

VoIP para el desarrollo

- Una guía para crear una infraestructura de voz en regiones en desarrollo



Autores:

Alberto Escudero-Pascual

Louise Berthilson

Tabla de Abreviaturas

| Abreviatura | Descripción |
|-------------|--|
| ATA | Adaptador Telefónico Analógico |
| DECT | Comunicación Digital Inalámbrica Mejorada Digital Enhanced Cordless Telecommunications |
| FXO | Foreign Exchange Office |
| FXS | Foreign Exchange Station |
| GSM | Sistema Global para Comunicaciones Móviles Global System for Mobile communication |
| IAX(2) | Protocolo de Intercambio de Asterisk (versión 2) |
| IETF | Grupo de Trabajo de Ingeniería de la Internet Internet Engineering Task Force |
| ITU/UIT | Unión Internacional de Telecomunicaciones International Telecommunications Union |
| IVR | Respuesta de Voz Interactiva Respuesta Vocal Interactiva Interactive Voice Response |
| NAT | Traductor de Direcciones de Red Network Address Translator |
| PBX (PABX) | Centralita Telefónica (Automática) Privada Private (Automatic) Branch Exchange |
| PCM/MIC | Modulación por Impulsos Codificados Pulse Code Modulation |
| PSTN/RTB(C) | Red de Telefonía Básica (Conmutada) Public Switched Telephone Network |
| QoS | Calidad de Servicio Quality of Service |
| RFC | Documento de Trabajo de Estandarización (Internet) Request For Comment |
| RTP | Protocolo de Tiempo Real Real-time Transport Protocol |
| SCCP | Protocolo de Control de Llamadas Skinny Skinny Call Control Protocol |
| SIP | Protocolo de Señalización de Sesión(es) Session Initiation Protocol |
| SS7 | Sistema de Señalización (versión) 7 Signalling System 7 |
| TA/ATA | Adaptador Telefónico Telephone Adapter |
| UDP | User Data Protocol |
| VoIP | Voz sobre IP. Telefonía IP |
| VSAT | Terminal de Pequeña Apertura (Comunicaciones por Satélite) Very Small Aperture Terminal |

Descripción de las ilustraciones



Centralita (PBX) con Asterisk



ATA



IAXy



Enrutador inalámbrico (wireless router)



Pasarela inalámbrica (wireless bridge)



Switch



Router



Teléfono analógico



Teléfono DECT



Teléfono VoIP



Softphone



Antena parabólica



VSAT terminal satélite



Satélite

Tabla de Contenidos

| | |
|---|----|
| 1. Introducción..... | 1 |
| 2. La poción mágica..... | 2 |
| 2.1 VoIP..... | 2 |
| 2.2 Estándares Abiertos y Código Libre..... | 3 |
| 2.3 Asterisk..... | 5 |
| 3. La receta..... | 5 |
| 3.1 PBX..... | 5 |
| 3.2 PSTN - RTB..... | 6 |
| 3.3 Señalización en telefonía tradicional..... | 7 |
| 3.3.1 Señalización analógica..... | 7 |
| 3.3.2 Señalización entre centrales telefónicas..... | 8 |
| 3.4 Señalización en telefonía IP..... | 8 |
| 3.4.1 Session Initiation Protocol (SIP)..... | 8 |
| 3.4.2 Servidores Proxy..... | 9 |
| 3.4.3 Protocolos en tiempo real y el NAT..... | 9 |
| 3.4.4 Inter-Asterisk eXchange (IAX)..... | 10 |
| 3.5 Equipamiento para VoIP..... | 11 |
| 3.5.1 Teléfonos VoIP..... | 11 |
| 3.5.2 Telefonía con Software – Soft Phones..... | 11 |
| 3.5.3 Tarjetas de interfaz a la RTB..... | 11 |
| 3.5.4 Adaptador para Teléfonos Analógicos (ATA)..... | 12 |
| 3.6 Codecs..... | 13 |
| 3.7 Calidad de Servicio..... | 13 |
| 3.7.1 Latencia..... | 13 |
| 3.7.2 Jitter – Fluctuaciones de velocidad..... | 14 |
| 4. Puesta en práctica – Creando tu propia PBX..... | 15 |
| 4.1 ¿Qué es lo que necesito?..... | 15 |
| 4.1.1 Consejos de instalación..... | 15 |
| 5. Instalando Asterisk..... | 16 |
| 5.1 Compilando Asterisk..... | 17 |
| 5.2 Descargando Asterisk..... | 18 |
| 5.3 Órdenes Básicas en Asterisk..... | 18 |
| 5.4 Ficheros de configuración..... | 19 |
| 5.5 Peers, Users y Friends..... | 20 |
| 6. ESCENARIO A | |
| - Red telefónica privada en una comunidad rural..... | 22 |
| 6.1 Antecedentes..... | 22 |
| 6.2 Configurando los clientes VoIP..... | 22 |
| 6.2.1 Biblioteca Comunitaria..... | 22 |
| 6.2.2 Hospital Regional..... | 23 |
| 6.2.3 Escuela Primaria..... | 24 |
| 6.2.4 Asociación de agricultores..... | 25 |
| 6.3 Configurando Asterisk..... | 27 |
| 7. ESCENARIO B | |
| - Conectando la RTB..... | 29 |
| 7.1 Incluir el soporte para la tarjeta TDM400P..... | 29 |
| 7.1.1 Gestión de llamadas entrantes desde la RTB..... | 31 |
| 7.1.2 Gestión de llamadas salientes por la RTB..... | 31 |
| 7.2 Añadiendo un terminal analógico a la PBX..... | 32 |
| 7.3 Actualización del plan de marcado..... | 33 |
| 8. ESCENARIO C | |
| - Conectando comunidades usando VoIP..... | 34 |

| | |
|---|----|
| 8.1 Problemas más comunes en enlaces vía satélite | 34 |
| 8.2 Interconectando dos servidores con Asterisk..... | 35 |
| 8.2.1 Telecentro | 35 |
| 8.2.2 Centro de capacitación..... | 36 |
| 8.2.3 La función de registro..... | 36 |
| 9. Para aprender más..... | 37 |
| 10. Conclusión..... | 37 |
| 11. Agradecimientos..... | 37 |

1. Introducción

No fue hasta la primavera del año 2004 cuando descubrí un programa libre * que era capaz de hacer exactamente lo que necesitaba. Por aquel tiempo vivía en Tanzania y tuve que sufrir, no sólo que la línea telefónica era inestable, sino que, además, las tarifas telefónicas eran totalmente desorbitantes. Tan pronto como conseguí un acceso a la Internet empecé a usar una de esas aplicaciones, tan propietaria como conocida, para hacer mis llamadas a Europa. Pronto me di cuenta que el programa no me daba la flexibilidad que yo quería porque lo que realmente necesitaba era poder hacer llamadas a través de mi línea telefónica en Suecia y poder ofrecer esa conexión de voz a mis vecinos.

La idea de usar la Internet como si fuera una red telefónica no era nueva, pero el proyecto que me daba toda esa flexibilidad era, desde luego, “revolucionario”. El programa que permitía que mi computadora se convirtiera en una centralita telefónica se llamaba “*Asterisk*.”¹ No necesité demasiado tiempo para descubrir que *Asterisk* era capaz de hacer mucho más de lo que nunca me pude imaginar. Mientras lo exploraba me daba cuenta del mundo de posibilidades y oportunidades que ofrecía, especialmente en las regiones en desarrollo. La sensación de aquel momento era muy parecida a la de mi primera conexión a la World Wide Web en 1994.

Sin ningún género de dudas aprender a poner en marcha un sistema con *Asterisk* fue una tarea que requirió gran perseverancia (sí, algunos me llaman obcecado). Empecé a escribir estas líneas de introducción a la voz IP porque, aunque ya existen algunos libros muy buenos sobre el tema, no he encontrado un documento que describa los conceptos más básicos en palabras que la mayoría de los mortales puedan entender. Esta “guía” pretende describirte los conceptos esenciales de la telefonía sobre IP y darte unos ejemplos concretos de su potencialidad en regiones en desarrollo. Como el reto al escribir este documento era crear un documento breve sin pecar de simplismo, una gran parte del esfuerzo ha sido intentar ser lo más pedagógico posible.

Ten paciencia. La persistencia es la clave del aprendizaje de un autodidacta.

Antes de describir cómo puedes crear tu propio sistema de telefonía, introducimos los conceptos básicos de telefonía sobre Internet (Secciones 2 y 3). Dedícale tiempo a leer estas dos secciones – a largo plazo, entender los conceptos es mucho más importante que instalar un programa u otro.

Las dos siguientes secciones están dedicadas a aquellos que quieran poner la teoría en práctica: montar tu propia centralita (Sección 4) e instalando y configurando los programas (Sección 5).

En lugar de listar todas las órdenes y configuraciones posibles, hemos seleccionado tres escenarios prácticos como ejemplos ilustrativos. Recuerda que el objetivo de este documento es ayudarte en tus primeros pasos. Los tres escenarios que hemos elegido son:

- Telefonía privada en una comunidad rural (Sección 6)
- Conectando una red local telefónica a la RTB (Sección 7)

1. *Asterisk* es un proyecto de código libre basado en las ideas del “Proyecto de Telefonía Zapata”
<http://www.asteriskdocs.org/modules/tinycontent/index.php?id=10>

- Interconectando dos comunidades remotas (Sección 8)

Para terminar hemos incluido algunas referencias útiles a algunos recursos que te ayudarán a aprender más (Sección 9).

2. La poción mágica

Tres son los elementos que te permitirán desplegar una infraestructura de telefonía: *VoIP*, *estándares* abiertos y los *programas libres y abiertos*.

2.1 VoIP

Una definición general de Voz sobre IP (también conocida como telefonía IP) es la posibilidad de transportar conversaciones telefónicas en paquetes IP. Cuando hablamos de “VoIP”, nos referimos a “la telefonía en Internet” en el sentido más amplio de la expresión. El término VoIP no se refiere a ninguno de los mecanismos concretos que existen para llevar las señales de voz de un sitio a otro en la red. Existen docenas de tecnologías que permiten hablar por la red. Las alternativas tecnológicas de VoIP se pueden dividir de una manera sencilla en dos grandes grupos: tecnologías cerradas-propietarias y sistemas abiertos. En el primer grupo de tecnologías nos encontramos con el conocido Skype o el ya legendario Cisco Skinny (SCCP).² En el segundo grupo de tecnologías nos encontramos con los estándares abiertos basados en SIP,³ H.323⁴ o IAX.⁵

H.323 es un protocolo desarrollado por la UIT que cobró cierta fama porque era el más usado por los grandes operadores en sus redes troncales. SIP ha incrementado su popularidad cuando las tecnologías de VoIP se han hecho más presentes en el “bucle local.”⁶ Últimamente hemos presenciado el nacimiento y el fuerte crecimiento de una nueva alternativa conocida como IAX. IAX2 (por ser la versión 2) está fuertemente influido por el modelo comunitario de desarrollo abierto y tiene la ventaja de haber aprendido de los errores de sus predecesores. IAX2 resuelve muchos de los problemas y limitaciones de H.323 y SIP. Aunque IAX2 no es un estándar en el sentido más oficial de la palabra (RFC),⁷ no sólo tiene el gran reconocimiento de la comunidad sino todos los pre-requisitos para convertirse en el remplazo (de facto) de SIP.

Una de las características esenciales de todos los protocolos tradicionales de voz sobre IP es el derroche de ancho de banda. Ese exceso de bits en la red es debido a la necesidad de enviar

-
2. Skinny es un protocolo de control para terminales. Originalmente desarrollado por Selsius Corporation y ahora bajo el control y diseño de Cisco Systems, Inc. Uno de los clientes más famosos de Skinny es la serie Cisco 7900 de teléfonos IP.
 3. El protocolo de inicio de sesión (SIP) es el resultado del trabajo del IETF y define el manejo de sesiones entre uno o más participantes.
 4. H.323 es un conjunto de recomendaciones de la UIT-T que define un grupo de protocolos para ofrecer sesiones audiovisuales en una red conmutada de paquetes. El H.323 se usa en el famoso programa Netmeeting.
 5. IAX2 es un protocolo de comunicación de voz IP que se usa en *Asterisk*, una centralita de código abierto y libre. IAX2 permite conexiones entre servidores *Asterisk* y clientes IAX2.
 6. El bucle de área local es un enlace físico que conecta al cliente con la terminación de la red de telefonía del proveedor de servicios de telecomunicaciones.
 7. El Request for Comments (RFC) (en castellano: solicitud de comentarios) es una serie de documentos numerados e informales que buscan construir consensos en favor de la estandarización de protocolos y servicios para la Internet.

información adicional en cada una de las cabeceras de los paquetes IP. Este problema tiene especial importancia en regiones en desarrollo donde el acceso a ancho de banda es limitado y los costes de conexión a Internet pueden llegar a ser hasta 100 veces mayor que en Europa o Norteamérica.⁸

Para que te hagas una idea del gasto adicional de ancho de banda necesario para enviar voz sobre Internet podemos citar como ejemplo que un audio comprimido de 5.6 kbit/seg necesita de hasta 18 kbit/seg. La diferencia entre los 5.6 y los 18 kbit/seg son esos bits en las cabeceras de los paquetes. Las cabeceras son toda esa información adicional que es necesaria para encaminar correctamente cada uno de los paquetes de voz al receptor. Una de las ventajas de IAX2 es que ha sido capaz de reducir considerablemente ese exceso de bits por paquete. Además, es capaz de agrupar los paquetes de distintas conversaciones, que van en una misma dirección en la red, en uno sólo. Al ser capaz de agregar múltiples paquetes de distintas conversaciones dentro de uno sólo, el exceso de información introducido por las cabeceras se reduce en cada una de las conversaciones.

Como resultado de las pruebas realizadas durante la elaboración de esta guía (usando una conexión telefónica a la red), evidenciamos las ventajas de utilizar IAX2 frente a la misma conversación usando SIP.⁹

2.2 Estándares Abiertos y Código Libre

No podríamos estar hablando de la libertad de construir nuestra propia red telefónica sin la existencia de los estándares abiertos y el código libre. Los estándares abiertos permiten que cualquiera pueda implementar un sistema con garantías de interoperabilidad. Gracias a esa interoperabilidad de nuestro diseño no sólo podemos crear nuestra red telefónica sino que, además, podemos conectarla a la red telefónica global. Con el código libre podemos aprender de experiencias parecidas, integrar sus soluciones y compartir nuestros propios resultados con los demás.

Una de las primeras preguntas que merece una respuesta es: ¿por qué deberías crear tu propia infraestructura de voz sobre IP y no seguir usando servicios gratuitos como Skype?

La respuesta es simple: sostenibilidad y flexibilidad. Los servicios gratuitos te pueden solucionar una necesidad a corto plazo pero nunca garantizar tu independencia o el control de tu propio proceso de aprendizaje y desarrollo. No se trata de una cuestión puramente técnica. El problema no es decidir cuál es la mejor de las tecnologías sino cuál es la que permite que las comunidades sean dueñas de su propio desarrollo y que puedan adaptarla a sus propias necesidades.

Es muy difícil imaginar un desarrollo sostenible sin transferencia de conocimiento y reapropiamiento tecnológico. Una solución basada en estándares abiertos y código libre no es

8. 1 Mbps en el Este de África cuesta más de 1000 USD/mes mientras que la misma capacidad en Suecia cuesta menos de 10 USD/mes.

9. Una conversación de voz IP usando un codec como el G.729 (8 Kbps) requiere unos 30 Kbps usando SIP y tan sólo 24 Kbps con IAX2. Si agregamos cinco llamadas simultáneas cada llamada se reduce a 13 Kbps.

sólo una buena solución desde un punto de vista puramente técnico sino que además permite la posibilidad de adaptación para mejorarse a la realidad local.

Para ser conscientes de la importancia de los estándares abiertos quizás sea bueno empezar presentando una definición de “estándar.” Un estándar es un conjunto de reglas, condiciones o requerimientos que describen materiales, productos, sistemas, servicios o prácticas. En telefonía, los estándares garantizan que todas las centrales de telefonía sean capaces de operar entre sí. Sin ese conjunto de reglas comunes un sistema de telefonía de una región sería incapaz de intercambiar llamadas con otro que esté, tan sólo, unos kilómetros más allá. Aunque muchos de los estándares de telefonía son públicos, los sistemas siempre han estado bajo el control de un grupo muy limitado de fabricantes. Los grandes fabricantes de sistemas de telefonía son los únicos capaces de negociar contratos a nivel regional o incluso nacional. Ésta es la razón que puede explicar porqué es muy común encontrar siempre el mismo tipo de equipos a lo largo de un mismo país.

Los equipos de telefonía tradicionales, además, tienen la particularidad de haber sido diseñados para realizar un conjunto de tareas muy concretas. Normalmente, son equipos informáticos con aplicaciones muy específicas. Aunque las reglas que gobiernan la telefonía (los estándares) son relativamente abiertas, no es el caso de los equipos informáticos que los implementan. Al contrario de los estándares, el funcionamiento interno siempre se mantiene en secreto.

Dentro de la “poción mágica de la telefonía” los estándares abiertos son un ingrediente necesario, pero lo que realmente ha permitido esta nueva “revolución” ha sido la posibilidad de emular la funcionalidad de los sistemas de telefonía tradicionales con un programa funcionando en un ordenador personal. Todos los elementos necesarios están a tu alcance:

- tienes el acceso a los programas y a los equipos que permiten el intercambio de conversaciones telefónicas.
- tienes una red abierta y pública para intercambiar esas llamadas (la Internet).
- tienes la posibilidad de modificar cada uno de los elementos para adaptarlos a tus propias necesidades.

2.3 Asterisk

Asterisk es una implementación libre de una centralita telefónica. El programa permite tanto que los teléfonos conectados a la centralita puedan hacer llamadas entre ellos como servir de pasarela a la red telefónica tradicional. El código del programa fue originalmente creado por Mark Spencer (Digium) basado en las ideas y el trabajo previo de Jim Dixon (proyecto de telefonía Zapata). El programa, sus mejoras y correcciones, es el resultado del trabajo colectivo de la comunidad del software (programas) libre. Aunque *Asterisk* puede funcionar en muchos sistemas operativos, GNU/Linux es la plataforma más estable y en la que existe un mayor soporte.

Para usar *Asterisk* sólo se necesita un ordenador personal (PC), pero si quieres conectarte a la red telefónica tradicional debes añadir el correspondiente periférico dedicado.

3. La receta

Esta sección resume los conceptos principales de VoIP. Entender cada uno los conceptos te va a ser muy útil cuando configures cualquier tipo de programa relacionado con telefonía IP. Aunque VoIP es una área enorme de conocimiento, hemos seleccionado cuidadosamente un número de conceptos esenciales. Esta sección incluye una descripción, básica pero sólida, de lo que necesitas saber para dar tus primeros pasos en la creación de un sistema de telefonía.

3.1 PBX

El término PBX o PABX es una de esas siglas que dicen bastante poco. PBX son las primeras letras del término inglés Private (Automatic) Branch Exchange. En palabras simples, el uso más común de una PBX es compartir de una a varias líneas telefónicas con un grupo de usuarios. Una PBX se emplaza entre las líneas telefónicas y los teléfonos (terminales de voz). La PBX tiene la propiedad de ser capaz de redirigir las llamadas entrantes a uno o varios teléfonos. De una manera similar, una PBX permite a un teléfono escoger una de las líneas telefónicas para realizar una llamada telefónica al exterior. De la misma forma que un enrutador (router) en Internet es responsable de dirigir los paquetes de un origen a su destino, una PBX es responsable de dirigir “llamadas telefónicas”.

La palabra “private” en la sigla PBX significa que el dueño del sistema tiene todo el control y decide como compartir los líneas exteriores con los usuarios.

Una PBX no sólo permite compartir un conjunto de líneas con un grupo de usuarios sino que también ofrece la posibilidad de crear servicios de valor añadido como transferencia de llamadas, llamadas a tres,¹⁰ pasarela de voz a correo¹¹ o servicios basados en una respuesta de voz interactiva(IVR)¹², etc.

Una PBX puede ser de gran utilidad en múltiples escenarios. Piensa en las regiones donde el acceso a la red telefónica implica caminar varias horas (sino días) a una cabina o Telecentro. Además, una situación muy común es que sólo exista una línea telefónica por edificio o por población. Una PBX (tu centralita) permitirá compartir esa línea e incluso extender la red telefónica a lugares remotos.

3.2 PSTN - RTB

PSTN es la Red Pública Telefónica Conmutada (Public Switched Telephone Network), “la red de redes telefónicas” o más conocida como “la red telefónica.” En castellano la PSTN es conocida como la red pública conmutada (RTC) o red telefónica básica (RTB). De la misma forma que Internet es la red global IP, la RTB es la amalgama de todas las redes conmutadas de teléfono. Una diferencia muy importante entre la RTB e Internet es la noción de “flujo de información”. En telefonía los flujos de información son cada una de las llamadas o conversaciones mientras que en Internet es cada uno de los paquetes de datos. Desde el punto de vista conceptual la RTB e Internet son muy diferentes y representan dos mundos y filosofías casi antagónicas. Si una conversación se efectúa en una RTB se tiene que reservar un canal (circuito) dedicado de 64 Kbps, pero en Internet la misma conversación puede coexistir con otros servicios de manera simultánea. Aunque esta diferencia pueda parecer irrelevante a primera vista, tiene grandes implicaciones de cara a la implementación de las tecnologías de la información tanto en regiones desarrolladas como en desarrollo. En el modelo tradicional, un “cable de cobre” proporciona acceso a la RTB y ofrece un sólo tipo de servicio: un canal analógico. Si ese mismo cable se usa para conectarse a una red conmutada de paquetes como Internet, se puede implementar cualquier tipo de servicio basado en el protocolo IP.

La RTB ha estado históricamente gobernada por estándares creados por la UIT,¹³ mientras que Internet es gobernada por los estándares del IETF.¹⁴ Ambas redes, la RTB e Internet usan direcciones para encaminar sus flujos de información. En la primera se usan números

10. Llamada-a-tres es la posibilidad de tener a más de dos personas hablando simultáneamente en la misma conversación.

11. Una pasarela de voz a e-mail permite grabar un mensaje de voz en un adjunto de correo electrónico (como si fuera un contestador automático). El mensaje se graba en un fichero de audio y se envía a una cuenta de correo.

12. Un sistema de voz interactivo (Interactive Voice Response) permite seleccionar una opción de un menú a través de la voz o del teclado del terminal.

13. La ITU o UIT en castellano es la Unión Internacional de Telecomunicaciones, una organización internacional responsable de estandarización, gestión del radio espectro y de la organización de acuerdos de interconexión entre países que permitan el intercambio de llamadas internacionales. La UIT es parte de la ONU y tiene una estructura de miembros formal.

14. El IETF (Internet Engineering Task Force) es un conjunto de grupos de trabajo responsables de estandarización de Internet. La organización es abierta, formada por voluntarios y sin ningún requerimiento formal para ser miembro.

teléfonicos para conmutar llamadas en las centrales telefónicas, en Internet se usan direcciones IP para conmutar paquetes entre los enrutadores (routers).

3.3 Señalización en telefonía tradicional

Las centrales telefónicas son los “routers” de la RTB. Un Foreign Exchange Office (FXO) es cualquier dispositivo que, desde el punto de vista de la central telefónica, actúa como un teléfono tradicional. Un FXO debe ser capaz de aceptar señales de llamada o ring, ponerse en estado de colgado o descolgado, y enviar y recibir señales de voz. Asume que un FXO es como un “teléfono” o cualquier otro dispositivo que “suena” (como una máquina de fax o un módem).

Un Foreign Exchange Station (FXS) es lo que está situado al otro lado de una línea telefónica tradicional (la estación). Un FXS envía el tono de marcado, la señal de llamada que hace sonar los teléfonos y los alimenta. En líneas analógicas, un FXS alimenta al FXO. El FXS utiliza alrededor de 48 voltios DC para alimentar al teléfono durante la conversación y hasta 80 voltios AC (20 Hz) cuando genera el tono de llamada (ring).

Una PBX que integra periféricos FXO y FXS puede conectarse a la RTB y incorporar teléfonos analógicos. Las líneas telefónicas que vienen del operador se tienen que conectar a una interfaz FXO. Los teléfonos se deben conectar a las interfaces FXS de la centralita.

En resumen, dos reglas fáciles que debes recordar son:

1. Un FXS necesita estar conectado a un FXO (como una línea telefónica necesita estar conectada a un teléfono) o viceversa.
2. Un FXS suministra energía (elemento activo) a un teléfono FXO (elemento pasivo)

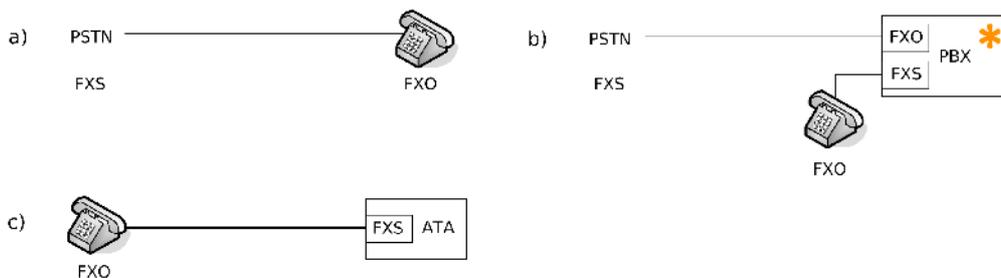


Imagen 1: a) Un teléfono analógico es un dispositivo FXO conectado a una línea telefónica (RTB) actuando como FXS b) Una PBX puede incorporar tanto interfaces FXS como FXO. c) Un adaptador telefónico o ATA actúa como un FXS.

3.3.1 Señalización analógica

Cada vez que usas una línea telefónica se intercambian un conjunto de “señales”. Las señales sirven para ofrecer información del estado de la llamada al usuario. Algunas de esas señales son el tono de marcado o el tono de línea ocupada. Estas señales se transmiten entre el FXS y el FXO haciendo uso de un protocolo conocido como “señalización”.

Por desgracia, existen muchas maneras de generar este tipo de señales. Cada uno de los mecanismos es conocido como “método de señalización”. Los métodos de señalización son diferentes de un lugar a otro, así que debes conocer de antemano el método de señalización que se usa en tus líneas telefónicas. Dos de los métodos de señalización más conocidos son el “loop start” y el “ground start”. Si desconoces el método de señalización que debes usar puedes empezar probando con “loop start”. Una consecuencia de configurar tu PBX con un método de señalización equivocado es que la línea telefónica se cuelga de manera inesperada.

3.3.2 Señalización entre centrales telefónicas

SS7 es un grupo de estándares desarrollados originalmente por la AT&T y la UIT que, entre otras cosas, se encargan de la gestión del establecimiento de llamadas y su encaminamiento entre centrales telefónicas en la RTB. Una cosa muy importante que debes entender es que en la red telefónica tradicional, la voz y las señales auxiliares (señalización) están claramente separadas. Esto significa que existe un “circuito” dedicado a voz y otro circuito independiente para el intercambio de las señales encargadas del establecimiento de las llamadas. Esta información “adicional” necesaria en cada llamada se intercambia usando un protocolo conocido como SS7.

El hecho de que la voz y la señalización están separadas significa que los flujos de información pueden tomar caminos físicos totalmente diferentes. Imagínate que las “conversaciones” pueden viajar por un cable mientras que los números de teléfono de los comunicantes se envían por otro. Este concepto es importante para entender la siguiente sección: señalización en telefonía IP.

3.4 Señalización en telefonía IP

Por herencia histórica, la señalización en voz sobre IP sigue unos principios muy parecidos a la señalización en RTB. Las señales y las conversaciones están claramente diferenciadas. En esta sección introducimos dos protocolos de VoIP que vamos a integrar en nuestra futura PBX: SIP e IAX2.

3.4.1 Session Initiation Protocol (SIP)

El protocolo de señalización de inicio de sesión, del inglés Session Initiation Protocol (SIP), es una especificación para Internet para ofrecer una funcionalidad similar al SS7 pero en una red IP. El protocolo SIP, desarrollado por el IETF, es responsable de establecer las llamadas y del resto de funciones de señalización. Recuerda que, cuando hablamos de señalización en el contexto de llamadas de voz, estamos hablando de la indicación de línea ocupada, los tonos de llamada o que alguien ha contestado al otro lado de la línea.

SIP hace tres cosas importantes:

1. Encargarse de la autenticación.

2. Negociar la calidad¹⁵ de una llamada telefónica.
3. Intercambiar las direcciones IP y puertos que se van utilizar para enviar y recibir las “conversaciones de voz”.

3.4.2 Servidores Proxy

Aunque dos dispositivos SIP (teléfonos IP) pueden comunicarse directamente, SIP normalmente hace uso de algunos elementos adicionales llamados “proxies” para facilitar el establecimiento de las llamadas. Un “proxy” opera como un representante (apoderado) que se encarga de negociar entre dos partes. Con la ayuda de un “proxy” puedes mover físicamente tu número de teléfono en Internet. Los números no están asociados a un sitio concreto sino que se pueden mover siempre y cuando notifiqemos al “proxy” de nuestra (nueva) ubicación. Como el “proxy” funciona como un intermediario es capaz de indicar a las partes dónde se encuentran los teléfonos. Este servidor intermedio en SIP aprende la posición de sus usuarios durante un proceso que se conoce como “registro”.

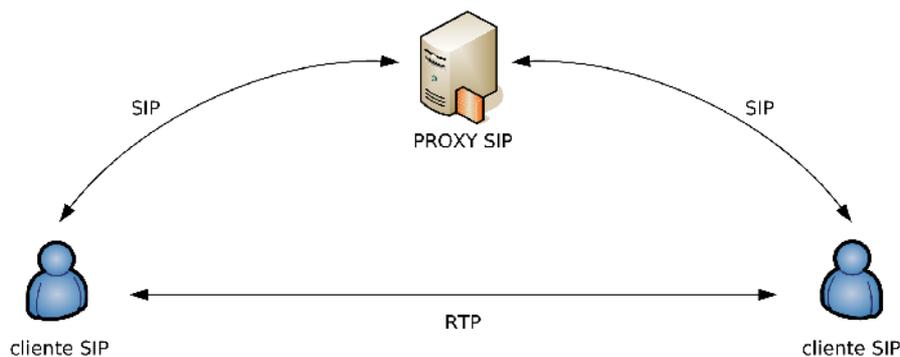


Imagen 2: El proceso de registro entre clientes y el servidor “proxy”. La señalización (SIP) y las conversaciones de voz (RTP) viajan por caminos diferentes.

3.4.3 Protocolos en tiempo real y el NAT

En Internet, las conversaciones que usan señalización de tipo SIP resultan en flujo constante de paquetes de pequeño tamaño entre los comunicantes. Estos paquetes de voz hacen uso de otro protocolo llamado RTP. El protocolo de transporte de tiempo real o Real-time Transport Protocol (RTP) es el encargado de llevar las conversaciones (la voz) de un lado a otro. En el RTP se define un mecanismo estándar para enviar audio y vídeo en Internet. De la misma forma que en una conversación existen dos flujos de voz, en una conversación en una red IP tenemos dos flujos de paquetes RTP.

Los Network Address Translators (NATs) son los grandes enemigos del RTP. Una red con un NAT consiste en varios ordenadores compartiendo, con el mundo exterior, una sólo dirección IP pública. Las máquinas situadas dentro de la red NAT usan direcciones “privadas”. Aunque

15. Una de las grandes diferencias entre la telefonía tradicional y la IP es que la calidad de servicio de una conversación se puede negociar.

el NAT permite conectar más fácilmente ordenadores a la red, lo hace al precio de no permitir una conexión puramente bi-direccional. El efecto de un NAT en voz sobre IP es que no se pueden recibir conexiones iniciadas desde el exterior.

Existen varios problemas relacionados con NAT y VoIP. El más común de los problemas es conocido como “audio en una sola dirección” (one-way audio). Como recordarás, una conversación está compuesta por dos flujos de paquetes RTP distintos. En presencia de un NAT, sólo el flujo de dentro a fuera no es bloqueado; el flujo de fuera a dentro no tiene la misma suerte y puede atravesar el NAT. La consecuencia: el que inicia la llamada desde dentro del NAT no puede escuchar a la otra parte. Si los dos comunicantes se encuentran dentro de NATs las cosas se complican aún más, hasta el punto de que ningún flujo de audio llega a su destino final.

Por desgracia, las direcciones IP privadas y los NAT están especialmente presentes en todos los lugares de las regiones en desarrollo. Configurar una red con señalización SIP y NATs no es trivial. Esta guía incluye algunos consejos generales en la sección que describe los escenarios prácticos.

3.4.4 Inter-*Asterisk* eXchange (IAX)

La segunda versión del protocolo de comunicación entre *Asterisks* (Inter-*Asterisk* eXchange) se conoce como IAX2.¹⁶ IAX2 es una alternativa al protocolo de señalización SIP. IAX2 fue creado como parte del desarrollo de la PBX *Asterisk*. A diferencia del SIP, que usa dos flujos de datos para voz y otros dos para señalización, IAX2 usa sólo un par de flujos donde voz y datos coexisten. Esta forma de enviar tanto las conversaciones como la señalización por el mismo canal se conoce como *in-band*, en contraste con el método que usa SIP, el *out-of-band*¹⁷.

Debido a su diseño, IAX2 es la opción más adecuada en regiones en desarrollo donde existen gran presencia de NATs. Además, IAX2 es capaz de empaquetar llamadas simultáneas en un sólo flujo de paquetes IP. Este mecanismo es conocido como “trunking” y su implementación resulta en ahorros en el consumo de ancho de banda.

El concepto de “trunking” se puede explicar con la siguiente metáfora: imagínate que necesitas mandar cinco cartas a gente que vive en otro país. Una posibilidad es usar un sobre por cada una de las cartas; la otra es usar un único sobre e incluir el nombre del destinatario final en la cabecera de cada una de las cartas. La agregación de llamadas en telefonía IP funciona de la misma forma y permite enviar múltiples cartas (llamadas) en un único sobre (paquete IP).

En resumen, el diseño de IAX2 es más adecuado para regiones en desarrollo por tres razones:

1. Reduce el uso de ancho de banda por llamada.
2. Está diseñado para operar en presencia de NATs (soporte nativo) y es más fácil de usar detrás de los cortafuegos.

16. IAX2 es un protocolo de telefonía IP que utiliza un reducido número de bits en las cabeceras y que está diseñado para permitir la comunicación entre centralitas y clientes *Asterisk*. El contenido de voz en los paquetes se envía usando una cabecera de tan solo 4 octetos (32 bits). Una cabecera más compleja de 12 octetos se utiliza con los paquetes de control y en algunos paquetes especiales de voz (uno por minuto aproximadamente).

17. La idea de enviar la señalización dentro del canal de voz (*in-band*) obliga a separar los paquetes de voz de los paquetes de señalización. Aunque este diseño requiere más gasto de procesamiento (CPU) ofrece mejores propiedades en presencia de cortafuegos y NATs.

3. Reduce aún más el ancho de banda cuando se realizan varias llamadas simultáneas (como resultado del “trunking”)

3.5 Equipamiento para VoIP

3.5.1 Teléfonos VoIP

Un teléfono de VoIP o teléfono IP es un equipo especialmente diseñado para conectarse a una red de telefonía IP. Los teléfonos IP pueden implementar uno o varios protocolos de voz sobre IP.¹⁸

Algunas de las características que debes tener en cuenta cuando compres un teléfono IP son:

- Ancho de banda reducido: inclusión de codecs de alta compresión (e.g. G.729, gsm, speex).
- Buena interfaz de administración: inclusión de interfaz web.
- Salida de audio: inclusión de salida externa de audio y soporte de manos-libres (para educación a distancia).

Existen muchos modelos en rango de precios de 100-120 USD que hacen mucho más de lo que vas a necesitar y funcionan perfectamente con *Asterisk*.¹⁹

3.5.2 Telefonía con Software – Soft Phones

Una alternativa al uso de equipos dedicados (físicos) de VoIP es el uso de programas para emularlos. Estos programas se conocen como “soft phones” y funcionan en cualquier ordenador personal. El único requerimiento es tener una tarjeta de sonido en funcionamiento y estar seguro de que el cortafuegos instalado en tu máquina no está bloqueando a la aplicación.²⁰

Si quieres reducir el ancho de banda usado por tus conversaciones elige un “soft phone” que tenga soporte para el protocolo IAX2 y activa un codec de alta compresión.

3.5.3 Tarjetas de interfaz a la RTB

Si quieres encaminar las llamadas de tus terminales de VoIP a la red telefónica tradicional (RTB) necesitas un periférico especializado en la PBX. Una solución modular para *Asterisk*, que permite conectar líneas y teléfonos analógicos, es una tarjeta PCI fabricada por Digium: *TDM400P wildcard* (la palabra inglesa wildcard significa “comodín”).

18. En Septiembre del 2006 ya existen varias compañías que han fabricado teléfonos IP con soporte IAX2.

19. Como parte de esta guía hemos evaluado de manera positiva los siguientes modelos con *Asterisk*: SwissVoice IP10S (150 USD), Thomson ST2030 (100 USD), Gulfstream Budgetome (75 USD) y el Cisco 7940 (300 USD).

20. Los siguientes teléfonos “softphones” han sido evaluados como parte de este guía: IAXClient (IAX2), X-Lite (SIP)

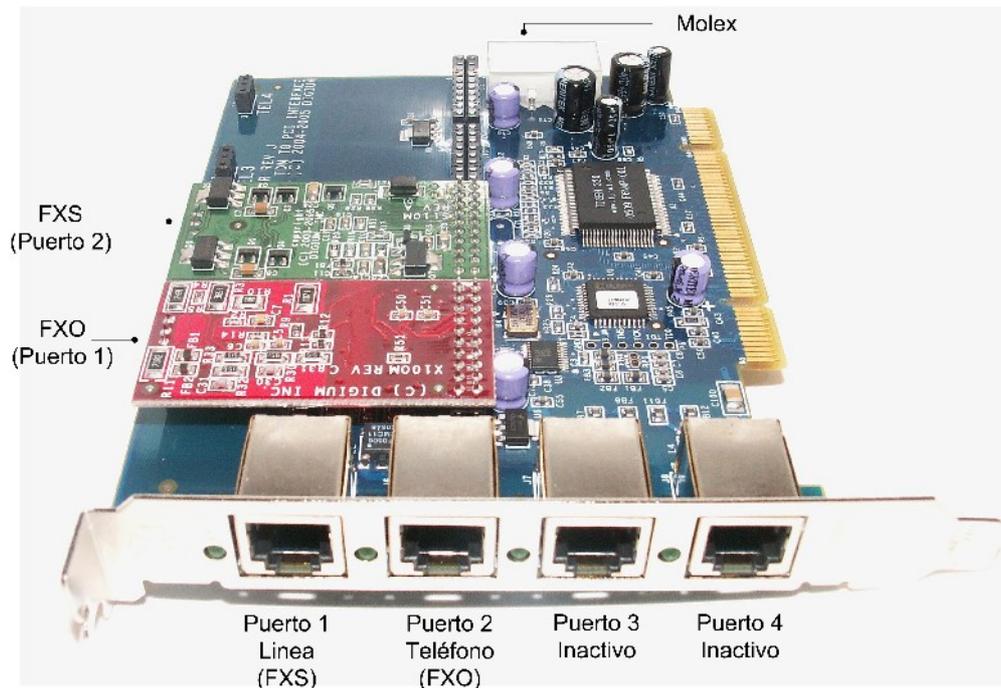


Imagen 3: Un tarjeta TDM400P con sus cuatro puertos. Los dos primeros puertos (Puerto 1: FXO, Puerto 2: FXS) están ocupados mientras que los dos últimos puertos (Puerto 3 y 4) están inactivos.

La tarjeta, conocida como “TDM wildcard” o simplemente “TDM”, tiene cuatro puertos y se llama “wildcard” porque permite insertar un máximo de cuatro expansiones. Las expansiones son módulos de unidades FXO o FXS. Desde el punto de vista práctico esto significa que a la TDM se le pueden conectar hasta cuatro líneas telefónicas (4 módulos FXO), o dos líneas entrantes (2 FXO) y dos teléfonos analógicos (2 FXS modules), etc.

Una buena idea es comprar inicialmente la versión de la TDM400P con dos módulos. Un módulo FXO (para conectar una línea telefónica) y un FXS (para conectar un teléfono analógico). Si en el futuro necesitas expandir la tarjeta siempre puedes añadir módulos extra más tarde.

3.5.4 Adaptador para Teléfonos Analógicos (ATA)

Un adaptador para teléfonos analógicos (ATA) o en breve, adaptador telefónico (TA) conecta un teléfono ordinario a una red de VoIP. Un ATA tiene un conector RJ-11 (el conector de teléfono) y un RJ-45 (el conector de red o Ethernet). Un ATA funciona como si fuera un adaptador FXS, por un lado habla con el teléfono analógico y por el otro opera en modo digital con la red de voz IP. Si quieres implementar una red en una región en desarrollo no es una mala opción utilizar ATAs en lugar de teléfonos IP. Los ATAs suelen ser más baratos y al ser más pequeños suelen ser más fáciles de “nacionalizar” en las aduanas. Otra de las ventajas de usar un ATAs es que puedes conectar cualquier tipo de aparato telefónico a la red IP, por

ejemplo, se pueden conectar una cabina telefónica (de monedas o tarjeta), un fax o un teléfono inalámbrico (DECT)²¹.

Una de las opciones tipo ATA que usa el protocolo IAX2 es el modelo s101i de Digium. Este ATA también se le conoce con el nombre de IAXy²².

3.6 Codecs

Un algoritmo compresor/de-compresor (codec) es un conjunto de transformaciones utilizadas para digitalizar la voz. Los codecs convierten tanto la voz en datos (bits) como los datos en voz. Un codec toma una señal analógica y la convierte en una señal digital en un formato binario (0s y 1s). Existen muchas formas de digitalizar audio y cada una de esas formas resulta en un tipo de codec. En general puedes asumir que a mayor compresión vas a obtener mayor distorsión (peor calidad). Un codec se considera mejor que otro cuando es capaz de ofrecer mejor calidad de voz usando la misma cantidad de ancho de banda.

Un circuito de la RTB (el teléfono de siempre) usa un codec conocido como Modulación por Impulsos Codificados (MIC) del inglés Pulse Code Modulation (PCM). El MIC es un codec de alta calidad que necesita 64 kbps. Dos estándares de compresión MIC son el *micro-law* (*u-law*) y el *a-law*. A estos estándares se les conoce también como G711u y G711a respectivamente. El *micro-law* se usa normalmente en Norteamérica y el *a-law* en Europa. La familia de codecs G711 no requieren de gran procesamiento y por eso están disponibles en la mayoría (si no todos) los equipos de voz IP.

En países en desarrollo, el uso del G.711 no es viable porque requiere demasiado ancho de banda. Debes considerar otro tipo de codecs que hagan un uso más efectivo del los recursos disponibles en la red.

Unas buenas opciones de codecs libres y que usan poco ancho de banda son el codec de GSM y el Speex. El G.729 es un codec propietario altamente robusto pero requiere de una licencia para su uso comercial.²³

3.7 Calidad de Servicio

La calidad de servicio o Quality of Service (QoS) es la capacidad de la red para ofrecer mejoras en el servicio de cierto tipo de tráfico de red. Uno de los grandes retos al implementar VoIP, especialmente en regiones en desarrollo, es garantizar que exista un ancho de banda constante para las conversaciones. Para ofrecer una buena calidad en la conversación, el ancho de banda que necesitan los dos flujos de tráfico se debe garantizar con independencia del estado del resto de las conexiones (incluso si la conexión a Internet está altamente ocupada). Cuando diseñes

21. Los siguientes adaptadores telefónicos (ATA) han sido evaluados como parte de esta guía: Sipura SPA-3000, GlobalTex IAD

22. El IAXy es una ATA de reducido tamaño con soporte IAX2. El ATA no soporta codecs de alta compresión.

23. G.729 es un codec de 8 kbps (aprox. 30 Kbps por conversación usando SIP). El codec fue desarrollado por un consorcio de organizaciones: France Telecom, Mitsubishi Electric Corporation, Nippon Telegraph and Telephone Corporation (NTT) y la Universidad de Sherbrooke. El precio del codec es de 10 USD. Ver <http://www.digium.com/en/products/voice/g729codec.php>

una red de voz IP debes intentar optimizar el ancho de banda, controlar las fluctuaciones de la red (jitter), y minimizar la latencia.²⁴

3.7.1 Latencia

Latencia es sinónimo de retraso, y mide el tiempo que tarda un paquete en viajar de un punto a otro. Para mejorar la calidad de las conversaciones de voz sobre IP es necesario reducir los retrasos al máximo, dando la máxima prioridad al tráfico de voz. Dar más prioridad a los paquetes de voz significa que se les deja “saltarse la cola” de salida y así ocupar una mejor posición que el resto de los paquetes que están esperando para ser transmitidos.

Si la comunicación requiere el uso de un enlace por satélite vas a tener que contar con, al menos, una latencia de 300 ms (0.3 segundos). Para poder reducir el retraso tienes que implementar buenas políticas de calidad de servicio en los enrutadores (routers) y conmutadores (switches) por los que atraviesa tu tráfico de voz. Aunque una conversación es técnicamente posible si existen dos o más enlaces de satélite entre los comunicantes, tienes que estar preparado para esperas del orden de un segundo. Una regla de oro para minimizar la latencia es colocar tu centralita (PBX) en el segmento menos congestionado o saturado de la red.

3.7.2 Jitter – Fluctuaciones de velocidad

En VoIP, el jitter es la variación del tiempo entre la llegada de distintos paquetes. Estas variaciones son debidas a la saturación de la red, la falta de sincronismo o los cambios dinámicos en las rutas. En redes con grandes cambios de velocidad se puede usar un “jitter buffer” para mejorar la calidad de la conversación. Un buffer es un espacio intermedio donde se almacenan los paquetes hasta su procesamiento. La idea básica del “jitter buffer” es retrasar deliberadamente la reproducción del sonido para garantizar que los paquetes más “lentos” hayan llegado. Los paquetes se almacenan en el buffer, se reordenan si es necesario y se reproducen a una velocidad constante. La calidad de voz mejora al precio de incrementar la latencia total.

Muchos equipos de VoIP te dejan ajustar el tamaño del “jitter buffer.” El buffer es ese área donde los paquetes se almacenan para luego ser enviados al procesador de voz en intervalos constantes. El tamaño del buffer se mide en milisegundos. Si el buffer es de 200 ms significa que introducimos un retraso (esperamos) ese tiempo antes de reproducir la voz.

Existen dos tipos de jitter buffers: estático y dinámico. Un buffer estático está implementado como parte del equipo y configurado de manera fija por el fabricante. El dinámico se configura usando un programa y lo puede cambiar el usuario. Un valor común del jitter buffer es de 100 ms. Al incrementar el buffer vamos a mejorar la calidad de la conversación pero no olvides que lo que estás haciendo es incrementar el retardo total (latencia). Debes buscar un valor de compromiso. Ten en cuenta que un retraso total muy por encima de 300 ms hace muy difícil tener una conversación.

24. Debes prestar atención especial a los casos donde vayas a usar VoIP en redes inalámbricas, como las que están basadas en IEEE 802.11b/g/a. En estas redes tienes que asegurarte que les das prioridad al tráfico de voz.

PARTE 2 – Manos a la obra

4. Puesta en práctica – Creando tu propia PBX

4.1 ¿Qué es lo que necesito?

Lo primero que vas a necesitar es un ordenador personal. Cualquier máquina fabricada después del año 2000 debe tener suficiente potencia para hacer funcionar *Asterisk*. A medida que tu sistema crece (especialmente si usas codecs de alta compresión) tendrás que considerar un buen procesador y memoria, pero para empezar cualquier máquina es buena. El ordenador debe funcionar con cualquier distribución del sistema operativo Linux.²⁵

La manera más barata de empezar es utilizar “softphones”. El primer ejercicio es aprender a configurar *Asterisk* para poder establecer una llamada entre dos “softphones” a través de tu PBX. Tus primeros ensayos los puedes hacer con dos ordenadores con tarjetas de sonido. Instala dos clientes de VoIP en cada uno de los ordenadores y usa un tercero para instalar y configurar *Asterisk*.

En el caso de que quieras usar equipos físicos de voz IP (teléfonos VoIP o ATA), o conectarte con la RTB vas a necesitar algunos de los componentes descritos en la Sección 3.5.²⁶

Si quieres construir una PBX portátil y de bajo consumo de energía consulta las soluciones basadas en placa base mini-ITX. Como parte de esta guía construimos una centralita usando el modelo EPIA M10000 (también conocida como Nehemiah). La placa base tiene dos ranuras PCI donde puedes conectar la tarjeta multiuso Digium TDM400P que es compatible con el estándar PCI 2.2.²⁷

4.1.1 Consejos de instalación

Ésta es una lista de consejos de instalación si necesitas instalar una tarjeta PCI TDM400P:

- Asegúrate de que tu máquina tiene una ranura PCI 2.2 libre.
- Si vas a usar un módulo FXS en la tarjeta TDM400P tienes que tener un cable de alimentación con un conector “molex” libre.²⁸
- Si vas a usar una placa base de tamaño reducido como las mini-ITX vas a necesitar una “raiser PCI”. La tarjeta “raiser” gira la posición de la TDM400P 90 grados. Es necesario girar la tarjeta para que entre dentro de la caja.

25. Si eres nuevo en el mundo de Linux puedes pedir un CD de la distribución Ubuntu. <http://www.ubuntu.com>

26. Un buen distribuidor de equipamiento de voz IP con descuentos especiales a organizaciones sin ánimo de lucro es <http://www.avanzada7.com>

27. Un “centralita portátil” te permitirá demostrar la tecnología y al mismo tiempo te ofrece la posibilidad de transportar tu propio demostrador como equipaje de mano.

28. Un conector del tipo Molex es un conector de alimentación de 4-pin de uso común en PCs. Los cables amarillos y rojos dan una tensión de +12V y +5V respectivamente, el cable negro es tierra.

- ¡No conectes una línea telefónica a un puerto FXS! Al conectar un puerto FXS con otro FXS se puede quemar el módulo.



Imagen 4: Una central telefónica portátil basada en una placa mini-ITX con una tarjeta Digium TDM400P

5. Instalando *Asterisk*

Asterisk no es sólo un programa de gran tamaño sino que, además, sigue creciendo al integrar más funcionalidades cada día. Su diseño y arquitectura es tan potente como flexible. La flexibilidad de *Asterisk* también implica cierta complejidad debido a las numerosas posibilidades y opciones. El programa es una herramienta tan potente que puede hacer probablemente todo lo que te puedes imaginar, pero los primeros pasos no son fáciles. Aprender a configurar *Asterisk* me recuerda a esas largas noches de hace diez años delante de la pantalla intentando configurar las primeras versiones del servidor web *Apache* o el *Sendmail*. *Asterisk* puede hacer tantas cosas que tu reto va a ser aprender a hacer bien tan sólo unas pocas.

La metodología que hemos tomado en esta guía no es la de listar todas los posibles órdenes, sino la de citar algunas durante la descripción de tres escenarios prácticos. Los escenarios han sido elegidos para que te sirvan de referencia en tus propias instalaciones. Los ejemplos, aunque básicos, te permitirán la creación de sistemas telefónicos mucho más complejos. Hemos decidido (intencionadamente) simplificar las configuraciones al máximo. Somos conscientes de que algunos de los ejemplos se pueden resolver de otras maneras, no olvides que eres libre de experimentar y explorar tú mismo.

5.1 Compilando *Asterisk*

Como cualquier programa libre, existen *dos* formas principales de instalarlo. El primer método consiste en descargar el código fuente de la red y compilar tu propia versión binaria. El segundo método consiste en descargar una versión ya compilada en forma de paquete.

Si decides compilar *Asterisk* desde su código fuente los siguientes consejos te pueden ser de utilidad:

- Descarga el código fuente de *Asterisk* de <http://www.asterisk.org>²⁹
- Para una versión básica no necesitas bajarte los paquetes de “add ons” o “sounds”.
- Para poder compilar *Asterisk* desde el código fuente es necesario tener un entorno de compilación en tu sistema. Asegúrate que tienes los siguientes paquetes instalados:
 - bison (un generador de analizadores sintácticos)
 - zlib y zlib-devel (bibliotecas de compresión – desarrollo)
 - ncurses y ncurses-dev (bibliotecas de utilidades de consola - desarrollo)
 - openssl y openssl-dev (libssl-dev) (SSL – bibliotecas de desarrollo)
 - libc6-dev (cabeceras y bibliotecas de desarrollo GNU C)
 - gcc y make (el compilador C de gnu y la utilidad make)

La compilación de *Asterisk* no es diferente de otro programa de código libre en Linux:

Para compilar:

```
# make
```

Para instalar:

```
# make install
```

Para instalar los “scripts” de arranque:

```
# make config
```

Para instalar los ficheros de configuración de ejemplo:

```
# make samples
```

Para instalar la documentación de desarrollo:

```
# make progdocs
```

Si quieres usar una tarjeta Digium Wildcard(tm) con *Asterisk* vas a tener que compilar e instalar un controlador llamado zaptel (módulo del kernel).

- Descarga el código fuente del Zaptel de <http://www.asterisk.org>. Por desgracia, el controlador de zaptel no forma parte del núcleo (kernel) de Linux y tienes que crearte tus propios módulos.
- Asegúrate de que las cabeceras del núcleo del kernel (paquete kernel-headers) está instalado en tu sistema.³⁰

29. A septiembre del 2006, la última versión estable de *Asterisk* es la 1.2.12. La última versión del controlador de la familia de tarjetas Zapata es la 1.2.9.

30. Puedes consultar la versión del núcleo/kernel de tu máquina usando la orden `# uname -a`. Por ejemplo si estas usando un sistema con Ubuntu Dapper (x386) tienes que instalar el siguiente paquete de cabeceras: `headers linux-headers-2.6.15-25-386`

5.2 Descargando *Asterisk*

También es posible descargar *Asterisk* ya compilado. El programa compilado (binario ejecutable) se obtiene en la forma de “paquete.” Un paquete contiene todos los ficheros necesarios para ejecutar un programa. Dependiendo de la distribución de Linux que estés usando existen paquetes en distintos formatos (rpm, deb o tgz). Si estás usando una distribución basada en Debian, puedes descargar e instalar *Asterisk* usando la utilidad “*apt*”. Ejecuta la orden *apt-get install* con los siguientes paquetes:

| Paquete | Descripción |
|----------------------------------|---|
| Asterisk-classic (obligatorio) | PBX en código libre – versión original de Digium |
| Asterisk-config (sugerido) | ficheros de configuración para <i>Asterisk</i> |
| Asterisk-dev (opcional) | ficheros de desarrollo para <i>Asterisk</i> |
| Asterisk-doc (sugerido) | documentación para <i>Asterisk</i> |
| Asterisk-sounds-extra (opcional) | ficheros adicionales de sonido para <i>Asterisk</i> |
| Asterisk-sounds-main (opcional) | ficheros de sonido para <i>Asterisk</i> |

A día de hoy, no existe una versión binaria (compilada) del controlador del núcleo “zaptel”. No tienes más opción que seguir el método descrito en la sección anterior. Descarga el código fuente del controlador del núcleo (zaptel kernel driver) y crea el módulo con la utilidad *make* y *make install*. No olvides que antes de compilar el controlador necesitas tener instaladas las cabeceras del núcleo (kernel) de Linux.

| Paquete | Descripción |
|---|--|
| zaptel (obligatorio) | Utilidades para Zaptel |
| zaptel-source (obligatorio) | Código fuente del controlador del núcleo Zaptel |
| linux-headers-2.6.15-25-386 (depende tu distribución) | Cabeceras del núcleo de Linux para Ubuntu Dapper x386 Kernel |

5.3 Órdenes Básicas en *Asterisk*

Asterisk tiene dos componentes internos: un servidor que normalmente funciona en segundo plano y un cliente (CLI) que supervisa el servidor. Tanto el servidor como el cliente se invocan usando la orden “*Asterisk*” pero utilizando distintos argumentos.

Una vez que tengas *Asterisk* instalado, tienes que aprender algunas órdenes básicas:

Arrancar/Parar *Asterisk* desde el arranque (run level)

```
#/etc/init.d/asterisk {start|stop}
```

Arrancar *Asterisk* desde la línea de órdenes

```
# asterisk
```

Arrancar el servidor en modo de depuración

```
# asterisk -vvvc
```

Arranca en modo de depuración o “verbose” (-vvv) y abre un cliente en modo consola (-c) (con un cliente en modo consola (CLI) puedes supervisar lo que esta pasando en el servidor.

Si el servidor está funcionando en segundo plano te puedes conectar usando el cliente con el argumento (r).

```
# asterisk -r
```

CLI órdenes básicas:

Recarga la configuración

```
#CLI>reload
```

Activa el modo de depuración para SIP o SIP o IAX2

```
#CLI> IAX2 debug  
#CLI> SIP debug
```

Desactiva el modo de depuración para SIP o IAX2

```
#CLI> IAX2 no debug  
#CLI> SIP no debug
```

Muestra el estado de “usuarios”, “peers”³¹ y “canales” para SIP o IAX2

```
#CLI> sip show users  
#CLI> sip show peers  
#CLI> sip show channels  
#CLI> iax2 show peers  
#CLI> iax2 show users  
#CLI> iax2 show channels
```

5.4 Ficheros de configuración

El número de ficheros de configuración que tienes que modificar para hacer funcionar *Asterisk* depende del tipo de tecnologías VoIP que quieras usar en tu instalación actual de manera simultánea. La lógica básica para configurar *Asterisk* se puede resumir en los dos pasos siguientes:

Paso 1: Define y Configura los canales de comunicación

Primero, tienes que definir y configurar el tipo de canales de comunicación que quieres usar. Una manera muy fácil de entender lo que es un canal de comunicación es imaginarse un “cable”. Los canales en telefonía IP no son los cables físicos, sino cables lógicos. Como Internet te permite tener muchas sesiones concurrentes en el mismo cable físico, podemos definir múltiples canales lógicos que operan simultáneamente en el mismo medio.

Recuerda que *Asterisk* te permite interconectar distintos dispositivos usando diferentes protocolos de VoIP. Los ficheros de configuración que necesitas preparar están asociados al tipo de tecnología VoIP que vayas a usar. No es mala idea instalar los ficheros “ejemplo” (samples) como referencia.

Paso 2: Define reglas para tus extensiones (Crear un plan de marcado)

El segundo paso es definir cómo van a interactuar cada uno de los canales entre sí. Una vez que has definido un canal garantizas que las conversaciones puedan entrar y salir de tu PBX

31. N.T. El término “peer” se traduce al castellano como “compañero”. En el caso concreto de *Asterisk* los términos “user” and “peer” se usan para clasificar los tipos de conexiones IP al sistema.

pero además tienes que definir cómo se encaminan cada una de esas conversaciones. Por ejemplo, puedes preferir que una llamada entrante desde la RTB se envíe automáticamente a un teléfono IP o, puedes definir una conexión entre dos teléfonos IP separados 20 kms a través de una red inalámbrica. Todo ese tipo de “inteligencia” entre los canales se debe crear en un fichero de configuración conocido como *extensions.conf*. El fichero de extensiones contiene todas esas reglas de gestión de llamadas a las que se conoce como el *plan de marcado* o *dial plan*.

Para que te hagas una idea más intuitiva de este tipo de conceptos, piensa en los sistemas de telefonía más antiguos. En esos sistemas, existía una persona (el operador) que era responsable de conectar físicamente los cables telefónicos entre dos terminales. Para que una llamada fluyera entre dos líneas de comunicación (canales) se necesitaba contactar primero con el operador (PBX) e informarle de nuestras intenciones. El fichero de extensiones en nuestra PBX suplanta el rol del operador tradicional.

En nuestros tres escenarios vamos a usar cinco ficheros de configuración:

| Fichero de Configuración | Descripción |
|--|---|
| /etc/asterisk/extensions.conf (siempre obligatorio) | Contiene el plan de marcado (dialplan). Interconecta los canales. |
| /etc/asterisk/sip.conf | Se usa para configurar canales tipo SIP (teléfonos SIP y proveedores SIP) |
| /etc/asterisk/iax.conf | Se usa para configurar canales tipo IAX2 (teléfonos IAX2 y proveedores IAX2) |
| /etc/asterisk/zapata.conf | Se usa para configurar las tarjetas de interfaz RTB tipo Zapata. <i>Asterisk</i> usa la configuración para habilitar el canal(es) de la tarjeta en el arranque |
| /etc/zaptel.conf | Configuración de bajo nivel de la tarjeta zaptel. Indica que dispositivo del tipo zaptel estamos usando. La utilidad Zaptel Configurator tool “ztcfg” usa este fichero de configuración antes de arrancar <i>Asterisk</i> |

5.5 Peers, Users y Friends

Uno de los temas más complicados de *Asterisk* (o al menos lo ha sido para mí durante mucho tiempo) es el del uso de la opción *peer*, *user* y *friend* en los ficheros *iax.conf* y *sip.conf*.

Los términos *peer*, *user* y *friend* se usan para clasificar las llamadas entrantes y salientes. Mientras que un “user” es una conexión que se autentifica con nuestra PBX (i.e. una llamada entrante), un “peer” es una llamada saliente. Los “users” nos llaman y nosotros llamamos a los

“peers.”³² Un “friend” es una conexión que se puede comportar tanto como “user” o como “peer”, es decir una conexión saliente o entrante.

Cuando nos llega una conexión entrante del tipo “user” o “friend” tenemos que decidir qué hacer con la conexión. El término “contexto” se usa para definir qué reglas o grupo de reglas del plan de marcado (*extensions.conf*) se deben aplicar a esa llamada concreta. El “contexto” de una llamada entrante se encarga de asociarla con un conjunto de reglas presentes en el plan de marcado. El “contexto” representa el punto de entrada de la llamada en el plan de marcado.

El fichero *extensions.conf* incluye todos los “números” que se pueden acceder desde la PBX en distintas secciones (contextos). Cada uno de los múltiples canales entrantes definidos en cada uno de los ficheros de configuración (*iax.conf*, *sip.conf*, *zapata.conf*) se asocian a cierta sección (contexto) del plan de marcado.

32. Tienes que tener en cuenta una de las excepciones a esta clasificación quizás simplista. Cuando uno de nuestros “peers” actúa como proxy de otros terminales IP, las llamadas entrantes desde ese “peer” se asocian a la sección “peer” correspondiente (en lugar de usar una sección tipo “user”). Esto es debido que cuando un “peer” actúa como proxy, no puede autenticarse en favor de sus clientes. El proxy puede redirigir las llamadas a tu centralita pero no puede autenticarse como el cliente final. *Asterisk* utiliza la dirección IP del “peer” para seleccionar la sección adecuada del fichero de configuración. En resumen, una llamada saliente siempre es tipo “peer”, una llamada entrante puede ser tipo “user”, o tipo “peer” cuando la llamada entrante procede de un proxy.

PARTE 3 – ESCENARIOS

6. ESCENARIO A - Red telefónica privada en una comunidad rural

6.1 Antecedentes

En nuestro primer escenario queremos instalar una PBX en el Telecentro de una comunidad rural y ofrecer telefonía IP a cuatro organizaciones de los alrededores. Después de completar la instalación, cada organización debe ser capaz de hacer llamadas telefónicas gratuitas³³ al Telecentro y a todos los socios conectados.

La siguiente tabla resume la información de cada una de las cuatro organizaciones y lista las cuatro tecnologías diferentes que se pueden usar para conectarse a la centralita. Con el objetivo de presentar un ejemplo lo más ilustrativo posible, hemos elegido una gran variedad de tecnologías de voz IP. En una implantación real debes considerar reducir el número de tecnologías con el fin de facilitar el soporte y mantenimiento.

| Organización | Tecnología | Extensión |
|------------------------|---|-----------|
| Biblioteca Comunitaria | Teléfono VoIP con protocolo SIP | 462 |
| Hospital Regional | ATA usando el protocolo SIP | 463 |
| Escuela de Primaria | ATA usando el protocolo IAX2 | 464 |
| Asociación Ganaderos | Dos teléfonos “Soft Phones” usando SIP y IAX2 | 465, 466 |

6.2 Configurando los clientes VoIP

Antes de describir como configurar la PBX empezamos con una descripción de la instalación de cada uno de los clientes de VoIP.

6.2.1 Biblioteca Comunitaria

El primer cliente está situado en la biblioteca pública de la zona. La biblioteca está situada alrededor de 1 km del Telecentro. En esta organización vamos a instalar un terminal de voz IP

33. Una vez que la infraestructura de comunicaciones se ha desplegado podemos establecer tarifas reducidas para las llamadas internas que permitan cubrir los costos de ampliación y mantenimiento. El uso de voz sobre IP nos permite trabajar en modelos de desarrollo comunitario en donde se puede considerar las llamadas internas como gratuitas.

con soporte para el protocolo SIP. El terminal está conectado directamente a nuestra PBX a través de una pasarela inalámbrica dedicada (enlace punto-a-punto en modo transparente o “bridge”). La dirección IP del terminal IP (192.168.46.2) está en el mismo segmento de red que nuestra PBX (192.168.46.1). Como el enlace inalámbrico está en modo transparente, la PBX y el terminal IP se comunican directamente y no tenemos que preocuparnos de los problemas relacionados con el NAT.

Para configurar cualquier tipo de terminal de voz IP consulta el manual del equipo y busca la manera de activar la interfaz de administración por Web.³⁴ Una vez que entres en la zona de administración busca cómo configurar por Web los siguientes parámetros básicos:

| Parámetro | Valor |
|------------------------------------|---------------|
| Dirección IP del teléfono VoIP | 192.168.46.2 |
| Dirección IP de la PBX (proxy SIP) | 192.168.46.1 |
| Registrar/Register | SI/YES |
| Nombre de Usuario (User/Auth name) | 462 |
| Caller ID | 462 |
| Clave/Password | 462pass |
| Codec | G.711 (u-law) |

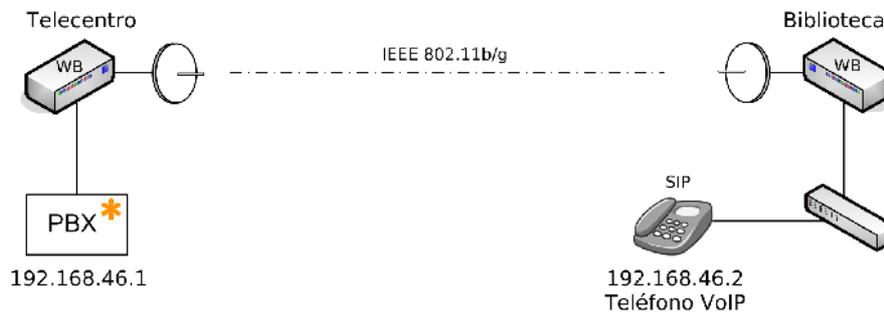


Imagen 5: La biblioteca comunitaria está conectada a la PBX a través de una pasarela inalámbrica. El enlace punto a punto permite al terminal de VoIP hacer y recibir llamadas.

6.2.2 Hospital Regional

El segundo cliente de nuestra red interna es un ATA situado en el hospital. El hospital local está situado al otro lado de la calle, en frente del Telecentro. La conexión entre el Telecentro y el hospital es un cable de par trenzado de 100 metros (ethernet Cat5). La manera de configurar un ATA no es muy diferente a la de configurar un teléfono VoIP. Siguiendo los mismos pasos que en el ejemplo anterior vamos a usar la interfaz de administración web y completar los siguientes parámetros.

34. Existen tres mecanismos básicos para configurar la dirección IP de un teléfono. Teclado: configurar la dirección IP de teléfono a través del teclado del terminal. DHCP: conectar el teléfono a una red con DHCP y extraer la dirección de la información ofrecida por el servidor DHCP. IP de fábrica: leer la documentación para conocer la dirección IP que viene por defecto en el terminal.

| Parámetro | Valor |
|------------------------------------|---------------|
| Dirección IP del ATA | 192.168.46.3 |
| Dirección IP de la PBX (SIP proxy) | 192.168.46.1 |
| Registrar/Register | SI/YES |
| Nombre de Usuario (User/Auth name) | 463 |
| Caller ID | 463 |
| Clave/Password | 463pass |
| Codec | G.711 (u-law) |

En lugar de conectar al ATA un teléfono tradicional decidimos usar un teléfono inalámbrico de tecnología DECT³⁵. La estación base se conecta al puerto RJ-11 del ATA. El resultado es que podemos tener cobertura telefónica DECT en cualquier parte del hospital. El ATA hace de puente entre el teléfono sin cables y la red de voz sobre IP.

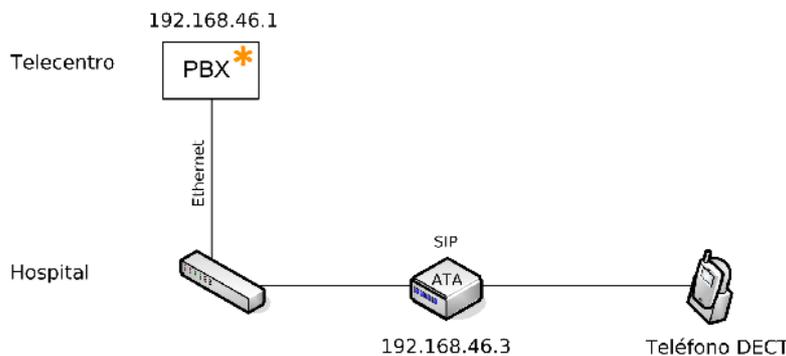


Imagen 6: El Hospital está conectado al Telecentro a través de un cable de par trenzado de 100 m. Un teléfono sin hilos (DECT) está conectado al ATA. El ATA se registra con la PBX usando el protocolo SIP.

6.2.3 Escuela Primaria

El tercer cliente de nuestra red es la escuela de primaria. La escuela está en un edificio adjunto al Telecentro y también se puede conectar con cable de par trenzado. En la escuela se coloca otro equipo ATA, en este caso el equipo usa el protocolo IAX2 en lugar de SIP.

En la escuela usamos el modelo de ATA (interfaz FXS) s101i o IAXy. En este cliente conectamos un teléfono ordinario al puerto RJ11 del adaptador telefónico.

El IAXy no incluye un interfaz de configuración por web. La manera más fácil y cómoda de configurar la unidad es utilizar el propio *Asterisk*. La primera vez que conectas el ATA a la red intenta obtener una dirección IP por DHCP. Mira la información (logs) que te da tu servidor

35. DECT, del inglés Digital Enhanced (formerly European) Cordless Telecommunications, es un estándar de comunicación inalámbrica para teléfonos portátiles que opera en 1.9 Ghz.

DHCP y toma buena nota de la dirección IP del IAXy. El siguiente paso es editar el fichero `/etc/asterisk/iaxprov.conf` incluyendo una sección parecida a la siguiente:

```
[iaxy_school]
ip: 192.168.46.4
netmask: 255.255.255.0
gateway: 192.168.46.1
codec: ulaw
server: 192.168.46.1.2
user: 464
pass: 464pass
register
```

Supongamos que tu servidor DHCP le asignó la dirección IP 192.168.46.100 al ATA. Para actualizar la configuración de la unidad escribe desde la consola de *Asterisk* la siguiente orden:

```
#asterisk -r <ENTER>
#CLI> iax2 provision 192.164.46.100 iaxy_school
```

Si no quieres usar el propio *Asterisk* para actualizar el IAXy puedes usar un programa de administración bajo Windows.³⁶

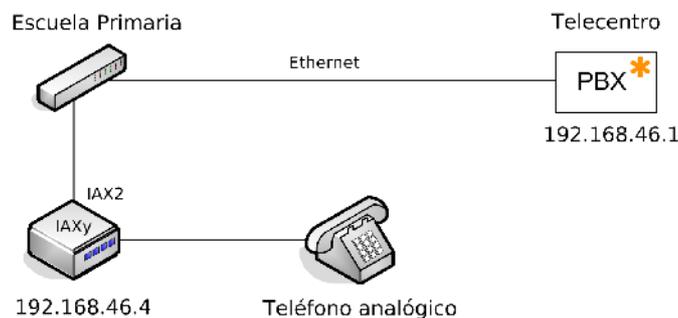


Imagen 7: La escuela primaria está conectada a la PBX a través de un cable de par trenzado. Un ATA con soporte de IAX2 conecta un teléfono analógico a la centralita.

6.2.4 Asociación de agricultores

El cuarto cliente de nuestra red es la asociación de agricultores que está situada en un edificio a 20 kms de nuestras oficinas centrales (Telecentro). La asociación ya tiene dos ordenadores conectados al Telecentro a través de un router NAT inalámbrico. El equipo inalámbrico tiene la dirección IP 192.168.46.5 y sirve una red interna (via NAT) con un rango de IPs (10.10.46.0/24). En las oficinas de la asociación vamos a instalar dos “Soft phones”, uno usando el protocolo SIP y otro el protocolo IAX2.

El elemento que requiere más atención es el “Soft Phone” que usa SIP. Tenemos que asegurar que el audio funciona en las dos direcciones.

En el “Soft phone” SIP:

- Activar el registro con la PBX (register).

36. Existe un programa para sistema operativo Windows para gestionar un teléfono de voz IP IAXy disponible en: <http://dacosta.dynip.com/asterisk>

Una guía para crear una infraestructura de voz en regiones en desarrollo
6. ESCENARIO A - Red telefónica privada en una comunidad rural

- Activar los mensajes que mantienen la conexión activa con la PBX (keep-alive packets).³⁷
- Activar la posibilidad de recibir audio por el mismo puerto que lo enviamos.

En la PBX:

- Configurar en *Asterisk* que el teléfono está dentro de un NAT.

Un buen “Soft Phone” que usa SIP y que funciona bien dentro de los NATs es el programa *X-Lite* de *Xten*.³⁸

El “Soft Phone” que usa IAX2 no necesita una configuración especial para funcionar dentro de un NAT. Lo único que tienes que asegurar es que el puerto de comunicaciones de IAX2, el UDP puerto 4569, no está bloqueado. Una buena opción de “Soft Phone” con soporte para IAX2 es *iaxcomm*.³⁹

Desde el punto de vista conceptual no existen diferencias entre la configuración de un programa cliente de telefonía IP en un ordenador y un terminal físico. Usa los valores de usuario/clave 465/465pass y 466/466pass en cada uno de los programas. Asegúrate que el codec G711 (u-law) está activado y que la dirección de la PBX (proxy) es la 192.168.46.1.

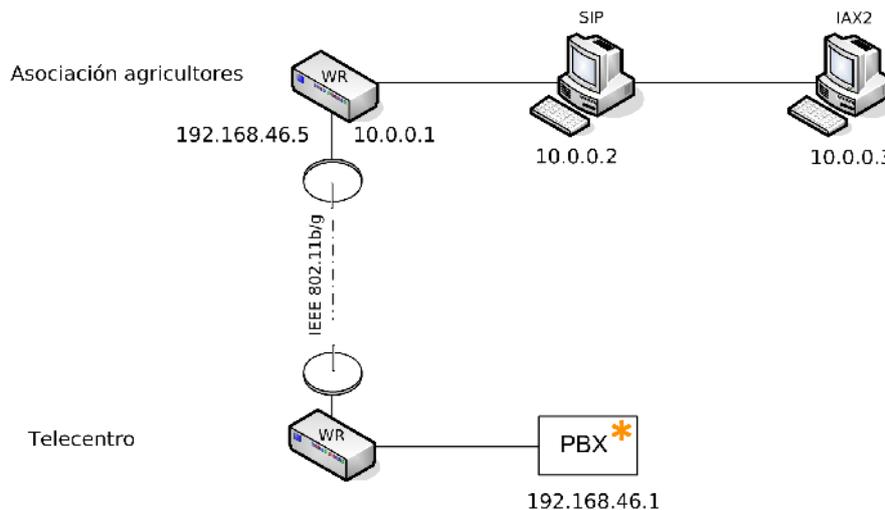


Imagen 8: La asociación de agricultores está conectada a la centralita a través de un NAT inalámbrico. Un programa de telefonía “soft phone” está instalado en cada uno de los ordenadores. El primer cliente usa el protocolo SIP mientras que el segundo usa IAX2.

37. Los paquetes “Keep-alive” son paquetes “vacíos” cuya única finalidad es asegurar que la conexión NAT se mantiene abierta/viva para recibir llamadas entrantes.

38. Hay disponible una versión gratuita del programa *X-ten* en: <http://www.xten.com/index.php?menu=download>

39. *IaxComm* se puede descargar desde: <http://iaxclient.sourceforge.net>

6.3 Configurando *Asterisk*

Paso 1: Definir y configurar los canales de comunicación

Es nuestro primer escenario vamos a usar dos tipos de canales de comunicación: SIP y IAX2. Por lo tanto, tenemos que editar los ficheros *sip.conf* y *iax.conf*.

Ten en cuenta que los comentarios dentro de los ficheros de comunicación en Asterisk comienzan por punto y coma (;).

En el fichero *sip.conf*, incluye los siguientes datos:

```
[462]
type=friend                ; Hacemos y recibimos llamadas
secret=462pass
context=internal_calls    ; Todas las "llamadas entrantes" están asociadas
                           ; al contexto internal_calls

host=dynamic
callerid=Library
disallow=all              ; Primero desactivamos todos los codecs
allow=ulaw                ; Luego activamos el/los codecs que podemos usar

[463]
type=friend
secret=463pass
context=internal_calls
host=dynamic
callerid=Hospital
disallow=all
allow=ulaw

[465]
type=friend
secret=465pass
context=internal_calls
host=dynamic              ; No sabemos la IP por adelantado.
                           ; Aprendemos la IP cuando el usuario se registra

callerid=Farmers1
disallow=all
allow=ulaw

nat=yes                   ; Opciones específicas para soporte NAT
qualify=yes               ; Se usan la IP,puerto del NAT
                           ; Tráfico "dummy" para mantener la conexión viva
```

Y en *iax.conf*, tenemos las siguientes opciones:

```
[464]
type=friend
secret=464pass
context=internal_calls
host=dynamic
callerid=School
disallow=all
allow=ulaw

[466]
type=friend
secret=466pass
context=internal_calls
host=dynamic
callerid=Farmers2
```

```
disallow=all  
allow=ulaw
```

Paso 2: Definir las reglas en el plan de marcado (crear las extensiones)

En el primer escenario tenemos todos los canales (users) asociados al mismo contexto (internal calls). Por lo tanto, sólo tenemos que definir un contexto en el plan de marcado en *extensions.conf* (ver más abajo).

```
[internal_calls]  
exten => 462,1,Dial(SIP/462)  
exten => 463,1,Dial(SIP/463)  
exten => 465,1,Dial(SIP/465)  
exten => 464,1,Dial(IAX2/464)  
exten => 466,1,Dial(IAX2/466)  
exten => t,1,Hangup() ; Extensión especial (Timeout)  
exten => i,1,Hangup() ; Extensión especial (Inválido)  
exten => s,1,Hangup() ; Extensión especial (Sin Destino)
```

La sintáxis del fichero de extensiones *extensions.conf* es muy intuitiva.

- Los corchetes *[nombre_contexto]* indican dónde empieza el contexto y su nombre de identificación. Los nombres de los contextos se han definido en los ficheros de canales de comunicación *sip.conf* y *iax.conf*. (Paso 1)
- Cada una de las secciones del plan de marcado está asociada a un contexto. Cada una de las líneas dentro del contexto tienen el formato:
exten => numero, prioridad, acción

En el ejemplo anterior estamos creando todas las extensiones (462 a 466) y poniéndolas disponibles dentro del contexto *[internal_calls]*. La orden *Dial()* crea un canal SIP o IAX2 con los “peers” de nombre 462 a 466.

7. ESCENARIO B - Conectando la RTB

Nuestro segundo escenario es el resultado de añadir al Escenario A un nuevo canal de comunicaciones. El objetivo es incluir en la red privada anterior un canal hacia la red telefónica tradicional (RTB). Para ello, es necesario añadir a la PBX una interfaz hacia la red telefónica.

En este ejemplo proponemos el uso de una tarjeta PCI TDM400P de *Digium* con un puerto FXO. Como recordarás la tarjeta TDM400P es un interfaz al que se le pueden incluir hasta cuatro módulos FXS/FXO. El uso de un módulo FXO te permitirá conectar la PBX a una línea telefónica analógica.



Imagen 9: El Telecentro usa una tarjeta TDM400P wildcard para (1) conectar la PBX a la RTB (módulo FXO) y (2) añadir una extensión al teléfono analógico (FXS module).

7.1 Incluir el soporte para la tarjeta TDM400P

Los pasos necesarios para poner en funcionamiento el interfaz TDM son cuatro.

Paso 1: Insertar la tarjeta PCI

El primer paso es conectar la tarjeta PCI de medio-tamaño en una de las ranuras libres de tu placa madre. Asegúrate que el conector de tipo *molex* (12/5 volt) del interfaz TDM está conectado a la fuente de alimentación de tu ordenador. La tarjeta TDM recibe corriente a través de un conector hembra conocido como *molex* (es el mismo tipo de conector de 4-hilos con el que se alimentan los discos duros IDE). Si tu fuente de alimentación no tiene un conector macho disponible tendrás que añadir un divisor de corriente (power splitter).

Paso 2: Instalar los controladores del dispositivo

El segundo paso es asegurarse que los controladores del dispositivo están disponibles (se compilaron correctamente y están cargados). Ejecuta el comando `#lsmod`, deberías ver el controlador `wctdm` cargado. Observa que el controlador `wctdm` depende del `zaptel` que a su vez depende del `crc_ccitt`.

```
# lsmod
zaptel                191748  7 wctdm
crc_ccitt             2304    3 hisax,zaptel,irda
```

Paso 3: Configurar la tarjeta TDM400P con la utilidad *ztcfg*

El tercer paso es configurar el dispositivo. Los controladores *wctdm* han sido diseñados para funcionar con una combinación cualquiera de módulos FXS y FXO. Para indicar al controlador que estamos usando un módulo del tipo FXO en el primer puerto de la tarjeta editamos el fichero */etc/zaptel.conf* con la configuración más básica:

```
fxs1s=1
loadzone=us      ; loadzone=es para España
defaultzone=us  ; defaultzone=es
```

La primera línea *fxs1s=1* significa que estamos usando señalización FXS del tipo *Loopstart* en el puerto 1. Recuerda que un módulo FXO necesita señalización FXS.

La segunda y tercera línea del fichero de configuración indican el tipo de “tonos” usados en la línea. El sonido y cadencia de los tonos de marcado o de línea ocupada varían de un país a otro.⁴⁰

Usamos *ztcfg*, una utilidad de configuración de dispositivos *zaptel* que se instala como parte del código fuente de *Asterisk* o el paquete *zaptel*. Ejecutamos */sbin/ztcfg* para cargar y ejecutar el fichero de configuración */etc/zaptel.conf*.

Al ejecutar la utilidad deberías obtener el siguiente resultado:

```
# ztcfg -vv
Zaptel Configuration
=====

Channel map:
Channel 01: FXS Loopstart (Default) (Slaves: 01)
1 channels configured.
```

Paso 4: Configurar *Asterisk* para usar el equipamiento Zapata

El cuarto y último paso es configurar *Asterisk* para que reconozca y use la tarjeta interfaz TDM. Esto creará un nuevo canal de comunicaciones. Editamos el fichero de configuración */etc/asterisk/zapata.conf* de la forma:

```
[channels]
usecallerid=yes
hidecallerid=no
callwaiting=no
threewaycalling=yes
transfer=yes
echocancel=yes
echotraining=yes

context=incoming_pstn
signalling=fxs_ls
channel => 1
```

40. La lista completa de las especificaciones de tonos por países está disponible en:
<http://www.itu.int/ITU-T/inr/forms/files/tones-0203.pdf>

Las tres últimas líneas del fichero *zapata.conf* son las más importantes para una configuración básica. La línea *context=incoming_pstn* indica que todas las llamadas entrantes por el canal de RTB se asociarán a ese contexto. Las siguientes dos líneas indican el tipo de señalización: *fxs_ls* (FXS con Loopstart) y que las llamadas llegan por el canal/puerto 1 de la TDM, *channel => 1*.

Una vez que tengas configurado este nuevo tipo de canal (TDM zapata) sólo te queda decidir cómo gestionar las llamadas entrantes y salientes de la RTB.

7.1.1 Gestión de llamadas entrantes desde la RTB.

En nuestro segundo escenario decidimos que el comportamiento ideal para las llamadas entrantes desde la RTB es el siguiente: una vez que una llamada entra por la línea analógica, queremos que se descuelgue un sistema automático de atención por tonos (IVR). El sistema (nuestro *Asterisk*) preguntará por la extensión deseada, una vez que el número de extensión se introduzca por el terminal (tonos DTMF) la llamada se encaminará a una de las extensiones.

La idea principal es permitir compartir un sólo número de la RTB con todas las extensiones. Esta “inteligencia” se implementa en el fichero *extensions.conf*. Añadimos una nueva sección (contexto) *[incoming_pstn]* de la siguiente forma:

```
[incoming_pstn]
exten => s,1,Answer() ; Contestamos la llamada entrante
exten => s,2,DigitTimeout(10) ; Configuramos los valores máximos para
; introducir el número de extensión

exten => s,3,ResponseTimeout(20)
exten => s,4,Background(vm-extension) ; Un mensaje de voz pregunta: ¿extensión?
exten => i,1,Goto(incoming_pstn,s,1) ; Repetir preguntar si extensión inválida
exten => t,1,Hangup() ; Colgar
include => internal_calls ; Pone a disposición todas las extensiones
internas
```

Nota: la versión final completa del fichero *extensions.conf* está disponible en la *Section 7.3*.

7.1.2 Gestión de llamadas salientes por la RTB

Para que cada uno de los teléfonos IP puedan usar la línea analógica de salida creamos un nuevo contexto *[outgoing_calls]*.

```
[outgoing_calls]
exten => _0.,1,Dial(Zap/1/${EXTEN:1})
exten => t,1,Hangup()
```

La extensión especial “_0.” significa que para alcanzar la RTB se debe empezar marcando el número “0”. La orden *Dial()* crea el puente entre la llamada IP desde los terminales y el canal analógico *Zap/1* (canal 1). Por último *\${EXTEN:1}* significa que el número que se marcará por el canal analógico es el número marcado desde los terminales internos IP sin el primer dígito. En nuestro caso se quita el “0” inicial al marcar por la RTB.

Una vez que hemos creado un nuevo contexto en el plan de marcado *[outgoing_calls]*, es necesario que los terminales tengan acceso a la extensión de salida. La manera más fácil de

conseguirlo es añadir una línea del tipo *include => outgoing_calls* al final del contexto *[internal_calls]*

```
include => outgoing_calls
```

7.2 Añadiendo un terminal analógico a la PBX

En nuestro primer escenario hemos configurado cinco terminales de VoIP en cuatro organizaciones, pero el Telecentro que aloja la PBX no tiene ninguna extensión de voz. La manera más simple de añadir al Telecentro su extensión de voz es instalar un módulo FXS en la PBX. La tarjeta TDM400P tiene 3 puertos libres, usamos el segundo puerto de la PCI para incluir el módulo FXS. Una vez que tengamos el módulo FXS configurado podemos conectar un terminal analógico (FXO) al puerto 2 de la PBX.

El proceso es simple; después de apagar la PBX, insertamos el módulo FXS en el segundo puerto de la tarjeta TDM. Después de arrancar el sistema, añadimos una línea más *fxo_ls=2* al fichero de configuración */etc/zaptel.conf*.

```
fxs1s=1
fxo1s=2
loadzone=us ; loadzone=es para españa
defaultzone=us ; defaultzone=es
```

Para asegurar que el segundo puerto (FXS) ha sido detectado tienes que usar la herramienta *ztcfg* (o incluirla en el arranque) con el siguiente resultado:

```
#ztcfg -vv
Zaptel Configuration
=====

Channel map:

Channel 01: FXS Loopstart (Default) (Slaves: 01)
Channel 02: FXO Loopstart (Default) (Slaves: 02)

2 channels configured.
```

Una vez que el sistema operativo puede hacer uso del nuevo puerto en la TDM, configuramos *Asterisk* en */etc/asterisk/zapata.conf* para que pueda hacer uso de un nuevo canal analógico (el teléfono). En el mismo fichero de configuración indicamos que las llamadas entrantes desde el teléfono analógico (puerto 2 de la tarjeta TDM) se deben asociar al contexto *[internal_calls]*

```
[channels]
usecallerid=yes
hidecallerid=no
callwaiting=no
threewaycalling=yes
transfer=yes
echocancel=yes
echotraining=yes

context=incoming_pstn
signalling=fxs_ls
channel => 1

;Añadimos un módulo FXS
```

```
context=internal_calls
signalling=fxo_ls
channel => 2
```

7.3 Actualización del plan de marcado

Necesitamos un nuevo plan de marcado con dos nuevas funcionalidades:

1. Permitir tanto llamadas entrantes como salientes por la RTB (canal zapata 1).
2. Incluir el nuevo terminal (canal zapata 2) a nuestro plan de marcado. El terminal analógico (FXO) en el Telecentro debe poder hacer y recibir llamadas.

El fichero *extensions.conf* para nuestro segundo escenario tiene el siguiente aspecto:

```
[incoming_pstn]
exten => s,1,Answer()
exten => s,2,DigitTimeout(10)
exten => s,3,ResponseTimeout(20)
exten => s,4,Background(vm-extension)
exten => i,1,Goto(incoming_pstn,s,1)
exten => t,1,Hangup()
include => internal_calls

[internal_calls]
exten => 461,1,Dial(Zap/2) ; Extensión 461 llama por el canal Zap 2
exten => 462,1,Dial(SIP/462)
exten => 463,1,Dial(SIP/463)
exten => 465,1,Dial(SIP/465)
exten => 464,1,Dial(IAX2/464)
exten => 466,1,Dial(IAX2/466)
exten => t,1,Hangup()
exten => s,1,Hangup()
exten => i,1,Hangup()
include => outgoing_calls ; Salida RTB disponible a todos los clientes

[outgoing_calls]
exten => _0.,1,Dial(Zap/1/${EXTEN:1}) ; Quita el 0 antes de marcar por RTB
exten => t,1,Hangup()
```

8. ESCENARIO C

- Conectando comunidades usando VoIP

En nuestro tercer y último escenario queremos conectar nuestro Telecentro con un centro de capacitación a distancia situado en otro país. La conexión de datos se realizará a través de un enlace de satélite tipo VSAT. Una vez que tengamos funcionando la conexión a Internet podemos usarla para realizar tanto llamadas internacionales al centro de capacitación como a otros destinos.

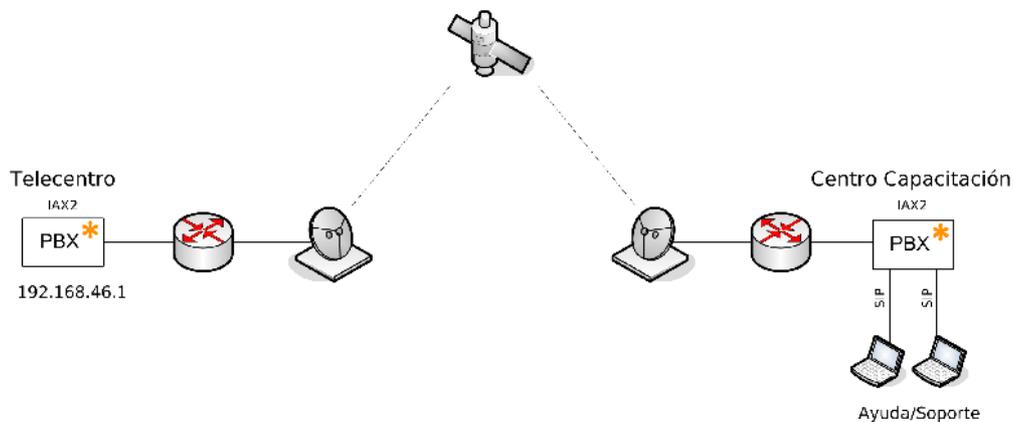


Imagen 10: Tanto el Telecentro como el Centro de Capacitación tienen su centralita Asterisk. Las centralitas están interconectadas gracias a un enlace de satélite tipo VSAT.

8.1 Problemas más comunes en enlaces vía satélite

La conexión a Internet desde el Telecentro tiene un ancho de banda muy limitado (128/64 Kbps) por lo que debemos diseñar una solución que haga un uso óptimo del ancho de banda disponible. En el siguiente ejemplo proponemos conectar dos centralitas usando *Asterisk* y el protocolo IAX2 de voz sobre IP. Además de usar IAX2 en lugar de SIP, utilizaremos un codec de alta compresión como el G.729. Para terminar, activaremos el “trunking” de llamadas que permite agregar varias llamadas simultáneas en el mismo flujo de paquetes IP.

Un escenario muy común en entornos con VSATs es la escasez de direcciones IP públicas que se ofrecen en la conexión a Internet. Si no tienes ninguna dirección pública IP disponible para tu centralita tienes que asegurarte de que el puerto de comunicaciones IAX2 es visible desde el exterior. Para hacer visible el puerto debes redirigir las conexiones al puerto UDP 4569 a la PBX.

- Si tu conexión usa un enrutador del tipo Cisco con NAT tienes que usar una configuración parecida a la siguiente.

```
#ip nat inside source static udp 192.168.46.1 4569 interface fastEthernet 0/0 4569
```

- Si tu enrutador usa Linux puedes redirigir las conexiones al puerto 4569 a tu máquina *Asterisk* (192.168.46.1) usando el programa de gestión de rutas: iptables

```
#!/sbin/iptables -t nat -A PREROUTING -p udp --dport 4569 -i eth0 -j DNAT --to-destination 192.168.46.1:4569
```

Independientemente del tipo de equipo que estés usando para encaminar tu tráfico a Internet, lo realmente importante es recordar que tu centralita tiene que ser accesible desde el exterior.⁴¹

8.2 Interconectando dos servidores con *Asterisk*

8.2.1 Telecentro

El fichero de configuración `iax.conf` en el Telecentro debería parecerse al siguiente:

```
[general]
bindaddr = 0.0.0.0
tos = lowdelay
disallow = all
allow = ulaw
allow = g729 ; Permitimos el uso del codec G.729

register => server2:server2pass@training_voip.org
; server2:server2pass es el usuario
; y clave que usamos para registrarnos

; Cuenta de usuario que usará el Centro de Capacitación para registrarse con
nosotros
[server1]
type=friend
user=server1
secret=server1pass
host=dynamic ; Aprendemos la dirección IP
; cuando se registran con nosotros

context=incoming_training_centre_calls
auth=md5 ; Añadimos seguridad en autenticación
disallow=all
allow=g729
trunk=yes ; Activamos el "trunking"
```

Para alcanzar el Centro de Capacitación desde nuestro Telecentro añadimos un nuevo contexto en el fichero `extensions.conf`. Cuando una llamada empieza por "99", redirigimos la conexión al centro de capacitación [*server1*] a través de Internet.

```
[outgoing_training_centre_calls]
exten => _99.,1,Dial(IAX2/server2:server2pass@server1/${EXTEN:2})
exten => _99.,2,Congestion ; En caso de fallo, sonido de congestionado
```

El siguiente paso es decidir lo que hacemos con las llamadas provenientes del Centro de Capacitación. En el ejemplo hacemos sonar el teléfono analógico (*Zap/2*) en el Telecentro.

41. El puerto por defecto de comunicaciones IAX2 es UDP 4569.

```
[incoming_training_centre_calls]
exten => _X.,1,Dial(Zap/2)           ; Llamadas desde el centro de capacitación
con cualquier                       ; número hacen sonar el teléfono analog.

del Telecentro
```

8.2.2 Centro de capacitación

El fichero `iax.conf` del centro de capacitación es similar:

```
[general]
bindaddr = 0.0.0.0
tos = lowdelay
disallow = all
allow = g729                       ; Usamos G.729
register => server1:server1pass@rural.telecentres.org ; Nos registramos como usuario server1

[server2]
type=friend
user=server2
secret=server2pass
host=dynamic

context=incoming_telecentres_calls
auth=md5
disallow=all
allow=g729
trunk=yes
```

Después de configurar el canal IAX2 en el Centro de Capacitación, vamos a añadir el contexto `[outgoing_telecentres_calls]` y `[incoming_telecentres_calls]` para gestionar las llamadas salientes y entrantes al/desde el Telecentro.

Creamos una regla en el plan de marcado para que las llamadas que empiecen por 88 se envíen al Telecentro (peer `[server2]` en `iax.conf`).

```
[outgoing_telecentres_calls]
exten => _88.,1,Dial(IAX2/server1:server1pass@server2/${EXTEN:2})
exten => _88.,2, Congestion
```

Las llamadas entrantes al Centro de Capacitación se redirigen a un centro de atención de usuario. En el siguiente ejemplo, las llamadas entrantes se reenvían al personal de ayuda usando un canal SIP (`SIP/support-desk`)

```
[incoming_telecentres_calls]
exten => _X.,1,Dial(SIP/support-desk) ; Llamadas entrantes se envían al centro
                                       ; de atención de usuario (Support Desk)42
```

8.2.3 La función de registro

No se pueden realizar llamadas a un “peer” hasta que se conozca su dirección IP. Imagínate la situación donde sólo una de las dos centralitas tiene la dirección IP fija. El proceso de registro permite al “peer” hacer pública su dirección IP actual. En nuestro ejemplo anterior hemos

42. La canal SIP que se ejecuta con `Dial()` hacia el `support-desk` (servicio de atención de usuario) tiene que configurarse en la centralita del Centro de Capacitación.

usado dos órdenes de registro aunque la función de registro no sea estrictamente necesaria porque los comunicantes tienen IP fijas. Para usar las direcciones fijas en lugar del proceso de registro sustituimos la opción `host=dynamic` por la opción `host=<ip_address or domain>`.

9. Para aprender más

- Uno de los mejores libros sobre *Asterisk* es: *Asterisk, The Future of Telephony*, Jim Van Meggelen, Jared Smith, Leif Madsen. O'Reilly 2005. Licencia Creative Commons. <http://www.oreilly.com/catalog/asterisk/>
Descarga libre en: <http://www.asteriskdocs.org/modules/tinycontent/index.php?id=11>
- Para no perder de vista lo que está pasando en el mundo de la telefonía IP puedes consultar: <http://www.oreillynet.com/etel/>
- El sitio “VoIP info” es una *wiki* enorme con cientos de consejos y ayudas; aunque encontrar lo que uno realmente necesita puede llevarte un poco de tiempo <http://www.voip-info.org/wiki-Asterisk+tips+and+tricks>

10. Conclusión

Esta guía es un intento de introducirte en el mundo de la telefonía IP. Esperamos que a través de algunos escenarios, haber sido capaces de hacerte consciente de las infinitas posibilidades que ofrece la telefonía IP en regiones en desarrollo. La unión de la telefonía IP con las tecnologías inalámbricas de bajo costo permite ofrecer servicios de voz y datos a regiones excluidas. Estas tecnologías promueven la creación de nuevas redes comunitarias, redes operadas y mantenidas por las comunidades.

Los ficheros incluidos como ejemplos pretenden servir de guía para ayudarte a poner en marcha tu primer sistema de telefonía. Ningún documento puede compararse con la experiencia personal; ¡ten paciencia!, tu perseverancia es la clave para tu aprendizaje. Y recuerda que no estás solo – siempre puedes pedir ayuda en los foros de discusión y compartir tu experiencia con otros. ¡Bienvenido/a a la una comunidad imparables de entusiastas de la telefonía IP!

¡Esperamos tu llamada!

11. Agradecimientos

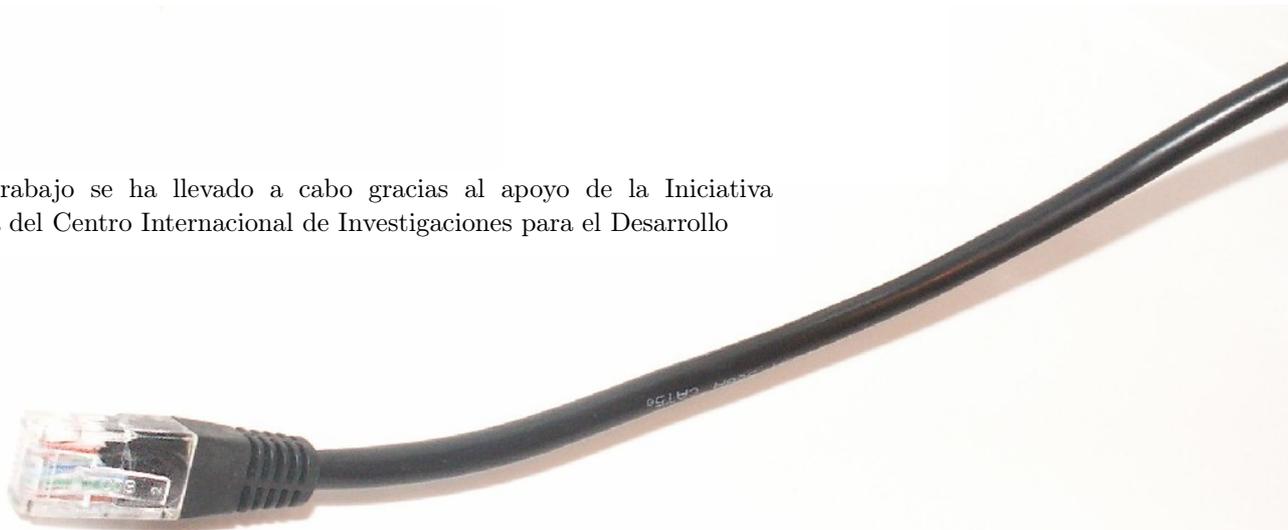
Nos gustaría agradecer a todos los que han hecho este trabajo posible:

Adel El Zaim (editor de árabe y francés), Anas Tawileh (traductor al árabe), Iñaki Cívico y Sylvia Cadena (editores al castellano), Johan Bilien (traductor al francés) y Martín Benjamin (editor al inglés).



Todo el material de este trabajo está licenciado bajo:
Creative Commons Attribution NonCommercial-ShareAlike 2.5
<http://creativecommons.org/licenses/by-nc-sa/2.5/>

Este trabajo se ha llevado a cabo gracias al apoyo de la Iniciativa
Acacia del Centro Internacional de Investigaciones para el Desarrollo



El Centro Internacional de Investigaciones para el Desarrollo de Canadá (IDRC) es una de las instituciones líderes en la generación y aplicación de nuevos conocimientos para resolver los retos de la desarrollo internacional. El IDRC cuenta con más de 30 años de colaboración con investigadores en regiones en desarrollo en su trabajo por crear sociedades más prósperas y equitativas.