Tipo de medio no soportado
	420	Extensión errónea
	480	No disponible temporalmente
	481	Segmento de llamada o transacción no existe
	482	Detectado bucle
	483	Demasiados saltos
	484	Dirección incompleta
	485	Ambiguo
	486	ocupado
Errores de servidor	500	Error interno de servidor
	501	Sin implementar
	502	Gateway erróneo
	503	Servicio no disponible
	504	Gateway fuera de tiempo
	505	Versión SIP no soportada
	600	Ocupado en todos partes
	603	Rechazado
	604	No existe en ningún sitio
	606	No aceptable

Ilustración 4. Códigos SIP

El protocolo SDP (Session Description Protocol) RFC 2327 se utiliza para describir sesiones multicast en tiempo real, siendo útil para invitaciones, anuncios, y cualquier

otra forma de inicio de sesiones. La propuesta original de SDP fue diseñada para anunciar información necesaria para los participantes y para aplicaciones de multicast MBONE (Multicast Backbone). Actualmente, su uso está extendido para el anuncio y la negociación de las capacidades de una sesión multimedia en Internet.

Puesto que SDP es un protocolo de descripción, los mensajes SDP se pueden transportar mediante distintos protocolos con SIP, RTSP, correo electrónico con aplicaciones MIME o protocolos como HTTP. Como el SIP, el SDP utiliza la codificación del texto. Un mensaje del SDP se compone de una serie de líneas, denominados campos, dónde los nombres son abreviados por una sola letra, y está en una orden requerida para simplificar el análisis. El SDP no fue diseñado para ser fácilmente extensible. A continuación se analizará una llamada. En una llamada SIP hay varias transacciones SIP. Una transacción SIP se realiza mediante un intercambio de mensajes entre un cliente y un servidor.

Las dos primeras transacciones corresponden al registro de los usuarios. Los usuarios deben registrarse para poder ser encontrados por otros usuarios. En este caso, los terminales envían una petición REGISTER, donde los campos from y to corresponden al usuario registrado.

El servidor Proxy, que actúa como Register, consulta si el usuario puede ser autenticado y envía un mensaje de OK en caso positivo. La siguiente transacción corresponde a un establecimiento de sesión.

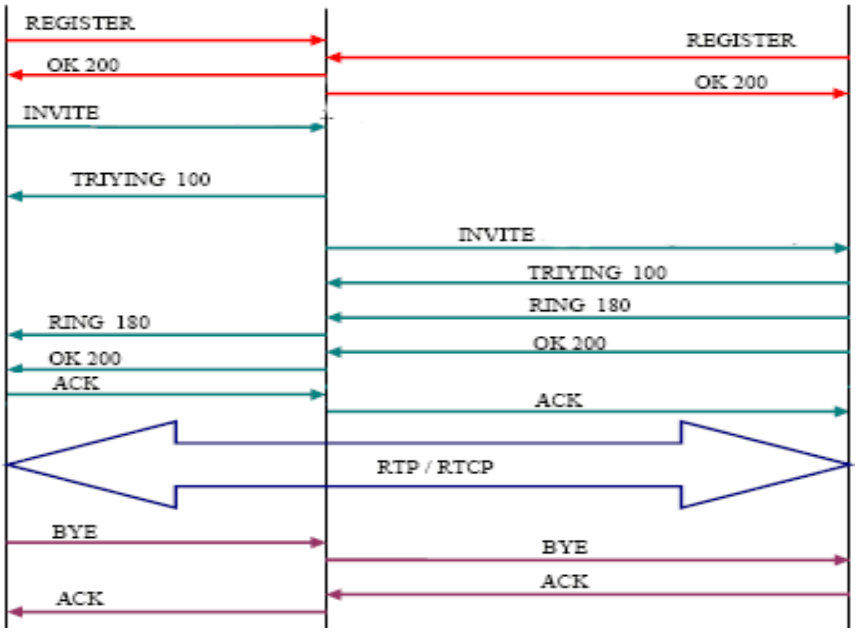


Ilustración 5. Sesión de Invitación y Aceptación de petición

Esta sesión consiste en una petición INVITE del usuario al proxy. Inmediatamente, el proxy envía un TRYING 100 para parar las retransmisiones y reenvía la petición al usuario B. El usuario B envía un Ringing 180 cuando el teléfono empieza a sonar y también es reenviado por el proxy hacia el usuario A. Por último, el OK 200 corresponde a aceptar la llamada (el usuario B descuelga) (9). En este momento la llamada está establecida, pasa a funcionar el protocolo de transporte RTP con los parámetros (puertos, direcciones, codecs, etc.) establecidos en la negociación mediante el protocolo SDP. La última transacción corresponde a una finalización de sesión. Esta finalización se lleva a cabo con una única petición BYE enviada al Proxy, y posteriormente reenviada al usuario B. Este usuario contesta con un OK 200 para confirmar que se ha recibido el mensaje final correctamente.

SIP es parte de los estándares de IETF y se modela en otros protocolos de Internet, tales como SMTP y HTTP. Se utiliza para establecer y cambiar entre unos o más usuarios en una red IP. En la Figura 2.8 se muestra las capas del protocolo SIP.

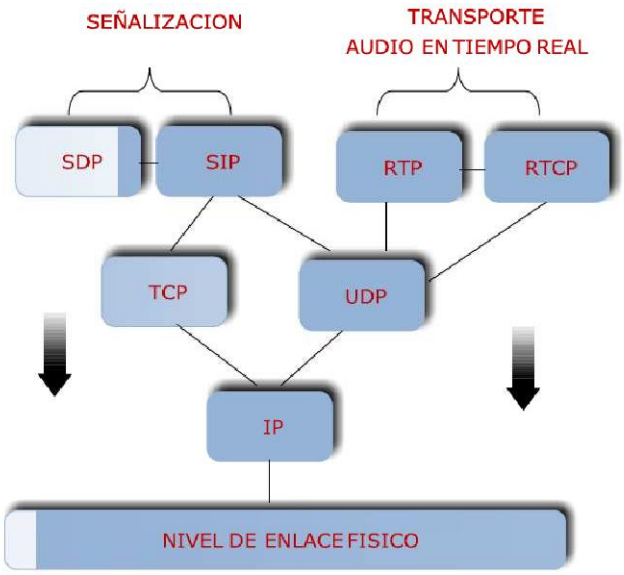


Ilustración 6. Modelo de capas de VoIP

2.2.2. RTP

RTP es el más popular de los protocolos de transporte de VoIP. Se especifica en el RFC 1889 bajo el título de "RTP: Un protocolo de transporte para aplicaciones en tiempo real." Este RFC describe RTP y RTCP. Pues los nombres sugerirían que estos dos protocolos son necesarios para soportar aplicaciones entiempo real como voz y vídeo. RTP funciona sobre la capa de UDP, que no evita pérdida de paquetes ni garantiza el orden correcto para la entrega de paquetes (1).

Los paquetes de RTP superan esos defectos incluyendo los números de serie que ayudan a RTP para detectar los paquetes perdidos y para asegurar la entrega del paquete en el orden correcto. Los paquetes de RTP incluyen una etiqueta de fecha/hora, donde indica el tiempo en que el paquete se envió de la fuente.

Esta etiqueta de fecha/hora ayuda con la sincronización del usuario de destino, en donde se calcula el retraso que se tiene y el jitter, dos detractores muy importantes de calidad de la voz. RTP no tiene la capacidad de corregir el retraso ni el jitter, pero proporciona la información adicional a un uso más alto de la capa, de modo que pueda hacer determinaciones en cuanto a cómo un paquete de voz o de datos se maneja lo mejor posible.

RTCP proporciona un número de mensajes que se intercambian entre los usuarios de sesión y que proporcionan la regeneración en la sesión. El tipo de información incluye los detalles, tales como los números de los paquetes perdidos de RTP, retraso y jitter entre llegadas. Mientras que los paquetes de voz se transportan en paquetes de RTP, los paquetes de RTCP transfieren la regeneración de la calidad. Siempre que una sesión de RTP se abra, una sesión (9) de RTCP también se abre, es decir, cuando un número de acceso de UDP se asigna a una sesión de RTP para la transferencia de los paquetes de los medios, otro número de acceso se asigna para los mensajes de RTCP. En la Ilustración 7 se muestran la pila de RTP (7).

Ilustración 7. Capas de RTP

2.2.3. RTCP

RTCP permite intercambios de información de control entre los participantes de la sesión con el fin de proporcionar la regeneración de la calidad. Esta regeneración se utiliza para detectar y para corregir problemas en la distribución. La combinación del multicast de RTCP y de IP permite al operador dela red monitorear la calidad. RTCP proporciona la información en la calidad de una sesión de RTP. RTCP autoriza a los operadores de la red para obtener la información sobre el retraso, el Jitter y la pérdida de paquetes, así tomar la acción correctiva para mejorar la calidad (3).

2.2.4. Software utilizado en telefonía IP.

Es válido hacer mención, que cuando se tiene el reto de crear soluciones tecnológicas, cuya implementación es una mezcla de Hardware y Software, se tienen dos opciones técnicamente válidas: Hardware y Software propietario o Hardware y Software Libre (Open Source). Las ventajas de usar Software Libre, no se remiten únicamente al aspecto económico, ya que a diferencia del Software

gratuito, el software libre ofrece siempre los archivos fuente, que le permiten a cualquier implementador de soluciones, usarlos de tal forma que puede adecuar esa aplicación a las necesidades específicas existentes. Una de las características más importantes de la implementación del Laboratorio para el Estudio de protocolos de VoIP, es que estará basado en software Libre. Existe una gran variedad de Software GPL o libre y se han seleccionado como herramientas de trabajo los más sobresalientes por su versatilidad, compatibilidad y desempeño (1).

2.3. FACTORES QUE DETERMINAN LA CALIDAD DE LA VOZ EN SISTEMAS VOIP

2.3.1. Latencia o Retardo

Una red Ethernet cableada o inalámbrica no fue diseñada para aplicaciones en tiempo real o con una garantía en la entrega de paquetes. La congestión de la red inalámbrica, sin diferenciación del tráfico, puede rápidamente hacer la voz inutilizable. Al procesamiento de la señal de voz en los puntos de envío y recepción, incluyendo el tiempo necesario para codificar o decodificar la señal de voz analógica o digital en el sistema de codificación de voz elegido, se le suma al retraso. La compresión de la señal de voz también aumentará el retraso, entre mayor sea la compresión mayor será el retraso. En caso de que los costos de ancho de banda no sean una preocupación, un prestador de servicios puede utilizar el códec G.711, que tiene una velocidad de descompresión de (64 Kbps), que representa un mínimo de retraso debido a la compresión (10).

En la parte de la transmisión, el retardo por paquetización es otro factor que debe tomarse en cuenta. El retraso de paquetización es el tiempo que tarda para formarse un paquete con los datos, cuanto mayor sea el tamaño del paquete se necesita más tiempo. El uso de tamaños de paquetes más cortos pueden reducir este retraso, pero esto provocará que se incremente la actividad en la red porque más paquetes han de ser enviados, y todos contendrán información similar en la cabecera. El equilibrio entre calidad de voz, el retraso por paquetización y el uso eficiente del ancho de banda son muy importantes a la hora de proveer un servicio de VoIP.

¿Cuánta retraso puede ser demasiado? De todos los factores que degradan las comunicaciones VoIP, el retardo es el mayor. La latencia de menos de 100 ms no afecta la voz. Sin embargo, la latencia superior a 120 ms es discernible para la mayoría, y en 150 ms la calidad de voz ha disminuido de forma notable. Un desafío para los futuros proveedores de servicios de VoIP es obtener una latencia de cualquier conversación en su red, que no sobrepase los 100 ms. Los seres humanos son perceptibles a los retrasos de más de alrededor de 200 ms. La ITU-T G.114 especifica que el retraso no debe ser superior a 150 ms en un sentido del envió de la información o 300 ms de ida y vuelta. El dilema es que, si bien aplicaciones por ejemplo (correo electrónico,) pueden tolerar una cantidad de retraso, por lo general estas aplicaciones tratan de consumir cada bit de la capacidad de la red que pueden. En contraste las aplicaciones de voz sólo necesitan pequeñas cantidades de la red, pero esa suma tiene que estar disponible en un momento inmediato (4).

El retraso experimentado por una llamada se produce en el lado de la transmisión, en la red y en lado de la recepción. La mayor parte del retraso en el lado de la transmisión es debido al retraso producido por el códec. En la red, la mayor parte del retraso se debe al tiempo de la transmisión (señalización y propagación) y el tiempo en las colas del ruteador. Por último, el jitter, el procesamiento y en algunas implementaciones añaden retraso en el lado de la recepción.

El retraso introducido por el codificador de voz puede dividirse en algorítmico y el retraso de procesamiento. El algoritmo de retraso se produce debido a la elaboración del bloque de procesamiento, ya que el codificador produce un conjunto de bits que representan un bloque de muestras de voz.

2.3.2. Paquetes perdidos.

En redes, un porcentaje de los paquetes pueden perderse o retrasarse, especialmente durante los períodos de congestión. Asimismo, algunos paquetes son descartados debido a errores durante la transmisión. Paquetes perdidos, retrasados, dañados y deteriorados, se ve reflejado en la calidad de voz.

En técnicas convencionales de corrección de errores utilizadas en otros protocolos, los bloques de datos que contengan errores se descartan, y los que recibe la computadora solicitan la retransmisión del paquete. De este modo el mensaje que es finalmente entregado al usuario no es exactamente el mismo mensaje que se originó. Porque sistemas VoIP son sensibles al tiempo y no pueden esperar para la retransmisión, los sistemas más sofisticados de detección y corrección de errores

utilizan sonido para llenar huecos en las llamadas. Este proceso es una parte de la voz del emisor y luego utilizando un complejo algoritmo para aproximar el contenido de los paquetes que faltan, el nuevo sonido de información es creado para mejorar la comunicación. De este modo, el sonido escuchado por el receptor no es exactamente el sonido de transmisión, sino más bien parte de los que han sido creados por el sistema para mejorar el sonido emitido (2).

La mayoría de las pérdidas de los paquetes se producen en los ruteadores, ya sea debido a las altas transferencias de carga o alta carga de enlace. En ambas situaciones, los paquetes en las colas podrían ser eliminados. Otra fuente de pérdida de paquetes son los errores en los enlaces de transmisión.

La configuración de errores y colisiones podrían también generar pérdidas de paquetes. En aplicaciones de tiempo no real, las pérdidas de paquetes se resuelven en la capa del protocolo de transmisión (TCP). Para la telefonía esto no es una solución viable ya que se volvería a transmitir los paquetes que llegan demasiado tarde y no sería de mucha utilidad.

Tal vez el principal desafío para VoIP es que en relación con las redes cableadas, los paquetes se reducen en una tasa excesiva (más de 30%). Esto puede conducir a la distorsión de la voz en la medida en que la conversación va siendo difícil. En pasarelas de VoIP diseñados para redes de cable, una soluciones usar un buffer de jitter.

2.3.3. Jitter

El jitter es una consecuencia de las redes de datos no orientadas a conexión y basadas en conmutación de paquetes. Como la información se modera en paquetes, cada uno de los paquetes puede seguir una ruta distinta para llegar al destino. El jitter se define técnicamente como "*la variación en el tiempo en la llegada de los paquetes, causada por congestión de red, perdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino". Las comunicaciones en tiempo real (como VoIP) son especialmente sensibles a este efecto* (8).

2.3.4. MOS

La industria telefónica emplea un sistema de calificación subjetiva conocido como MOS, para medir la calidad de sus conexiones telefónicas. Las técnicas de medición se definen en el ITU-T P.800 y se basan en las opiniones de muchos ensayos hechos por voluntarios que escuchan una muestra de tráfico de voz y califican la calidad de la transmisión. Los voluntarios escuchan una variedad de muestras de voz, donde se les piden considerar diversos factores, como pérdida de paquetes, ruido en el circuito, eco, distorsión, retraso de paquetes y otros problemas de transmisión. Luego los voluntarios califican las muestras de voz, con una calificación de 1 a 5, siendo 5 "excelente" y 1 "malo". (1)Las muestras de voz son conferidas al MOS. Una puntuación de 4 en el MOS, significa tener una calidad igual de buena, que en la red pública telefónica (4).

Ilustración 8. MOS de Usuario

2.4. ASTERISK

Asterisk es una aplicación de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica (PBX). Como cualquier PBX, se puede conectar un número fijo de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una RDSI tanto básicos como primarios.

Mark Spencer, de Digium, originalmente creó Asterisk y actualmente es su principal desarrollador, junto con otros programadores que han contribuido a corregir errores y añadir novedades y funcionalidades (3).

Asterisk incluye muchas características anteriormente sólo disponibles en costosos sistemas propietarios PBX como buzón de voz, conferencias, IVR, distribución automática de llamadas, y otras muchas más. Los usuarios pueden crear nuevas funcionalidades escribiendo un dialplan en el lenguaje de script de Asterisk o

añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación soportado por Linux.

2.4.1. Historia

Asterisk fue creada en 1999 por Mark Spencer de la empresa Digium y concedida a la comunidad con licencia libre tras lo cual se han recibido muchas colaboraciones y mejoras por parte de muchos desarrolladores libres y empresas sin solicitar nada a cambio.

Poco a poco, esta aplicación se ha convertido en la evolución de las tradicionales centralitas analógicas y digitales permitiendo también integración con la tecnología más actual: VoIP. Asterisk se convierte así en el mejor, más completo, avanzado y económico sistema de comunicaciones existente en la actualidad.

Otro aliciente es su capacidad de ser programada, permitiendo realizar labores que hasta el día de hoy lo llevaban realizando sistemas extremadamente costosos y complicados y, gracias a Asterisk, esta misma labor se realiza de una forma más económica lo que fomenta el uso de sistemas libres como Linux y estándares abiertos como SIP (4).

2.4.2. Esquema Conceptual

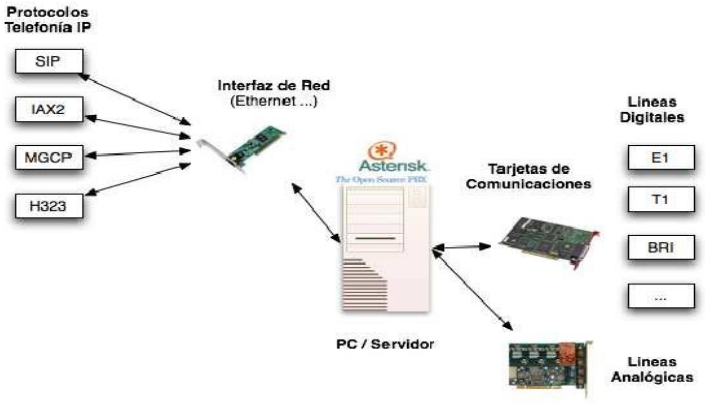


Ilustración 9. Esquema conceptual de central de VoIP. Fuente Digium

2.4.3. Arquitectura Base

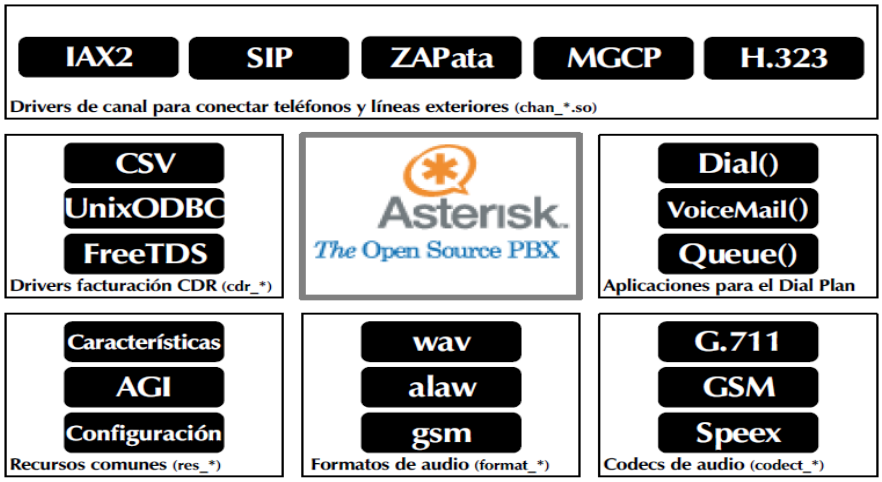


Ilustración 10. Arquitectura Base de Asterisk. Fuente DIgium

2.4.4. Funcionalidades

Tipo Centralita

Algunas de las funcionalidades, tipo centralita, más interesantes:

Música en espera

Registro de llamadas en BD Buzón de Voz por Mail Llamada en espera

Salas de Conferencia

Caller ID

Buzón de Voz personal Bloqueo de Caller ID Colas de llamada Timbres distintivos

Colas con prioridad (1)

Funcionalidades Avanzadas

IVR: Interactive Voice Response, gestión de llamadas con menús interactivos.

LCR: Least Cost Routing, encaminamiento de llamadas por el proveedor VoIP más económico.

AGI: Asterisk Gateway Interface, integración con todo tipo de aplicaciones externas.

AMI: Asterisk Management Interface, gestión y control remoto de Asterisk.

Configuración en base de datos: usuarios, extensiones, proveedores. (1)

Requisitos Técnicos del sistema

Previa la instalación de Asterisk, es necesario contar con los requerimientos mínimos para poder ser instalado.

Procesador a 500MHz (Pentium3) con 128 MB en RAM

2GB en disco duro como mínimo. Recomendados:

Procesador a 1.5 GHz (Pentium 4)

256 MB en RAM

10 GB en disco duro.

Elección del sistema operativo

Asterisk puede ser instalado en las siguientes plataformas: GNU/Linux 2.x MacOSX 10.x

BSD

MS Windows (1)

En este documento se detallará la instalación en plataformas GNU/Linux debido a que la telefonía es un servicio totalmente crítico y la elección de la

plataforma donde se instalará Asterisk es clave. La estabilidad de las plataformas BSD y GNU/Linux está más que probada por infinidad de usuarios.

Administración

Arranque

Asterisk es un demonio que se ejecuta en segundo plano. Se invoca con el comando *"asterisk"*, una vez ejecutado devuelve el control de la shell, haciendo un 'detach' se puede comprobar que se está ejecutando correctamente con un listado de procesos habitual:

ps aux | grep Asterisk

Conexión al CLI

En este punto el programa Asterisk está en funcionamiento en la configuración de */etc/Asterisk.*

Asterisk soporta un intérprete de comandos (CLI: Command Line Interface), del estilo de muchos routers y para conectarse basta con ejecutar el comando:

asterisk r

El intérprete de comandos de Asterisk es bastante potente, y permite controlar y monitorizar gran parte de la situación de la centralita. Soporta el empleo de la tecla <Tabulador>, al estilo de las consolas de UNIX/GNU Linux, por lo que para ver un

listado de todos los comandos disponibles, basta con presionar varias veces la tecla. (1)

Para ver los posibles argumentos de un comando o completar un parámetro largo o complicado.

Como primer comando del CLI, se puede verificar la versión de Asterisk instalada:
CLI> show version

Asterisk 1.6..1 built by root @ pbuxubuntu01 on a i686 running Linux on 20060117

23:08:46 UTC

Obtención del tiempo en ejecución:

CLI> show uptime

System uptime: 5 weeks, 5 days, 2 hours, 29 minutes, 28 seconds

CLI> detención

Es posible realizar una desconexión del CLI de Administración con 'quit'. Asterisk continuará ejecutándose en segundo plano.

Para detener al propio Asterisk desde el CLI, se puede utilizar el comando stop, en sus tres variantes:

stop now: Detiene Asterisk al momento

stop when convenient: Detiene Asterisk cuando no haya carga. (1)

stop gracefully: Detiene asterisk cuando no haya carga y deja de aceptar peticiones de llamadas a partir de este momento.

Verbose

Nivel de “Verbose” es el valor que indica la cantidad de mensajes que se recibirán sobre los eventos generales del sistema. Cuanto más alto, más información sobre lo que sucede en la centralita se recibirá, este nivel, se puede establecer de varias formas:

Al arrancar el demonio:

sudo asterisk vvvvvv

Al conectarse al demonio:

sudo asterisk rvvvvvvvv

Desde el CLI:

CLI> Set Verbose 30

Debug

Nivel de “Debug” es el valor que indica la cantidad de mensajes que se recibirán sobre los eventos generales del sistema, pero utilizado normalmente para depurar problemas de drivers o de aplicaciones. (1)

Este nivel, se puede establecer de varias formas: Al arrancar el demonio:

sudo asterisk dddd

Al conectarse al demonio:

sudo asterisk rdddd

Desde el CLI:

CLI> Set Debug 30

Terminología

Canal: Es una conexión que conduce una llamada entrante o saliente en el sistema Asterisk. La conexión puede venir o salir hacia telefonía tradicional analógica o digital o VozIP.

Por defecto Asterisk soporta una serie de canales, los más importantes: H.323, IAX2, SIP, MGCP: Protocolos VozIP

Console: GNU Linux OSS/ALSA sound system. Zap: Lineas analógicas y digitales (1).

Dialplan: Se trata de la configuración de la centralita Asterisk que indica el trayecto que sigue una llamada desde que entra o sale del sistema hasta que llega a su punto final.

Se trata en líneas generales del comportamiento lógico de la centralita.

Extension: En telefonía tradicional, las extensiones se asocian con teléfonos, interfaces o menús. En Asterisk, una extensión es una lista de comandos a ejecutar.

Las extensiones se acceden cuando:

Se recibe una llamada entrante por un canal dado. El usuario que ha llamado marca la extensión.

Se ejecuta un salto de extensiones desde el Dialplan de Asterisk.

Contexto (Context): El Dialplan o lógica de comportamiento de Asterisk se divide en uno o varios contextos. Un contexto es una colección de extensiones.

Los contextos existen para poder diferenciar el 'lugar' donde se encuentra una llamada, para:

Aplicar políticas de seguridad: Asterisk no se comporta igual cuando llama un usuario y marca el 1 y cuando un usuario local marca el mismo

Menús y submenús diferenciados. (1)

En general, es una forma de diferenciación.

Aplicación (Application): Asterisk ejecuta secuencialmente los comandos asociados a cada extensión. Esos comandos son realmente aplicaciones que controlan el comportamiento de la llamada y del sistema en sí. Algunos ejemplos:

Hangup: Colgar la llamada.

Monitor: Comenzar la grabación a disco de la llamada. Dial: Realiza una llamada saliente.

Goto: Salta a otra extensión o contexto. PlayBack: Reproduce un fichero de sonido.

Configuración de Asterisk

Asterisk puede configurarse desde varios puntos, los más importantes son:

Pare desde el propio CLI

Desde los ficheros de configuración (.conf) en /etc/asterisk

La configuración se carga al iniciar Asterisk, por lo que para aplicar cualquier cambio será necesario recargarla, para ello basta con ejecutar el comando reload en el cli:

CLI> reload

Ficheros de Configuración más importantes

Asterisk se configura desde múltiples ficheros de configuración, cada uno para una determinada área los más importantes son:

Fichero de configuración maestro: asterisk.conf Fichero de configuración de módulos: modules.conf Canales:

iax.conf: Canales Inter Asterisk eXchange sip.conf: Canales SIP

zapata.conf: Telefonía analógica y digital h323.conf: Canales H323

mgcp.conf: Canales MGCP Dialplan:

extensions.conf: El propio Dialplan.

features.conf: Dialplan para métodos complementarios (transferencias, call parking, grabación de llamadas bajo demanda)

Configuración de aplicaciones del Dialplan: meetme.conf: Para salas de conferencias. musiconhold.conf: Configuración de la música en espera. queues.conf: Configuración de Colas de llamadas. voicemail.conf: Configuración de los buzones de Voz.

Configuración para canales de Voz IP SIP

Los ficheros a manipular son sip.conf e iax.conf, la instalación crea ficheros de ejemplo con la syntaxis bastante comentada a modo de guía.

SIP.CONF

En este fichero se definen: Variables generales de SIP. Clientes SIP.

Servidores SIP. Sección General

En primer lugar existe la sección [general], donde se definen variables globales y aspectos por defecto para todos los canales SIP.

La syntaxis es la siguiente: [general] variable1=valor1 variable2=valor2

register => usuario : password @ servidorregistrar register => ....

Register pide a Asterisk que registre su presencia en el SIP, de esta forma, el proveedor sabrá 'donde estamos', solo vale para esa localización. En ningún caso es suficiente para poder hacer llamadas.

Las variables generales más importantes son:

allow y disallow: indican los codecs permitidos / no permitidos. dtmfmode: permite especificar el método por el cual seenviaran los tonos (digitos pulsados durante la conversación), valores posibles:

nat: Informa a Asterisk del tipo de NAT en el que se encuentra. externip: Dirección Pública tras el NAT.

context: Contexto por defecto donde entraran las llamadas entrantes por SIP.

port: Puerto en el que escuchar (5060). Clientes y Servidores

En sip.conf se definen tanto los clientes que se conectarán a Asterisk, como los proveedores que se utilizaran para encaminar llamadas. Conceptualmente, se distinguen (versión 1.2): (1)

user: Envía llamadas a Asterisk

peer: Recibe llamadas de Asterisk (proveedor). friend: Recibe y Envía llamadas (usuario).

La syntaxis para definir un friend o un peer es: [nombre]

type = friend / peer variable = valor viarable2 = valor

Las variables más importantes que deben ser configuradas inicialmente son:

type: peer / friend

context: Contexto donde entraran las llamadas generadas. nat: Indica si el usuario o peer se encuentran tras un nat. host: IP remota o dynamic.

username: nombre de usuario. secret: contraseña de acceso.

allow y disallow: Configuraciones de codecs específicas para cada friend/peer.

qualify: Evalúa el estado del extremo SIP para conocer su accesibilidad y latencia.

Verificación de la configuración con el CLI

Mediante el comando “reload” en el CLI de Asterisk, significa que recargue la configuración. Aunque es posible recargar de forma independiente:

CLI> sip reload

Una vez recargada, se puede comprobar los “friends” que hemos definido con el comando:

sip show users.

Para ver los “peers” definidos:

sip show peers

Es importante recalcar que los “friends” son también “peers”, ya que pueden recibir y enviar llamadas. (1)

Desde el CLI, se puede consultar si Asterisk se ha 'registrado' correctamente en los registros configurados en la sección general con el comando:

sip show registry

2.4.5. Introducción al Dialplan

Cuando un usuario marca un determinado número la manera en la que se puede llamar utilizando alguno de los proveedores configurados es mediante el Dialplan. (1)

El Dialplan es el corazón del comportamiento de Asterisk, en él se configura toda la lógica en lenguaje natural, un ejemplo muy sencillo podría ser el siguiente:

Cuando un usuario marca un número:

Si el número empieza por 0, llamar al destino utilizando un proveedor externo.

Si el número tiene 3 cifras y empieza por 1, llamar a un determinado usuario del a centralita.

Si cuando llamamos a ese usuario, no entra la llamada en 60 segundos, se reproduce un mensaje de alerta.

En situaciones normales, el dialplan se puede complicar considerablemente.

Arquitectura del dialplan

El dialplan se define en extensions.conf, su “forma” genérica se asemeja al esquema de la figura:

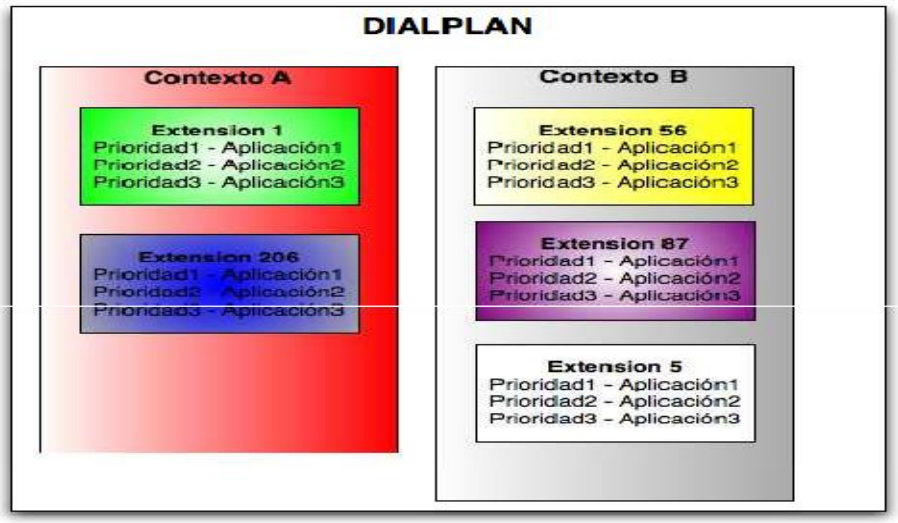


Ilustración 11. Escenario de Dialplan

Aspectos a tener en cuenta

Si no existe la prioridad N + 1, Asterisk no salta a la siguiente prioridad (N+2). Existen aplicaciones como (1)Goto que modifican el flujo de la ejecución. Algunas extensiones especiales:

s: Extensión por defecto cuando una llamada entra en un contexto sin número destino asociado.

i: Cuando el usuario marca una extensión incorrecta. t: Cuando se produce un timeout.

Es posible analizar cómo ha 'leído' Asterisk el fichero extensions.conf desde el CLI, con esto confirmamos posibles errores de syntaxis, etc .. El comando es:

CLI> show dialplan [contexto]

Ejemplo:

CLI> show dialplan desde_usuarios

[Context 'desde_usuarios' created by 'pbx_config']

'_1XX' => 1. Macro(llamarusuario|${EXTEN}) [pbx_config] Include => 'servicios' [pbx_config]

Include => 'fijos' [pbx_config] Include => 'moviles' [pbx_config] irontur*CLI>= 1 extensions (1 priorities) in 1 context. =

Detalles sobre extensiones

Las extensiones son los dígitos, el destino de llamada que ha marcado el usuario cuando llama. Cuando un usuario SIP marca el 105, el flujo de ejecución salta a la extensión 105 en el contexto asociado a ese usuario SIP.

Asterisk, cuando recibe una llamada, la procesa en una determinada extensión. Pero puede quedarse a la espera (“marque el 1”, “marque el 2”...) y saltar a la extensión que marca la llamada entrante: Caso de los menús IVR (1)

Manejo de Extensiones

La syntaxis general en el dialplan es:

exten => EXTENSION, PRIORIDAD, Aplicación

En el caso de llamadas internas o funcionamiento simple, las extensiones son conocidas.

Pero cuando un usuario llama a un número que no se pre-conoce se debe utilizar patrones en las extensiones.

Patrones de Coincidencia

Para indicar patrones, se utiliza el carácter: "_" Se pueden utilizar:

X: Indica un dígito del 0 al 9

Z: Indica un dígito del 1 al 9

N: Indica un dígito del 2 al 9 [129] Indica el 1, 2 o 9

. Indica uno o más caracteres (¡Atención! Coincide con las extensiones especiales: h,i,t .., recomendable: _X.)

Ejemplos:

Fijos Nacionales: exten=> _9XXXXXXXX Internacionales: exten=> _00.

Variables

En el Dialplan de Asterisk existen variables, que pueden ser modificadas por el propio Asterisk en su ejecución lógica o por comandos expresos del Dialplan, las aplicaciones pueden cambiar variables.

Los tipos de variables son:

Globales: Declaradas en extensions.conf (o por comando). Canal: Son propias a cada canal.

Entorno: Variables de entorno (UNIX Like). La sintaxis de una variable es:

${variable}

Manejo de Variables Asignación de variables: SetVar(Variable=valor) SetGlobalVar(Variable=valor) Manejo de cadenas:

Subcadenas: ${Variable : offset : longitud }

Devuelve la subcadena de variable que comienza en offset y con la longitud especificada.

Ejemplo:

${ 123456789:2:3} devuelve 345

Longitud:

${LEN(Variable)} Concatenación:

${Variable1}${Variable2}

Variables de canal definidas automáticamente

Listado de variables más importantes:

${CALLERID}: Caller ID actual, nombre y número.

${CONTEXT}: Contexto actual.

${EXTEN}: Extensión actual.

${CHANNEL}: Canal actual.

${DIALSTATUS}: Estado de la llamada: unavailable, congestion, busy, noanswer, answer, cancel, hangup.

${DATETIME}: Hora actual.

Un comando útil para ver el contenido es NoOp: NoOp (${VARIABLE})

Nos mostrará en el CLI el valor.

Expresiones

Es posible utilizar expresiones en las llamadas a aplicaciones (principalmente: GotoIf)

Syntaxis:

$[expr1 operador expr2]

Operadores Lógicos: |(or) , &(AND)

Operadores de Comparación: =, !=, <, >, <=, >= Operadores Aritméticos: +, -, *, /, %

Ejemplos:

exten => 1,1,SetVar(total=$[1 + 1])

exten => 1,2,GotoIf($[${CALLERID}=123456]?10:20)

Funcionalidades

Toda la secuencia y programación del dialplan es el verdadero núcleo del sistema centralita, si bien, las siguientes funcionalidades se configuran en features.conf:

Transferencias de llamadas: transferencia de llamadas entre diversos usuarios, independientemente de la tecnología que usen.

Call Parking: Parking de llamadas.

Call Pickup: Auto-transferencia de un teléfono que esté sonando.

Música en Espera

Asterisk puede poner un canal dado en espera ('HOLD'), principalmente en las siguientes situaciones:

Durante una transferencia.

Durante una llamada si se ha especificado el parámetro 'm', que indica que no se oirá tono de llamada sino música en espera.

Durante una espera en el parking.

Si la aplicación MusicOnHold o WaitMusicOnHold ha sido llamada desde el

DialPlan

Si el destino de la llamada ha solicitado explícitamente que la llamada sea puesta en espera

Es posible tener distintos tipos de música en espera. La música en espera se configura en musiconhold.conf

Configuración para utilizar el formato nativo:

Es necesario compilar asterisk-addons (en concreto el directorio format_mp3)

En /etc/asterisk/modules.conf debe indicarse la precarga del módulos:

preload => format_mp3.so

En musiconhold.conf, es para el tipo de música en espera.

(suponiendo para el modo default): [default]

mode = files

directory = /var/lib/asterisk/mohmp3

Colas de llamadas

Una llamada entrante puede ser enviada a una cola de llamadas, que será gestionada por determinados usuarios. Se utilizan mucho en entornos tipo

'callcenter', con los canales tipo de Agentes (que hacen 'login en el sistema'). Las colas pueden comportarse de forma distinta:

Suena todos los teléfonos hasta que alguno descuelgue. Los teléfonos van sonando en orden

Existen colas con prioridad. Las colas de llamadas se configuran en queues.conf:

Registro de llamadas

Asterisk permite llevar un control exhaustivo de todas las llamadas que se han realizado o recibido. Es interesante para control propio de facturación,

independiente del proveedor (sino lo somos). Permite realizar estadísticas. Este control se denomina: CDR, Call Detail Record

El registro del CDR se escribe por defecto en el fichero

/var/log/asterisk/cdr-csv/Master.csv

Existen extensiones al cdr: cdr_mysql por ejemplo, que permiten almacenar los registros en una base de datos.

El CDR se configura en el fichero cdr.conf, para el módulo de MySQL, se utiliza cdr_mysql.conf

Para confirmar el estado del CDR desde el CLI, se puede ejecutar:

CLI> cdr status

Existe muchas aplicaciones que permite gestionar el CDR. Desarrollar una propia no es realmente muy complejo.

Algunas aplicaciones open source:

Astbill: Es una de las mejores aplicaciones opensource para tarificación, control de cuentas y llamadas.

Areski Stat v2: Se trata de una aplicación para listar y realizar estadísticas de las llamadas realizas o enviadas.

A2Billing

Iabslite: Irontec Asterisk Billing system (próximamente).

2.5. OPENWRT

OpenWrt se describe como una distribución de Linux para dispositivos embebidos. (11). En lugar de tratar de crear un único estático firmware, OpenWrt proporciona un sistema de ficheros totalmente modificable con la gestión de paquetes. Esto le libera de la selección y la aplicación de configuración proporcionada por el vendedor y le permite personalizar el dispositivo mediante el uso de paquetes para adaptarse a cualquier aplicación. Para el desarrollador, OpenWrt es el marco para crear una aplicación sin tener que construir un firmware completo alrededor de ella, esto significa para los usuarios la capacidad de una personalización completa, para utilizar el dispositivo de una manera nunca imaginada.

El soporte fue limitado originalmente al modelo Linksys WRT54G, pero desde su rápida expansión se ha incluido soporte para otros fabricantes y dispositivos, incluidos el Netgear, D-Link, ASUS y algunos otros. El router más popular sigue siendo el Linksys WRT54G y el ASUS WL500G. OpenWrt utiliza principalmente una interfaz de línea de comando, pero también dispone de una interfaz WEB en constante mejora. El soporte técnico es provisto como en la mayoría de los proyectos de Software Libre, a través de foros y su canal IRC. (11)

El desarrollo de OpenWrt fue impulsado inicialmente gracias a la licencia GPL, que obligaba a todos aquellos fabricantes que modificaban y mejoraban el código, a liberar éste y contribuir cada vez más al proyecto en general. Poco a poco el software ha ido creciendo y se encuentran características implementadas que no tienen muchos otros fabricantes de dispositivos comerciales para el sector no profesional, tales como QoS, VPN y otras características que dotan a OpenWrt de un dispositivo realmente potente y versátil, apto para utilizar los hardware donde corre OpenWrt no sólo para utilizarlos como routers, sino como servidores de archivo, nodos P2P, servidores de WEBcams, firewall o puertas de acceso VPN.

2.6. PLACA BCM5354

La placa a utilizar pertenece a Broadcom, Broadcom Corporation es uno de los principales fabricantes de circuitos integrados para comunicaciones de banda ancha de los Estados Unidos. Fundada en 1991 por Henry Samueli (presidente del consejo de administración y CTO) y Henry Nicholas, salió al mercado bursátil en 1998 y actualmente emplea a más de 4.000 personas en todo el mundo.

Broadcom forma parte del Top 20 mundial de empresas de semiconductores (12).

El router a la placa Broadcom BCM5354 / 240 MHz. A continuación en detalle en la tabla 3:

SoC	Ram	Flash	Network	USB	Serial	JTag
Broadcom BCM5354 / 240MHz	16MiB	4MiB	4x1	Yes	Yes	?

Tabla 3. Especificación de Placas

Architecture:	MIPS
Vendor:	Broadcom
Bootloader:	CFE
System-On-Chip:	BCM5354KFBG
CPU/Speed:	Broadcom BCM3302 / 240MHz
Flash-Chip:	4MB MX 29LV320CB
Flash size:	4 MiB
RAM:	Samsung K4S281632I-UC60 / 166 MHz / 16 MiB
Wireless:	Broadcom 5354 (core revision 13) 802.11b/g (integrated)
Ethernet:	Switch in CPU
USB:	Yes
Serial:	Core supports 2 serial ports, only 1 is available on the PCB
JTAG:	?

Tabla 4. Detalles de Placas

2.6.1. Placa BCM5354KFBG

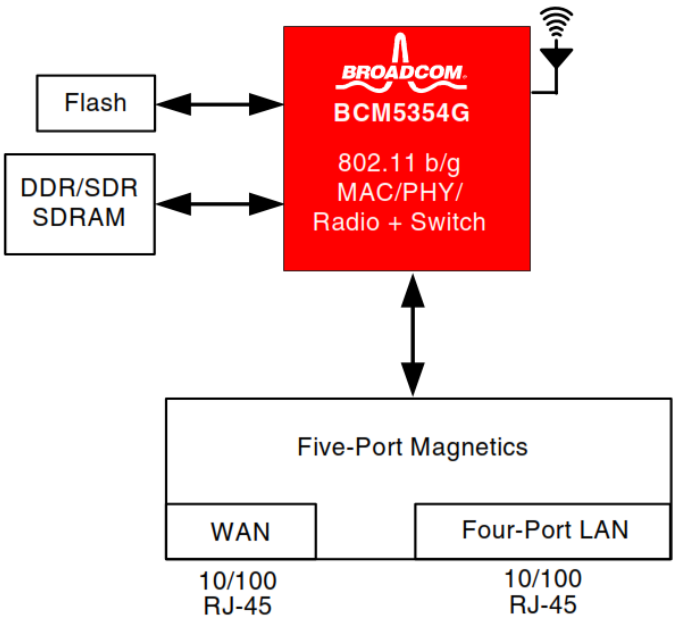


Ilustración 12. Diagrama base de Placa. Fuente: Broadcom

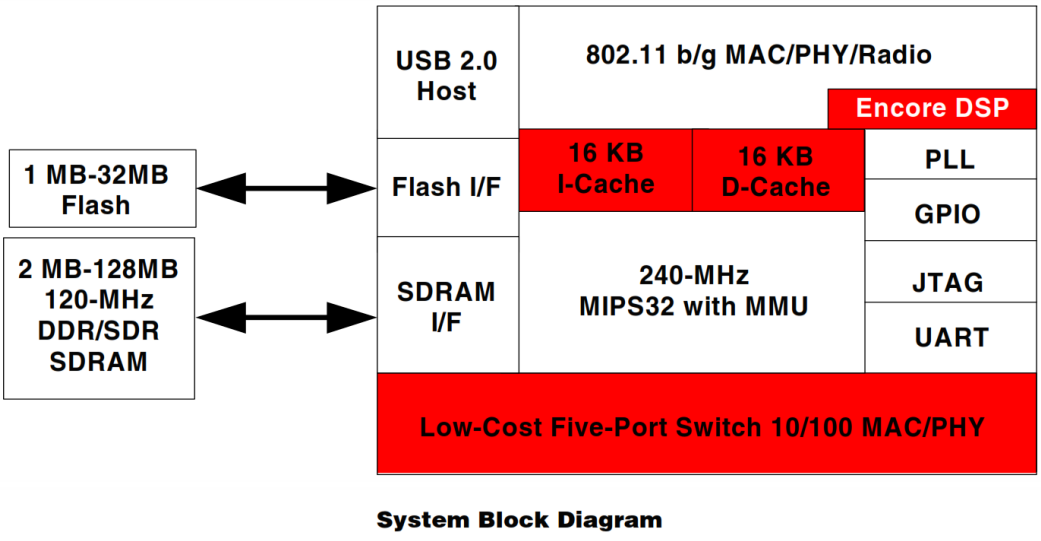


Ilustración 13. Sistema de Bloques de la Placa. Fuente: Broadcom

2.6.2. MIPS

Con el nombre de MIPS (siglas de Microprocessor without Interlocked Pipeline Stages) se conoce a toda una familia de microprocesadores de arquitectura RISC desarrollados por MIPS Technologies. (11)

Los diseños del MIPS son utilizados en la línea de productos informáticos de SGI; en muchos sistemas embebidos; en dispositivos para Windows CE; routers Cisco; y videoconsolas como la Nintendo 64 o las Sony PlayStation, PlayStation 2 y PlayStation Portable.

Las primeras arquitecturas MIPS fueron implementadas en 32 bits (generalmente rutas de datos y registros de 32 bits de ancho), si bien versiones posteriores fueron implementadas en 64 bits. Existen cinco revisiones compatibles hacia atrás del conjunto de instrucciones del MIPS, llamadas MIPS I, MIPS II, MIPS III, MIPS IV y MIPS 32/64. En la última de ellas, la MIPS 32/64 Release 2, se define a mayores un conjunto de control de registros. Así mismo están disponibles varias "extensiones", tales como la MIPS-3D, consistente en un simple conjunto de instrucciones SIMD en coma flotante dedicadas a tareas 3D comunes, la MDMX(MaDMaX) compuesta por un conjunto más extenso de instrucciones SIMD enteras que utilizan los registros de coma flotante de 64 bits, la MIPS16 que añade compresión al flujo de instrucciones para hacer que los programas ocupen menos espacio (presuntamente como respuesta a la tecnología de compresión Thumb de la arquitectura ARM) o la

reciente MIPS MT que añade funcionalidades multithreading similares a la tecnología HyperThreading de los procesadores Intel Pentium 4 (11).

Debido a que los diseñadores crearon un conjunto de instrucciones tan claro, los cursos sobre arquitectura de computadores en universidades y escuelas técnicas a menudo se basan en la arquitectura MIPS. El diseño de la familia de CPU's MIPS influiría de manera importante en otras arquitecturas RISC posteriores como los DEC Alpha.

2.7. SISTEMA EMBEBIDO

Un sistema embebido es un sistema informático aplicado, a diferencia de otros tipos de sistemas informáticos, como ordenadores personales (PC) o supercomputadoras. Sin embargo, se dará cuenta de que la definición de "sistema integrado" es fluida y difícil de precisar, a medida que evoluciona constantemente con los avances en la tecnología y una disminución drástica en el costo de la aplicación de diversos componentes de hardware y software. En los últimos años, el campo ha superado muchos de sus descripciones tradicionales. Debido a que el lector probablemente se encontrará con algunas de estas descripciones y definiciones, es importante entender el razonamiento detrás de ellos y por qué puede o no puede ser exacta hoy, y ser capaz de hablar de ellos con conocimiento de causa. (12)A continuación se presentan algunas de las descripciones más comunes de un sistema embebido:

- Los sistemas embebidos son más limitadas en el hardware y / o la funcionalidad del software de una computadora personal (PC). Esto es válido para un subconjunto significativo de la familia de sistemas integrados de los sistemas informáticos. En términos de limitaciones de hardware, esto puede significar limitaciones en el rendimiento de procesamiento, consumo de energía, la memoria, la funcionalidad del hardware, y así sucesivamente. En el software, esto significa típicamente limitaciones con respecto a una aplicaciones de PC-menos, aplicaciones a escala reducida, ningún sistema operativo (OS) o un sistema operativo limitado, o menos código de nivel de abstracción. Sin embargo, esta definición es sólo parcialmente cierto hoy como tableros y software que se encuentran típicamente en las PC del pasado y del presente han sido re envasado en diseños más complejos integrados del sistema (12).
- Un sistema integrado está diseñado para realizar una función específica. Mayoría de los dispositivos integrados están diseñados principalmente para una función específica. Sin embargo, ahora vemos dispositivos como asistente de datos personales (PDA) / híbridos de teléfonos móviles, que son los sistemas integrados diseñados para ser capaz de hacer una gran variedad de funciones primarias. Además, los últimos televisores digitales incluyen aplicaciones interactivas que realizan una amplia variedad de funciones generales no relacionadas con la función "TV", pero igual de importantes, como el correo electrónico, navegación web y juegos.

- Un sistema embebido es un sistema informático con los requisitos más altos de calidad y fiabilidad que otros tipos de sistemas informáticos. Algunas familias de los dispositivos integrados tienen un umbral muy alto de los requisitos de calidad y fiabilidad. Por ejemplo, si el controlador del motor de un coche se estrella durante la conducción en una autopista ocupada o un mal funcionamiento de dispositivos médicos críticos durante la cirugía, problemas muy serios resultan. Sin embargo, también hay dispositivos embebidos, tales como televisores, juegos y teléfonos celulares, en la que un mal funcionamiento es un inconveniente, pero no suele ser una situación peligrosa para la vida (12).
- Algunos dispositivos que se llaman sistemas embebidos, tales como PDAs o pads web, no son realmente los sistemas embebidos. Hay una cierta discusión en cuanto a si los sistemas informáticos que resuelven algunas, pero no todas las definiciones tradicionales de sistemas embebidos en realidad son sistemas o algo más incrustado. Algunos sienten que la designación de estos diseños más complejos, tales como PDAs, como los sistemas integrados es impulsada por el marketing no técnico y profesionales de ventas, en lugar de ingenieros. En realidad, los ingenieros incorporados están divididos en cuanto a si estos diseños son o no son los sistemas, a pesar de que actualmente estos sistemas son a menudo discutidas como tal entre estos mismos diseñadores embebidos. Sea o no las definiciones incrustados tradicionales deben seguir evolucionando, o un nuevo campo de

los sistemas informáticos pueden designada para incluir estos sistemas más complejos en última instancia, ser determinados por otros en la industria. Por ahora, ya que no hay un nuevo campo de los sistemas informáticos destinados a los diseños que se encuentran entre el sistema integrado tradicional y los sistemas de PC de propósito general apoyada por la industria, apoya la visión evolutiva de los sistemas integrados, que abarca este tipo de sistema informático diseños.

2.7.1. Diseño de Sistemas Embebidos

Al acercarse al diseño de los sistemas embebidos de la arquitectura desde el punto de vista de la ingeniería de sistemas, varios modelos se pueden aplicar para describir el ciclo de diseño de sistemas embebidos (13). La mayoría de estos modelos se basan en una o alguna combinación de los siguientes modelos de desarrollo:

- El modelo del Big Bang, en el que esencialmente no hay planificación o procesos en el lugar antes y durante el desarrollo de un sistema.
- El modelo de codificación y revisión, en la que se definen los requisitos del producto, pero no hay procesos formales están en su lugar antes del inicio del desarrollo.
- El modelo de cascada, en la que hay un proceso para el desarrollo de un sistema en pasos, donde los resultados de un flujo de paso en el siguiente paso.

- El modelo en espiral, en el que hay un proceso para el desarrollo de un sistema en pasos, y a lo largo de las diversas etapas, se obtiene la retroalimentación y se incorpora de nuevo en el proceso.

CAPÍTULO III

INTEGRACIÓN DE LA APLICACIÓN DE VOIP EN EL SISTEMA EMBEBIDO BASADO EN LINUX

En este capítulo se describe el proceso de análisis, integración y funcionamiento del sistema embebido para realizar llamadas de VoIP con diferentes dispositivos conectados a la red.

3.1. Requisitos y soporte de dispositivos

El hardware a utilizar es una placa Broadcom BCM5354 / 240 MHz que tiene una memoria RAM de 16MiB, una memoria flash de 4MiB, 4 puertos Ethernet y un puerto USB.

Según la página oficial de OpenWrt y las características de la placa a utilizar, se tiene que utilizar la distribución de OpenWrt-brcm-2.4 conocido también como backfire 10.03.1.

3.2. Instalación del Sistema Embebido

Para la instalación se debe conocer el dispositivo en específico a utilizar y conocer cómo funciona según el diseño del flash.

Existen 4 métodos de instalación:

- Método 1: vía OEM firmware

  Este método consiste en abrir una interfaz web en el explorador del dispositivo y utilizar la opción de “Firmware Upgrade” con una imagen de OpenWrt. A veces este método no siempre funciona por el hecho de que se necesita una imagen específica para la placa del dispositivo y no siempre está disponible.

- Método 2: vía Bootloader y un puerto Ethernet

  No todos los bootloaders son fáciles de implementar ya que algunos dispositivos utilizan un cliente TFTPy un servidor TFTP, otros un cliente FTP y servidor FTP y algunos un servidor web con un protocolo XMODEM.

- Método 3: vía Bootloader y un puerto serial

Es similar al anterior método anterior pero con un puerto serial en vez de un puerto Ethernet.

- Método 4: vía JTAG

  Este método consiste en Instalar una imagen ramdisk en la memoria principal.

  El método a utilizar en nuestro caso para la placa BCM5354 es el método 2: vía Bootloader y un puerto Ethernet. A continuación los pasos:

  1. Primeramente apague el dispositivo, presione y mantenga el botón de reseteo, encienda el dispositivo.
  2. Suelte el botón cuando la led de encendido empiece lentamente a parpadear.
  3. El dispositivo ahora se encuentra en FRM (Firmware Restoration Mode) y puede ser flasheado a un nuevo firmware mediante tftp o ftp.
  4. En el computador configure la interfaz con una dirección ip estática entre 192.168.1.2 – 192.168.1.254 /24 y Gateway 192.168.1.1; no importa el dns.
  5. Conecte el router con el computador mediante un cable de red Ethernet
  6. Compruebe que se comunican entre sí haciendo ping.

7. En la línea de comandos ingrese el siguiente comando: tftp –i 192.168.1.1 put <archivo firmware de OpenWrt>
8. Una vez que aparezca el mensaje de transferencia correcta, reinicie el dispositivo.
9. Listo, abra un navegador web para configurar el dispositivo.

3.3. Configuración de dispositivo con OpenWrt

Una vez reiniciado el router ya estará listo con la nueva versión del firmware.

Se prepara el pc para acceder por ethernet, esta vez tiene que estar configurado en rango 192.168.1.xxx, se puede usar por ejemplo la 192.168.1.100, el router estará configurado con la IP fija 192.168.1.1, sin contraseña y la WiFi desactivada.

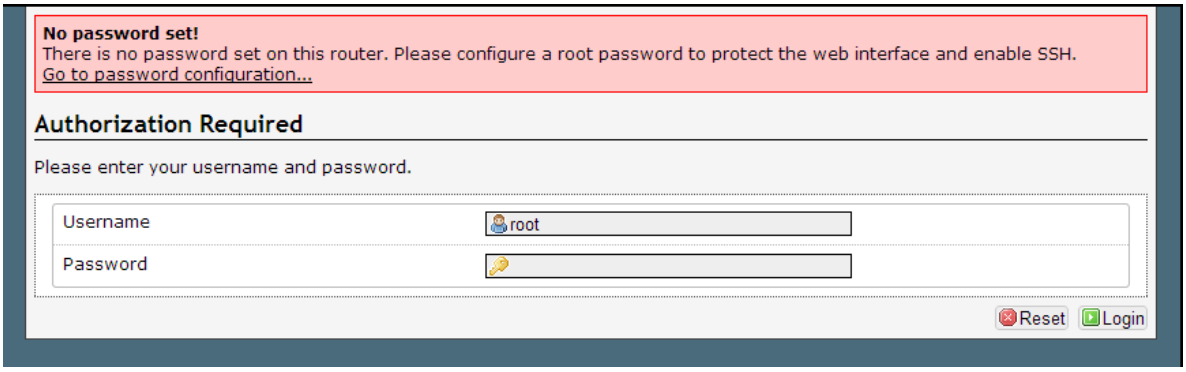


Ilustración 14. Login de Usuario

Lo primero al ingresarse es cambiar el password de acceso, esto también activará el acceso por ssh. Si la primera vez no se puede ingresar por web, se puede hacer por telnet, desde ahí cambiar el password y ya tener el acceso ssh accesible.

Una vez habilitado el ssh se puede abrir una conexión y configurar la red:

```
uci set wireless.radio0.channel=11

uci set wireless.radio0.hwmode=11bg

uci set wireless.radio0.txpower=20

uci set wireless.radio0.country=ES

uci set wireless.radio0.disabled=0

uci set wireless.@WiFi-iface[0].device=radio0

uci set wireless.@WiFi-iface[0].network=lan

uci set wireless.@WiFi-iface[0].mode=ap

uci set wireless.@WiFi-iface[0].ssid=nombreWiFi

uci set wireless.@WiFi-iface[0].encryption=psk

uci set wireless.@WiFi-iface[0].key="clave de la WiFi"

uci set dhcp.lan.interface=lan

uci set dhcp.lan.ignore=1

uci set dhcp.wan.interface=wan

uci set dhcp.wan.ignore=1

#uci set network.lan.nat=1

uci set network.lan.gateway=192.168.1.1
```

uci set network.lan.ipaddr=192.168.1.2

uci set network.lan.netmask=255.255.255.0

uci set network.lan.dns="8.8.4.4 8.8.8.8"

uci commit

/etc/init.d/network restart

Lo siguiente a configurar es el almacenamiento USB ya que como memoria flash sólo dispone de 4MiB, necesitamos una memoria USB formateada en un sistema de archivo compatible con linux, puede ser ext2, ext3 o ext4.

opkg update

opkg install kmod-usb-storage block-mount kmod-fs-ext3

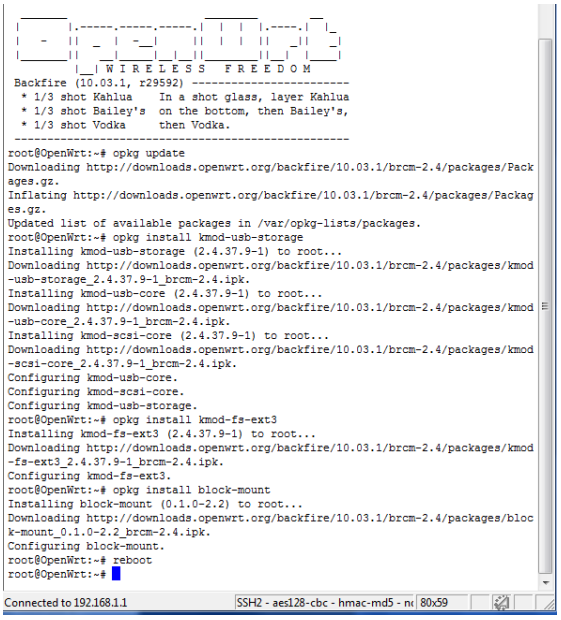


Ilustración 15. Instalación de paquetes

Con los comandos anteriores se obtiene una actualización de la lista de paquetes del repositorio, y se instalan los paquetes necesarios para montar un dispositivo USB.

En esta versión de backfire OpenWrt existe un bug que no deja que se monte los dispositivos USB, hay que hacerlo de manera manual, para eso se hace lo siguiente:

Crear un punto de montaje

#mkdir /mnt/usbdisk

Obtener la información del disco

#fdisk -l

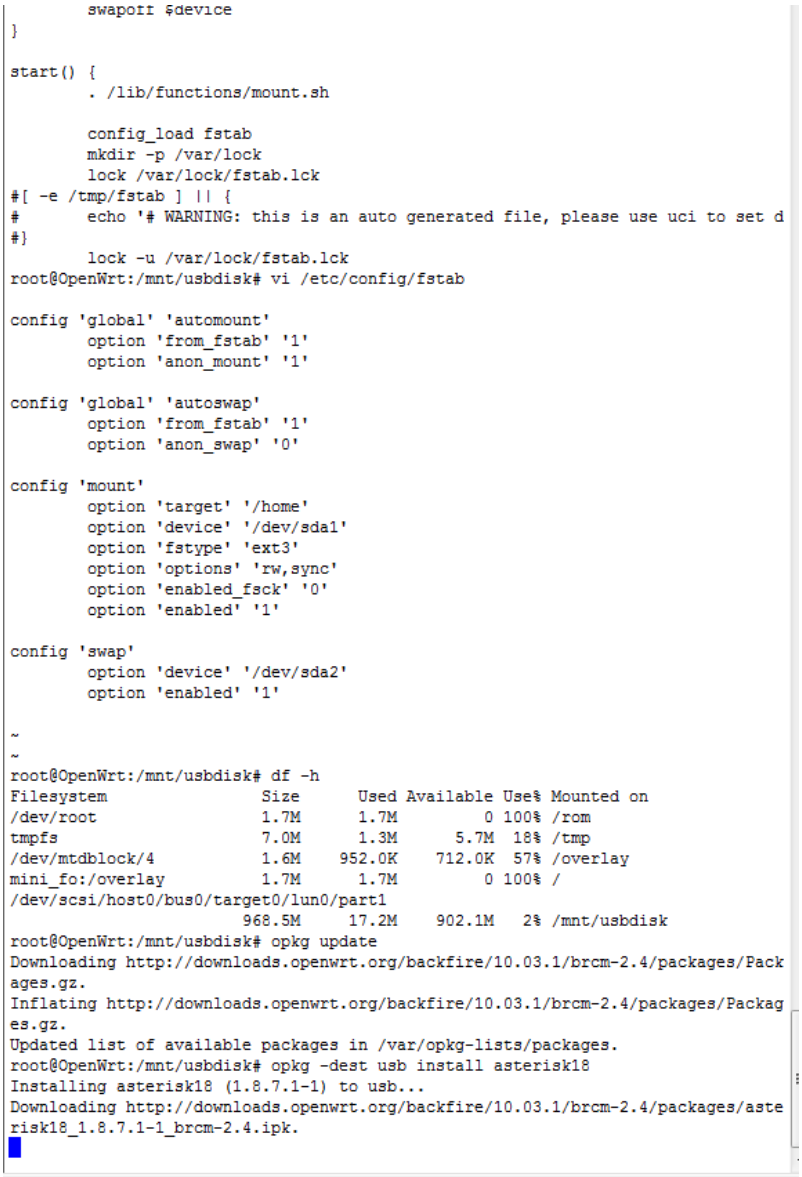
>/dev/scsi/host0/bus0/target0/lun0/part1

Montar el dispositivo

#mount /dev/scsi/host0/bus0/target0/lun0/part1 /mnt/usbdisk

Ir al directorio

#cd /mnt/usbdisk



```
        swapoff $device
}

start() {
        . /lib/functions/mount.sh

        config_load fstab
        mkdir -p /var/lock
        lock /var/lock/fstab.lck
#[ -e /tmp/fstab ] || {
#       echo '# WARNING: this is an auto generated file, please use uci to set d
#}
        lock -u /var/lock/fstab.lck
root@OpenWrt:/mnt/usbdisk# vi /etc/config/fstab

config 'global' 'automount'
        option 'from_fstab' '1'
        option 'anon_mount' '1'

config 'global' 'autoswap'
        option 'from_fstab' '1'
        option 'anon_swap' '0'

config 'mount'
        option 'target' '/home'
        option 'device' '/dev/sda1'
        option 'fstype' 'ext3'
        option 'options' 'rw,sync'
        option 'enabled_fsck' '0'
        option 'enabled' '1'

config 'swap'
        option 'device' '/dev/sda2'
        option 'enabled' '1'

~
~
root@OpenWrt:/mnt/usbdisk# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                 1.7M      1.7M         0 100% /rom
tmpfs                     7.0M      1.3M      5.7M  18% /tmp
/dev/mtdblock/4           1.6M    952.0K    712.0K  57% /overlay
mini_fo:/overlay          1.7M      1.7M         0 100% /
/dev/scsi/host0/bus0/target0/lun0/part1
                        968.5M     17.2M    902.1M   2% /mnt/usbdisk
root@OpenWrt:/mnt/usbdisk# opkg update
Downloading http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/packages/Pack
ages.gz.
Inflating http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/packages/Packag
es.gz.
Updated list of available packages in /var/opkg-lists/packages.
root@OpenWrt:/mnt/usbdisk# opkg -dest usb install asterisk18
Installing asterisk18 (1.8.7.1-1) to usb...
Downloading http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/packages/aste
risk18_1.8.7.1-1_brcm-2.4.ipk.
```

Ilustración 16. Montaje de memoria externa

Se debe editar el archivo /etc/init.d/fstab, se debe poner la línea de sleep en 20 y poner un # en la línea del echo

Y por último se debe crear un enlace entre archivos.

ln -s /tmp/fstab /etc/fstab

Ahora se debe dar permisos de root al dispositivo usb, para ello se debe duplicar los datos de la siguiente manera:

Copie los archivos necesarios de la memoria flash a la nueva partición raíz de la plantilla actual (JFFS2) a las nuevas rootfs (suponiendo que el sistema de ficheros para los nuevos rootfs externos está montado en / mnt/sda1):

tar -C /overlay -cvf - . | tar -C /mnt/sda1 -xf –

Luego se configure el /etc/config/fstab

config mount

option target /mnt # This is ignored once is_rootfs is set to 1

option device /dev/sda1

option fstype ext3

option options rw,sync

option enabled 1

option enabled_fsck 0

option is_rootfs 1

Reiniciamos el dispositivo y notamos que /overlay se encuentra en la nueva partición montana en la memoria usb

Lo último es configurar el archivo de la configuración de los opkg para que la instalación sea directa en el usb y no en la memoria flash del dispositivo.

```
rc1/ar71xx/generic/packages
dest root /
dest usb /mnt/usb
dest ram /tmp
lists_dir ext /var/opkg-lists
option overlay_root /overlay
```

Ilustración 17. Configuración de memoria externa

3.4. Instalación de la Aplicación de VoIP en el sistema embebido

Vamos a requerir de asterisk 1.8 como aplicación de llamadas sobre VoIP para ello ingresamos los siguientes comandos:

opkg update

opkg install asterisk18

```
                        968.5M     17.2M    902.1M   2% /mnt/usbdisk
root@OpenWrt:/mnt/usbdisk# opkg update
Downloading http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/packages/Pack
ages.gz.
Inflating http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/packages/Packag
es.gz.
Updated list of available packages in /var/opkg-lists/packages.
root@OpenWrt:/mnt/usbdisk# opkg -dest usb install asterisk18
Installing asterisk18 (1.8.7.1-1) to usb...
Downloading http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/packages/aste
risk18_1.8.7.1-1_brcm-2.4.ipk.
Installing libopenssl (0.9.8r-1) to usb...
Downloading http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/packages/libo
penssl_0.9.8r-1_brcm-2.4.ipk.
Installing libncurses (5.7-2) to usb...
Downloading http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/packages/libn
curses_5.7-2_brcm-2.4.ipk.
Installing libpopt (1.7-5) to usb...
Downloading http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/packages/libp
opt_1.7-5_brcm-2.4.ipk.
Configuring libopenssl.
Configuring libncurses.
Configuring libpopt.
Configuring asterisk18.
root@OpenWrt:/mnt/usbdisk#
Connected to 192.168.1.1    SSH2 - aes128-cbc - hmac-md5 - n    80x59
```

Ilustración 18. Instalación de la Aplicación de VoIP

Asterisk tiene una línea de comandos para arrancar y monitorear

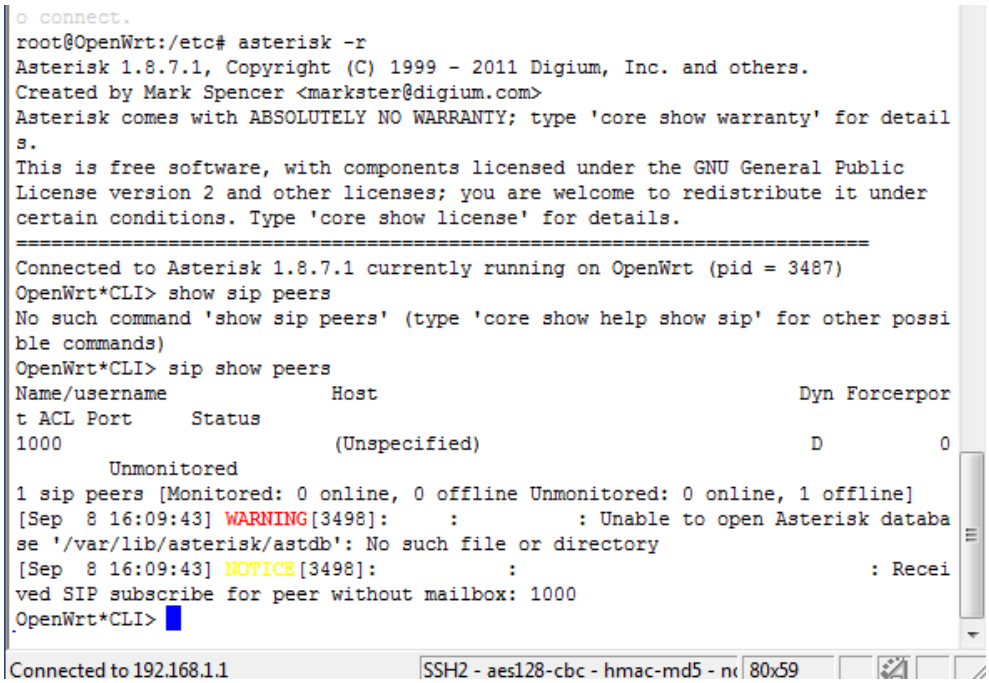


Ilustración 19. Funcionamiento de la Aplicación

3.5. Configuración de SIP

Para la configuración de Asterisk se encuentra en las siguientes ubicaciones:

/etc/init.d/asterisk

/etc/default/asterisk

/etc/asterisk/asterisk.conf

/etc/asterisk/features.conf

/etc/asterisk/modules.conf

/etc/asterisk/extensions.conf

/etc/asterisk/manager.conf

/etc/asterisk/logger.conf

/etc/asterisk/rtp.conf

/etc/asterisk/indications.conf

/etc/asterisk/sip_notify.conf

/etc/asterisk/sip.conf

Los arvhivos que se van a configurar son los de extensions.conf y sip.conf, a continuación un ejemplo rápido de configuración:

extensions.conf

[others]

[phones]

exten => _1000, 1, Dial(SIP/1000, 20)

exten => _1001, 1, Dial(SIP/1001, 20)

exten => _1002, 1, Dial(SIP/1002, 20)

exten => _1003 1, Dial(SIP/1003, 20)

esten => _1004 1, Dial(SIP/1004, 20)

sip.conf

[general]

srvlookup=no

bindport = 5060

bindaddr = 0.0.0.0

videosupport=yes

context = others

[1000]

type=friend

context=phones

secret=1000

qualify=yes

port=5060

nat=yes

host=dynamic

dtmfmode=rfc2833

callerid=device <301>

dial=SIP/1000

canreinvite=no

disallow=all

allow=ulaw

allow=gsm

allow=h261

allow=h263

Ilustración 20. Configuración de Protocolo SIP

3.6. Configuración de los softphones

Para la configuración de los softphones se han configurado tres tipos de dispositivos:

- Un teléfono Cisco IP
- Un SmartPhone con configuración SIP
- Un software para simulación SIP, X-lite.

Para la configuración de los dispositivos los principal datos a introducir son la ID de usuario, el dominio en cual está la aplicación de VoIP, y si requiere la información de autenticación.

X-lite

Para la configuración del software de simulación los datos a registrar fueron:

User ID: 1000

Dominio: 192.168.1.1

Password: contraseña

Display Name: Prueba

Autentication name: 1000

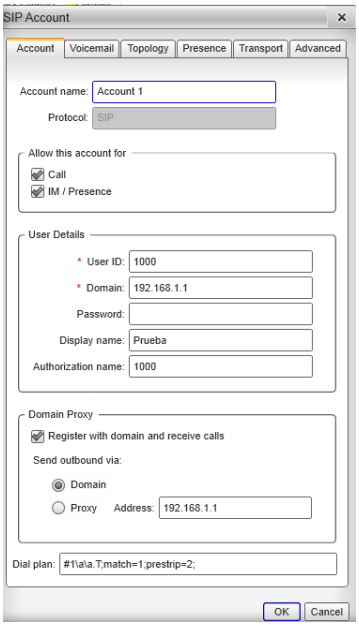


Ilustración 21. Configuración de Sotfphone

Dispositivo móvil (SmartPhone)

Para la configuración del SmartPhone se introdujeron los siguientes datos:

Host: 192.168.1.1

Username: 1001

Password: contraseña

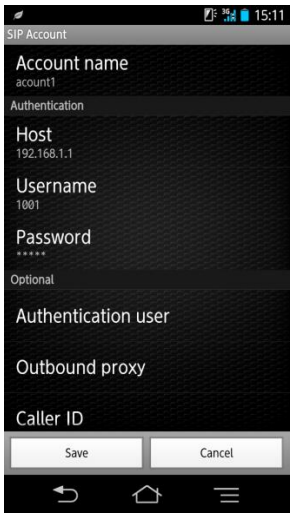


Ilustración 22. Configuración de SmartPhone

Teléfono IP

Para la configuración del teléfono IP se ingresó los datos correspondientes:

Host: 192.168.1.1

User: 1002

Password: constraseña

Ilustración 23. Configuración de Telefono IP

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1. Ejecución de Pruebas

Una vez instalada la red y en funcionamiento el sistema embebido con la aplicación de VoIP se debe ejecutar las pruebas integrando los diferentes dispositivos en telefonía IP, para verificar su perfecto funcionamiento y validar la hipótesis.

Para la realización de las pruebas se realizaron varias llamadas entre los dispositivos conectados a la red WiFi con una duración entre 60 y 120 minutos para probar cada uno de los parámetros de evaluación que se indican más adelante.

A continuación unas ilustraciones de las llamadas realizadas:

Ilustración 24. Prueba de llamadas hacia teléfono IP

En la Ilustración 24 se está realizando una llamada desde el teléfono IP configurado con SIP hacía el dispositivo móvil.

En la Ilustración 25 se puede observar la llamada entrante desde el teléfono IP en el dispositivo móvil.

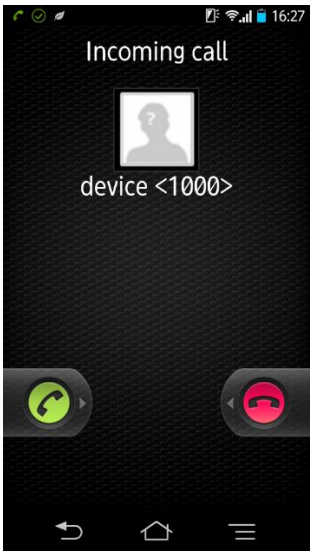


Ilustración 25. Prueba de llamada hacia SmartPhone

En la Ilustración 26 se puede visualizar la llamada entrante en el dispositivo móvil.

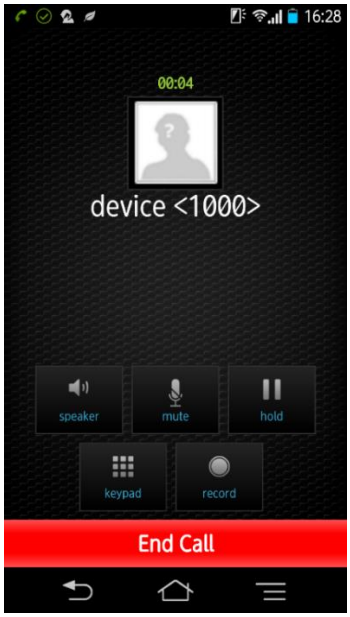


Ilustración 26. Llamada en SmartPhone

En la Ilustración 27 se puede visualizar la llamada entrante en el softphone.

Ilustración 27. Llamada hacia el sotfphone

4.2. Parámetros de Evaluación

Para definir los parámetros de evaluación se debe considerar los indicadores de la operacionalización de las variables que son: accesibilidad y disponibilidad.

Para la accesibilidad se tiene que comprobar que todos los dispositivos a utilizar son compatibles, se conectan a la red wireless sin problema y tienen conectividad entre sí.

Para la disponibilidad se deben evaluar los principales parámetros de la red:

- Latencia y paquetes perdidos
- Test de ancho de banda
- Jitter

4.3. Instrumentos de Evaluación y Validación

Los Instrumentos de evaluación y validación es todo recurso hardware o software que ayuda a medir, obtener datos y realizar los cálculos necesarios para determinar los parámetros a evaluar.

4.3.1. Jperf

Jperf es una herramienta de prueba de red de uso general que puede crear flujos de datos TCP y UDP y medir el rendimiento de una red.

Jperf permite al usuario configurar los distintos parámetros que pueden ser utilizados para realizar pruebas en una red, alternativamente, para optimizar el ajuste o una red. Jperf tiene un cliente y servidor de la funcionalidad, y se puede medir el rendimiento entre los dos extremos, ya sea unidireccional o bidireccionalmente.

4.3.2. Cmd

Es el intérprete de comandos en OS/2 y sistemas basados en Windows NT. Es el equivalente de command.com en MS-DOS y sistemas de la familia Windows 9x.

A diferencia de su antecesor (command.com), este programa es tan sólo una aplicación, no es una parte del sistema operativo y no posee la función de cargar la configuración al arrancar el sistema.

Muchas funciones que se realizan desde la interfaz gráfica de algún sistema operativo son enviadas al cmd que es el encargado de ejecutarlas.

4.3.3. Ssh client

Es el protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos existe un Servidor X (en sistemas Unix y Windows) corriendo.

4.3.4. Wireshark

Es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. Wireshark incluye un completo lenguaje para filtrar y la habilidad de mostrar el flujo reconstruido de una sesión de TCP

4.3.5. Monitoreo

Para realizar el monitoreo de la red se pueden utilizar muchas herramientas con el objetivo de medir y ver la el estado de la red.

4.3.6. Ping

Ping utiliza el protocolo ICMP (Internet Control Message Protocol) son mensajes para determinar si un host remoto está activo o inactivo, el tiempo de ida y vuelta de datos de los paquetes y si existen paquetes perdidos.

Prueba de envío y recibo de paquetes hacia el dispositivo inalámbrico utilizando ping en la Ilustración 28 y 29.

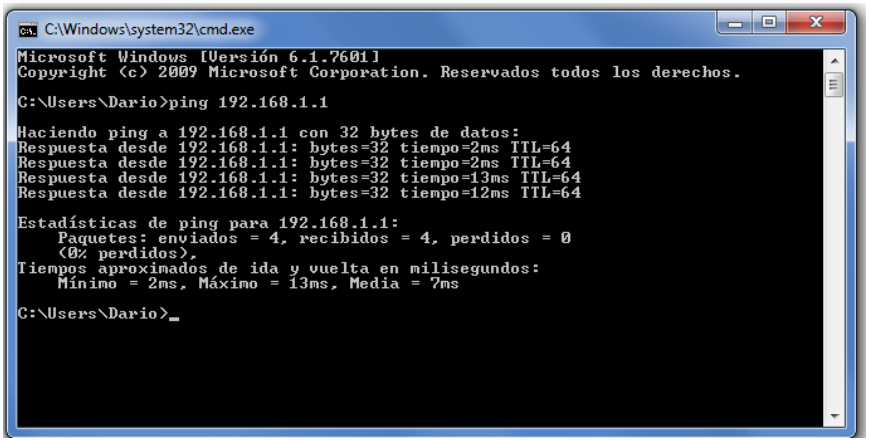


Ilustración 28. Prueba de envío y recibo de paquetes hacia dispositivo inalámbrico

Prueba de envío y recibo de paquetes hacia softphones

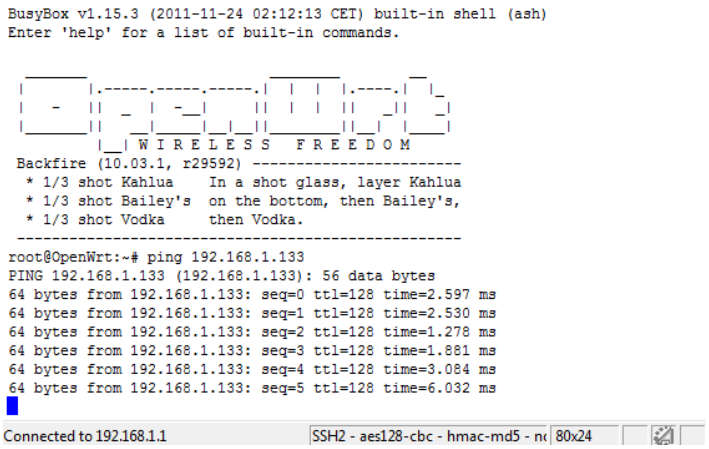


Ilustración 29. Prueba de envío y recibo de paquetes hacia sotfphone

4.3.7. Ancho de Banda

El ancho de banda es la cantidad de datos transferidos de un lugar a otro en un tiempo determinado, con ello se determina si entre el dispositivo inalámbrico y los dispositivos móviles existe cuellos de botella.

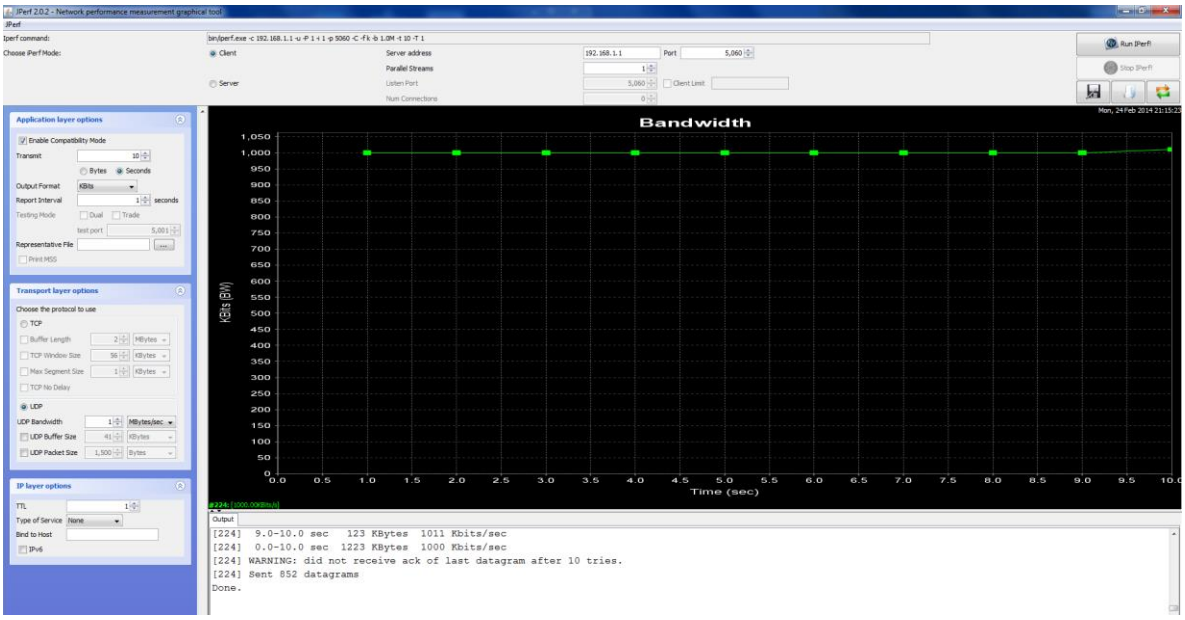


Ilustración 30. Prueba de ancho de banda

4.3.8. Jitter

Se denomina jitter a la variabilidad del tiempo de ejecución de los paquetes. Este efecto es especialmente molesto en aplicaciones multimedia en Internet como radio por Internet o telefonía IP, ya que provoca que algunos paquetes lleguen demasiado pronto o tarde para poder entregarlos a tiempo.

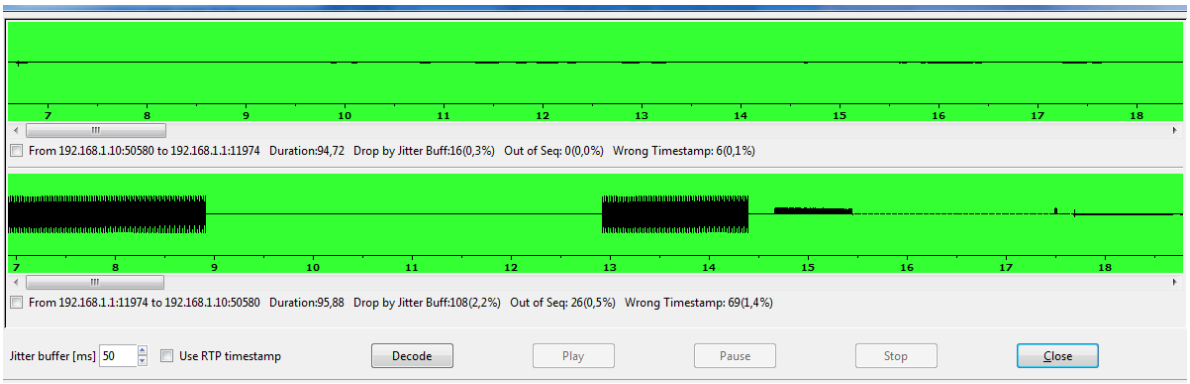


Ilustración 31. Prueba de Jitter

4.4. Análisis e Interpretación de Resultados

En base a los resultados obtenidos en el monitoreo de la red es posible analizar e interpretar los parámetros a medir antes mencionados en la operacionalización de las variables y poder así demostrar la hipótesis, a continuación la interpretación:

4.4.1. Latencia y Paquetes perdidos.

Enlace	N° Paquetes enviados	Paquetes Perdidos (%)	Tiempo de respuesta (ms)		
			Min	Max	Promedio
Dispositivo Inalámbrico	4	0	2	13	7
Dispositivo móvil	4	0	1.396	2.019	3.617

Tabla 5. Latencia y paquetes perdidos

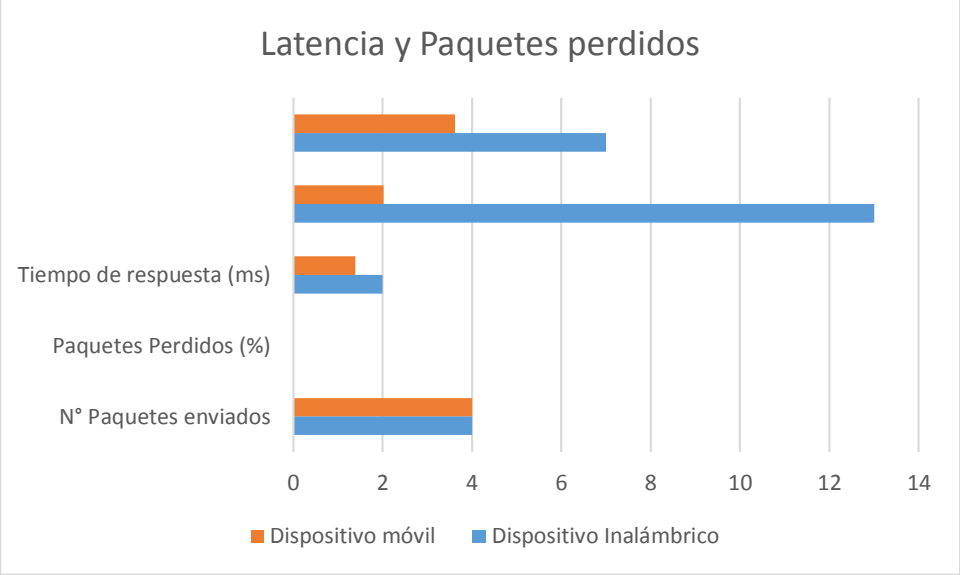


Ilustración 32. Latencia y paquetes perdidos

Para la latencia y los paquetes perdidos se realizó un ICMP desde el dispositivo móvil hacia el dispositivo inalámbrico y viceversa, lo que muestra que en el tiempo de respuesta el dispositivo inalámbrico es mucho más rápido el envío y recepción de paquetes que el dispositivo móvil. No se pierde ningún paquete en la prueba, por lo tanto todos los paquetes enviados se reciben con éxito. Como conclusión se puede decir que existe una conectividad constante sin pérdida de paquetes entre el dispositivo inalámbrico y el dispositivo móvil.

4.4.2. Ancho de Banda

Enlace	Tasa de Transmisión (Kbps)	Intensidad de la Señal (dBm)	Banda Ancha (Kbps)
Dispositivo Móvil	1220	-42	1000

Tabla 6. Ancho de Banda

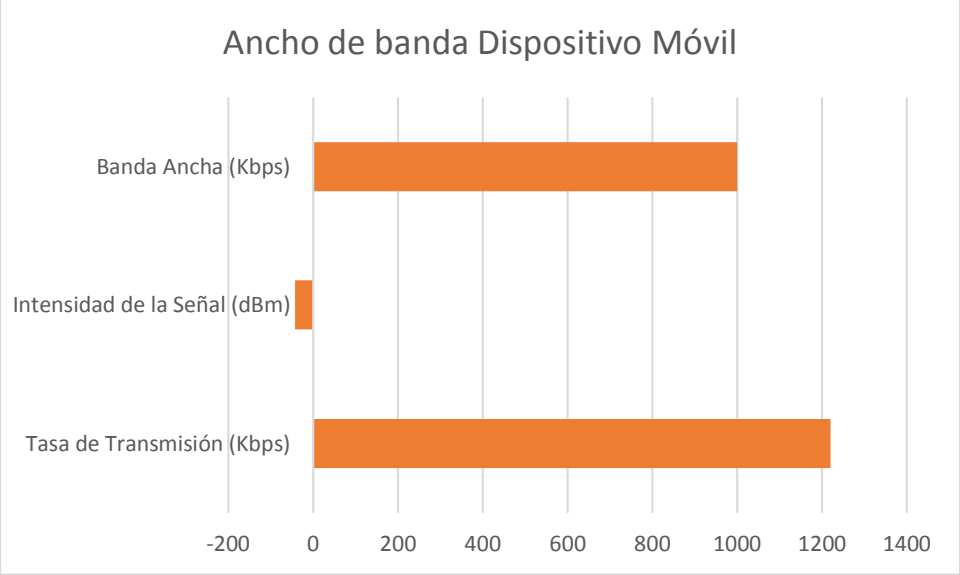


Ilustración 33. Ancho de banda de dispositivo móvil

En el ancho de banda se realizó la prueba con la herramienta Jperf, lo que principalmente se puede observar es que la tasa de transmisión es más alta que el ancho de banda promedio por lo que existe un pequeño retardo en la transmisión de datos haciendo un pequeño cuello de botella pero a medida que se sigue utilizando el ancho de banda se sigue normalizando hasta alcanzar una estabilidad entre el ancho de banda y la tasa de transmisión.

La intensidad de la señal esta con muy buen nivel de dBm[2] por lo que no habría pérdida de señal.

4.4.3. Jitter

Enlace	Duración (s)	Descenso por Jitter (ms)	Fuera de Secuencia (%)	Mal marca de tiempo (%)
Dispositivo Móvil	95	0.3	0.0	0.1
Dispositivo Inalámbrico	95	2.2	0.5	1.4

Tabla 7. Jitter

[2] El indicador de fuerza de la señal recibida tiene una escala de 0 a -80. Siendo 0 la señal ideal, difícil de logara en la práctica, -40 a -60 la señal idónea con tasa de transferencia estables, -60 enlace bueno, -70 enlace normal y -80 la señal mínima aceptable.

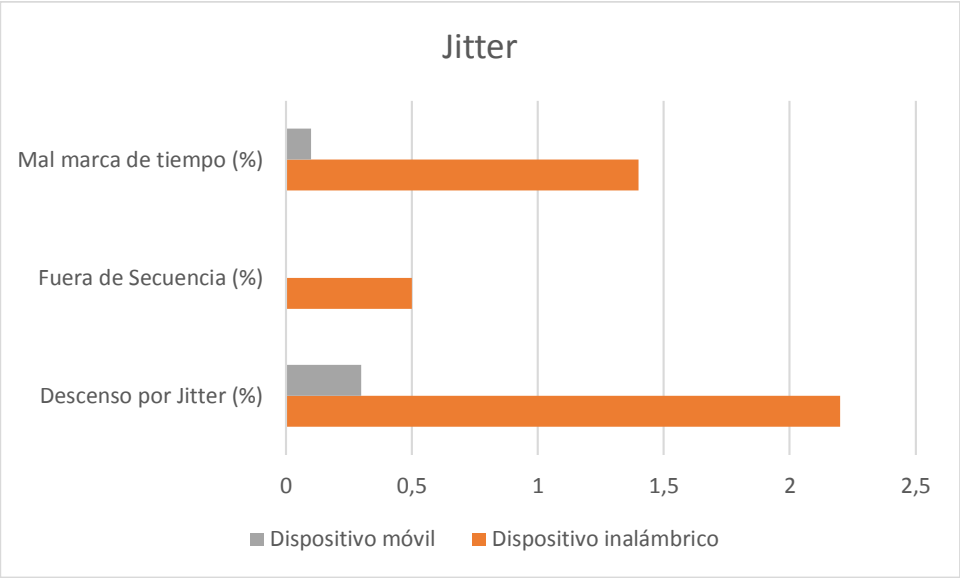


Ilustración 34. Jitter en dispositivos

Para el jitter se utilizó la herramienta wireshark y se observa que los porcentajes en que afecta a los paquetes son bajos por lo que la calidad de voz entre llamadas es de buena calidad sin mencionar que casi no existe retardo de voz entre el emisor y el receptor.

Resumen[3]

Parámetros / Dispositivo	Jitter (ms)	Pérdida de Paquetes (%)	Wrong Timestamp (%)
Dispositivo Inalámbrico	2.2	0	1.4
Dispositivo Móvil	0.3	0	0.1
Media	1.25	0	0.75

Tabla 8. Resumen de Pruebas

La conexión de jitter tiene una media de 1.25 ms, lo que indica que se produce un flujo constante de datos por los que las llamadas de VoIP tienen una buena calidad

La pérdida de paquetes de datos tiene una media de 0.0 % lo que indica que la transferencia de datos es correcta.

Wrong Timestamp es de 0.75 lo que indica que el retardo en las llamadas de VoIP es casi imperceptible.

4.4.4. Tabla de MOS (Mean Opinion Score)

La puntuación media de opinión (MOS) es una indicación de la calidad de una conversación de voz. El MOS varía de 1 a 5, siendo 1 el peor y 5 la mejor. El MOS

[3] Para mayor detalle revise los anexos

está basado el estándar de calidad de voz que dice que el jitter tiene que ser menor a 5 ms, la pérdida de paquetes menor al 1% y la Wrong Timestamp menor al 5% y subjetivamente se le asigna letras del abecedario (A, B, C, D, F) siendo “A” la mayor calificación y “F” la menor calificación

Tabla MOS	
A	Mayor a 4.37
B	Entre 4.28 y 4.37
C	Entre 4.00 y 4.27
D	Entre 2.5 y 3.99
F	Menor a 2.49

Tabla 9. MOS

Según la tabla MOS y los datos obtenidos de las pruebas la calificación que obtendría es de la letra A con un resultado de 4.4 como se puede evidenciar en la Ilustración 35.

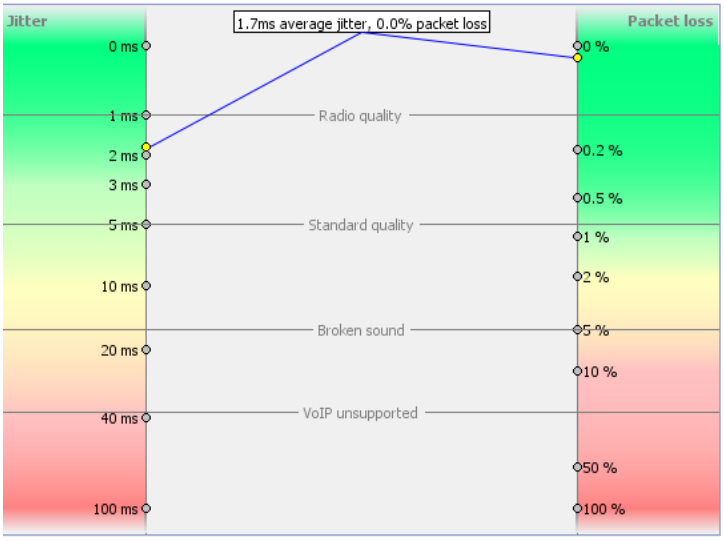


Ilustración 35. Resultados MOS

4.5. Prueba de Hipótesis

El propósito de la prueba de hipótesis es determinar si el valor supuesto (hipotético) de un parámetro, debe aceptarse como verosímil con base a evidencias muestrales.

4.5.1. Formular la hipótesis nula (Ho) y la hipótesis alternativa (Ha)

Hipótesis nula: La integración de software libre de VoIP en un sistema embebido con el uso de dispositivos móviles no permitirá implementar una solución eficiente de comunicación empresarial.

Hipótesis alternativa: La integración de software libre de VoIP en un sistema embebido con el uso de dispositivos móviles permitirá implementar una solución eficiente de comunicación empresarial.

4.5.2. Seleccionar el nivel de significación

Existen tres niveles de significación:

α = 1% = 0.01 (Investigación altamente significativa)

α = 5% = 0.05 (Investigación significativa)

α = 10% = 0.1 (Investigación poco significativa)

Para los cálculos de esta investigación se escogerá el valor de α = 0.01 porque representa una investigación altamente significativa.

4.5.3. Determinar la técnica y la prueba estadística

La técnica y prueba estadística a utilizar es la Prueba de Chi-Cuadrado cuya fórmula es:

$$x^2 = \sum \left[\frac{(n_i - n_i^*)^2}{n_i^*} \right]$$

Ecuación 1. Fórmula de Chi-Cuadrado

Dónde: ni= Frecuencia real

ni* = Frecuencia esperada

v = (k-1)(j-1) = grados de libertad (k = filas, j = columnas)

4.5.4. Calcular los datos muestrales

Parámetros / Dispositivo	Jitter	Paquetes Recibidos	Wrong Timestamp	Total
Dispositivo Inalámbrico	2.2	100	1.4	103.6
Dispositivo Móvil	0.3	100	0.1	100.4
Total	2.5	200	1.5	204

Calculamos $n_i^* = n.p$

$$n_1^* = \frac{2.5\ (103.6)}{204} = 1.27$$

$$n_2^* = \frac{2.5\ (100.4)}{204} = 1.23$$

$$n_3^* = \frac{200\ (103.6)}{204} = 101.57$$

$$n_4^* = \frac{200\ (100.4)}{204} = 98.43$$

$$n_5^* = \frac{1.5\ (103.6)}{204} = 0.76$$

$$n_6^* = \frac{1.5\ (100.4)}{204} = 0.74$$

n_i^*	n_i^*	$n_i - n_i^*$	$(n_i - n_i^*)^2$	$\frac{(n_i - n_i^*)^2}{n_i^*}$
2,20	1,27	0,93	0,87	0,68
0,30	1,23	-0,93	0,87	0,70
100,00	101,57	-1,57	2,46	0,02
100,00	98,43	1,57	2,46	0,02
1,40	0,76	0,64	0,41	0,53
0,10	0,74	-0,64	0,41	0,55
204,00			∑	**2,52**

Calculamos los grados de libertad:

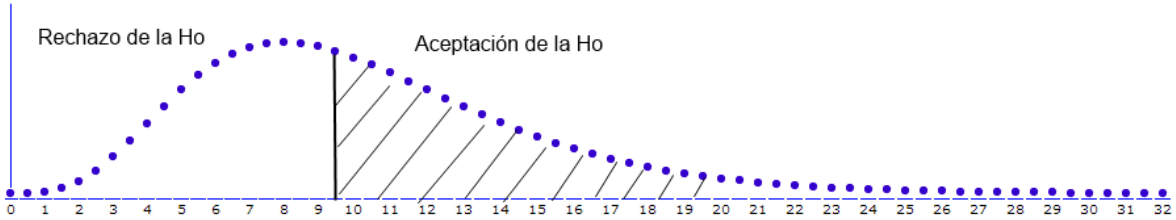
v = (k-1) (j-1) = (2-1)(3-1) = 2

4.5.5. Toma de decisión estadística

Para un valor de grados de libertad de 2 y un nivel de significancia del 1% el valor de la tabla de la prueba de chi cuadrado es de 9.21.

Como $X^2 = 2.52$, cae en el área de rechazo de Ho, Se rechaza la hipótesis nula y se acepta la hipótesis alternativa Ha. Es decir, la diferencia es altamente significativa.

Como se puede apreciar en la Ilustración 36, se visualiza la región de aceptación y la región de rechazo de la hipótesis según el nivel de significancia del 1%.



$$X^2 = 2.52 \quad y \quad X^2_{0.01,2} = 9.21$$

Ilustración 36. Toma de decisión según Chi Cuadrado

CONCLUSIONES

- Se integró satisfactoriamente la aplicación de VoIP en un sistema embebido con el uso de GNU/Linux para interconectar dispositivos móviles mediante el método Bootloader y los comandos de Linux antes mencionados.
- En el escenario propuesto de ambiente de pruebas la integración de la aplicación de VoIP con el uso de GNU/Linux en un sistema embebido destinado a dispositivos móviles permitió implementar una solución eficiente por los resultados obtenidos de comunicación entre 3 dispositivos conectados simultáneamente y con llamadas entre una duración de 60 a 120 segundos.
- Los estudios realizados y publicados en los estándares de comunicación de VoIP y relacionados, indican que el jitter tiene que ser menor a 5 ms, la pérdida de paquetes menor al 1% y el Wrong Timestamp menor al 5%, con los resultados obtenidos se demuestra que la red opera dentro de lo recomendado.
- Se aseguró la disponibilidad del servicio para la interconexión de los dispositivos móviles a través de los resultados obtenidos con los parámetros de medición antes mencionados.

RECOMENDACIONES

- Asegurarse del tipo de dispositivo inalámbrico a utilizar ya que no todos los dispositivos con compatibles o tienen las características disponibles para embeber el sistema e instalar las aplicaciones de VoIP
- Se tiene que tener un conocimiento medio-alto de linux y sus comandos; al mismo tiempo para editar, instalar y configurar los archivos de las aplicaciones de VoIP.
- En la red inalámbrica se pueden hacer la interconexión de todos los equipos e integrarlos sin dificultad sean estos Teléfonos IP, softphones, SmartPhones. La única consideración es que soporten los mismos protocolos y códecs de comunicación.
- La ejecución de esta tecnología aunque tiene puntos a favor acerca de su implementación, tiene sus debilidades como su utilización en grandes y muy grandes empresas, por el número de usuarios y lo limitado que puede ser el hardware a utilizar.
- A pesar de las facilidades que se ofrece, es importante recordar que este sistema entrega herramientas para desarrollar una central telefónica desde cero, por lo que se debe prestar gran atención al valor del conocimiento, es decir, los profesionales a cargo de configurar el servidor de comunicaciones, quienes pueden resultar difíciles de conseguir, por la instalación y el mantenimiento de la central. Un

ahorro excesivo en este aspecto puede resultar en una central poco confiable y de baja calidad.

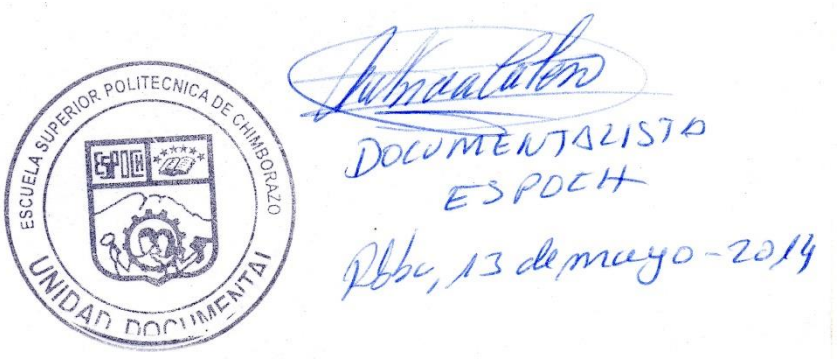
- Debido a la apertura que ofrece el código abierto, existe una gran cantidad de productos que se ofrecen en base al servidor de comunicaciones y sistemas operativos Linux. Una de las tareas más importantes para quien desee configurar su propio sistema es la selección de los paquetes y herramientas necesarias.
- La investigación realizada puede ser la base para la implementación en pequeñas y medianas empresas de un sistema de llamadas de VoIP con pocos recursos.

RESUMEN

Investigación sobre la interconexión de dispositivos móviles con una aplicación de VoIP dentro de un sistema embebido para lograr una mejor comunicación dentro de la pequeña y mediana empresa, es objetivo a alcanzarse.

El hardware utilizado es un router inalámbrico con una placa Broadcom BCM5354, sistema operativo Linux OpenWrt Backfire 10.03 y Asterisk 1.6 como aplicación de VoIP. Con el método experimental se instaló el sistema embebido mediante la técnica Bootloader y un puerto Ethernet; y para la aplicación de VoIP se utilizó comandos de instalación de Linux.

Los resultados obtenidos indican que se integró satisfactoriamente en un 100% los dispositivos utilizados en el escenario de pruebas, estos resultados se consiguen mediante los indicadores propuestos siendo éstos: jitter debe ser menor a 5 ms, habiéndose obtenido 1.25 ms; pérdida de paquetes debe ser menor al 1%, se obtuvo 0% y Wrong Timestamp debe ser menor al 5%, se alcanzó 0.75%. Por lo que se calificó a la interconexión con la letra A, la más alta de la tabla MOS de pruebas estándar de calidad de VoIP, quedando demostrada la hipótesis de que es posible integrar software libre de VoIP en un sistema embebido con el uso de dispositivos móviles obteniéndose una solución eficiente de comunicación empresarial.



Palabras clave: / SISTEMAS EMBEBIDOS / INTERCONEXIÓN DE DISPOSITIVOS MÓVILES / PROTOCOLO DE VOIP / COMUNICACIÓN EMPRESARIAL /

ABSTRACT

Investigation about the interconnection of mobile devices with a VoIP application within an embedded system for achieving a better communication in small and medium enterprises is the objective to fulfill it.

The used hardware is a wireless router with Broadcom plate BCM5354, OpenWrt Backfire 10.03 and Asterisk 1.6 Linux operating system as VoIP application. With the experimental method the embedded system was installed by means of the Bootloader and Ethernet port technique; Linux setup commands were used for the VoIP application.

The obtained results show that it was satisfactorily integrated the used devices in the test scenario at 100%, these results are achieved through the proposed indicators such as: jitter has to be less than 5 ms, it was obtained 1.25 ms; Packet loss have to be less than 1%, it was obtained 0% and Wrong Timestamp has to be less than 5%, it was arrived at 0.75%. So that the interconnection was assessed over letter A, the highest score of MOS chart of VoIP standard quality test, it is possible to integrate VoIP free software within an embedded system with the use of mobile devices obtaining an efficient solution of business communication, therefore the hypothesis has been demonstrated.

Keywords: / EMBEDDED SYSTEMS / INTERCONNECTIONS OF MOBILE DEVICES/ VOIP PROTOCOL / BUSINESS COMUNICATION

REFERENCIAS BIBLIOGRÁFICAS

1. **Van Meggelen, Jim, Madsen, Leif y Smith, Jared.** *Asterisk The Future of Telephony.* Mike Loukides. Sebastopol : O'Reilly, 2007. ISBN-13: 978-0-0596-51048-0.
2. **Samrat Ganguly, Sudeept Bhatnagar.** *VoIP: Wireless, P2P and New Enterprise Voice over IP.* West Sussex : Wiley, 2008. ISBN-978-0-470-31956-7.
3. **Wallingford, Ted.** *Switching to VoIP.* Sebastopol : O'Reylli, 2005. ISBN-0-596-00868-6.
4. **Digium.** Get Started Asterisk. *Sitio web de Asterisk.* [En línea] 2012. http://www.asterisk.org/.
5. **Gomillion, David y Dempter, Barrie.** *Building Telephony Systems With Asterisk.* Birmingham : Packt, 2005. ISBN-1-904811-15-9.
6. **Verdezoto, Daniel.** *Implementación de un Call Center sobre una central de VoIP Basada en Asterisk. Tesis Ing Electrónico.* Sangolqui - *Quito. Escuela Superior Politécnica del Ejército*, Facultad de Electrónica, 2008.
7. **Elastixtech.** Elastixtech. *Elastixtech.* [En línea] 2013. http://elastixtech.com/protocolo-sip/.
8. **VoIP Info.** Organización VoIP Info. *Sitio web de VoipInfo.* [En línea] 2012. http://www.voip-info.org/wiki/view/H.323.
9. **3CX.** Central telefónica de 3CX. *Central telefónica de 3CX.* [En línea] 2012. http://www.3cx.es/.
10. **Buenas Tareas.** Acerca de Buenas Tareas. *Sitio Web Buenas Tareas.* [En línea] 2012. http://www.buenastareas.com/ensayos/Aprendizaje-Receptivo-Se-Asemeja-Al-M%C3%A9todo/693787.html.
11. **OpenWrt.** OpenWrt. *OpenWrt.* [En línea] OpenWrt, 2014. https://openwrt.org/.
12. **A., David A. Pérez.** *Sistemas Embebidos y Sistemas.* Caracas, 2009.
13. **Givargis, Frank Vahid and Tony.** *Embedded System Design-A Unified Hw-Sw Approach.* California, 1999.
14. **Broadcom.** Broadcom. *Broadcom.* [En línea] Broadcom, 2014. https://www.broadcom.com/.

ANEXOS

A continuación se detalla el número de pruebas realizadas en cada dispositivo que fueron altamente similares:

Dispositivo Inalámbrico

Parámetros / Pruebas	Jitter (ms)	Pérdida de Paquetes (%)	Wrong Timestamp (%)
Prueba 1	2.2	0	1.3
Prueba 2	2.3	0	1.4
Prueba 3	2.2	0	1.4
Prueba 4	2.2	0	1.5
Prueba 5	2.2	0	1.4
Prueba 6	2.2	0	1.4
Prueba 7	2.2	0	1.4
Prueba 8	2.2	0	1.3
Prueba 9	2.1	0	1.5
Prueba 10	2.2	0	1.4
Media	2.2	0	1.4

Dispositivo móvil

Parámetros / Pruebas	Jitter (ms)	Pérdida de Paquetes (%)	Wrong Timestamp (%)
Prueba 1	0.4	0	0.1
Prueba 2	0.3	0	0.1
Prueba 3	0.2	0	0.1
Prueba 4	0.3	0	0.1
Prueba 5	0.3	0	0
Prueba 6	0.4	0	0.1
Prueba 7	0.3	0	0.1
Prueba 8	0.3	0	0.1
Prueba 9	0.3	0	0.2
Prueba 10	0.2	0	0.1
Media	0.3	0	0.1