

## VOZ SOBRE IP

Jorge Del Carpio Salinas, Percy Yaranga Quispe, Ángel Cucho Human,  
Instituto de Investigación de la Facultad de Ingeniería Eléctrica y Electrónica  
E-mails: jdelcarpio@uni.edu.pe, percyuni@yahoo.com, angecu77@hotmail.com

### .RESUMEN

*La posibilidad de transmitir voz por redes basadas en el protocolo IP, con todas las dificultades que plantea y las oportunidades que ofrece, tales como la integración de voz y datos, es una etapa fundamental de la convergencia del sector de las telecomunicaciones y también refleja la convergencia entre dos tecnologías de redes que han surgido en condiciones de política y reglamentarias muy diferentes. El objetivo de este proyecto es básicamente probar con un software de comunicación, cómo es que redes basadas en IP responden adecuadamente cuando a través de ella se manda voz. La aplicación es implementada sobre el sistema operativo Linux, debido a que Linux es totalmente compatible y de muy buen rendimiento. Además Linux es de libre distribución.*

### ABSTRACT

*The main goal of this project is to demonstrate by means of a communication software that IP-based networks can be used to transmit speech signals (voice) efficiently. For validation purposes, an application is implemented using the Linux operating system. We have used Linux because it makes possible compatibility and shows good performance. Linux is also a shareware software.*

### INTRODUCCIÓN

#### Protocolo VoIP

Está constituido por dos tipos de protocolos:

#### Protocolos que soportan transporte de ruta de paquetes de audio.

- Ofrecen información de temporización para asegurar una reproducción de audio consistente.
- Ofrece retroalimentación de calidad de servicio (QoS) con respecto a la red adyacente.

#### Protocolos que soportan la señalización de llamadas y las funciones de control.

- Proporcionan la configuración y cancelación de la llamada, direccionamiento y enrutamiento, servicios de información adicionales y métodos para trabajar otro tipo de señalización.

Protocolos de ruta de audio VoIP universalmente aceptados.

RTP.- Protocolo de transporte rápido.

#### Descripción del Estándar H. 323

“La red LAN sobre la cual los terminales H. 323 se comunican puede ser un solo segmento o un anillo, o pueden ser múltiples segmentos con topología compleja. Se hace notar que la operación de los terminales H.323 sobre segmentos múltiples de una red LAN (incluyendo Internet) puede resultar con una pobre performance. Los posibles recursos mediante los cuales la calidad del servicio puede ser garantizada en estos tipos de redes LAN y redes Internet, están más allá del alcance de esta recomendación».

#### Importancia del H. 323

Esto puede apreciarse sobre la base de los siguientes aspectos:

- Cooperación entre la industria de las Telecomunicaciones y la industria de la Computación.
- Las innovaciones se realizan más rápidamente.
- Potencial del Mercado.

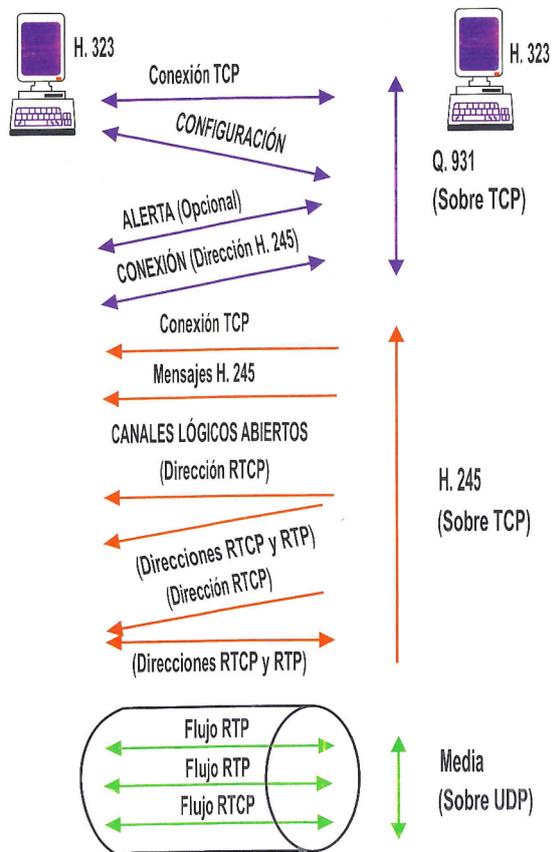


Fig 1. Secuencia de llamada con H.323.

## SEÑALIZACIÓN SIP

El protocolo de iniciación de sesión (SIP, Session Initiation Protocol), es un estándar de la IETF.

Como se explica en extracto de la SIP RFC: “El Protocolo de iniciación de sesión (SIP) es un protocolo de (señalización) de control de aplicaciones de capas para crear, modificar y cerrar sesiones con uno o más participantes. Estas sesiones incluyen conferencias multimedia, llamadas por teléfono y distribución multimedia por Internet, etc.

Algunos de los protocolos más destacados (excluyendo los esfuerzos relacionados con QoS) de esta arquitectura son:

- RTP y RTCP.

- El protocolo de flujo de tiempo real (RTSP, Real-Time Streaming Protocol), proporciona un formato de descripción estándar para el intercambio de capacidad de los medios (como los codecs de voz sobre VoIP).

**Atributos del SIP.**- SIP se diseñó con una solución a largo plazo para las conferencias multimedia y la telefonía en Internet. Se han tenido en cuenta muchas consideraciones en el desarrollo del protocolo para asegurarse que es una plataforma viable para futuras comunicaciones que se basan en Internet:

- Simplicidad
- Eficiencia
- Escalabilidad
- Flexibilidad
- Soporte de movilidad
- Programa de usuario
- Extensión

## SISTEMA OPERATIVO LINUX

El sistema operativo Linux ha pasado a ser una alternativa viable para cualquiera que tenga una PC. Este sistema aporta todo el poder y la flexibilidad de una estación de trabajo Unix, así como un juego completo de aplicaciones para Internet y una interfaz de escritorio completamente funcional, además se puede utilizarlo con cualquier intranet local, estableciendo una localización ftp o web para su red.

A diferencia del sistema Unix, Linux se distribuye de forma gratuita bajo una licencia pública general de GNU, tal como lo especifica la Free Software Foundation, poniéndolo a disposición de cualquiera que lo desee utilizar. El hecho de que Linux sea gratis da a la gente, a veces, la impresión equivocada de que algún modo es inferior a un sistema operativo profesional y, sin embargo, Linux es, de hecho, una versión para PC Unix. Para apreciar realmente a Linux en todo lo que vale, puede ser necesario conocer el contexto especial en el que fue desarrollado el Unix. Este, a diferencia de la mayoría de los demás sistemas operativos, fue desarrollado en un entorno académico y de investigación.

Con el fin de diferenciarse de otras versiones de Linux, cada distribución añade algunas utilidades adicionales. Dado que la mayoría de las utilidades avanzadas ya están incorporadas, la mayoría de mejoras de una

distribución concreta tienen la finalidad facilitar la instalación, la configuración y el uso de Linux. Red Hat se ha desmarcado de otras distribuciones de Linux en los siguientes aspectos:

- Empaquetado del software
- Facilidad de instalación
- Estilos de script de nivel de ejecución UNIX System V
- Herramientas de administración.

### Estructura del Sistema

Linux es el kernel (núcleo) de un sistema operativo compartido. El núcleo del sistema es un programa que siempre está residente en memoria y, entre otros, brinda los siguientes servicios:

- Controla los recursos del hardware.
- Controla los dispositivos periféricos (discos, terminales, impresoras, etc.)
- Permite a un sistema de archivos que administra el almacenamiento de información (programas, datos, documentos, etc.)

En un sentido más amplio, Linux abarca también un conjunto de programas estándar, como pueden ser:

- Compilador de lenguaje C (gcc)
- Editor de texto (vi)
- Interprete de órdenes (sh, ksh, csh)
- Programas de gestión de archivos y directorios (cp, rm, mkdir, etc).

Este kernel está escrito en lenguaje C en su mayor parte, aunque coexistiendo con código en ensamblador.

En el tercer nivel se encuentran programas estándares de cualquier sistema UNIX (vi, who, etc) y programas generados por el usuario (a.out programa ejecutable estándar creado por defecto por el compilador gcc, o cualquier otro programa ejecutable).

Por encima del tercer nivel tenemos aplicaciones que se sirven de otros programas ya creados para llevar a cabo su función. Estas aplicaciones no se comunican directamente con el kernel. Un ejemplo de estos programas es el compilador de lenguaje C (gcc) Cuando invocamos el compilador mediante una línea

como la siguiente:

```
$gcc archivo_entrada.c -o archivo_salida gcc va a
invocar de forma secuencial los programas:
```

- Preprocesador de C
- Compilador de C
- Ensamblador
- Linker

En efecto, Linux es un sistema multiusuario y multitarea. Como multitarea, se puede pedir al sistema que realice varias tareas al mismo tiempo. Mientras se ejecuta una tarea, se puede trabajar con otra. Por ejemplo, se puede editar un archivo mientras se imprime otro. No es necesario esperar a que se termine de imprimir éste antes de poder editar aquel. Como sistema multiusuario, admite el que varios usuarios trabajen con él simultáneamente, cada uno interactuando con el sistema por medio de su propio terminal.

Además el shell proporciona una interfaz entre el núcleo y el usuario. Se puede describir éste como un intérprete: interpreta las órdenes que introduce el usuario y las envía al núcleo. El núcleo, el shell y el sistema de archivos de archivos forman en conjunto la estructura básica del sistema operativo. Con estos tres elementos puede ejecutar programas, gestionar archivos e interactuar con el sistema.

### Arquitectura del Sistema Operativo Linux

Los dos conceptos centrales sobre los que se basa la arquitectura de Linux son los ficheros y los procesos. El kernel está pensado para facilitarnos servicios relacionados con el sistema de ficheros y con el control de procesos. Los tres niveles que vamos a estudiar en la arquitectura del sistema: hardware, kernel y usuario. Las llamadas al sistema y su librería asociada representan la frontera entre los programas del usuario y el kernel. La librería asociada a las system calls es el mecanismo mediante el cual podemos invocar una system call desde un programa C. Esta librería se enlaza por defecto al compilar cualquier programa C. Los programas escritos en lenguaje ensamblador pueden invocar directamente las system calls sin necesidad de ninguna librería intermedia.

Las llamadas al sistema se ejecutan en modo kernel

(para muchos microprocesadores, modo supervisor) y para entrar en este modo hay que ejecutar una sentencia en código máquina conocida como trap (o interrupción software).

Es por esto que las system calls pueden ser invocadas directamente desde el ensamblador y no desde C.

Por último, el módulo de control del hardware es la parte del kernel encargada del manejo de las interrupciones y de la comunicación con la máquina. Los dispositivos pueden interrumpir a la CPU mientras está ejecutando un proceso. Si esto ocurre, el kernel debe reanudar la ejecución del proceso después de atender a la interrupción. Las interrupciones no son atendidas por procesos, sino por funciones especiales, codificadas en el kernel, que son invocadas durante la ejecución de cualquier proceso.

En Linux, una orden se considera como un proceso, una tarea que debe ser realizada. Un sistema Linux puede ejecutar varios procesos al mismo tiempo, al igual que manejar varios usuarios a la vez. Hay órdenes para examinar y controlar procesos, aunque normalmente se reserven para operaciones de administración del sistema. En realidad, los procesos incluyen no sólo las ordenes que ejecutan el usuario, sino también todas las tareas que el sistema debe efectuar por si mismo para seguir funcionando.

### Hardware en Linux

Esto es quizá lo más importante de todo. Todas las grandes virtudes y ventajas de Linux no servirán de mucho si la PC en donde se instalará no posee el hardware adecuado y en buen estado. Si el hardware y/o periféricos tienen problemas en MS-DOS/Windows, o alguna de las funciones no trabaja adecuadamente, será muy difícil que lo haga en Linux. Es conveniente revisar varios puntos, además de la documentación que acompaña a cada versión.

Antes de proceder a instalar Linux, se debe determinar que hardware se tiene instalado, principalmente los intervalos de entrada / salida (IO o E/S), intervalos de interrupción (IRQ) y acceso directo a memoria (DMA) de cada componente de su equipo, sobretodo de la tarjeta de sonido y periféricos.

Es indispensable determinar al menos el modelo exacto de la tarjeta de video y, si existiese, el modelo

de la tarjeta SCSI.

Se puede fácilmente determinar la configuración del hardware de la PC si se examina la solapa de Administrador de Dispositivos» en «Propiedades del Sistema» en Windows.

Después de esto, solo hará falta diagnosticar si el hardware esta trabajando adecuadamente. Es muy común que algunas tarjetas madre «reconstruidas» o mal manufacturadas, y que tienen integrados tarjeta de audio, video, modem y/o red, sufran de desperfectos en algunos de los componentes. Ejemplo: algunos modelos de tarjetas madre fabricadas por empresas «desconocidas» en Taiwán o China poseen defectos de fabricación que involucran a la tarjeta de audio «on-board».

## TRATAMIENTO DE VOZ CON LINUX

### Uso de la tarjeta de sonido

El proceso de adquisición de la voz en su forma natural (es decir analógica) es realizado en la etapa del cliente el cual actúa como el agente usuario llamante. La utilización de la plataforma Linux permite que la obtención de las muestras sea realizada a través de la tarjeta de sonido que el cliente posee.

En lo que respecta a la solución en hardware proporciona las ventajas que no ofrecen las soluciones de software, es decir, velocidad y precisión en el procesamiento de datos al poseer un componente especializado en esta aplicación. Pero implica una mayor complicación en la implementación, debido a que la programación de un dispositivo de hardware es orientada a bajo nivel.

Un ejemplo de este tipo de implementación sería el montaje de tarjeta de Procesamiento Digital de señales (DSP) que con sólo el hecho de mejorar la interfaz PC-DSP (la comunicación es totalmente asíncrona) se podría mejorar la transmisión de voz sobre IP, también obtener audio de más alta calidad y hasta aplicaciones de seguridad en comunicaciones telefónicas.

Desde luego el proyecto en primer término se maneja en la implementación en el ámbito de software; dejando por ello a la tarjeta de sonido la obtención de muestras (etapa del cliente) y posterior recuperación

en forma analógica (etapa del servidor). De esta manera la información será adquirida mediante un micrófono.

Es bueno recalcar que las tarjetas de sonidos incorporan procesadores especializados o coprocesadores que se conocen como DSPs. Una memoria ROM el cual incorpora las principales instrucciones de funcionamiento del DSP, mientras que una pequeña cantidad de memoria RAM le facilita el almacenamiento en tiempo de ejecución.

Las prestaciones del DSP no son tan importantes respecto de la calidad del sonido que almacena y reproduce una tarjeta, como los conversores DAC Y ADC que también incorpora. Son estos componentes de la tarjeta de audio los que se encargan de convertir el sonido de un formato a otro y, por lo tanto, de determinar capacidad y calidad de los datos en ambos formatos.

### Configuración de la Tarjeta de Sonido para el Transporte de Voz

Puesto que el sistema operativo sobre el cual se maneja este hardware es Linux (Red Hat) es necesario saber que los dispositivos están en el directorio de Linux: /dev y específicamente en /dev/mixer, /dev/dsp, /dev/audio, y /dev/midi. Esto es importante conocer puesto que para acceder a las funcionalidades de la tarjeta de sonido y la posterior programación de ella, las distribuciones de Linux proporcionan unos controladores que son sencillos de utilizar. De esta forma, en Linux existen unos dispositivos virtuales que aíslan las características del hardware del dispositivo real, de entrada y salida. Es por eso que se programarán funciones que permitan la adquisición de sonido y la reproducción de sonido digital en Linux, usando el driver de la tarjeta de sonido. Los parámetros más importantes de la tarjeta de sonido son:

- a. El tamaño del DMA
- b. El número de canales PCM a utilizar
- c. El número de bits y
- d. La frecuencia de muestreo

La tarjeta de audio, quizá es algo de entre lo más difícil de configurar, es importante, en lo que al proyecto se refiere. Cabe aclarar que las tarjetas de audio o video baratas y de calidad regular, como la

tarjeta de audio CMI8330 suelen no ser totalmente funcionales o suelen presentar defectos de manufactura que se evidencian claramente en Linux. Una herramienta llamada sndconfig (comando del linux) puede ayudar a configurar la tarjeta de sonido .para usar sndconfig, tiene que entrar como root.

Una vez como root, abrir una ventana Xterm y teclear sndconfig en el prompt. Esta herramienta sirve para configurar la tarjeta de sonido para el sistema Red Hat Linux. Para navegar a través de los botones Yes, No, Cancel usando el tabulador, Tab y la tecla Enter. Aunque muchas tarjetas de sonido son soportadas, no todas son compatibles -o compatibles totalmente, encontrará una lista en <http://www.redhat.com/hardware/>. Si la utilidad detecta una tarjeta sonora plug and play, procederá automáticamente a configurar la tarjeta dando ejemplos sonoros.

### SOCKET DE COMUNICACIÓN

Se define socket como la combinación de la dirección IP del cliente o servidor y del número de puerto del cliente o servidor respectivamente. Puesto que la comunicación entre el cliente y el servidor se efectuara sobre la arquitectura de una Intranet, sobre la cual se efectuará una transferencia de datos utilizando protocolo TCP/IP; la comunicación se efectúa utilizando Sockets TCP en lugar de Sockets UDP con el fin de ganar control de sesión y manejo de errores (anteriormente) pero, actualmente UDP se utiliza en Voz sobre IP para transportar el trafico de voz real. El protocolo TCP no es práctico por que no se necesitan ni el control del flujo ni la retransmisión de paquetes de audio de voz, con TCP para Voz sobre IP, la latencia en que caería a la espera de los acuses de recibo de retransmisión haría que la calidad de voz fuera inaceptable.

El sistema de control es Linux. De este modo el servidor abrirá un Socket disponible y quedará a la espera del establecimiento de alguna comunicación con el cliente. Durante este tiempo el servidor debe esperar indefinidamente, manejando el Socket mediante un demonio de red. Cuando el cliente desea transmitir datos establece una petición de conexión con el servidor, es decir, intenta realizar una conexión en el Socket del servidor. El canal de comunicación es bidireccional, debido a que aparte del flujo de información del cliente hacia el servidor también existe flujo de información del servidor hacia el cliente, en

la que se incluye la señalización y acuse de recibo.

## APLICACIONES SOBRE UNA INTRANET

Crear una LAN y configurarla para que sea realmente útil en la prueba del software de comunicación está compuesto de tres partes:

1. Configurar el hardware: Esto entraña elegir una topología de red, adquirir el equipo necesario e instalarlo (añadir tarjetas y conectar cables).
2. Configurar Ethernet: Red Hat Linux debe ser capaz de reconocer la tarjeta Ethernet del ordenador, instalar un controlador para ello y habilitar su uso bajo Linux. En lo que respecta a las tarjetas con soporte, esto se realiza fácilmente durante la instalación de Red Hat Linux).
3. Configurar TCP/IP: para usar la mayoría de las aplicaciones de trabajo de red y las herramientas que vienen con Red Hat Linux, se debe configurar TCP/IP. Este protocolo no sólo permite comunicar con los ordenadores de la LAN, sino también con cualquier ordenador con el que se pueda contactar en esta LAN, modem u otra conexión de red.

Utilizando el comando `ifconfig`, se puede determinar si la tarjeta ethernet está lista y en funcionamiento. Para verificar si ésta se puede ver o acceder a otra maquinas de la Red Lan.

La forma de usar este comando es:

```
Ifconfig interfaz [direccion [parametros]]
```

*interfaz* es el nombre de la interfaz y *direccion* es la dirección IP que se asigna a dicha interfase. La dirección puede estar en forma de cuaterna o usando un nombre que `ifconfig` buscará en `/etc/hosts`.

Si `ifconfig` es ejecutado añadiendo únicamente el nombre de la interfaz, presentará la información de la configuración de dicha interfaz. Si se ejecuta sin parámetros, presenta todas las interfaces configuradas hasta el momento; usando la opción `-a` fuerza a `ifconfig` a incluir la información de las interfaces inactivas. A modo de ejemplo, la consulta de la configuración de la interface Ethernet `eth0` sería:

```
# ifconfig eth0
eth0 Link encap 10Mbps      Ethernet
```

```
HWaddr 00:00:C0:90:B3:42
      inet addr 172.16.1.2 Bcast
172.16.1.255 Mask    255.255.255.0
      UP BROADCAST    RUNNING
MTU 1500 Metric 0
      RX packets 3136 errors 217 dropped
7 overrun 26
      TX packets 1752 errors 25 dropped
0 overrun 0
```

Los campos MTU y Metric informan sobre los valores actuales de la MTU (Unidad Máxima de Transferencia) y de la métrica para una interface dada. El valor de la métrica es usado tradicionalmente por algunos sistemas operativos para calcular el coste de una ruta. Linux no usa este valor por el momento, pero lo define por razones de compatibilidad.

## GNOPHONE (LINUX)

Gnophone, es una fuente libre de teléfono internet que permite llamar a otro usuario de gnophone usando la red de internet. Este código fuente esta escrito en C y hay que adaptarlo al hardware que se tenga y luego realizar la compilación, en caso de no adaptarse al sistema se debería bajar los parámetros establecidos. Este software usa codificación GSM en el cual la voz generada por la persona, es filtrada y luego muestreada a razón de 8kHz (8000 muestras / seg.) cada muestra se codifica en 8 bits, lo que nos da una tasa de bits de 64 kbps. Estos 64kbps se dividen en grupos de 20ms, los cuales pasan después por un proceso de compresión. Este proceso de compresión convierte la señal de audio PCM en 13kbps.

Debido a que los canales de audio pueden distorsionar la información, se agregan bits de protección de error. Estos bits incrementan la cantidad de data a 22.8kbps, para ayudar a proteger la data cuando la señal es pobre.

Uno de las grandes preocupaciones en un sistema de transmisión digital de voz es la performance del codificador de voz. Si el codificador de voz no codifica y decodifica exactamente los datos de voz, súmese a ello el ruido o el error. La performance del codificador de voz GSM se ejecuta bien en condiciones de radio pobre.

Generalmente los códigos fuente en Linux vienen en

extensión tar.gz el cual es un formato de comprensión en Linux. Para descomprimirlo se usa el comando tar.

```
Shell>$tar zxvf nombre_archivo.tar.gz
```

En este caso el nombre del archivo sería

```
gnophone.tar.gz.
```

Luego de descomprimirlo se ubica en el directorio donde esta el archivo y teclear .

```
Shell>./configure
```

```
Shell>$make
```

```
Shell>$make install
```

Luego para usar el programa hay que correr el demonio (el cual es un programa) que hace que este software pueda correr como aplicación.

Make es una herramienta utilizada para controlar los procesos de construcción y revisión de software, make automatiza el tipo de software que se crea y cuando se lo desarrolla, permitiendo al programador concentrarse exclusivamente en la redacción del código fuente. Además permite ahorrar muchísimo tipeo, ya que contiene decisiones lógicas integradas que invocan al compilador de C de Linux con las opciones y argumentos que sean adecuados.

En primer lugar, los proyectos compuestos por múltiples archivos de código fuente requieren invocaciones del compilador que son largas y complejas .make simplifica esto mediante la invocación de estas líneas de comando en un makefile. Un makefile es un base de datos en formato texto que contiene reglas que le indican a make que construir y como hacerlo.

Cada regla consiste en lo siguiente:

Un target, es lo que make trata de crear (target es habitualmente el archivo binario o el archivo objeto).

- \* Una lista de una o más dependencias, generalmente archivos requeridos para construir el target.
- \* Una lista de comandos a ser ejecutados con el fin de crear el target a partir de las dependencias entre archivos especificados.

## INTRANET

En el ámbito de la intranet se ha trabajado en el dominio de UNI (uni.edu.pe), en este nivel se pudo comprobar que la transmisión de voz por ser una red interna es aceptable, no así la comunicación fuera de este dominio, donde existe retardo y mucha distorsión; esta prueba se realizó con una PC de la intranet de la UNI y otro exterior a ella (con IPs válidos). Los servidores de la UNI así como el ancho de banda con que cuenta, es pobre para una transmisión de paquetes de voz. Cuando se utiliza servidores proxies, no se puede establecer la comunicación debido a las limitaciones de los softwares utilizados.

Aspectos adicionales.- Debido a que el transporte de datos y sobre todo voz es muy “pesado” el hardware requerido para esto tiene que ser óptimo y de alta capacidad de procesamiento (Pentium II ó III). En lo posible, deben ser originales, que son las que brindan mayor posibilidad de que el sistema operativo (Linux) posea sus drivers (controladores) para que pueda funcionar óptimamente; de no ser así, una posibilidad sería compilar el núcleo de Linux o crear un programa que lo pueda soportar en lenguaje C.

## OBSERVACIONES

En el proceso de transmisión de voz, es necesario distinguir diferentes tipos de aplicaciones para la implementación de una red que pueda soportar este tipo de comunicación, por ejemplo:

- Voz sobre IP, transmisión vocal por el Protocolo Internet. Término genérico para definir la tecnología.
- Telefonía IP, prestación de servicio telefónico por una red conmutada de paquetes IP que emula o sustituye a la RTPC (Red Telefónica Pública Conmutada)
- Telefonía Internet, prestación de servicio telefónico usando parcial o totalmente la red pública Internet.

En un principio nos dimos cuenta que el primer problema a afrontar era que el programa de comunicación a utilizar tendría que ser soportado por el hardware con que se cuenta.

En especial la tarjeta de sonido tendría que contar con módulos dentro del Kernel de Linux, para que pueda ser configurada con todos sus parámetros;

llámese sintetizador de voz, mezclador, etc. Se llegó a disponer de una Red Lan y hacer pruebas a través de varios softwares de comunicación en Windows y en Linux, habiéndose comprobado el retardo, la calidad de señal que se presentan en este tipo de comunicaciones. En este tipo de comunicación cliente a cliente la validación en servidor apropiado es importante, así como poseer IPs válidos.

### CONCLUSIONES

En lo que se respecta a la calidad de voz transmitida en la comunicación, todavía no ha n podido ser manejados los problemas de retardo, eco, entre otras cosas, pero podemos decir que la comunicación es aceptable.

Lo que sucede es que la red establecida actualmente tiene prioridad más en la transmisión de datos, que la de voz; por ejemplo, los routers que existen para el encaminamiento de datagramas dejan en cola a paquetes de voz y dan prioridad a los de datos.

En lo que respecta a los softwares compilados,

generalmente éstos se manejan con el criterio de cliente a cliente (IP válidos), es decir, depende de los softwares que dan permiso para la transmisión, dependiendo de la implementación del tipo de red con que se cuenta (p. e. los servidores proxies en algunos casos no permiten la comunicación).

### REFERENCIAS

1. **Lan McLean**, TCP/IP, "La Biblia de TCP/IP", Editorial: Anaya, Multimedia, Ohio 2000.
2. **Jonathan Davidson y James Peters**, Cisco Systems. VoIP, "Fundamentos de Voz sobre IP", Princeton 2000.
3. Linux, "Manual de Referencia" **Richard Peterson**. Editorial: Osborne/ McGraw Hill, Ohio 2001.
4. **Robert L. Ziegler**, Linux, "Firewalls Linux" Editorial: Prentice may, San José Usa 1998
6. **José Luis Raya**, "Redes Locales y TCP/IP, Computec / Rama", Panama 1996.
7. **Scout Keagy**, "Integración de redes de voz y datos", Cisco Systems, Ciscopress.com Angel, San José Usa 2001.