# TRABAJO DE GRADO

# Análisis de una aplicación de Voz sobre IP versión 6

Alumno: Carlos Espósito
Director: Ing. Luis Marrone

UNLP
Facultad de Informática
2004

TES
04/13
DIF-02939
SALA

UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMATICA
Biblioteca
50 y 120 La Plata
catalogo.info.unlp.edu.ar
biblioteca@info.unlp.edu.ar

DIF-02939

DONACION Facultad
$
Fecha 16-10-07
Inv. E. Inv. B. 002939
TES
04/13

*A mis padres*

# Agradecimientos

Quisiera agradecer a todos quienes durante tantos años me guiaron como alumno, primero de la Facultad de Ciencias Exactas y ahora de Informática, contando a profesores, ayudantes y personal no docente, con quienes en muchos casos he construído una relación de simpatía mutua. A mi hermana María, hoy docente de esta Universidad, que estuvo a cargo de la corrección ortográfica. Pero muy especialmente quiero agradecer al Ing. Luis Marrone, quien aportó, además de su conocimiento, un apoyo y un empuje fundamental para la concreción de este trabajo sin el cuál el mismo no hubiera sido posible.

# Indice

# Listado de figuras

# Prefacio

Este trabajo de grado tiene como finalidad analizar una aplicación de tiempo real que corra sobre el protocolo IPv6, para descubrir cómo utiliza las modificaciones y ventajas que este protocolo propone respecto de su antecesor. Se eligió para el análisis una aplicación de voz sobre IP, en busca de evaluar cómo se desempeña IPv6 respecto de calidad de servicio, movilidad y demás facilidades que se desarrollarán a lo largo del trabajo.

En el Capítulo 1 se describe IPv6, poniendo el énfasis en las características salientes que se analizarán luego. El Capítulo 2 es una reseña de la tecnología de voz sobre IP donde se explica cuál es la problemática más importante a tener en cuenta. En el Capítulo 3 se describe el análisis hecho sobre la aplicación, las herramientas usadas y las primeras conclusiones.

Para la lectura de este trabajo se dá por supuesto que el lector conoce qué es una red de computadoras que usan protocolo TCP/IP en versión 4, y domina las aplicaciones comunes implementadas sobre el mismo. Además se supone que conoce las distintas topologías de red y los tipos de red más usuales.

Se han usado pocas traducciones de los términos en inglés, debido a que en general son vocablos ampliamente usados y cuya traducción literal hace más dificil en vez de simplificar la comprensión de la cosa o acción mencionada. En caso de haber traducciones, se menciona el vocablo original entre paréntesis.

Como norma, la primera referencia a una sigla estará acompañada de la frase completa entre paréntesis, con la sóla excepción de las siglas muy conocida y aceptadas. Las siguientes referencias se hacen sólo a la sigla.

Durante la elaboración de este trabajo la definición de la estructura de la dirección IPv6 mostrada en la figura 1-5 fue modificada por la IETF. Este cambio no está reflejado en el texto, aunque sí se hace una mención al final.

# Capítulo 1

# INTRODUCCION A IPV6

El crecimiento y la evolución de la red Internet han requerido el desarrollo de nuevos estándares de red para adecuar la tecnología a las necesidades actuales y prever futuros desarrollos. El protocolo IP versión 6 –también conocido como Ipng, por *IP next generation*- es el estándar aceptado para el reemplazo del actual protocolo versión 4.

La primera motivación que surgió para reemplazar la versión 4 fue el agotamiento del espacio de direcciones. Aunque los 32 bits que poseen las direcciones actuales parecían suficientes, la asignación hecha, sumada al enorme crecimiento de la red en los últimos años, están en vías de acabar con ellas. Por otra parte, las aplicaciones que vienen, requerirán direcciones IP para cada dispositivo real o virtual de forma de poder comunicarse con él. No es arriesgado pensar que una sola computadora vaya a tener varias –muchas- direcciones asignadas en la pantalla, la memoria, los discos, además de las placas de red o las impresoras.

Pero el espacio de direcciones no fue la única motivación de la aparición de IPv6. Aunque la versión 4 en sus múltiples agregados incorpora funciones como seguridad, calidad de servicio o movilidad –por mencionar algunas- se torna complicado trabajar con ellas cuando se desea usar varias de estas funciones simultáneamente. Se hizo por lo tanto necesario que el nuevo protocolo incorporara estas funciones, ya que cada vez más aplicaciones requieren de ellas para operar. Una de las funciones que más importa a este trabajo es la **calidad de servicio (QoS)**, ya que todas las aplicaciones de transmisión en tiempo real (como la voz o el video) necesitan que se les garantice a sus datos llegar cuanto antes, sin importar la fiabilidad de la transmisión efectuada. Al contrario de aplicaciones de transmisión de bytes –como mail o ftp- éstas pueden soportar errores en la entrega de los datos.

IPv6 continúa siendo un protocolo de entrega de datagramas sin conexión o no orientado a conexión. Las capas de red superiores en el modelo OSI de interconexión de sistemas abiertos -comúnmente llamado simplemente modelo OSI- deberán encargarse de garantizar que todos los paquetes lleguen a destino y sin errores. Precisamente, el hecho que sea un protocolo sin conexión lo obliga a implementar alguna forma de entrega confiable, a través de la calidad de servicio. En otras palabras, se intenta "conectar" un protocolo sin conexión.

Como es de esperar, la nueva versión incorpora mecanismos para comunicarse con la versión 4. Todas las combinaciones entre redes son posibles (v4-v6-v4, v6-v4, v6-v4-v6, etc.) y el pasaje entre una versión y otra es totalmente transparente para el usuario. Sólo requiere algún trabajo en los servidores y routers que comunican una red dada con el resto de Internet. Los hosts individuales buscarán su servidor de configuración para obtener automáticamente

dirección IP, DNS, máscara, etc. Esta característica hace sumamente fácil la configuración, en contraste con su antecesor, que requiere que se ingresen los datos en forma manual muchas veces, aumentando el riesgo de errores en el tipeo u omisiones de algún dato.

El protocolo IPv6 fue definido por la IETF (Internet Engineering Task Force) en la RFC 2460 (Deering y Hinden, Dic. 1998) en conjunto con una serie de RFCs que la acompañan. Entre las varias instituciones que aportan al desarrollo de IPv6, una de las más importantes es el Foro IPv6, al cual adhieren gran cantidad de instituciones internacionales y grandes corporaciones de la industria, cuyo fin es el de promover el uso y la aplicación de IPv6, hasta llegar al reemplazo total de la actual versión 4.

## 1.1 Características principales

Las características principales de IPv6 (Palet Martínez, *Tutorial de IPv6*, pág.5) son:

- Direcciones: asigna 128 bits a la dirección, o sea 4 veces el tamaño actual.
- Calidad de servicio
- Seguridad
- Autoconfiguración: IPv6 se dice que es "Plug & Play"
- Facilidad de ruteo: El gran espacio de direcciones permite rutear más eficientemente que el IPv4.
- Renumeración: Se puede cambiar de proveedor de enlace y hacer que todo se renumere automáticamente.
- Movilidad: Un dispositivo con una IP determinada puede "moverse" de red. Las redes inalámbricas requieren esta función.
- Fragmentación más eficiente: el reensamblado de fragmentos se hace en el extremo final de la ruta y no en la punta de cada enlace.
- Paquetes grandes: aumenta el tamaño máximo del paquete, lo que permite hacer más eficientes redes de MTU grandes.

## 1.2 El header IPv6

El header IPv6 tiene dos características fundamentales que lo distinguen de su antecesor. Como ya se mencionó, las direcciones fuente y destino tienen una longitud de 128 bits lo que provoca que se incremente el tamaño del encabezado. Este incremento se traduce en un mayor ancho de banda para la transmisión del encabezado con respecto a su antecesor. Como contrapartida, IPv6 elimina en primer lugar, algunos campos del encabezado y en segundo lugar, elimina las opciones, que pasan a tener encabezados propios. Esta particularidad hace mucho más flexible el protocolo, ya que se pueden diseñar las opciones de encabezado que se deseen. Hay seis opciones distintas ya definidas, y está abierta la posibilidad de que se incorporen nuevas extensiones.

Si bien el encabezado IPv6 sin opciones mide 40 bytes –contra 20 de su antecesor- tiene una alineación de 64 bits, lo que significa que se procese más velozmente en los

procesadores que soportan ese tamaño de palabra. Además, al sacarse las opciones y ponerlas en encabezados separados, se logra tener una longitud fija de encabezado que facilita su procesamiento por routers intermedios entre un host y otro.

Los campos del encabezado IPv6 se muestran en la siguiente figura:

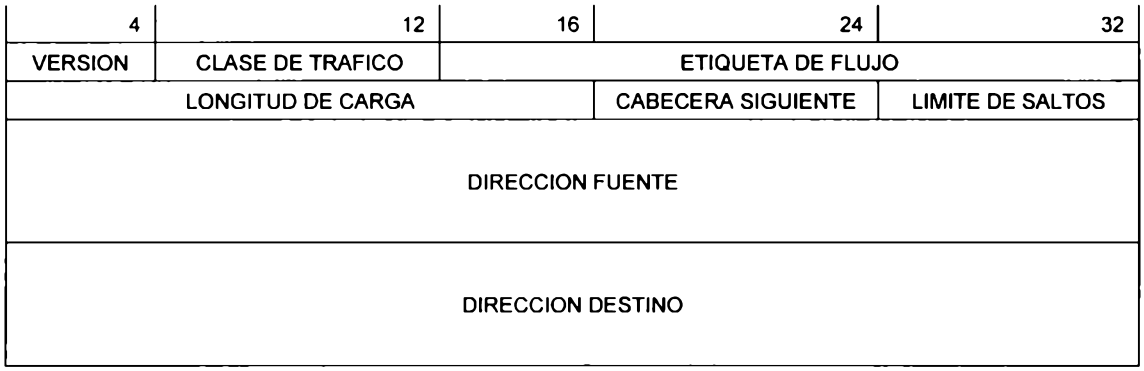



Figura 1-1: El formato del encabezado IPv6

Como puede verse, el encabezado consta de 40 bytes, 8 para la primera parte y 16 para las direcciones fuente y destino.

Versión: Allí siempre irá un 6, indicando la versión del protocolo.

Clase de tráfico: La clase de tráfico definirá la prioridad y el tratamiento que se dará a los paquetes. Junto con la etiqueta de flujo se podrá diferenciar la calidad de servicio requerida para el paquete.

Etiqueta: Se usa para identificar conjuntos de paquetes. Todos los paquetes del conjunto tendrán la misma etiqueta, que se establece aleatoriamente.

Longitud de carga: Especifica el tamaño del paquete que sigue al encabezado. Los encabezados de extensión son considerados parte el paquete en el cálculo de la longitud.

Cabecera siguiente: Como se verá en la sección Encabezados de extensión, el campo indica qué cabecera sigue inmediatamente después este encabezado. Si no hay opciones, entonces el siguiente encabezado será TCP (o UDP, ICMP, etc.). Si hay opciones, entonces la cabecera siguiente será la de esa opción.

Límite de saltos: Al igual que en la versión 4, este límite se decrementa hasta llegar a cero en cada salto. Cuando llega a cero, el paquete se descarta. Este campo permite controlar la carga de la red si un paquete entra en un bucle que se haya formado accidentalmente. A su vez, el valor de este parámetro deberá ser lo suficientemente amplio para garantizar que un paquete no se descarte en una ruta larga.

Dirección fuente: Al igual que en la versión 4, contiene la dirección del host que origina el paquete a enviar, pero en 128 bits.

Dirección destino: En este campo irá siempre la dirección de destino final del paquete, excepto en el caso que una extensión de ruteo esté presente.

### 1.2.1 Direcciones

Una de las características principales de IPv6 es el nuevo tamaño de las direcciones que permite tener una cantidad infinitamente más grande. Haciendo una cuenta simple,

podemos calcular que hay 6 x $10^{23}$ direcciones por metro cuadrado en la tierra (Morton, 1997, pág. 18). Aunque parece exagerado suponer que alguna vez semejante cantidad de direcciones puedan ser usadas, el motivo de tener tal espacio tiene que ver con el ruteo. La versión 4, en su crecimiento desordenado y vertiginoso llevó la red a tener enormes tablas de ruteo en los routers centrales. Con el nuevo espacio de direcciones se puede crear un mecanismo de asignación de direcciones con múltiples niveles de jerarquía. Esto permite que las tablas y los algoritmos de ruteo se vuelvan mucho más simples y menos costosos de procesar. El prefijo de una red IPv6 será la ruta a esa red.

La notación de las direcciones de IPv6 se hace en formato hexadecimal, en campos de cuatro dígitos separados por dos puntos. Ya que muchas direcciones van a contener muchos ceros, la notación puede simplificarse eliminando los ceros del comienzo en cada campo. Cuando sólo tengo ceros en uno o varios campos, se anota "::". Por ejemplo, la dirección

DEAD:BEEF:0000:0000:0000:0073:FEED:F00D

puede anotarse como

DEAD:BEEF::73:FEED:F00D

o bien, la dirección

0000:0000:0000:0000:0080:0000:FFAA:CCCC

se anota

::80:0000:FFAA:CCCC.

A diferencia de IPv4, las direcciones no se asignan a hosts sino que se asignan a interfaces dentro de un host. Los distintos tipos de direcciones se analizan en detalle en 1.2.2.

### 1.2.2 Encabezados de extensión

IPv6 introduce una mejora importante en el tratamiento de las opciones del encabezado, ya que las trata como extensiones. De esta forma logra resolver las cuestiones de generalidad y eficiencia; generalidad, porque el protocolo necesita mecanismos adicionales para manejar la fragmentación, el ruteo de fuente o la autenticación; y eficiencia, porque el encabezado básico es el mínimo indispensable: si una aplicación requiere algún servicio adicional lo implementa a través de una extensión.

La figura siguiente muestra cómo es el funcionamiento de los encabezados de extensión:

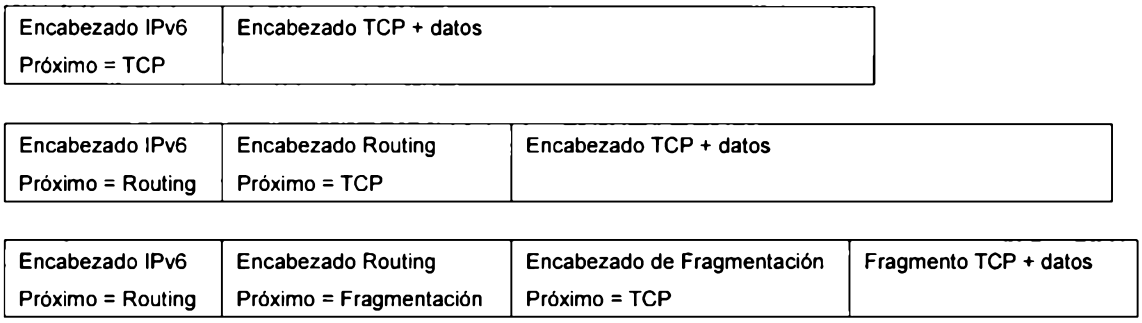



Figura 1-2: Encadenamiento de los encabezados de extensión

Aquí se muestra claramente cómo se engancha una extensión con otra. El encabezado principal contiene en el campo Cabecera Siguiente (visto en 1.2) el tipo de encabezado que se encuentra a continuación, si es una extensión o bien si es un protocolo de

capa superior. A su vez, cada encabezado de extensión tiene un campo que indica qué protocolo le sigue, hasta llegar al protocolo de capa superior: allí se terminan las extensiones. Esta flexibilidad hace que sea posible incluir cuantas extensiones sean necesarias para una aplicación, y quedan a cargo del implementador las consideraciones respecto de la eficiencia que se pierda.

Hasta el momento sólo hay definidos seis tipos de extensiones:

- Opciones para Salto en Salto
- Ruteo
- Fragmentación
- Opciones para el Destinatario
- Autenticación
- Encapsulado de seguridad de la carga (Encapsulating security payload)

La RFC 2460 define el formato que deben tener los encabezados de Opciones para el Destinatario, dando flexibilidad para la creación de nuevas funciones. Los encabezados de extensión sólo son procesados por los extremos de una conexión, evitando procesamiento innecesario en todos los nodos intermedios de la ruta que sigue un paquete, en pos de la eficiencia. Sólo hay una excepción a esta regla y es para el encabezado de Opciones de Salto en Salto, que deberá ser ubicado inmediatamente después del encabezado principal. Esta extensión -y todas las que la sigan- deberá ser procesada por cada nodo intermedio de una conexión.

Aunque la RFC aconseja un orden para las extensiones, ese orden no es obligatorio. Es obligatorio sin embargo, procesar las extensiones en el orden en que aparecen, una por una; no se puede elegir qué extensión procesar primero.

### 1.2.3 Interoperatividad con IP versión 4. Transición

El protocolo IPv6 necesita convivir durante mucho tiempo con su versión anterior. Para lograr un funcionamiento adecuado de la red con ambas tecnologías interoperando, se establecieron distintos mecanismos; en la RFC 2893 se definen algunos de esos mecanismos, entre los que se destacan:

**Doble pila:** Consiste en el uso simultáneo de ambos protocolos en host y routers, llamados "nodos IPv6/IPv4" que administran pilas separadas. De esta forma, cualquier otro nodo IPv4 o IPv6 puede comunicarse con estos nodos.

Un nodo IPv6/IPv4 tendrá dos direcciones, una de cada protocolo, y podrá enviar y recibir ambos tipos de paquetes. Además este mecanismo se podrá combinar con el mecanismo de túnel o el de encapsulado de direcciones, como se verá en los párrafos siguientes.

**Túneles IPv4/IPv6:** Este mecanismo sirve para transmitir paquetes IPv6 sobre una infraestructura IPv4, agregando al paquete original un nuevo encabezado IPv4. Hay cuatro formas de operación de los túneles:

- Host a router
- Router a router
- Router a host
- Host a host

En las primeras dos opciones, el destino final no es el mismo que el final del túnel, por lo que se requiere que el nodo de origen del paquete arme el túnel de acuerdo con alguna configuración previamente establecida. Este tipo de túneles se denomina "configurado".

En las otras opciones, el destino final del paquete es el mismo que el fin del túnel, por lo que no se requiere ninguna configuración en el nodo de origen. Basta con establecer como dirección IPv6 destino la dirección IPv4 del host destino encapsulada en el formato IPv6 como se verá en la sección siguiente.

## 1.3 Asignación del espacio de direcciones

Las direcciones de IPv6 se clasifican según estas categorías:

- Unicast: direcciones asignadas a una interfaz individual.
- Anycast: direcciones para designar a un grupo de interfaces. El paquete llegará a una cualquiera del grupo, usualmente la más cercana. Se usa para designar grupos de servidores redundantes, donde si se "cae" el primero, el siguiente recibirá el paquete.
- Multicast: direcciones para designar a muchas interfaces. El mismo paquete llega a todas las interfaces. Reemplaza a las direcciones de broadcast de la versión 4, aunque tiene algunas diferencias.

La tabla siguiente muestra cómo está asignado inicialmente el espacio de direcciones. Puede verse que sólo el 15% de las direcciones posibles están asignadas, el resto será usado en el futuro.

| Estado | Prefijo | Espacio |
|---|---|---|
| Reservado | 0000 0000 | 1/256 |
| No asignado | 0000 0001 | 1/256 |
| Reservado NSAP | 0000 001 | 1/128 |
| Reservado IPX | 0000 010 | 1/128 |
| No asignado | 0000 011 | 1/128 |
| No asignado | 0000 1 | 1/32 |
| No asignado | 0001 | 1/16 |
| Direcciones Unicast Globales Agregables | 001 | 1/8 |
| No asignado | 010 | 1/8 |
| No asignado | 011 | 1/8 |
| No asignado | 100 | 1/8 |
| No asignado | 101 | 1/8 |
| No asignado | 110 | 1/8 |
| No asignado | 1110 | 1/16 |
| No asignado | 1111 0 | 1/32 |
| No asignado | 1111 10 | 1/64 |
| No asignado | 1111 110 | 1/128 |
| No asignado | 1111 1110 0 | 1/512 |

| | | |
|---|---|---|
| Direcciones Unicast de Enlace local | 1111 1110 10 | 1/1024 |
| Direcciones Unicas de Sitio Local | 1111 1110 11 | 1/1024 |
| Direcciones Multicast | 1111 1111 | 1/256 |

Figura 1-3: Asignación del espacio de direcciones IPv6

En las siguientes secciones se explican los formatos de estas direcciones, su uso y sus funciones.

### 1.3.1 Direcciones unicast locales

Una dirección unicast identifica una interfaz dentro de un nodo en forma única.

Hay varios formatos de direcciones unicast de acuerdo con los diferentes usos que se les den:

Dirección sin especificar: es la dirección :: o 0:0:0:0:0:0:0:0. Esta dirección no debe ser usada por ningún host y sirve para establecer la dirección inicial de una interfase antes de poder configurar la dirección que le corresponde (por mecanismos de configuración automática).

Dirección de loopback: Como en IPv4, existe la dirección ::1 que se utiliza para prueba dentro de un host.

Dirección de túnel IPv6/IPv4: el formato ::*dir_ipv4* se usa en los túneles IPv6/IPv4 explicados en la sección 1.2.3, para poder pasar tráfico IPv6 a través de una infraestructura IPv4.

Direcciones IPv4 en IPv6: el formato ::FFFF:*dir_ipv4* sirve para que los nodos IPv4 puedan operar en una red IPv6. Se denomina "mapeo de direcciones IPv4 a IPv6".

Direcciones 6to4: el formato 2002:*dir_ipv4*:XXXX:*interface_id* fue definido en la RFC 3056 como otro mecanismo de transición hacia IPv6 para que un host pueda comunicarse a través de una red IPv4 sin necesidad de establecer un túnel.

Direcciones NSAP: Permiten mapear direcciones OSI NSAP en IPv6 como mecanismo de transición.

Direcciones IPX: Igual que el anterior, sirven para mapear direcciones IPX en IPv6.

Dirección link-local: La dirección link-local sirve para direccionar paquetes dentro de una red de un mismo link. Los routers no hacen pasar fuera de la red paquetes con la dirección fuente o destino del tipo link-local. Se usa para autoconfiguración de direcciones, descubrimiento de vecinos o cuando no hay routers.

Dirección site-local: Trabaja en forma similar al anterior, sólo que el ámbito de uso es todo un sitio, y no simplemente un link.

Dirección unicast global agregable: Este formato de direcciones sirve para soportar que un sitio se conecte tanto a un proveedor como a una entidad llamada exchange con un mecanismo de agregado. Este doble mecanismo permite que un sitio cambie de proveedor sin necesidad de reconfigurar todos los números de la red, como se describe en la sección siguiente.

### 1.3.2 Mecanismo de agregación de direcciones unicast locales

En la versión 4 actual, un sitio se conecta a uno o más proveedores que le proveen direcciones. Cuando el sitio desea cambiar de proveedor, se ve obligado a reconfigurar todos los números IP de su red. El mecanismo por el cual el proveedor asigna direcciones se denomina agregación (aggregation).

Con el uso de sistemas autónomos, la reconfiguración debido a un cambio de proveedor puede evitarse. Una organización que posee un sistema autónomo, tiene todas sus IP's relacionadas con ese sistema autónomo y puede conectarse y desconectarse de los proveedores que desee sin necesidad de reconfigurar IP's; sólo se actualizan las tablas de ruteo. Pero los sistemas autónomos agregan complejidad a los protocolos de ruteo y por tanto agregan ineficiencia. En IPv6 se pretende que la dirección IP encierre los identificadores de red necesarios para que con tablas simples y fácilmente actualizables se pueda hacer el ruteo.

La versión 6 incorpora una nueva entidad, llamada exchange, donde un sitio puede conectarse y, a través de ella, llegar a su o sus proveedores.

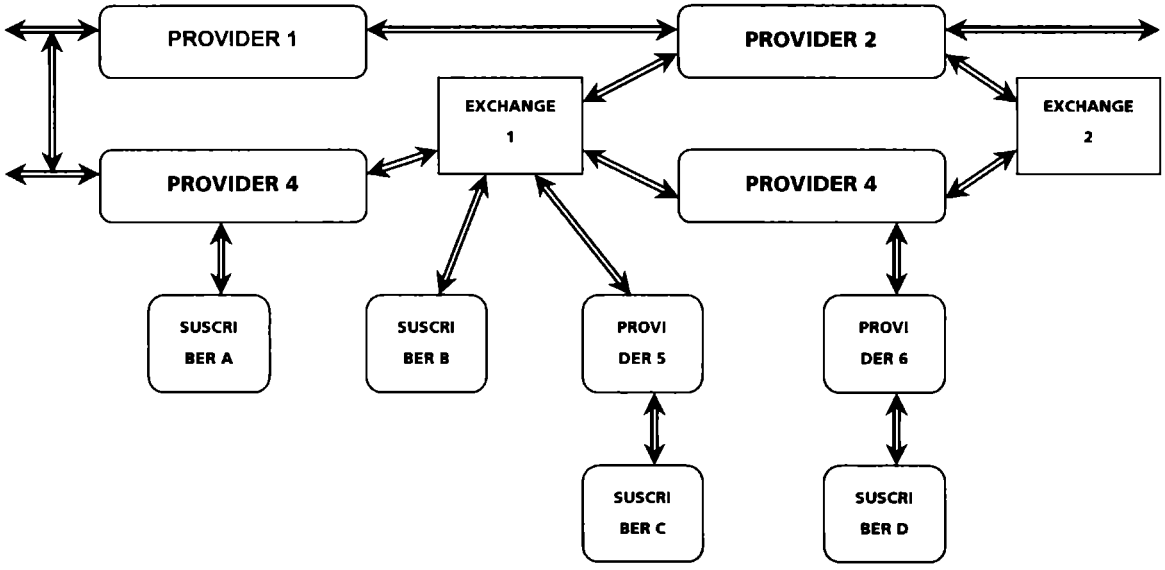



Figura 1-4: Proveedores y suscriptores conectados a través de exchanges (Hinden y otros, 1998)

El formato de direcciones unicast locales agregable está diseñado para soportar tanto proveedores como exchangers. Un exchange va a asignar direcciones IPv6 a las organizaciones conectadas a él. De esta forma, cualquier organización puede cambiar de proveedor (conectada a través del exchange) sin cambios de IP's.

Como se ve en la figura, las entidades se organizan de una manera jerárquica. Tanto los proveedores indicados en la figura como P1, P2, P3 y P4 como los exchangers son entidades que forman la *Topología Pública* de la red; los proveedores menores -típicamente ISP's- y otras organizaciones (P5, P6, Sa, Sb y Sc) forman cada uno su *Topología de Sitio* interno a ellos; por último se encuentran suscriptores (Sd, Se y Sf) que sólo tienen un *Identificador de Interfaz* en el link al cuál se conectan.

A continuación, se muestra la estructura de una dirección unicast local agregable:

| 3 | 13 | 8 | 24 | 16 | 64 bits |
|---|---|---|---|---|---|
| FP | TLA ID | RES | NLA ID | SLA ID | Identificador de interfaz |

FP: Format Prefix 001
TLA ID: Top Level Aggregation Identifier
RES: reservado
NLA ID: Next Level Aggregation Identification
SLA ID Site Level Aggregation Identification

Figura 1-5: Estructura de dirección IPv6 (Deering y Hinden, Jul. 1998) .

Los TLA ID se encuentran al tope de la jerarquía de ruteo. Los routers que forman el backbone principal conocen información de ruteo a todos los TLA ID activos. Se prevén 8192 TLA ID's, pero ese número puede aumentarse avanzando sobre el campo reservado siguiente o bien creando un nuevo Format Prefix. El número es arbitrariamente chico para evitar grandes tablas de ruteo en los routers principales de la red.

Cada organización administradora de un TLA ID puede a su vez organizar jerárquicamente su red asignando a organizaciones de segundo nivel un NLA ID. Quien administra los NLA ID's puede armar jerarquías libremente, o bien puede usar el espacio entero de identificadores disponibles (24 bits) en forma plana. Armando jerarquías se gana en tablas de ruteo más eficientes, mientras que haciendo asignaciones planas se gana en flexibilidad.

Al igual que con los TLA ID's, los administradores de NLA ID's asignan los SLA ID's a organizaciones más pequeñas, para que armen sus propias subredes y su propia jerarquía. Las 65536 subredes disponibles (16 bits) son más que suficientes para cualquier organización que administra redes en la actualidad.

Finalmente el Identificador de Interfaz de 64 bits sirve para asignar en forma unívoca un número a cada host conectado a una subred. Puede usarse la numeración que se desee, pero se recomienda usar la dirección de enlace al link (MAC address). Por ejemplo, en una red Ethernet se usa el número de 48 bits que identifica al hardware de red y en el resto se agregan ceros.

### 1.3.3 Direcciones anycast

Las direcciones anycast identifican varias interfaces, típicamente de varios nodos. Un paquete con destino a una dirección anycast se ruteará a la interfaz más cercana que identifique esa dirección, de acuerdo al protocolo de medición de distancia.

Las direcciones anycast se toman del espacio de direcciones unicast. Basta con asignar a dos interfaces la misma dirección unicast para que se transforme en anycast.

El uso esperado de las direcciones anycast es identificar un grupo de routers pertenecientes a un determinado proveedor, y así poder obligar a un paquete a tomar una ruta de ese proveedor, especificando la dirección anycast de ese proveedor en un header de ruteo. También pueden usarse para identificar grupos de routers de una determinada subred o un dominio de ruteo.

Debido a que el mal uso de direcciones anycast provoca múltiples trastornos en la red, se pueden usan sólo en routers y no se pueden establecer como direcciones de origen en un paquete.

Hay una dirección anycast obligatoria para todos los routers de una subred y es la dirección Subnet-Router anycast. El formato es:

| n bits | 128-n bits |
|---|---|
| prefijo de subred | 000000000000000 |

Figura 1-6: Formato de dirección subnet-router anycast.

La dirección está formada por la dirección de la subred y ceros en lugar del identificador de interfaz. Un paquete enviado a esta dirección deberá ser recibido por el router de la subred más cercano al originante.

### 1.3.4 Direcciones multicast

Las direcciones multicast identifican a un grupo de nodos. Esto significa que un paquete dirigido a una dirección multicast llegará a todos los nodos que esa dirección represente. El formato es:

| 8 | 4 | 4 | 112 bits |
|---|---|---|---|
| 1111 1111 | Flags | Scope | ID de grupo |

Figura 1-7: Formato de direcciones IPv6 multicast.

Los bits de flag serán 0000 para direcciones permanentes asignadas por la autoridad de direcciones o 0001 para direcciones temporarias, que sólo tienen aplicación dentro de un ámbito local.

Los bits de alcance (scope) sirven para definir el alcance que tiene el grupo de nodos representados por una dirección multicast. Los alcances definidos hasta ahora son:

1 alcance local al nodo

2 alcance local al link

5 alcance local al site

8 alcance local a la organización

E alcance global

Para ilustrar el funcionamiento de las direcciones multicast se puede tomar un ejemplo. Está definido que el grupo de los servidores de tiempo (NTS) tiene el ID 101, entonces la dirección

FF02::101

representa a todos los servidores de tiempo que se encuentran en el mismo link del emisor; la dirección

FF0E::101

representa a todos los servidores de tiempo de toda la internet.

Hay un formato especial de dirección multicast que se denomina Solicited-node multicast address. El formato es

FF02::1:FFxx:xxxx

donde las x se reemplazan por los últimos 24 bits de la dirección anycast o unicast de la interfaz. Cada nodo debe unirse a cada dirección Solicited-node multicast que le corresponde por cada interfaz que tenga asignada una dirección IPv6.

## 1.4 Ruteo de paquetes en IPv6

El esquema de direcciones que incluían la estructura TLA/NLA descripta en 1.3.2 hacía suponer que las políticas de ruteo a seguir estarían atadas a esta estructura. Pero según la reciente RFC 3587 (Hinden y otros, 2003), TLA/NLA no es la mejor solución técnica para este estado de difusión de IPv6. Además, esta estructura de direcciones condiciona a las autoridades de asignación de números en la forma y políticas de distribución del espacio, por lo que tampoco es aceptada.

Como reemplazo a la estructura TLA/NLA se eligió que sean los Regional Internet Registries (RIRs) los que asignen las direcciones con una política conjunta, de forma muy similar a como sucede en la actualidad con las direcciones IPv4.

El esquema de sistemas autónomos (Autonomous Systems o AS) vigente se mantiene en IPv6 y como consecuencia, el ruteo es muy similar al de IPv4. La ventaja respecto del ruteo mencionada en 1.1 quedará para un desarrollo posterior.

Un sistema autónomo es un grupo de redes que tienen una administración común. Dentro de un AS la información de ruteo se distribuye de acuerdo a un Interior Gateway Protocol (IGP); RIP y OSPF son protocolos de tipo IGP. Entre ASes, la información de ruteo viaja de acuerdo a un Exterior Gateway Protocol, como es el caso de BGP-4. Estos tres protocolos tienen sus respectivas extensiones para IPv6, y también se está desarrollando un nuevo IGP para IPv6 llamado IS-IS. Cada sistema autónomo define qué protocolo de ruteo IGP desea implementar .

### 1.4.1 RIPng

El RIPng es la versión para IPv6 del conocido protocolo RIP para IPv4. Es un protocolo basado en el algoritmo vector-distancia conocido como Bellman-Ford, y se usa en redes pequeñas de hasta 15 saltos.

Las únicas modificaciones que introduce RIPng respecto de RIP son el tamaño de la dirección y el chequeo de seguridad (Malkin y Minnear, 1997). Como IPv6 maneja desde el encabezado las cuestiones relativas a la seguridad, RIPng usa esas facilidades. El resto del protocolo tiene la misma forma de funcionamiento que su antecesor RIP.

### 1.4.2 OSPFv3

El otro IGP más usado es el OSPF. La RFC 2740 describe los cambios que deben hacerse sobre OSPFv2 para IPv4 para su funcionamiento en IPv6, creando la versión 3 de OSPF. Algunos de estos cambios son (Hagen, 2002, p.152):

- Proceso sobre link, no sobre subred: en IPv6 dos o más nodos de distintas subredes pueden compartir un link, y necesitarán cambiar información de ruteo sobre un mismo link.
- Cambio en la semántica: Los routers que se identificaban por la dirección IP en OSPFv2 pasan a identificarse por el Router ID, que se mantiene en 32 bits.
- Alcance del "flooding": La información de ruteo se actualiza mediante el intercambio de Link State Advertisments (LSAs) por un mecanismo de "flooding" (lit. inundado). El alcance del flooding define a quiénes se entregarán los LSAs. La nueva versión incorpora nuevos alcances necesarios en IPv6.
- Soporte de múltiples instancias en un mismo link: múltiples ASs pueden compartir un mismo link y necesitan que dos OSPFs se puedan ejecutar en forma conjunta sobre ese link.
- Uso de la dirección link-local: el nuevo OSPF asume que todas las interfaces de los routers poseen una dirección unicast link-local en IPv6, y la usa para mensajes de descubrimiento de vecino entre otros.
- Autenticación: se sacó de OSPF y se usan las funciones de seguridad de IPv6.

### 1.4.3 BGP

El Border Gateway Protocol es el protocolo más usado para intercambiar información de ruteo entre ASs; es del tipo EGP. No hay una definición particular de BGP para IPv6, sino que existe una extensión a BGP4, definido en la RFC 2858, que introduce modificaciones y agregados para soportar múltiples protocolos, como IPX e IPv6, y donde se eliminan todas las dependencias con IPv4.

La forma de usar las extensiones multiprotocolo de BGP en IPv6 está definida en la RFC 2545, y tiene que ver con el tratamiento de las direcciones unicast site-local y unicast link-local que introduce IPv6. El resto del funcionamiento del protocolo se ajusta a lo definido en la RFC 2858.

## 1.5 Calidad de servicio (QoS) en IPv6

Los distintos mecanismos para brindar calidad de servicio pueden agruparse en dos grandes arquitecturas: Integrated Services y Differentiated Services (Hagen, 2002, p.115). Ambas trabajan de acuerdo a distintas políticas de tráfico. La primera establece una determinada QoS de extremo a extremo de una conexión, mientras que la otra se basa en administrar el tráfico en cada nodo que recorre un flujo de paquetes.

En IPv6 se incorporaron o mejoraron distintos mecanismos para brindar QoS eficientemente. Como IP forma básicamente una red no orientada a conexión, muchas veces se hace imposible que esa red asegure una determinada calidad. No hay hasta el momento una solución única a este problema.

### 1.5.1 Integrated Services: RSVP.

Los primeros pasos para brindar calidad de servicio fueron dados buscando imitar en una red IP las prestaciones de una red orientada a conexión, como la red telefónica, precisamente pensando en transmitir flujos de voz y video por Internet.

La arquitectura de Integrated Services se basa en hacer una reserva de todos los elementos que hacen a la calidad de servicio, como ancho de banda o buffers necesarios, de extremo a extremo de un flujo de paquetes. Para ello se requiere tener un *protocolo* para comunicar los requerimientos de QoS de un determinado flujo, y la *infraestructura de nodos* que soporten los pedidos de ese protocolo.

Entre otros, los servicios de Controlled load y Guarenteed, especificados en las RFCs 2211 y 2212, proveen definiciones para que los nodos soporten los pedidos de QoS.

El RSVP (Resource Reservation Protocol) definido en la RFC 2205 (Braden y otros, 1997) establece el mecanismo por el cuál una aplicación o un router se comunican los pedidos/respuestas de QoS. Es un protocolo de nivel superior al IP, pero no es un protocolo de transporte sino que es de control, de forma similar a como trabaja ICMP. Los pedidos de QoS los hace en una sola dirección, generalmente del receiver hacia el sender.

Si bien RSVP trabaja *sobre* IP y no es parte del conjunto de IPv6, es un protocolo nacido para responder a los nuevos requerimientos de la red, y forma parte de la nueva Internet. RSVP utiliza el campo *next header* de IPv6 para un más eficiente procesamiento de la reserva por parte de cada router, como se verá en la sección 1.5.3.

La principal desventaja de la reserva de recursos es que se requiere que *toda* la red acepte y responda favorablemente a los pedidos de QoS. Esto generalmente se puede obtener en una red de administración centralizada, pero muy difícilmente se da en redes más complejas; en suma, va a pasar mucho tiempo hasta que una arquitectura como ésta se pueda implementar en toda la Internet.

### 1.5.2 Differenciated Services

La otra gran arquitectura desarrollada para brindar calidad de servicio es Diferenciated Services. Esta arquitectura asevera que la prioridad y el tipo de servicio de un paquete son suficientes para que cada nodo de la red administre el tratamiento de esos paquetes sensibles a la calidad de servicio, y les dé un tratamiento preferencial frente a otros.

Esta arquitectura no hace reserva de recursos ni elige rutas determinadas, lo que la hace muy flexible y sencilla para su uso e implementación, especialmente en redes amplias o de administraciones diferentes. El problema que surge es que no siempre puede brindar la calidad de servicio requerida por una aplicación, especialmente cuando se trata de voz y video, con sólo dar prioridad de paso a determinados paquetes por sobre otros.

El campo Clase de Tráfico del encabezado IPv6 fue redefinido en la RFC 2474 (Nichols y otros, 1998) como el campo Differenciated Services (DS), cuya estructura es:

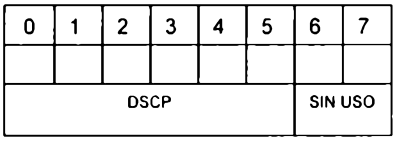



Figura 1-8: Campo Clase de tráfico.

Los primeros seis bits del campo DS definen el *codepoint DS* (DSCP), mediante el cuál los routers seleccionan el comportamiento por cada salto (per-hop behavior, o PHB) para cada paquete. Los 64 codepoints posibles se dividen en tres grupos:

- 32, (máscara xxxxx0) de uso estandar
- 16, (máscara xxxx11) de uso experimental o local
- 16, (máscara xxxx01) de uso experimental o futuro uso estandar

La RFC 2474 define, dentro del primer grupo, ocho selectores de clase (máscara xxx000) recomendados, donde los de mayor número tienen precedencia sobre los menores al momento de ser tratados por los routers. Esto significa que un paquete con DSCP igual a 101-000 tendrá precedencia, es decir, pasará primero, sobre un paquete con DSCP 001-000

### 1.5.3 Otros elementos IPv6 de QoS

Las arquitecturas descriptas usan distintos elementos definidos en la arquitectura IPv6 para llevar a cabo su tarea:

- Flujos: Un flujo es una secuencia de paquetes para los cuáles una aplicación requiere un tratamiento especial. Cada flujo se identifica por las direcciones fuente y destino y por el *flow label* que es un número aleatorio común a todos los paquetes de un flujo. El tratamiento especial se comunica a los nodos de las formas vistas en las secciones anteriores.
- Encabezado de extensión de ruteo: un encabezado de ruteo sirve para indicar una secuencia de direcciones IP que deben ser alcanzadas por el paquete que lo contiene. Esta es una forma de indicar una ruta específica para un paquete. Si se conoce que determinados routers soportan el manejo de QoS, puede usarse esa ruta específica en un flujo que la requiera.
- Encabezado de extensión hop-by-hop: Un paquete con un encabezado de extensión hop-by-hop indica a todos los nodos intermedios en el camino, que deben procesar las opciones de ese encabezado y los que le siguen. Esta es una forma muy eficiente, ya que se hace a nivel IP, de comunicar cuestiones relativas al ruteo, sin afectar la performace global. El RSVP usa el encabezado hop-by-hop para comunicar los parámetros necesarios a cada router.

### 1.5.4 Otros enfoques sobre QoS

Resulta interesante mencionar, en función del análisis de este trabajo, otras líneas de estudio sobre la cuestión de Calidad de Servicio:

- QoS en los sistemas finales: este enfoque supone que la red debe seguir entregando paquetes haciendo el "mejor esfuerzo", y que la calidad debe ser proporcionada por el sistema que usa la red, que deberá implementar mecanismos para evitar el delay, jitter, etc. Si bien es una forma muy simple de tratar las cuestiones de calidad, no es efectiva para uso en aplicaciones multimedia reales.
- QoS por servicios diferenciados: en IPv6 se pueden armar distintos grupos multicast de acuerdo a las distintas clases de tráfico. Por ejemplo, se puede codificar audio en cuatro calidades diferentes y transmitirlos a cuatro grupos distintos. Cada usuario se suscribirá al grupo que mejor se adapte por el ancho de banda a usar, de acuerdo al estado de la ruta entre él y la fuente.

## 1.6 Movilidad

La cantidad de dispositivos móviles existentes, como teléfonos celulares, palmtops o notebooks, van en franco crecimiento. La convergencia de redes hace suponer que en un futuro no muy lejano todos estos dispositivos van a requerir una o varias direcciones IP, y además van a requerir funcionar siempre y en todo momento, sin importar en qué lugar se encuentre, o que esté en movimiento. Así sucede con los teléfonos celulares, que hacen roaming cuando pasan de una celda a otra sin perder continuidad en la comunicación.

IPv4 posee una implementación de movilidad basada en la creación de dos direcciones para un dispositivo móvil: la primera llamada dirección de sitio y una segunda llamada dirección Care-of. La dirección de sitio es fija, mientras que la dirección care-of varía de acuerdo a qué punto de conexión esté accediendo el dispositivo. Para lograr que todo funcione, se requiere también un Home Agent y un Foreing Agent. Cuando un paquete es enviado al dispositivo móvil, se envía a la dirección IP de sitio. El Home Agent, que previamente conoce la dirección Care-of del dispositivo, encapsula el paquete con un encabezado cuyo destino es la dirección care-of del dispositivo. Cuando el paquete le llega al dispositivo, este lo desencapsula y lo procesa como un paquete normal.

El dispositivo móvil busca su dirección care-of mediante mensajes llamados Agent Advertisements enviados por el Foreing Agent (agente lejano), que se encuentra en el punto de conexión temporario del dispositivo. Luego envía un requerimiento de registro al Home Agent (agente de sitio) con la dirección care-of, algunos parámetros y el tiempo de vida de la registración. Todo este registro se denomina Binding, y es respondido por el Home Agent mediante una *respuesta de registro*. Cuando el dispositivo se mueve a un nuevo punto de acceso, se requiere que haga un nuevo binding con el Home Agent. Todos estos registros se encriptan con firmas digitales para evitar problemas de seguridad.

IPv6 mejora en forma significativa el manejo de los dispositivos móviles. IPv4 hace un túnel con los paquetes, provocando una comunicación triangular, como se ve en la figura:

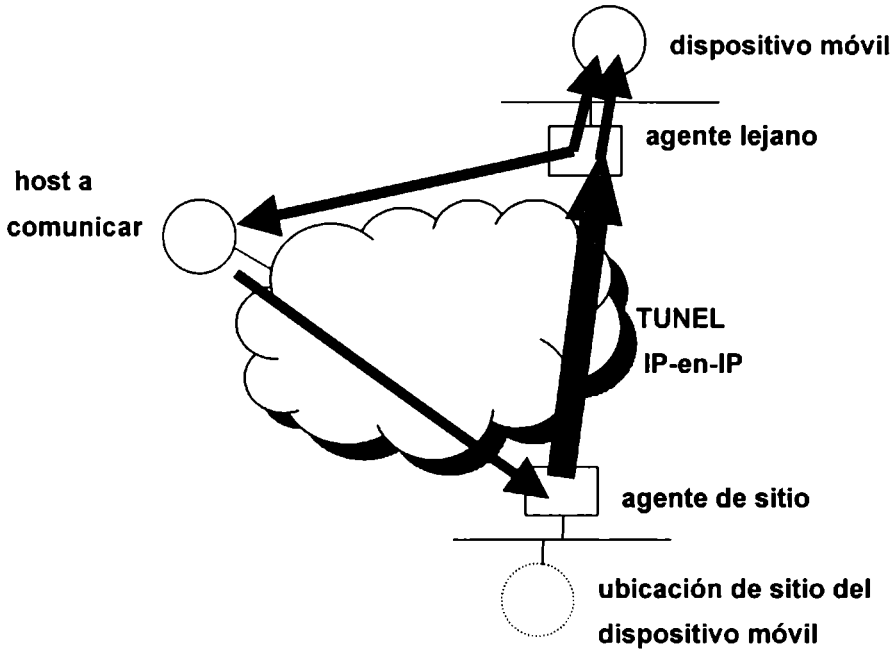



Figura 1-9: Movilidad IPv4 (Teitelbaum, 2003).

En IPv6, el Home Agent utiliza el encabezado de *extensión de ruteo* y de *opciones al destinatario* para encapsular los paquetes con la dirección care-of. El dispositivo móvil no tiene la necesidad de poner la dirección de sitio como origen de un paquete, sino que pone la dirección care-of, evitando de esta forma la comunicación triangular:

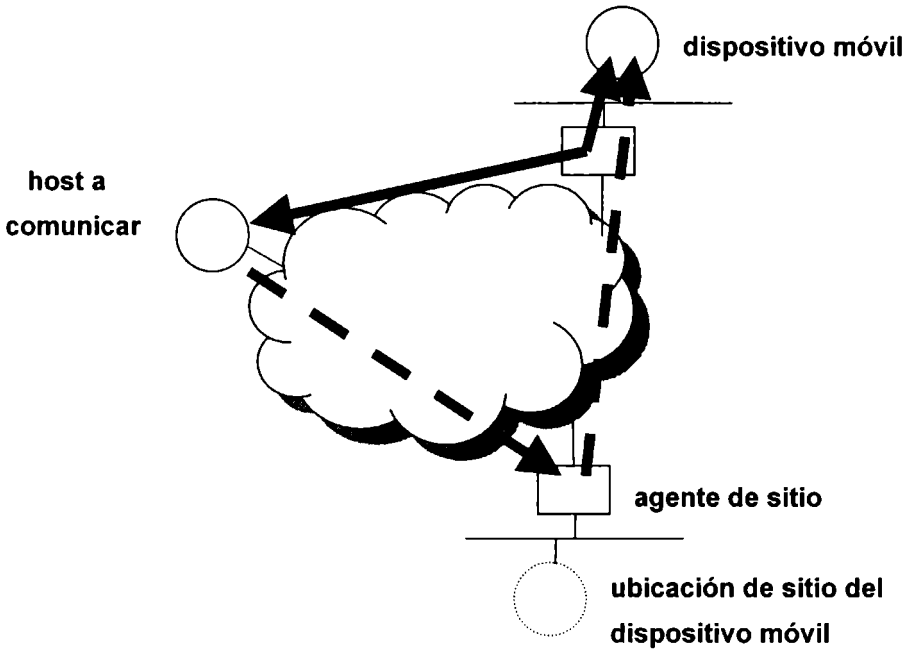



Figura 1-10: Movilidad IPv6 (Teitelbaum, 2003).

Tampoco es necesario el Foreing Agent en IPv6 ya que descubrimiento de vecino y autoconfiguración son funciones estándar del protocolo, que en IPv4 debía hacerlas ese agente.

En la actualidad el desarrollo de movilidad IPv6 sigue siendo experimental, y requiere de mayor desarrollo especialmente en cuestiones de seguridad. El crecimiento de las redes wireless y de los celulares de tercera generación conectados a Internet prometen un desarrollo muy importante del tema movilidad IP.

## 1.7 Multicast IPv6

Los grupos de nodos identificados por direcciones multicast adquiere singular importancia en aplicaciones de VoIP, ya que son muy útiles cuando se necesita hacer una conferencia o videoconferencia múltiple. Un sólo paquete dirigido a una dirección multicast es recibido por todos las interfaces de todos los nodos unidos a esa dirección multicast. También es una forma de emular una transmisión broadcast, como pueden ser las transmisiones de radio y TV.

Multicast existe como agregado para IPv4 pero es estándar en IPv6. El formato de direcciones multicast fue mostrado en 1.3.4. En la RFC 2375 (Hinden y Deering, 1998) se define la asignación inicial de direcciones multicast, que se muestra a continuación:

| Node-Local Scope | |
|---|---|
| FF01:0:0:0:0:0:0:1 | All Nodes Address |
| FF01:0:0:0:0:0:0:2 | All Routers Address |
| **Link-Local Scope** | |
| FF02:0:0:0:0:0:0:1 | All Nodes Address |
| FF02:0:0:0:0:0:0:2 | All Routers Address |
| FF02:0:0:0:0:0:0:3 | Unassigned |
| FF02:0:0:0:0:0:0:4 | DVMRP Routers |
| FF02:0:0:0:0:0:0:5 | OSPFIGP |
| FF02:0:0:0:0:0:0:6 | OSPFIGP Designated Routers |
| FF02:0:0:0:0:0:0:7 | ST Routers |
| FF02:0:0:0:0:0:0:8 | ST Hosts |
| FF02:0:0:0:0:0:0:9 | RIP Routers |
| FF02:0:0:0:0:0:0:A | EIGRP Routers |
| FF02:0:0:0:0:0:0:B | Mobile-Agents |
| FF02:0:0:0:0:0:0:D | All PIM Routers |
| FF02:0:0:0:0:0:0:E | RSVP-ENCAPSULATION |
| FF02:0:0:0:0:0:1:1 | Link Name |
| FF02:0:0:0:0:0:1:2 | All-dhcp-agents |
| FF02:0:0:0:0:1:FFXX:XXXX | Solicited-Node Address |
| **Site-Local Scope** | |
| FF05:0:0:0:0:0:0:2 | All Routers Address |
| FF05:0:0:0:0:0:1:3 | All-dhcp-servers |
| FF05:0:0:0:0:0:1:4 | All-dhcp-relays |
| FF05:0:0:0:0:0:1:1000 -FF05:0:0:0:0:0:1:13FF | Service location |

Figura 1-11: Asignación de direcciones multicast.

En IPv4 hay un protocolo específico para la administración de multicast llamado Internet Group Management Protocol (IGMP). En IPv6 la administración se hace a través del ICMPv6, y se denomina Multicast Listener Discovery (MLD). Este protocolo posee tres tipos de mensajes: Query, Report y Done, y se envían siempre con un alcance Link-local. Los mensajes Query los envía un nodo que desea saber qué Listeners de un determinado grupo hay en el link, o qué grupos tienen listeners en el link. Los mensajes Report son la respuesta a un mensaje Query. La RFC 2710 define el formato de los mensajes y muestra varios diagramas de transición de estado.

Los paquetes dirigidos a una dirección multicast requieren ser ruteados, y para eso se han definido varios protocolos para IPv4:

- Distance Vector Multicast Routing Protocol (DVMRP) es una versión multicast similar a RIP, y tiene los mismos problemas de escalabilidad.
- Multicast Extension a OSPF (MOSPF) es una extensión de OSPF para soportar multicast.

- Protocol Independent Multicast (PIM) es el protocolo más usado en Internet de ruteo de multicast.
- Border Gateway Multicast Protocol (BGMP) opera más eficientemente que el anterior y sirve a su vez para ruteo entre dominios.

Como multicast es estándar en IPv6, también el ruteo multicast figura en los protocolos de ruteo para IPv6, como OSPFv3. El resto de los protocolos están en adaptación.

El campo de alcance de multicast en IPv6, sumado al gran espacio de direcciones disponibles, hacen que sea ideal para implementar esta técnica, ya que le brinda escalabilidad y flexibilidad.

# Capítulo 2

# VOZ SOBRE IP

Las comunicaciones de voz tradicionales, establecidas sobre circuitos conmutados permanentes son de buena calidad y altamente confiables, pero a su vez son muy ineficientes. El canal de transmisión se usa sólo una pequeña porción del tiempo en que dura la comunicación gracias a que no se realiza compresión de ningún tipo, y los espacios en silencio no son aprovechados ni suprimidos. Por otra parte, la implantación de redes de voz y datos muchas veces se hace en conjunto, en general dentro de la misma organización.

La necesidad de optimización -por cuestiones de costo- del uso de los canales de voz, sumado a la conveniencia de desarrollar una sola red de voz y datos integrados, llevaron al desarrollo de las comunicaciones de voz mediante conmutación de paquetes, transmitidos a través de redes de datos. La red de conmutación de paquetes más ampliamente difundida es IP, para la cual se desarrollaron varias tecnologías de transmisión de voz y más recientemente, de video.

Otro factor que ayuda a la convergencia de redes es el desarrollo del hardware; mientras que no hace muchos años el costo de un teléfono convencional era muy inferior al costo de una computadora con facilidades multimedia, hoy en día existe una amplia variedad de dispositivos de bajo costo para conectar a una PC o a la red y realizar comunicaciones multimedia, con prestaciones muy superiores a las de un teléfono común.

Hay distintos tipos de conexión de voz sobre IP (VoIP) de acuerdo a los extremos que intervienen:

- PC a PC: mediante un software y el equipo multimedia de la máquina, puede establecerse una comunicación de voz y video.
- PC a teléfono: una máquina puede comunicarse con un teléfono conectado a la RTPC (Red Telefónica Pública Conmutada). Los paquetes "ingresan" a la red telefónica mediante un "gateway".
- Teléfono a teléfono: Una comunicación puede tener en su recorrido algún tramo que pasa por la red, mediante el uso de dos gateways.

Hay variantes a estos esquemas que se logran haciendo combinaciones. Por ejemplo, puede hacerse una conexión PC a PC pasando en un tramo por dos gateways y así evitar un router congestionado. Por otro lado, se están desarrollando nuevos productos cada vez más accesibles que permiten difundir el uso de VoIP como alternativa a la red convencional. Ya hay gran cantidad de modelos de teléfonos IP que tienen el aspecto de un teléfono convencional, pero por dentro es en realidad una computadora que debe conectarse a una red IP para

funcionar. Además requiere que esa red IP tenga conectados una serie de dispositivos que le permitirán a ese teléfono hacer llamadas hacia otros teléfonos tanto IP como convencionales.

En general el uso más difundido que tiene actualmente VoIP es el de reemplazar los costosos circuitos de 64 Kb de la telefonía tradicional para comunicaciones de larga distancia nacional e internacional. Tanto proveedores de servicios de larga distancia, como empresas con sedes en distintos puntos lejanos usan en algunos casos VoIP en sus enlaces interurbanos. Suponiendo que una empresa requiere un enlace de datos entre dos de sus sedes, se logra un interesante ahorro de costos agregando al enlace de datos una conexión VoIP, que además permite tener una funcionalidad mejor.

## 2.1 Los elementos de un sistema de VoIP

Cuando apareció la necesidad de hacer transmisiones multimedia a través de una red IP, cuyo exponente más notorio es Internet, comenzaron a desarrollarse distintos protocolos que posibilitaran esa tarea. IP es una red no orientada a conexión, que hace entrega de paquetes de acuerdo al "mejor esfuerzo", es decir, no se garantiza la calidad de servicio de la red. Está diseñada para el envío de datos, los que no son afectados si se produce algún delay, llegan en paquetes desordenados o son necesarias retransmisiones. Pero las aplicaciones multimedia sí se ven afectadas por estas cuestiones.

Por otra parte, la comunicación en IP se produce entre dos extremos que están esperando para establecer una conexión. Por ejemplo, un usuario envía un mail a un servidor que está escuchando en forma permanente para recibirlo. Si el servidor estuviera fuera de servicio por alguna razón no hay forma de enviar ese mail. Este comportamiento es totalmente diferente en el caso de una conexión telefónica, ya que quien recibe la comunicación no está escuchando en forma permanente el auricular sino que hay que avisarle con un timbre que levante el auricular para comunicarse.

Estos dos tópicos dan cuenta de la necesidad de establecer una serie de protocolos que resuelvan estas cuestiones, divididos en tres categorías (Sisalem y otros):

- **Señalización:** para encontrar usuarios, verificar su presencia, establecer, modificar y cerrar una sesión.
- **Transporte de medios:** para transportar en forma paquetizada audio y video.
- **De soporte:** localización de gateways, DNS, reserva de recursos, accounting, etc.

Varios protocolos se han desarrollado en cada uno de estos tópicos. Como habitualmente sucede en estos casos, no hay consenso sobre cuál adoptar ni tampoco total conectividad/compatibilidad entre ellos, y en la mayoría de los casos ninguna compatibilidad.

En la figura siguiente se muestran los protocolos relacionados con VoIP, muchos de los cuáles se analizan en detalle en este trabajo:

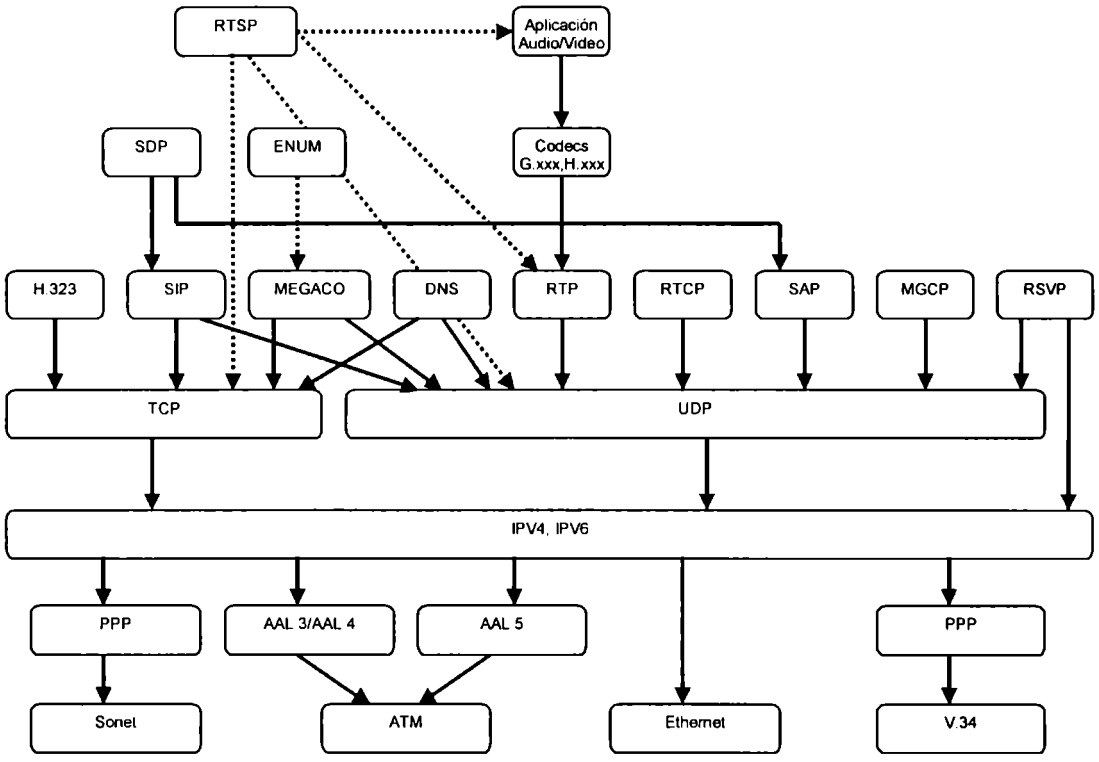



Figura 2-1: Protocolos relacionados con VoIP (extraído de Miller, 2002, pág. 122).

En el caso de la *señalización*, hay dos grupos de protocolos que son los más aceptados hasta el momento como estándares del mercado: H.323 y SIP.

En el caso del *transporte de medios*, ambos estándares usan el protocolo RTP (Real-time Transport Protocol) para transmitir la información codificada de audio y video.

### 2.1.1 H.323

La ITU (International Telecommunication Union) desarrolló la Recomendación H.323 llamada Packet-Based Multimedia Communications Systems, en la que se describen terminales y otras entidades necesarias para realizar transmisiones multimedias en redes de paquetes que no proveen Calidad de Servicio garantizada. No está estrictamente pensada para IP, aunque es el protocolo de red más usado en esta aplicación. H.323 hace uso de una serie de otras recomendaciones que son complementarias; aquí se referirá a H.323 como al conjunto de esas recomendaciones. Las más importantes son: H.225.0 (Call signalling protocols and media streams packetization for packet based multimedia communications systems), H.245 (Contol protocol for multimedia communication), Audio y video codecs G.711, G.722, G.723.1, G.728, G.729, H.261 y H.263, T.120 (Data protocols for multimedia conferencing) entre otros. H.323 es muy completa, compleja y detalla todos y cada uno de los ítems a tener en cuenta en una transmisión multimedia. Enfoca el problema desde el punto de vista de la comunicación telefónica y de video a llevar a cabo.

### 2.1.2 SIP / IETF

Del lado de Internet, la IETF (Internet Engeneering Task Force) desarrolló paralelamente su conjunto de RFCs para tratar el tráfico multimedia, entre las que se mencionan la RFC 3261: Session Initiation Protocol (SIP), RFC 2327: Session Description Protocol (SDP), RFC 3263: SIP: Locating SIP servers y RFC 3015: Megaco Protocol Version 1.0 entre otras. Esta última, llamada Megaco, fue desarrollada en conjunto con la ITU y también se la conoce como Megaco/H.248 por su nombre relacionado con la nomenclatura de la ITU. Este protocolo define cómo es internamente un gateway para conectar una red VoIP con la red telefónica pública. Debido a su origen, SIP y los protocolos asociados enfocan el problema del transporte multimedia desde el punto de vista de la Internet. En la figura siguiente puede verse una representación de ambos conjuntos de protocolos, donde pueden apreciarse las relaciones entre ellos:

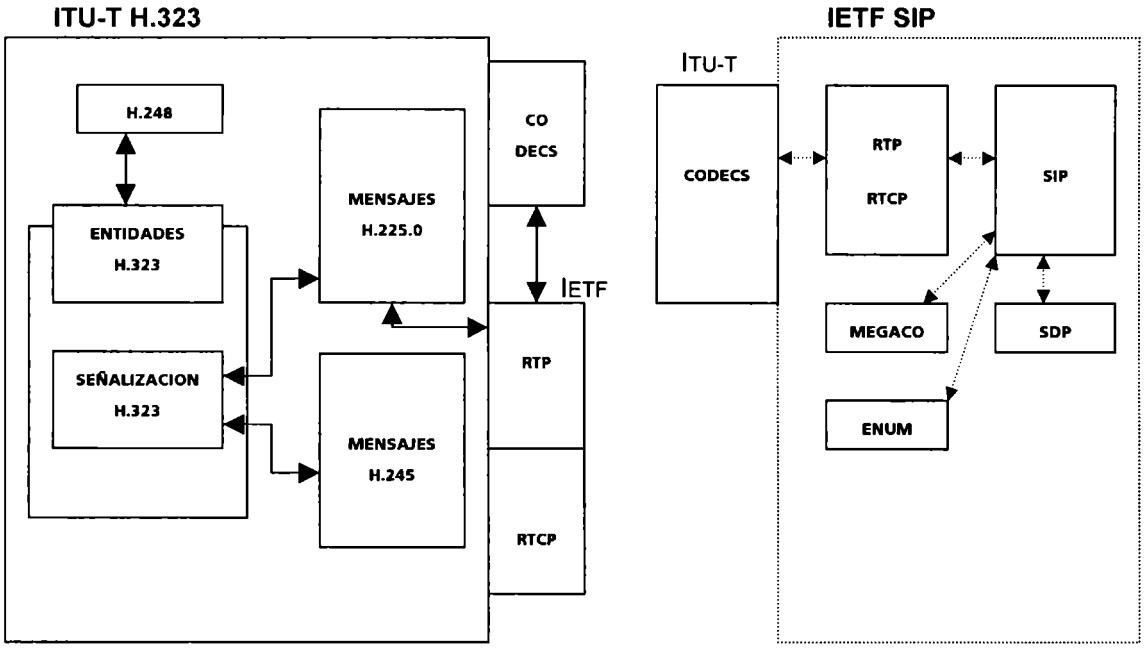



Figura 2-2: Relaciones entre protocolos IETF e ITU-T

### 2.1.3 RTP

Como ya se ha visto, la red IP no ofrece garantía de Calidad de Servicio. Los paquetes son puestos en un extremo de la red, y hacen el mejor esfuerzo para llegar al extremo destino, pero nadie garantiza en qué tiempo lo haga, y ni siquiera que realmente lleguen. Esos servicios se dejan para servicios capas superiores de la red, como por ejemplo TCP. En UDP en cambio, tampoco existe esta garantía, por lo que en general el usuario final es quien termina corrigiendo estos problemas. Dan cuenta de ello las numerosas veces que un usuario se ve obligado a hacer "Reload" cuando está navegando por la web.

Esta característica de la red IP deriva en tres problemas (Miller, 2002, pag.303) cuando se requiere transmitir en tiempo real que son:

- Packet lost: paquetes que se pierden.
- Packet jitter: paquetes que llegan a destino en forma desordenada, ya que no recorrieron la misma ruta.

- Packet delay: paquetes que llegan con mucho retraso y hacen que, por ejemplo, no se pueda mantener una conversación en forma fluída.

Una aproximación a la solución de estos problemas derivó en el diseño de un protocolo de transporte de datos en tiempo real, el RTP (Real-time Transport Protocol), que es un protocolo que provee servicios extremo a extremo de envío de datos de aplicaciones de tiempo real. Fue desarrollado por la IETF y definido en la RFC 3550. Lo acompaña en la misma RFC el protocolo RTCP (Real-time Transport Control Protocol), que provee servicios de monitorización de la calidad de servicio y registro de participantes de una sesión. Mediante la utilización de ambos protocolos en el transporte de streams de medios se logra resolver los problemas descriptos, o bien detectarlos para tomar las medidas correctivas (ej: disminuir la calidad de audio o video, tomar una ruta menos congestionada, etc.).

En la sección 2.4.1 se mostrarán en detalle las funciones del protocolo, el encabezado y su modo de operación.

## 2.2 Funcionamiento de H.323

### 2.2.1 La Recomendación por dentro

La Recomendación H.323 describe los componentes de un sistema H.323, que incluyen Terminales, Gateways (GW), Gatekeepers (GK), Multipoint Controllers (MC), Multipoint Processors (MP) y Multipoint Control Units (MCU). También define los procedimientos y mensajes de control para que estos componentes se comuniquen (International Telecommunication Union, Nov. 2000a).

Las Terminales son los componentes que proveen la capacidad de transmitir audio y video punto a punto o en conferencia múltiple con otras terminales. Una computadora con micrófono, cámara y parlantes corriendo el programa Microsoft Netmeeting es un ejemplo de Terminal H.323. Los Gateways sirven para interconectar la red de paquetes sin conexión con otras redes, como por ejemplo PSTN o ISDN y traducir los formatos de la comunicación multimedia de una a otra red. Los Gatekeepers proveen funciones de control de admisión y traducción de direcciones, además de participar en el inicio de las sesiones. Los MCs, MPs y MCUs dan soporte para hacer conferencia múltiple.

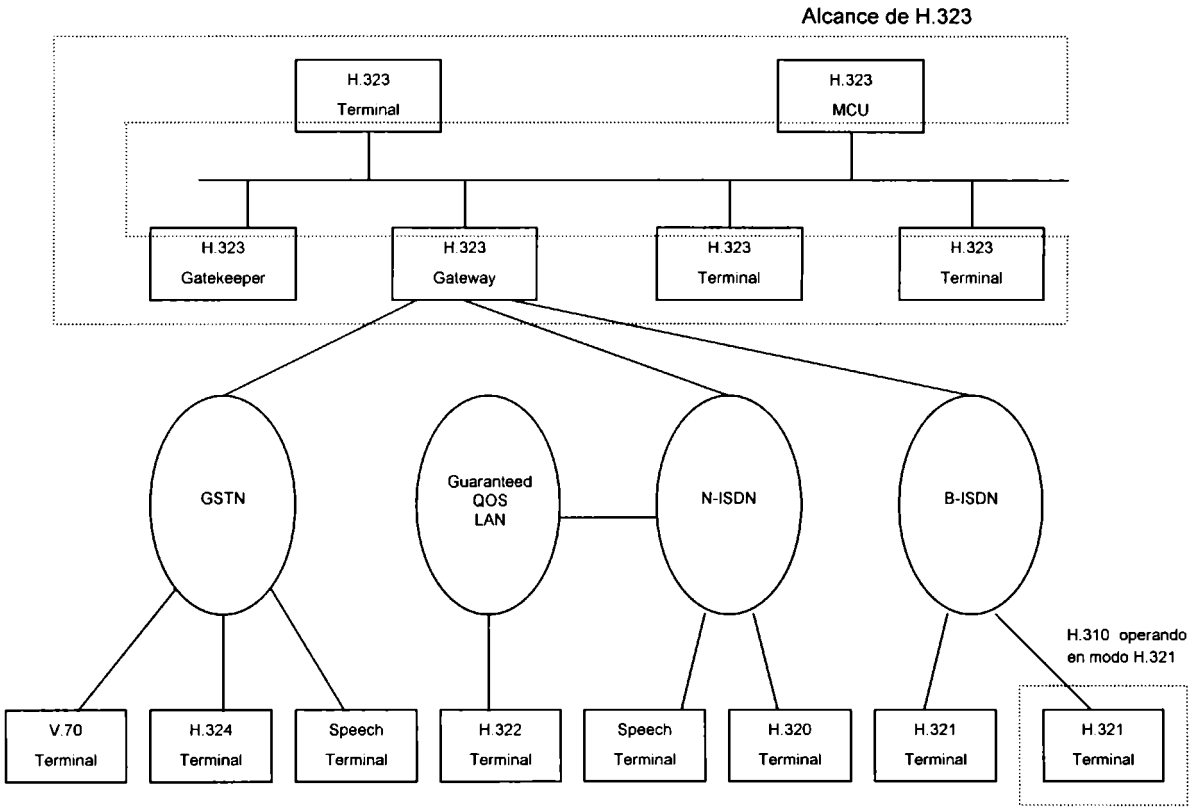



Figura 2-3: Interoperatividad de terminales H.323 (ITU, Nov.2000, secc. 1)

### 2.2.2 Entidades H.323

### 2.2.3 Terminales

Una Terminal H.323 está compuesta por los elementos que se muestran a continuación:

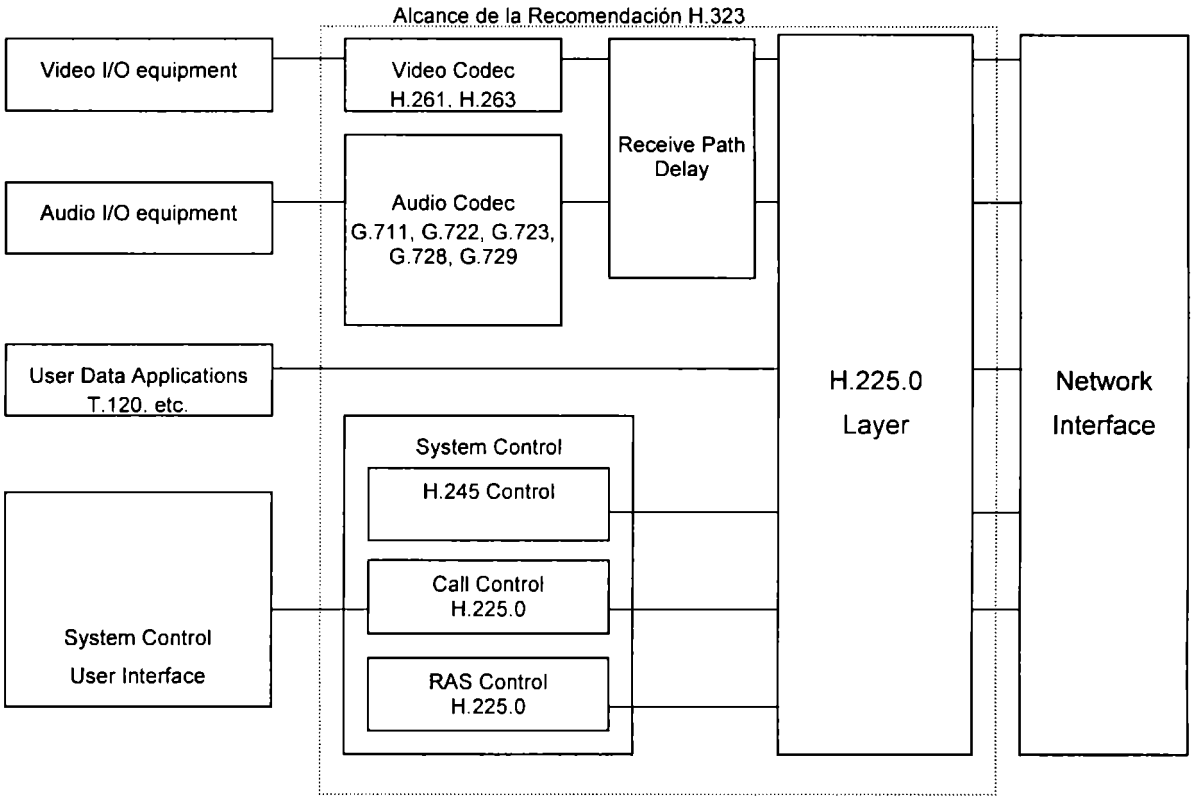



Figura 2-4: Elementos de un terminal H.323

En la figura se distinguen dos zonas específicas: la vinculada a los datos en la parte superior, y la vinculada al establecimiento y control de llamada en la parte inferior.

En la zona de datos, los audio y video codecs mencionados codifican la señal que reciben de los dispositivos respectivos y la envían mediante RTP (en la figura, dentro de H.225.0 Layer) hacia la red. Lo propio hacen con la recepción y decodificación de datos, que envían a dispositivos de salida. Entremedio de los codecs y H.225.0 se encuentra el Receive Path Delay, que hace de buffer para sincronización de paquetes y entre distintas señales que deben viajar juntas (por ejemplo, sincronizar la imagen con el sonido). El T.120 Data channel se establece en el momento del establecimiento de una conexión H.323. Se utiliza para enviar archivos, hacer acceso a base de datos y demás mientras una conversación o conferencia está en curso.

En la zona de establecimiento y control de llamada, se distinguen el Call Control H.225.0 que se usa para iniciar una llamada, el Register, Admission and Status (RAS) Channel (o RAS Control) que se usa en la comunicación con el Gatekeeper, y el H.245 Control usado en el intercambio de capacidades durante una comunicación, en el mantenimiento de canales lógicos y mensajes de control de flujo entre otros.

### 2.2.4 Gateway

Un Gateway (GW) es un punto extremo de un sistema H.323 que realiza la translación de una transmisión entre la red de paquetes y otras redes de conmutación de circuitos (Switched Circuit Networks o SCN), como son la red telefónica conmutada (Public Switched Telephone Network o PSTN) o ISDN. Esa translación incluye los formatos de transmisión y los

procedimientos de comunicación, y se realiza en ambas direcciones, esto es, desde la red hacia la SCN o desde la SCN hacia la red.

La PSTN está formada por una red de circuitos que transmiten los streams de audio y por una red paralela de control y señalización denominada SS7 (Signalling System 7).

El GW debe hacer la conversión de los formatos de transmisión, de los procedimientos y señalización del establecimiento de la llamada y de los procedimientos y señalización del control de llamada. En particular, debe convertir los mensajes H.245 en H.242 (el homónimo para SCN), y de H.225.0 Call signalling en los estándares el sistema de señalización de SCN, Q.931, Q.2931 y otras.

Hay distintos modelos de descomposición de un Gateway, según cómo se presten las funciones, pero todos siguen el mismo modelo genérico:

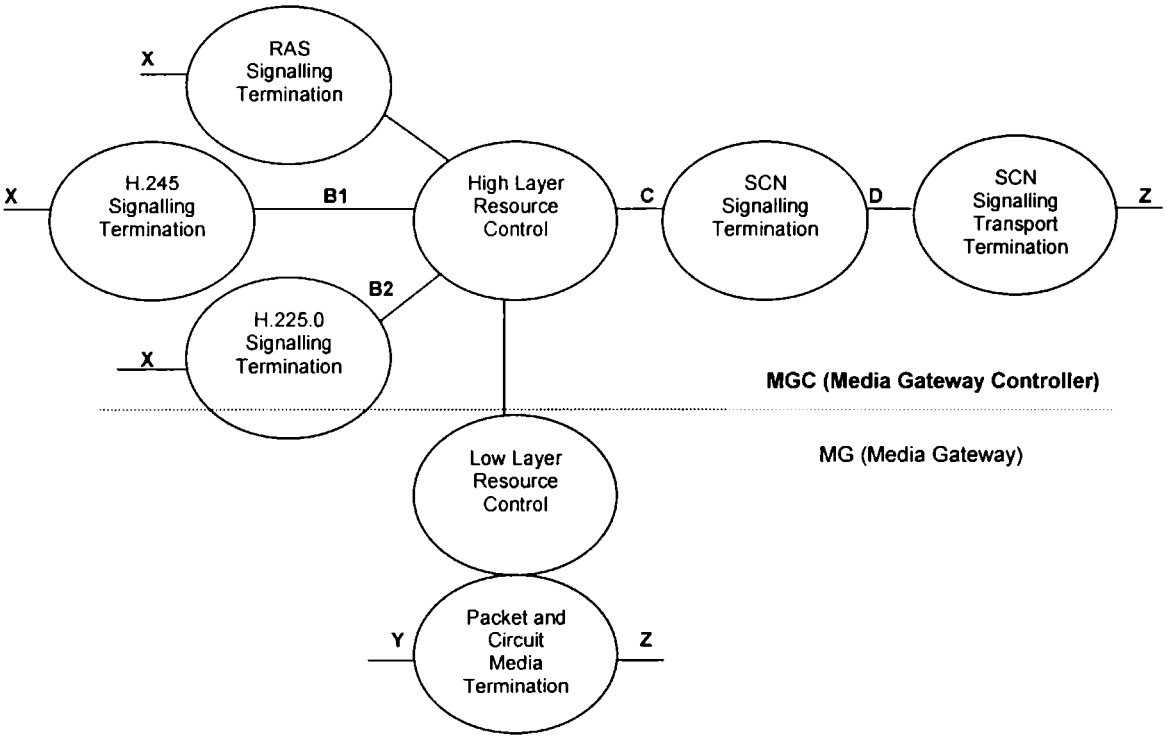



Figura 2-5: Arquitectura de un gateway

Las funciones de la parte superior de la figura se agrupan en el Media Gateway Controller (MGC) mientras que las de la parte inferior forman el Media Gateway (MG).

Las funciones del MGC son:

- intercambiar los mensajes RAS H.225.0 con el Gatekeeper
- manejar la interfaz de señalización SS7
- manejar la interfaz de señalización H.323

El MG se encarga de:

- hacer de terminal de la red de paquetes
- hacer de terminal de la SCN
- opcionalmente puede manejar señalización H.323 y señalización de SCN.

La Recomendación H.323 explica todos los paradigmas de división de funciones que un GW deber realizar entre un MG y un MGC, y si estas son unidades físicamente separadas o

no. En General, el MG se encarga de manejar los recursos de bajo nivel, mientras que el MGC maneja los recursos de alto nivel.

### 2.2.5 Gatekeeper

El Gatekeeper es una entidad opcional en un sistema H.323. Proporciona servicios de control de llamada a los terminales en una Zona. Una Zona es precisamente, el conjunto de Terminales, Gateways y MCUs gobernados por un Gatekeeper. Es totalmente independiente de la topología de la red, y puede estar formada por varios segmentos unidos por routers. Un sólo Gatekeeper puede estar presente en una zona.

El Gatekeeper provee los siguientes servicios:

- Traducción de direcciones: el Gatekeeper encuentra la dirección de transporte a partir de un alias.
- Control de Admisión (o registración): mediante mensajes ARQ/ACF/ARJ, autoriza el acceso a la red de acuerdo a diferentes criterios.
- Control de Bandwidth: mediante mensajes BRQ/BCF/BRJ, trabaja en el control de que no se exceda el ancho de banda utilizado.
- Management de Zona: estas funciones deben ser provistas a Terminales, Gateways y MCUs que se registren con el Gatekeeper.
- Opcionalmente puede brindar servicios de directorio, reserva de bandwidth, autorización de llamada, control de señalización de llamada entre otros.

Obligatoriamente, el Gatekeeper debe comprender los mensajes mencionados, aunque no está obligado a actuar de una determinada manera. Puede simplemente contestar que sí a todos los requerimientos sin ejercer control de ningún tipo.

### 2.2.6 Multipoint Controller

El Multipoint Controller provee funciones de control para soportar conferencias entre tres o más puntos extremos. Lleva registro de las capacidades que poseen los extremos para recibir y establece el Selected Communication Mode (SCM) que es el modo de comunicación que todos los puntos extremos adoptarán. Puede ocurrir que el MC defina dos SCMs, y que un grupo de nodos se comunique de acuerdo a uno y otro grupo al otro. Por ejemplo, si un grupo soporta video y otro no, puede transmitirse video sólo a aquellos que lo soportan. Además puede cambiar el o los SCMs de acuerdo a quiénes se sumen o abandonen la conferencia.

El Multipoint Controller usa mensajes H.245 para intercambiar las capacidades con los puntos extremos. También define el modo de conferencia, esto es, si se hará centralizada o descentralizada.

El MC puede ubicarse sólo en una MCU, dentro de un Gatekeeper, dentro de un Gateway, o en una Terminal.

### 2.2.7 Multipoint Processor

El MP recibe los streams de audio y video de los puntos extremos de una conferencia múltiple, los procesa, y los retorna a esos puntos. El procesamiento involucra el mezclado o switching de señales de audio y video. Mezclado significa juntar las señales de todos los participantes de la conferencia en una sola, y switching significa transmitir una señal por vez.

El mezclado de audio puede incluir filtrar ruidos y atenuar señales molestas. Si lo que se hace en cambio es switching de audio, se toma el canal de acuerdo a cómo se decida en la implementación.

El mezclado de video significa poner todas las imágenes en una matriz que divida la pantalla. En el switching de video se puede pasar cada imagen en intervalos de tiempo, o bien seguir a quien tenga uso de la palabra en la conferencia.

El MP puede hacer también conversión de formatos. Puede ubicarse en un Gateway o en una MCU.

### 2.2.8 Multipoint Control Unit

Una MCU es un punto extremo que provee soporte de conferencias múltiples. Consiste en un MC y tipicamente uno o varios MPs. Se comunica mediante mensajes H.245, y puede soportar conferencias múltiples centralizadas o descentralizadas.

Una MCU puede estar junto con un Gatekeeper o un Gateway, pero su funcionalidad es totalmente independiente, sólo comparten físicamente un gabinete.

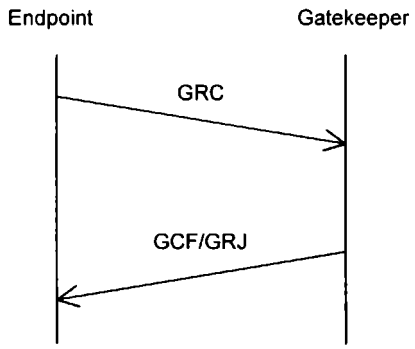
### 2.2.9 Operación en H.323

Una comunicación entre dos Terminales H.323 se divide en 5 fases: Establecimiento de llamada, Intercambio de facilidades (capabilities), Establecimiento de la comunicación audiovisual, Servicios de llamada y Terminación.

En todas estas fases se usan mensajes definidos en detalle en los protocolos H.225.0 y H.245 para llevar a cabo las funciones que se requieren. RTP y RTCP están incluidos en H.225.0 como un anexo (ITU, Nov. 2000b, pág. 101). En general, la información de control viajará por canales confiables y los streams de audio y video por canales no confiables, según puede verse en la figura 2.1.

En H.323 el destino de los paquetes se identifica por el par dirección de red y el identificador TSAP (Transport Service Access Point), según el esquema de direccionamiento usado en el modelo OSI. En un entorno TCP/IP, la dirección de red será la dirección IP y el identificador TSAP será el port number de TCP o UDP.

Un paso previo a establecer una comunicación es definir si hay o no un Gatekeeper en cual registrarse. Esto se hace mediante el envío de mensajes H.225.0 en el canal RAS:




GCF: Gatekeeper Request

GCF/GRJ: Gatekeeper Confirm / Reject

Figura 2-6: Descubrimiento de Gatekeeper.

Luego la terminal y el Gatekeeper se intercambian mensajes de Registration Request (RRQ), que el Gatekeeper contesta mediante Registration Confirm o Reject (RCF / RRJ). Para finalizar las operaciones, cualquiera de las partes envía un Unregister Request (URQ) respondido por un Unregister Confirm o Reject (UCF / URJ). También pueden intercambiar otros mensajes como Location Request (LRQ) para determinar la información de contacto, o Admission Request (ARQ) para solicitar un determinado ancho de banda.

A continuación se describen las cinco fases de una llamada H.323:

- Fase A: Call Setup

Existen varias alternativas de establecer una conexión, de acuerdo a que existan o no Gatekeepers a los cuales los terminales se registran. Se presentarán dos modelos representativos; el resto puede deducirse de estos dos:

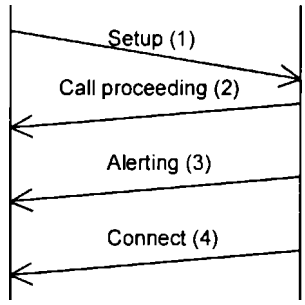



Figura 2-7: Call setup básico, sin Gatekeepers

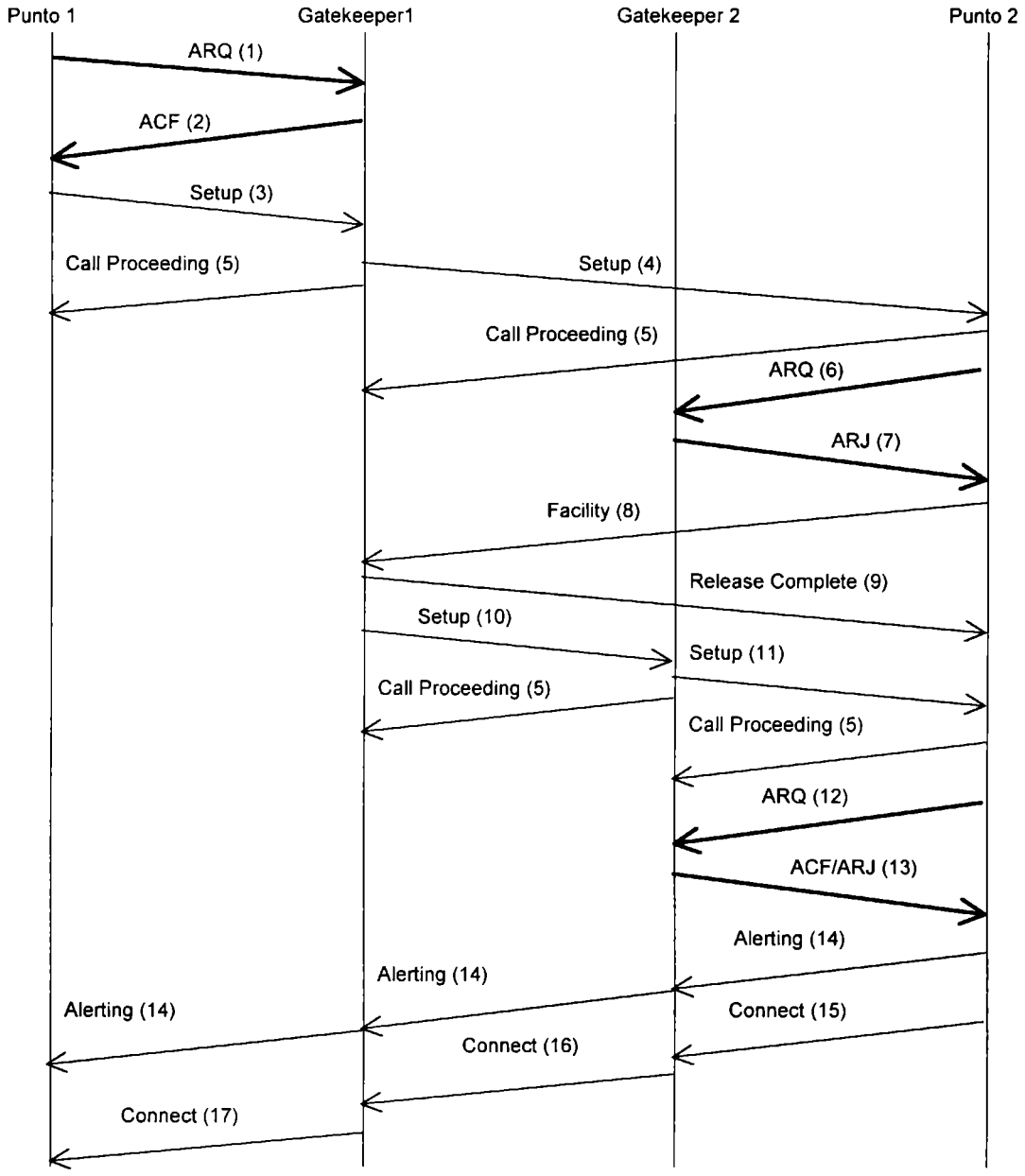



Figura 2-8: Call setup con registro en dos Gatekeepers y ruteo de señalización a través de ambos Gatekeepers.

Los mensajes relativos al establecimiento de la llamada están definidos en H.225.0 y son enviados a través del Call Control Channel (ver figura 2.4). Los relativos al registro con el Gatekeeper también son mensajes H.225.0 y son enviados a través del RAS channel.

El Gatekeeper puede estar o no presente, y puede intervenir o no en el ruteo de la señalización de la llamada, a su elección. En la figura 2.8 se muestra el caso más completo, donde existen Gatekeepers en ambos puntos, y participan en el ruteo de la señalización de la llamada:

- El punto 1 inicia la secuencia ARQ / ACF con su Gatekeeper. En el ACF, el Gatekeeper le informa al punto la dirección de transporte para el Call Signalling Channel, que es la suya.

- Luego el punto 1 envía un mensaje de Setup (3) a esa dirección (o sea, a su Gatekeeper 1)
- El Gatekeeper 1 envía el mensaje de Setup (4) al punto 2, quien devuelve una señal de Call proceeding (5) para avisar que ya recibió la petición de llamada. Previamente ese Call Proceeding también fue enviado por el Gatekeeper 1 al punto 1.
- El punto 2 inicia su registro con su Gatekeeper 2. En este caso, el Gatekeeper 2 contesta ARJ (Admission Reject) con su dirección de transporte para el Call Signalling Channel y la indicación que la llamada va a ser administrada por él.
- El punto 2 entonces informa de este cambio al Gatekepeer 1 mediante el mensaje Facility (8), y se cierra con un Release Complete (9). A partir de aquí, ambos Gatekeepers se comunicarán la señalización de la llamada.
- El Gatekeeper 1 redirige el Setup (10) al Gatekeeper 2, quien lo envía al punto 2.
- El punto 2 comienza la secuencia ARQ / ACF.
- Si es aceptada la llamada, se envía un Alerting (14) que emula el tono que recibimos en el auricular del teléfono mientras el otro extremo está sonando.
- Una vez que la llamada es atendida por el usuario del punto 2, se envía un mensaje Connect (15, 16 y 17). En el mensaje de Connect, el punto 2 envía la dirección de transporte del Control Channel H.245 que se usará en la siguiente fase.

- Fase B: Intercambio de facilidades (capabilities) del inicio

En esta fase se requiere abrir el H.245 Control Channel, para intercambiar las facilidades (capabilities) con que cuentan ambos puntos extremos, esto es, verificar si ambos tienen video, qué ancho de banda requieren y demás.

Este intercambio se hace mediante el envío de un mensaje H.245 terminalCapabilitySet, que deberá ser el primer mensaje enviado de la serie. Luego podrán enviarse otros mensajes H.245.

A continuación del intercambio de facilidades, debe determinarse la condición Master / Slave de cada punto en la comunicación, mediante mensajes H.245 masterSlaveDetermination/Ack.

En esta fase también se determina si estamos dentro de una conferencia múltiple. Si el intercambio de mensajes de esta fase es satisfactorio, se puede pasar a la fase siguiente. Sino, la comunicación se corta de acuerdo a la fase E.

- Fase C: Establecimiento de la comunicación audiovisual

Aquí se abre un canal lógico H.245 por cada stream de datos a enviar. La información de control se envía a través de ese canal lógico, indicando qué protocolo de transporte usan los datos que se envían, típicamente RTP/RTCP. Este canal lógico envía la dirección TSAP del canal de control RTCP. El canal lógico usa un protocolo de transporte confiable, como TCP.

Los datos se envían entre puntos en streams RTP, sobre un protocolo de transporte no confiable como UDP. Dentro del paquete RTP, los medios se codifican de acuerdo a algún codec que se negocia en la fase B. Los codecs se analizan en 2.4.

- Fase D: Servicios de llamada

Durante la llamada pueden requerirse cambios en las facilidades que fueron establecidas en la fase B. En esta fase, que puede resultar innecesaria, se realizan cambios

de ancho de banda, control de estado de los puntos extremos, expansión de una conferencia múltiple, o una pausa en una comunicación.

- Fase E: Terminación de llamada

  La terminación de la llamada consta de:

  - discontinuar la transmisión de video y/o audio y/o datos.
  - transmitir y esperar un mensaje H.245 endSessionCommand por el canal de control por cada canal de control abierto
  - cerrar el Call Signalling Channel mediante un mensaje Release Complete.

Cualquiera de los puntos extremos puede terminar una llamada, iniciando los procesos de terminación.

## 2.3 Funcionamiento de SIP/IETF

Si bien en mucha bibliografía se menciona el sistema como SIP, emulando lo que sucede con H.323, SIP por sí solo no serviría para completar una comunicación multimedia. El título refiere al SIP/IETF para englobar al conjunto de protocolos desarrollados por la IETF que se usan en una comunicación multimedia sobre una red IP, incluyendo SIP, RTP, RTCP, SDP, MEGACO, RSVP y otros. Por ejemplo, el transporte de los streams de audio y video se hacen mediante RTP, y SIP no interviene en esa fase, ya intervino para contactar al usuario, entregar la dirección IP, negociar el protocolo de transporte, el puerto y el codec a usar (junto con el SDP).

SIP fue elegido por el 3GPP como el protocolo para celulares de 3ra. generación o GSM. El 3GPP es un acuerdo de colaboración entre importantes organizaciones que definen estándares para lograr criterios comunes en el desarrollo de la tecnología celular de 3ra. generación (Teitelbaum, 2003).

### 2.3.1 El protocolo SIP por dentro

SIP es un protocolo de nivel de aplicación que lo único que realiza es el establecimiento de una sesión entre dos o más participantes, para descubrir la presencia de ellos. El resto de las funciones para que una conexión multimedia tenga lugar las realizan otros protocolos. SIP está basado en TCP/IP y tiene la misma estructura de mensajes de otros protocolos que corren sobre TCP/IP como HTTP o SMTP. A diferencia de lo que sucede con H.323, no requiere de otros protocolos para funcionar. Tampoco es de uso exclusivo para telefonía; hay otras aplicaciones, como juegos interactivos o educación a distancia que utilizan SIP para establecer una sesión.

SIP no define Gateways ni MCUs. Hay algunas implementaciones comerciales que refieren a Gateways SIP análogamente a como refieren Gateways H.323, pero esta referencia es errónea. El traslado de señales entre la red IP y la SCN se hace según el protocolo Megaco/H.248, aunque un dispositivo de este tipo necesariamente debe incluir un User Agent, que es un componente de SIP, como se verá más adelante.

Tampoco hace reserva de recursos; esto se hace mediante RSVP, y ni siquiera define qué características tendrá la sesión; SDP se usa para informar al extremo cómo será.

SIP está formado por un conjunto de primitivas básicas, con las cuáles pueden implementarse otros servicios, por ejemplo, redireccionar llamados no conocidos a una secretaria o responder con una página web si no está disponible el usuario.

Una sesión SIP se establece entre dos User Agents mediante el envío de mensajes definidos en la RFC de SIP, y pueden intervenir o no otros componentes como son los proxy servers, redirect servers, registar servers y location servers.

### 2.3.2 Entidades SIP

El *User Agent* es la entidad básica de SIP, o terminal SIP. Está compuesto por dos partes, el User Agent Client (UAC) y el User Agent Server (UAS). El User Agent Client será quien genere el pedido de una sesión. El User Agent Server está escuchando pedidos de otros UACs y es el encargado de responder a esos pedidos.

Además de los User Agents, SIP define servidores que cumplen distintas funciones dentro de una red. El *Proxy server* es el encargado de rutear mensajes SIP de requerimiento (SIP Requests) hacia un UAS, y mensajes SIP de respuesta (SIP Responses) hacia un UAC. Un mensaje puede atravesar varios SIP Proxies, cada uno de los cuales tomará decisiones de ruteo hasta el destino final. No deben confundirse las decisiones de ruteo de SIP con un protocolo de ruteo de paquetes. Por ejemplo, el SIP Proxy recibe de un UAC un mensaje SIP enviado a cae@iptel.org y lo reenvía al servidor SIP del dominio iptel.org. En el camino de vuelta, el Proxy recibirá la respuesta lo que le permite controlar la sesión. Los proxies que controlan la sesión se denominan statefull proxies, mientras que los que se limitan a rutear mensajes, pero no guardan registro de sesiones se denominan stateless.

El Proxy se vale del Location Service para descubrir la dirección IP, protocolo de transporte y puerto, que son los datos necesarios para rutear un mensaje hacia el Proxy del destino final.

No siempre el Proxy rutea los mensajes hacia el destino final. En ocasiones, es conveniente redireccionar la llamada de sesión. Esto lo hace el Redirect server (que se encuentra en general en cohabitación con el Proxy server) mediante la devolución al UAC originante de un mensaje de error que incluye la nueva dirección de la persona a la que se está llamando. Todo este funcionamiento se ilustra en la sección siguiente.

El Location Service trabaja sobre la base de una tabla que contiene usuarios y su ubicación específica. Esa tabla es mantenida por el Registar server, y se forma a partir del envío por parte de cada usuario de sus datos mediante mensajes *Register* a ese servidor. El Registrar también puede cohabitar con otros servidores SIP.

Cuando se requiere conectar una red IP a otra red como la SCN, al igual que en H.323 se necesitará un Gateway. El protocolo Megaco es el protocolo desarrollado por la IETF para la operación de gateways.

### 2.3.3 Señalización SIP

La comunicación entre dos entidades SIP se realiza por medio de mensajes de *request* y *response*. Cuando se genera un ida y vuelta de mensajes de request y response se establece un diálogo, que deviene en sesión una vez que hay acuerdo entre las partes.

Los mensajes de *request* se agrupan en:

INVITE: Cuando una entidad desea comenzar una sesión, envía este mensaje que contiene una descripción de la sesión solicitada, incluyendo la dirección del usuario invitado.

ACK: Es la confirmación de que una solicitud INVITE ha sido aceptada.

OPTIONS: Se usa en la negociación de capabilities.

BYE: Se usa para solicitar cancelación de una sesión.

CANCEL: Elimina un request pendiente.

REGISTER: Se usa para enviar los datos de un usuario al servidor de registro.

Las posibles respuestas tienen códigos con el mismo formato que en HTTP, y son:

1xx, Información: avisa sobre la recepción de un request mientras está siendo procesado.

2xx, Exito: confirma que un request ha sido recibido, evaluado y aceptado.

3xx, Redirección: avisa que se necesita otra acción para procesar el request.

4xx, Error en el cliente: rechaza el request debido a errores de sintaxis.

5xx, Error en el server: rechaza un request válido por problemas internos del servidor.

6xx, Fallo general: el request no se puede procesar de ninguna manera.

La RFC 3261 (Rosenberg y otros, 2002) especifica con detalle el formato de cada mensaje, y los campos que debe y puede contener cada uno. Cada campo se denomina *header SIP*. En los ejemplos siguientes se muestra el conjunto mínimo de headers requeridos:

Mensaje INVITE:

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

Figura 2-9: Mensaje Request de INVITE.

El primer renglón especifica el nombre del mensaje (INVITE) y la dirección del destinatario. Las direcciones sip se denominan SIP o SIPS URI, por SIP o Secure SIP Uniform Resource Identification. Puede verse que son similares a una dirección de e-mail. Finalmente se incluye el protocolo y la versión.

Via contiene la dirección en la cuál el usuario del request espera recibir respuesta junto con un parámetro que identifica la transacción.

To contiene el nombre literal del destinatario y la dirección a donde originalmente se envía el mensaje.

From contiene el nombre literal y la dirección del remitente. Al final se agrega un tag formado por un string aleatorio que también se usa para identificación.

Call-ID contiene un identificador global para esta llamada. Este identificador se obtiene de combinar un string aleatorio con la dirección IP o el nombre del host que llama. De esta forma,

con el To, el tag del From y este Call-ID se establece una relación entre llamante y llamado que es lo que se denomina diálogo.

CSeq es la secuencia de comando, formada por un entero y un nombre de método request. Sucesivos requests dentro de un diálogo verán incrementados en uno el CSeq.

Contact muestra la dirección que representa la ruta directa para contactar al llamante. El Via header es usado como destino de las respuestas, mientras que este header se usa como destino de futuros requests.

Max-Forwards es el límite del número de saltos permitidos.

Content-type contiene la descripción del cuerpo del mensaje.

Content-lenght contiene el largo del cuerpo del mensaje medido en bytes.

Puede observarse que los detalles de la sesión, como son el tipo de medio transportado, codec usado, protocolo de transporte de medios o el ratio de muestras (sampling rate) usado no son parte del mensaje SIP. Para transmitir toda esa información se usa el SDP (Session Description Protocol), otro protocolo de la familia IETF. El protocolo SDP se transmite en el cuerpo del mensaje SIP; en el encabezado sólo transmite datos referidos a la sesión.

Los mensajes de respuesta tienen un formato parecido al mostrado anteriormente, sólo que se agrega la ruta que siguió el mensaje para llegar a destino, como se verá en la sección siguiente. El mensaje se muestra a continuación:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bKnashds8;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com; branch=z9hG4bK77ef4c2312983.1;
received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com; branch=z9hG4bK776asdhds ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>; tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131
```

Figura 2-10: Mensaje de OK.

La primera linea contiene el código de respuesta y la razón de la respuesta. En este caso no es instructiva (OK) pero en caso de errores es importante la razón más allá del código retornado.

Los tres headers Via siguientes muestran el camino seguido por el mensaje OK. Aquí puede verse que mensaje fue de pc33.atlanta.com al proxy de atlanta.com que es bigbox3.site3.atlanta.com y de allí al proxy de biloxi.com que es server10.biloxi.com. Futuros requests deberán seguir esta ruta.

El resto de los headers son copiados del header de INVITE.

### 2.3.4 SDP

El SDP (Session Description Protocol, RFC 2327) define un formato para describir una sesión multimedia, y se utiliza cuando se anuncia o se inicia una sesión. La descripción de la sesión se refiere a la información relacionada con los streams de medios que se van a transmitir, que permite a los usuarios participar de una sesión.

Este protocolo no contiene encabezados, sólo define letras clave a las que se asignan valores con la descripción de la sesión. Las más importantes (Handley y otros, 1998) se muestran a continuación:

Descripción de sesión

v= versión

o= identificación del owner.

s= nombre de la sesión.

c= información de la conexión, versión y dirección del IP

Descripción de tiempo

t= tiempo que lleva activa la sesión.

Descripción del medio

m= nombre del medio y dirección de transporte (protocolo y puerto)

b= bandwidth

k= clave de encriptación

a= atributos del medio (codificación usada)

En los ejemplos del capítulo 3 se ilustra el uso de SDP.

### 2.3.5 Protocolos de control de gateways: Megaco y MGCP

La arquitectura interna de un gateway fue definida por la ETSI (European Telecommunications Standards Institute), otra de las entidades que trabajan sobre estándares, en colaboración con la ITU y con la IETF. El modelo de gateway propuesto divide al mismo en tres funciones como se muestra en la figura:

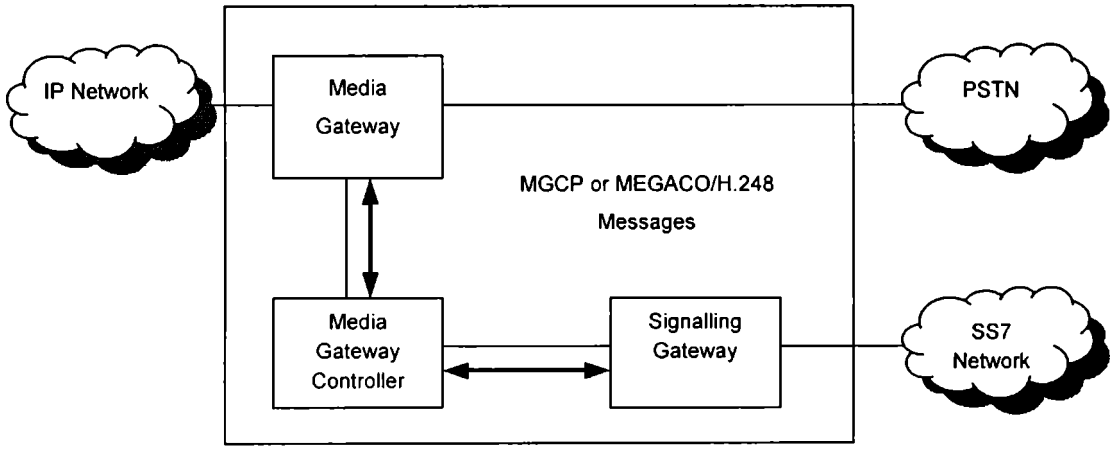



Figura 2-11: Arquitectura de un Gateway según ETSI (Miller, 2002, pág. 238)

El Media Gateway es el encargado de la conversión de la información transportada en paquetes RTP/UDP/IP en una señal que entienda la PSTN, por ejemplo, voz codificada PCM. El Signaling Gateway es el encargado de convertir la señalización del lado de la red de paquetes en señales SS7, entendibles por la PSTN. Puede observarse la analogía existente entre este modelo de Gateway y el Gateway H.323 mostrado en 2.2.2.2.

El protocolo MGCP (Media Gateway Control Protocol) fue desarrollado por la IETF para definir los comandos y parámetros que se pasan entre el MGC (Media Gateway Controller), el MG (Media Gateway) y el SG (Signaling Gateway). El MGC es nombrado como Call Agent en MGCP; de él se distribuyen señales por un lado hacia el SG y medios por otro hacia el MG. Si necesitamos un Gateway para una red SIP, el MGC deberá comprender la señalización SIP y SDP. Aquí puede verse claramente la independencia que SIP tiene de cualquier Gateway entre la red IP y otras SCNs. El MGCP es el protocolo más ampliamente difundido entre los productos comerciales disponibles. Por ejemplo, en IPTel.org puede encontrarse una lista de Gateways comerciales catalogados por su soporte para SIP/H.323/MGCP.

Como se menciona anteriormente, el protocolo Megaco fue desarrollado por la IETF en conjunto con la ITU, para la comunicación dentro de un Gateway. Básicamente, hace la misma tarea que MGCP pero con un enfoque conceptual distinto. Usa la misma descomposición en funciones mostrada en la figura 2.11 pero hace una abstracción distinta del modelo de conexión: MGCP aplica los comandos sobre *conexiones* mientras que Megaco lo hace sobre *Terminaciones* relativas a un *Contexto*. Esta diferencia los hace distintos e incompatibles; además Megaco es independiente del protocolo de transporte de la red de paquetes, mientras que MGCP es TCP/UDP/IP dependiente. Otra ventaja de Megaco es que tiene también el soporte de la ITU, aunque MGCP es el más elegido en las aplicaciones comerciales.

### 2.3.6 Operación para una comunicación multimedia en SIP/IETF

Previo a cualquier operación, cada usuario realiza su registro como se muestra a continuación:

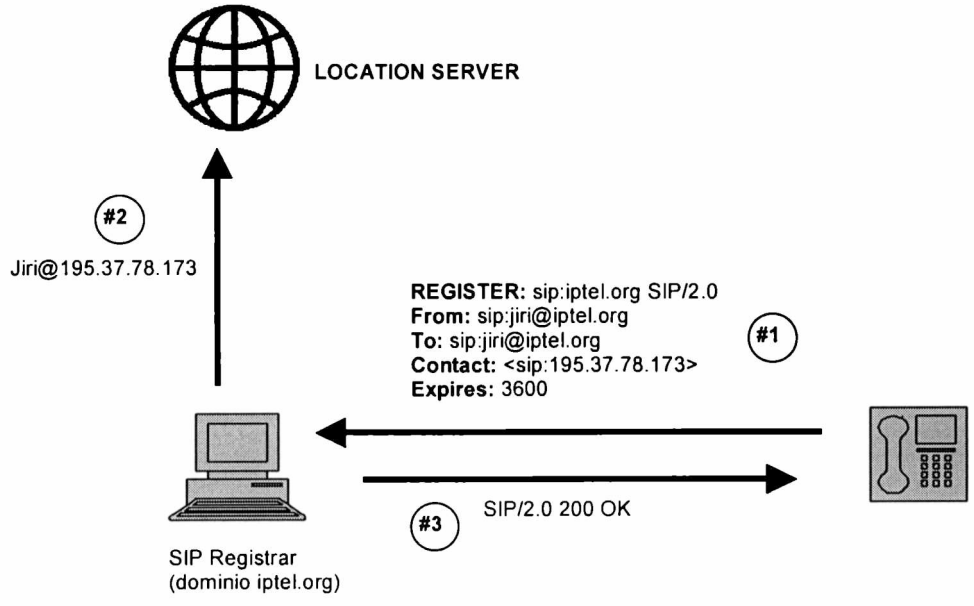



Figura 2-12: SIP. Registro de usuario

1. El usuario envía un mensaje REGISTER al servidor de Registro.
2. El servidor lo guarda en la tabla que será consultada por el Location Server
3. Luego devuelve al usuario el OK del REGISTER.

Este proceso se realiza sin la real intervención del usuario. Por ejemplo, si tenemos un softphone SIP, el registro se hará automáticamente cuando el software sea abierto.

La operación básica de una comunicación entre dos usuarios que pretenden comunicarse mediante User Agents SIP se muestra en la siguiente figura:

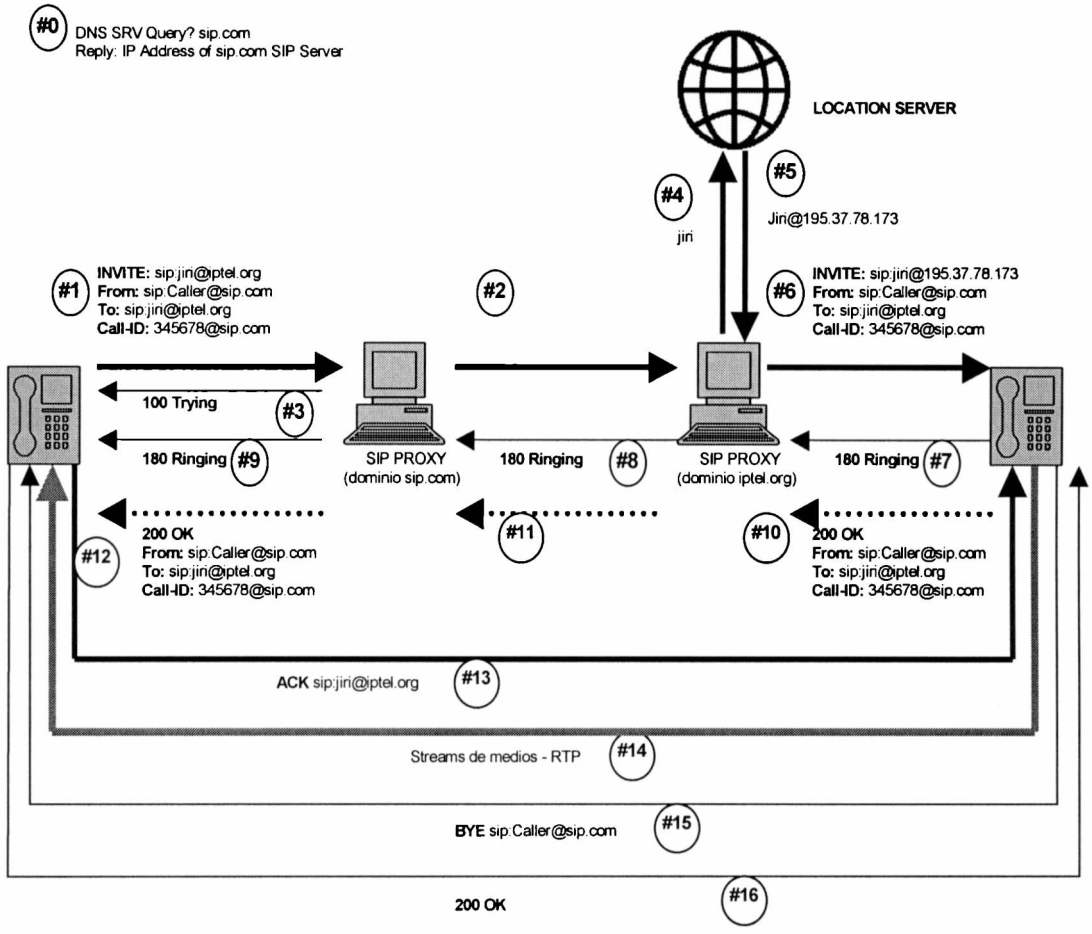



Figura 2-13: Comunicación SIP/IETF (versión modificada de la extraída de Sisalem y otros, pág. 20)

0. El usuario consulta al DNS por el proxy server correspondiente a su dominio. El descubrimiento del proxy server se explica en detalle en la RFC 3262.
1. Envía un INVITE al proxy server correspondiente a su dominio (proxy.sip.com), con destino final el usuario jiri@iptel.org. En este INVITE se incluye información de la sesión, de acuerdo al protocolo SDP.

2. El proxy proxy.sip.com envía el INVITE al proxy correspondiente del dominio iptel.org.
3. También envía un TRYING al usuario llamante para avisar que recibió la solicitud y que está en proceso.
4. El proxy de iptel.org busca la ubicación del usuario jiri en el Location Server.
5. El Location server retorna la dirección IP del usuario jiri.
6. El proxy de iptel.org envía el INVITE al usuario. Aunque no se muestra en la figura, ambos proxies intervinientes informan en headers Via que el mensaje pasó por ellos.
7. El usuario llamado avisa que está sonando con RINGING.
8. Los proxies envían el RINGING al usuario llamante. En un sistema tradicional telefónico, esto emula el tono de llamada que emite el auricular.
9. Idem anterior
10. El usuario llamado atiende. Se envía el OK hacia el usuario llamante incluyendo la información SDP de la sesión abierta.
11. Idem anterior hacia el proxy.
12. Idem anterior hacia el usuario llamante.
13. El usuario llamante envía el ACK indicando que está todo listo para iniciar la conferencia.
14. Comienza la transmisión de streams de medios. La señal se codifica de acuerdo al codec elegido (comunicado entre las partes como información de la sesión) y se agrega a un paquete RTP, que se envía de un lado al otro usualmente por UDP. En esta etapa la sesión ya está establecida, no se usan headers SIP.
15. Cuando la conferencia ha terminado, cualquiera de las partes comienza a cerrar la sesión (en este caso comienza el usuario llamado) enviando un mensaje BYE directamente al otro usuario.
16. El usuario requerido envía el OK a la finalización de la comunicación.

El caso de comunicación presentado es un caso típico donde no se existen ítems que salgan de lo normal. Por ejemplo, el usuario puede no atender la llamada, puede no tener encendido el dispositivo llamado, puede haber errores en la negociación de los codecs, y otras alternativas. Los diferentes códigos de respuesta son usados para cada uno de estos problemas.

Un usuario puede haberse mudado, y tal como sucede con la red telefónica convencional, el usuario llamante se notifica de esto mediante un mensaje. El Redirect Server es quien se encarga de enviar este mensaje como ha sido visto anteriormente.

El ejemplo trata la comunicación entre dos entidades o User Agents dentro de una red IP. En el caso de necesitar una conexión con otra red, por ejemplo la telefónica, el Proxy elegirá un Gateway (descripto en 2.3.5) como destino de la sesión SIP, en particular el User Agent dentro del Gateway. Para descubrir que una sesión tiene una terminación fuera de la red IP, el Location Server usa el protocolo Enum, que se muestra en la sección 2.5.2.

## 2.4 Detalle de los protocolos de transporte

### 2.4.1 RTP/RTCP

RTP (Real-time Transport Protocol) es un protocolo que brinda servicios de transporte de datos en tiempo real. Lo acompaña un protocolo de control, el RTCP (Real-time Transport Control Protocol). Estos servicios son:

- Identificación del tipo de payload
- Numeración de secuencia
- Timestamp
- Monitorización de la entrega

Un paquete RTP consta de un encabezado donde figuran diversos parámetros que sirven para prestar los servicios mencionados y en el cuerpo, o payload, lleva los datos que requieren ser transmitidos en tiempo real, usualmente datos de audio o video. Un stream de audio o video se divide en pequeñas fracciones, se encapsulan las fracciones en paquetes RTP, y se envían al destinatario. Durante una transmisión en tiempo real, se envían a intervalos regulares paquetes RTPC que incluyen datos estadísticos de la transmisión y datos de descripción de los participantes de la transmisión.

Es necesario destacar que RTP no provee ninguna garantía de QoS, como puede ser asegurar que los paquetes lleguen en un determinado tiempo; deja esas cuestiones a las capas inferiores de la red. El protocolo corre usualmente sobre UDP, aunque su definición es independiente de la capa de transporte empleada. De UDP usa las capacidades de multiplexado y checksum. El puerto UDP que se usa para transmitir paquetes RTP lo negocian las aplicaciones que se comunican, pero siempre debe ser par. El puerto RTPC es el número siguiente del puerto negociado.

La siguiente figura muestra el encabezado RTP. Del estudio del encabezado se deduce qué son y cómo se prestan los servicios mencionados anteriormente:

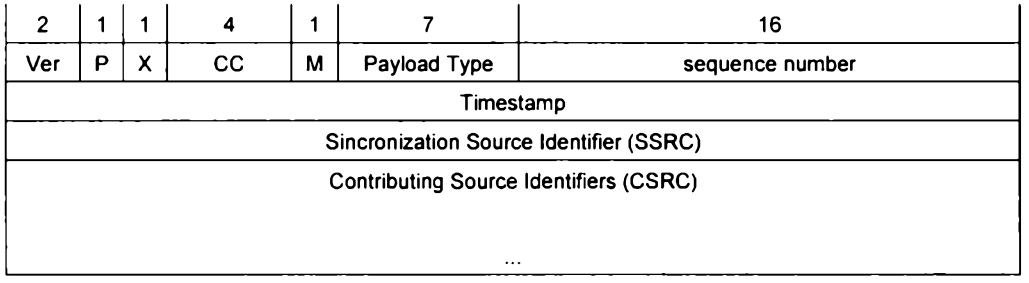



Figura 2-14: Encabezado RTP (Schulzrinne y otros, 2003, pág. 13)

Ver: Versión

P: Padding: indica si hay bytes de padding. El último byte tiene la cuenta de bytes de padding.

X: Extensión: indica si hay un header de extensión a continuación.

CC: CSRC count: número de CSRCs presentes.

M: Marker: el uso lo define un profile separado.

Payload Type: La identificación del *tipo de payload* es usada para conocer qué tipo de datos son transportados en un paquete, en qué forma están codificados. A la definición de RTP la acompaña la RFC 3551, que es un profile donde se especifica el listado de códigos de payload y su asociación con el formato del payload, usualmente el codec transportado, junto con el comportamiento que debe tenerse para ese tipo de payload. La RFC 3551 especifica payloads con control mínimo; cada nuevo tipo de payload o payloads que requieran tratamiento especial se especifican en RFCs separadas.

Sequence number: se incrementa en uno en cada paquete enviado. Con el número de secuencia se puede reconstruir la secuencia de paquetes y también establecer la posición de un paquete decodificado más tarde que otros. De esta forma se evita tener que decodificar los paquetes secuencialmente.

timestamp: es el instante en que es tomado el primer byte del payload. Se toma de un reloj que crece monótona y linearmente. Se usa para sincronización con otras fuentes de datos (ej: video, o audio de una conferencia múltiple) y cálculo de jitter.

SSRC: identifica la fuente de sincronización. Es elegido aleatoriamente por la fuente que genera los streams de paquetes RTP, y es única para todos los streams dentro de un espacio de tiempo y números de secuencia. El destinatario agrupa los streams por el SSRC para la reproducción (en el caso de audio o video).

CSRC: es una lista de SSRCs que identifican qué fuentes están mezcladas en el payload. Cuando un destinatario en una conferencia múltiple no puede recibir paquetes de todos los participantes por limitaciones de bandwidth, un mixer intermedio se encarga de juntar todas las fuentes en un sólo stream y lo envía a ese destinatario con la lista de todas las fuentes en este campo.

Audio y video deben transmitirse en sesiones RTP separadas, con distinto SSRC, a pesar de generarse en la misma fuente.

El tipo de payload posibilita codificar audio en varias calidades distintas y transmitir en distintos grupos ese audio. El usuario elige suscribirse y recibir el audio del grupo que más le conviene, como se menciona en 1.5.4.

El RTCP tiene las siguientes funciones (Schulzrinne y otros, 2003, pág 19-20):

1. Provee feedback de la calidad de la distribución de datos; esta es su función principal.

2. Transmite un identificador de nivel de transporte de los participantes en una conferencia, llamado canonical name o CName. Con este CName, más un timestamp que lleva el paquete RTCP, el destinatario puede identificar múltiples streams de una misma fuente y sincronizar audio y video.

3. Al llevar los datos de todos los participantes de una conferencia, puede adecuar los ratios de envío de acuerdo a la cantidad de participantes.

Los paquetes RTCP son de distintos tipos y se envían por lo general un conjunto de paquetes de varios tipos. Esos tipos son:

- SR (sender report): un usuario que envía datos en forma activa transmite y recibe estadísticas mediante SR. Las estadísticas incluyen cantidad de paquetes perdidos, round trip, jitter, timestampings entre otros datos.
- RR (receiver report): un usuario que no envía datos en forma activa pero participa de la conferencia envía estadísticas mediante RR.
- SDES (source description): un usuario envía su identificación, que incluye el Cname y otros datos, mediante SDES.

- BYE: un usuario indica final de la conferencia mediante el envío de un BYE.
- APP: específico para funciones de aplicaciones.

En 3.2 se ilustra el uso de paquetes RTCP.

### 2.4.2 Audio codecs

La información de audio se modula desde su fuente de origen usando distintos codecs, que varían en cuanto a la cantidad de muestras que toman, compresión usada, calidad esperada entre otras. Estos codecs fueron desarrollados por la ITU-T y se usan como estándares en una amplia variedad de aplicaciones. A continuación se muestran los más importantes con una breve descripción de sus características:

**G.711:** opera a 64 Kbps, es el de mejor calidad y más básico de todos, pero el más costoso. Usa Pulse Code Modulation (PCM), cada frame contiene 125 ms. de audio.

**G.722:** opera a 48, 56 o 64 Kbps.

**G. 723.1:** opera a 5.3 y 6.4 Kbps. Usa el Algebraic Code Excited Linear Prediction (ACELP) y produce frames de 30 ms. y un delay total de 37.5 ms. Es muy usado ya que fue adoptado por el IMTC (International Multimedia Teleconferencing Consortium) para su uso en VoIP.

**G.726:** opera entre 16 y 40 Kbps, produciendo un frame de 0,125 ms. y un delay de 0,125 ms.

**G.728:** opera a 16 Kbps, produciendo un frame de 0,625 ms. y un delay de 0,625 ms.

**G.729:** opera a 8 Kbps y produce un frame de 10 ms. con un delay de 15 ms. Usa el Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP). Fue pensado originalmente para ambientes wireless.

**G.729A:** es igual al anterior, sólo que usa una versión más simple de CS-ACELP. Fue adoptado por el IMTC para comunicaciones de voz y datos integradas.

La mayoría de los productos comerciales soporta estos codecs, especialmente el G.711 y el G.723.1.

### 2.4.3 Video codecs

Estos son los video codecs que definió la ITU-T:

**H.261:** opera en múltiplos de 64 Kbps ($p$ x 64, con $p$ entre 1 y 30), lo que da como resultado un bit rate entre 40 Kbps y 2 Mbps. Este codec es obligatorio para las entidades H.323.

**H.263:** se basa en el anterior, con un mecanismo de compresión adicional, y es optativo para las entidades H.323.

## 2.5 Detalle de otros protocolos involucrados en VoIP

En 2.1 se menciona que son importantes otros protocolos de soporte para llevar a cabo una comunicación multimedia en Internet. Los dos protocolos dentro de este grupo que resultan más significativos son RSVP y Enum. Como ya se ha mencionado, RSVP se usa para gestionar y administrar reserva de recursos de red a fin de asegurar la calidad de servicio. El otro se usa para emular en una red TCP/IP el sistema de numeración telefónica convencional.

### 2.5.1 RSVP

En 1.5.1 ya mostramos el funcionamiento del RSVP. Ahora nos detendremos en la semántica de los mensajes del protocolo.

RSVP define dos tipos de mensajes básicos:

- Reservation Request (Resv): se envían desde el destino hacia el origen para crear la ruta y la reserva de recursos en todos los nodos intermedios que recorre.
- Path: se envían desde el origen al destino luego de un mensaje Resv para recorrer el camino prescripto. Se guarda información en el mensaje de cada nodo recorrido.

Otros mensajes definidos por RSVP sirven para control y transmisión del estado de una reserva:

- PathErr: reporta al origen un error al procesar un mensaje Path.
- ResvErr: reporta al destino un error al procesar un mensaje Resv.
- PathTear: notifica la baja de un mensaje Path.
- ResvTear: notifica la baja de un mensaje Resv.
- ResvConf: confirmación del requerimiento de reserva.

Los mensajes RSVP pueden enviarse mediante UDP o bien directamente encapsulados en un paquete IP.

### 2.5.2 Enum

Los sistemas de telefonía tradicionales se identifican con números del tipo 54-221-422-2222, mientras que los dispositivos de VoIP lo hacen con direcciones IP. Para lograr traducir el número telefónico en direcciones IP se definió el proceso Enum (por electronic number).

Enum está especificado en la RFC 2916. El funcionamiento de Enum se basa en el DNS; cuando un usuario se intenta comunicar a un número telefónico, Enum consulta si ese número está asociado a una dirección IP. Si la asociación existe, le devuelve al usuario esa dirección, mientras que si no existe, supone que se trata de un número convencional y entonces redirige la llamada a la PSTN vía un gateway.

# Capítulo 3

# Análisis de la aplicación de Voz sobre IP versión 6

Hasta aquí se han visto conceptos generales del protocolo IPv6 y de Voz sobre IP. En este capítulo se analiza una aplicación de VoIP que soporta IPv6, para determinar si realmente esa aplicación usa en forma consistente las ventajas que IPv6 puede aportar a aplicaciones de tiempo real como las de voz y video, o si se limita a adecuar el tamaño de direcciones y algunos otros parámetros menores.

Se ha elegido para este análisis una aplicación llamada Bonephone. Ha sido desarrollada por el *iptel.org*, una organización dependiente del Instituto FRANHOUSER de origen alemán. Los detalles de la instalación de este software se mencionan en el anexo B. Una vez instalado en dos computadoras, se procedió a establecer una comunicación entre ambos programas por algunos minutos, mientras se realizaba una captura de paquetes que corrían por el segmento de red que unía ambas computadoras. Con esa captura, se pudo efectuar el análisis de la información transmitida, los protocolos utilizados y las opciones usadas en cada protocolo, para poder determinar como es el comportamiento del software en un entorno IPv6.

## 3.1 Análisis de paquetes transmitidos

La figura siguiente muestra la secuencia de paquetes enviados entre los dos sistemas durante la conversación hasta el corte final hecho por uno de los usuarios. Se han omitido paquetes que no son parte importante de esta conversación, y de todos los paquetes de datos RTP y RTCP (más de mil) sólo se muestran algunos pocos, que representan al resto, ya que todos comparten las mismas características y sólo cambia el área de Payload, o sea, los datos de la voz codificada.

Esta secuencia es una versión resumida de la secuencia presentada en 2.3.6 (figuras 12 y 13) ya que no estamos en presencia de proxies, ni redirect servers ni location servers. Sólo hay una comunicación directa entre ambas computadoras que se realiza a usuarios que se conocen y que se sabe cuál es su ubicación y disponibilidad.

| No. | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|
| 11 | 3.75 | 2002:c850:220b:: | 2002:c850:2220:: | SIP/SDP | Request: INVITE sip:root@[2002:c850:2220::], with session description |
| 16 | 4.67 | 2002:c850:2220:: | 2002:c850:220b:: | SIP | Status: 180 Ringing |
| 45 | 8.74 | fe80::2d0:9ff:fe... | 2002:c850:2220:: | ICMPv6 | Neighbor solicitation |
| 46 | 8.74 | 2002:c850:2220:: | fe80::2d0:9ff:fe... | ICMPv6 | Neighbor advertisement |
| 50 | 9.20 | 2002:c850:2220:: | 2002:c850:220b:: | SIP/SDP | Status: 200 OK, with session description |
| 52 | 9.61 | 2002:c850:220b:: | 2002:c850:2220:: | SIP | Request: ACK sip:root@[2002:c850:2220::] |
| 155 | 56.05 | 2002:c850:220b:: | 2002:c850:2220:: | RTCP | Receiver Report |
| 170 | 64.17 | 2002:c850:2220:: | 2002:c850:220b:: | RTCP | Receiver Report |
| 176 | 69.41 | 2002:c850:2220:: | 2002:c850:220b:: | RTP | Payload type=ITU-T G.711 PCMU, SSRC=808055695, Seq=2820, Time=986469228, Mark |
| 177 | 69.43 | 2002:c850:2220:: | 2002:c850:220b:: | RTP | Payload type=ITU-T G.711 PCMU, SSRC=808055695, Seq=2821, Time=986469388 |
| | ⋮ | | | | |
| 1342 | 146.81 | 2002:c850:220b:: | 2002:c850:2220:: | RTP | Payload type=ITU-T G.711 PCMU, SSRC=687951629, Seq=14000, Time=638805707 |
| 1343 | 146.86 | 2002:c850:2220:: | 2002:c850:220b:: | RTCP | Sender Report |
| 1344 | 146.86 | 2002:c850:2220:: | 2002:c850:220b:: | RTP | Payload type=ITU-T G.711 PCMU, SSRC=808055695, Seq=3048, Time=987546188 |
| | ⋮ | | | | |
| 1690 | 155.36 | 2002:c850:2220:: | 2002:c850:220b:: | RTP | Payload type=ITU-T G.711 PCMU, SSRC=808055695, Seq=3306, Time=987614348 |
| 1700 | 158.34 | 2002:c850:2220:: | 2002:c850:220b:: | RTCP | Sender Report |
| 1701 | 158.59 | 2002:c850:2220:: | 2002:c850:220b:: | RTCP | Receiver Report |
| 1702 | 158.60 | 2002:c850:220b:: | 2002:c850:2220:: | RTCP | Sender Report |
| 1707 | 161.63 | 2002:c850:2220:: | 2002:c850:220b:: | SIP | Request: BYE sip:root@[2002:c850:220b::]: 5060;transport=UDP |
| 1713 | 164.97 | 2002:c850:220b:: | 2002:c850:2220:: | SIP | Status: 200 OK |

Figura 3-1: Secuencia de paquetes de la comunicación

La comunicación comienza con un mensaje de Request INVITE (11) enviado desde [2002:c850:220b::] hacia [2002:c850:2220::], este último le envía un mensaje de Status Ringing (16) indicando que el llamado está esperando ser atendido, luego envía un mensaje de Status OK (50) indicando que la conversación puede comenzar y finalmente el originante responde con un Request ACK (52) y comienza el envío de datos en ambas direcciones, usando los protocolos RTP y RTCP. Una vez concluida la conversación (a partir del frame 1690), se produce un intercambio de paquetes RTCP que cierran el canal de datos y luego un paquete Request BYE que finaliza la sesión. En el medio de la comunicación aparecieron mensajes ICMPv6 que no son parte de la comunicación pero se dejaron como ilustración. Estos mensajes son producto de una rutina de IPv6 de descubrimiento de vecinos.

El primer mensaje de INVITE fue enviado por el usuario *root* de [2002:c850:220b::] (o voipv6, el nombre del host) al usuario *root* de [2002:c850:2220::] (o voip2). En el anexo B se explica por qué ambos usuarios son iguales y por qué no se usan nombres en lugar de las largas direcciones IPv6.

```
Frame 11:
Internet Protocol Version 6
   Source address: 2002:c850:220b::
   Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 32773 (32773), Dst Port: 5060 (5060)
Session Initiation Protocol
      INVITE sip:root@[2002:c850:2220::] SIP/2.0
      CSeq: 12 INVITE
      Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
      From: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
      To: "" <sip:root@[2002:c850:2220::]>
```

```
    Via: SIP/2.0/UDP [2002:c850:220b::]:5060
    Contact: "Carlos Espositov6" <sip:root@[2002:c850:220b::]:5060;transport=udp>
    Subject: empty
    Content-Type: application/sdp
    Content-Length: 178
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): root 1075326941736 1075326941737 IN IP6 [2002:c850:220b:0:0:0:0:0]
    Session Name (s): BonePhone
    Connection Information (c): IN IP6 [2002:c850:220b:0:0:0:0:0]
    Time Description, active time (t): 0 0
    Media Description, name and address (m): audio 9032 RTP/AVP 0
    Media Attribute (a): rtpmap:0 PCMU/8000
```

Los mensajes SIP se envían por UDP en el puerto 5060, como se observa en la línea de "UDP". Este mensaje de INVITE incluye la información de la sesión que se desea establecer en la parte del SDP. Los datos más importantes que figuran aquí son la descripción y atributos del medio a transmitir (en Media Attribute), en este caso, audio codificado PCMU/8000, que se transmitirá en paquetes RTP en el puerto 9032. La información del puerto es muy importante ya que no existe un puerto estandar para RTP sino que se usa el que se informa en esta sección.

A continuación, el host llamado envía un mensaje Ringing indicando que espera a que el usuario atienda la comunicación. Puede verse que el "Call ID:" se mantiene sin modificaciones, y será así hasta el final de la sesión:

```
Frame 16:
Internet Protocol Version 6
   Source address: 2002:c850:2220::
   Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 1029 (1029), Dst Port: 5060 (5060)
Session Initiation Protocol
    SIP/2.0 180 Ringing
    Via: SIP/2.0/UDP [2002:c850:220b::]:5060
    From: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
    To: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
    Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
    CSeq: 12 INVITE
    Date: Wed, 28 Jan 2004 06:55:53 GMT
    Contact: "" <sip:root@[2002:c850:2220::]:5060;transport=udp>
    Content-Length: 0
```

Una vez que el usuario llamado atiende la comunicación, haciendo click en un botón en la interfaz del Bonephone, se envía un mensaje de OK. En este mensaje se incluye la descripción de la sesión que el usuario llamado desea establecer; los datos son similares al del Frame 11 con la excepción del puerto, que será el 9022. Esto significa que el host [2002:c850:2220::] escuchará los paquetes RTP que le quieran enviar en el puerto UDP 9022, y con la codificación PCMU/8000, como se observa en el campo Media Attribute:

```
Frame 50:
Internet Protocol Version 6
   Source address: 2002:c850:2220::
   Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 1029 (1029), Dst Port: 5060 (5060)
Session Initiation Protocol
    SIP/2.0 200 OK
    Via: SIP/2.0/UDP [2002:c850:220b::]:5060
    From: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
    To: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
    Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
    CSeq: 12 INVITE
    Content-Type: application/sdp
    Date: Wed, 28 Jan 2004 06:55:58 GMT
    Contact: "" <sip:root@[2002:c850:2220::]:5060;transport=udp>
    Content-Length: 178
```

```
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): root 1075316158226 1075316158227 IN IP6 [2002:c850:2220:0:0:0:0:0]
    Session Name (s): BonePhone
    Connection Information (c): IN IP6 [2002:c850:2220:0:0:0:0:0]
    Time Description, active time (t): 0 0
    Media Description, name and address (m): audio 9022 RTP/AVP 0
    Media Attribute (a): rtpmap:0 PCMU/8000
```

La comunicación comienza cuando el host llamante envía el ACK del mensaje de OK anterior:

```
Frame 52:
Internet Protocol Version 6
  Source address: 2002:c850:220b::
  Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 32773 (32773), Dst Port: 5060 (5060)
Session Initiation Protocol
    ACK sip:root@[2002:c850:2220::] SIP/2.0
    CSeq: 55 ACK
    Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
    From: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
    To: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
    Via: SIP/2.0/UDP [2002:c850:220b::]:5060
    Contact: "Carlos Espositov6" <sip:root@[2002:c850:220b::]:5060;transport=udp>
    Subject: OK
    Content-Length: 0
```

En la pantalla de la aplicación aparecen los datos del usuario llamado (o llamante, según sea el caso) y se activa un fondo verde indicando que una conversación está en curso.

Como se vio en 2.4.1, el host [2002:c850:220b::] también escuchará paquetes RTCP en el puerto UDP 9033, mientras que el host [2002:c850:2220::] hará lo propio en el puerto UDP 9023. La comunicación se inicia con un intercambio de varios paquetes RTCP que sólo contienen la descripción de la fuente en el campo SDES. Puede verse que hay dos paquetes RTCP, uno a continuación de otro, que forman un paquete compuesto. En este caso, el primer paquete que aparece, el Reception Report, sólo se envía a fin de cumplir con las restricciones de la RFC 3550 pero no incluye ninguna información relevante:

```
Frame 155:
Internet Protocol Version 6
  Source address: 2002:c850:220b::
  Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 9033 (9033), Dst Port: 9023 (9023)
Real-time Transport Control Protocol
  Version: RFC 1889 Version (2)
  Padding: False
  Reception report count: 0
  Packet type: Receiver Report (201)
  Length: 1
  Sender SSRC: 687951629
Real-time Transport Control Protocol
  Version: RFC 1889 Version (2)
  Padding: False
  Source count: 1
  Packet type: Source description (202)
  Length: 9
  Chunk 1, SSRC/CSRC 687951629
    Identifier: 687951629
    SDES items
      Type: CNAME (user and domain) (1)
      Length: 21
      Text: root@2002:c850:220b::
      Type: NOTE (note about source) (7)
      Length: 0
      Text:
      Type: NAME (common name) (2)
```

```
    Length: 4
    Text: root
```

Según puede verse en la figura 3.1, le siguen a estos una gran cantidad de paquetes RTP enviados en ambas direcciones, que contienen en el Payload pequeños fragmentos de audio codificado, que la aplicación decodificará para entregar al sistema de sonido de la computadora:

```
Frame 176:
Internet Protocol Version 6
   Version: 6
   Traffic class: 0x00
   Flowlabel: 0x00000
   Payload length: 180
   Next header: UDP (0x11)
   Hop limit: 64
   Source address: 2002:c850:2220::
   Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9022 (9022), Dst Port: 9032 (9032)
   Source port: 9022 (9022)
   Destination port: 9032 (9032)
   Length: 180
   Checksum: 0x4ba0 (correct)
Real-Time Transport Protocol
   Version: RFC 1889 Version (2)
   Padding: False
   Extension: False
   Contributing source identifiers count: 0
   Marker: True
   Payload type: ITU-T G.711 PCMU (0)
   Sequence number: 2820
   Timestamp: 986469228
   Synchronization Source identifier: 808055695
   Payload: F3F3F4F4F6F6F4F3F4F4F4F6F5F6F5F5...
```

En este ejemplo, el paquete es enviado por el host [2002:c850:220b::]. Puede observarse que todos los paquetes enviados por este host, tanto RTP como RTCP, tienen el mismo Synchronization Source identifier (SSRC). No hay Contributing souce identifiers (CSRC) ya que la fuente de datos proviene de un sólo host, y no fue mezclada con otras. Como se vio en 2.4.1, el Sequence number sirve para ordenar los paquetes en el destino, mientras que el Timestamp se utiliza para sincronizar este stream de audio con otros streams de audio o video, si estuvieran presentes. Los paquetes subsiguientes, que no se muestran aquí, incrementan en uno el Sequence number.

El Payload type indica el codec bajo el cuál fue procesado el audio, tal como negociaron ambos hosts mediante SDP en los Frames 11 y 50 analizados precedentemente.

El encabezado UDP proporciona el servicio de Checksum de todo el paquete, y la identificación del puerto donde los datos fueron enviados. El encabezado IPv6 se analiza en detalle en la sección siguiente, ya que es el motivo de estudio de este trabajo.

Durante la comunicación, aparecen distintos paquetes RTCP de tipo Sender Report o Receiver Report, que incluyen el reporte estadístico de cómo está recibiendo los datos el host que envía el paquete en cuestión. Aquí se muestra un ejemplo de Sender Report enviado por el host [2002:c850:2220::]:

```
Frame 1343:
Internet Protocol Version 6
   Source address: 2002:c850:2220::
   Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9023 (9023), Dst Port: 9033 (9033)
Real-time Transport Control Protocol
   Version: RFC 1889 Version (2)
   Padding: False
```

```
    Reception report count: 1
    Packet type: Sender Report (200)
    Length: 12
    Sender SSRC: 808055695
    Timestamp, MSW: 3284305096
    Timestamp, LSW: 927266574
    RTP timestamp: 987546188
    Sender's packet count: 228
    Sender's octet count: 39216
    Source 1
      Identifier: 687951629
      SSRC contents
        Fraction lost: 0 / 256
        Cumulative number of packets lost: 0
      Extended highest sequence number received: 14000
        Sequence number cycles count: 0
        Highest sequence number received: 14000
      Interarrival jitter: 418
      Last SR timestamp: 2934439515
      Delay since last SR timestamp: 330789
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Source count: 1
    Packet type: Source description (202)
    Length: 18
    Chunk 1, SSRC/CSRC 808055695
      Identifier: 808055695
      SDES items
        Type: CNAME (user and domain) (1)
        Length: 21
        Text: root@2002:c850:2220::
        Type: NOTE (note about source) (7)
        Length: 0
        Text:
        Type: TOOL (name/version of source app) (6)
        Length: 37
        Text: RAT v4.2.22 Linux 2.4.18-1-686 (i686)
```

Puede observarse que no hay paquetes perdidos (Fraction lost: 0/256), y que el valor de jitter (Interarrival jitter: 4,18 ms) es muy bajo. En el campo TOOL del Source Descriptor (SDES) se menciona la aplicación que usa el Bonephone para manejar la codificación del sonido.

Cuando la conversación finaliza y uno de los dos usuarios realiza un click en el botón Hang up de la aplicación, se suceden los envíos de dos paquetes por parte del host que corta: uno RTCP de tipo BYE (o Goodbye, tipo 203) para cerrar el canal lógico del RTP/RTCP y el otro es un mensaje de Request tipo BYE que cierra la sesión SIP. Ambos paquetes se muestran a continuación:

```
Frame 1701:
Internet Protocol Version 6
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9023 (9023), Dst Port: 9033 (9033)
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Reception report count: 0
    Packet type: Receiver Report (201)
    Length: 1
    Sender SSRC: 808055695
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Source count: 1
    Packet type: Goodbye (203)
    Length: 1
    Identifier: 808055695

Frame 1707:
```

```
Internet Protocol Version 6
   Source address: 2002:c850:2220::
   Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 1030 (1030), Dst Port: 5060 (5060)
Session Initiation Protocol
      BYE sip:root@[2002:c850:220b::]:5060;transport=UDP SIP/2.0
      CSeq: 13 BYE
      Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
      From: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
      To: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
      Via: SIP/2.0/UDP [2002:c850:2220::]:5060
      Contact: "Carlos Espositov6" <sip:root@[2002:c850:2220::]:5060;transport=udp>
      Content-Length: 0
```

Para finalizar, el host que recibió el mensaje BYE, lo acepta enviando un mensaje OK. La comunicación finaliza y la aplicación regresa a su pantalla inicial en ambas computadoras:

```
Frame 1713:
Internet Protocol Version 6
   Source address: 2002:c850:220b::
   Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 32774 (32774), Dst Port: 5060 (5060)
Session Initiation Protocol
      SIP/2.0 200 OK
      Via: SIP/2.0/UDP [2002:c850:2220::]:5060
      From: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
      To: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
      Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
      CSeq: 13 BYE
      Date: Wed, 28 Jan 2004 09:58:22 GMT
      Contact: "Carlos Espositov6" <sip:root@[2002:c850:220b::]:5060;transport=udp>
```

Aunque lo que se ha mostrado da la sensación de que los datos de establecimiento y control de llamada ocupan muchos paquetes y por lo tanto, mucho ancho de banda, tomando una estadística puede apreciarse que no es así. La figura siguiente muestra el uso que tuvo la red durante la comunicación auditada:

| | Packets | % of Packets | Bytes | % of bytes |
|---|---|---|---|---|
| Internet Protocol version 6 | 1375 | 100% | 315900 | 100% |
| User Datagram Protocol | 1358 | 98,7% | 314118 | 99,4% |
| Session Initiation Protocol | 6 | 0,4% | 3002 | 0,9% |
| Real-time Transport Control Protocol | 48 | 3,4% | 5980 | 1,8% |
| Real-time Transport Protocol | 1304 | 94,8% | 305136 | 96,5% |
| Internet Control Message Protocol v6 | 17 | 1,2% | 1782 | 0,5% |

Figura 3-2: Estadística de protocolos de la comunicación

Casi el 95% de los paquetes son fragmentos de streams de audio, y sólo el 5% se usa en control de la llamada. A su vez, de ese 5% más de la mitad lo usa RTCP para control de la calidad del canal; el establecimiento y mantenimiento de la sesión ocupa sólo unos pocos paquetes. La relación de rendimiento mejora aún más si se mide en bytes. En ese caso, casi el 97% de los bytes transmitidos corresponden a paquetes de datos.

Los paquetes UDP que contienen fragmentos de audio dentro de paquetes RTP, son en su gran mayoría de 180 bytes de longitud. El encabezado RTP es de 12 bytes y el UDP de 8 bytes, o sea que quedan 160 bytes de datos. Si sumamos el encabezado IP (44 bytes), el UDP y el RTP se obtiene que un paquete completo está formado por un encabezado de 64 bytes para transmitir 160 bytes de datos.

## 3.2 Los paquetes IPv6 transmitidos

Cada uno de los paquetes RTP que contenían porciones del stream de audio PCMU se transmitió dentro de un paquete IPv6 en el nivel de red, todos con el mismo encabezado excepto por las direcciones de fuente y destino -dependiendo hacia qué host era transmitido el paquete- y por la longitud del payload que se modificó en cada paquete. A continuación se muestra uno de los encabezados IPv6 completo, con el valor de todas sus opciones:

```
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 180
  Next header: UDP (0x11)
  Hop limit: 64
  Source address: 2002:c850:2220::
  Destination address: 2002:c850:220b::
```

Figura 3-3: El encabezado IPv6 transmitido por la aplicación Bonephone

Una de las ventajas que ofrece IPv6 para aplicaciones multimedia está relacionada con el manejo de Calidad de Servicio. De acuerdo a lo expresado en 1.5.2, el campo *Clase de tráfico* define el tipo de tráfico que se está enviando, y se le da un tratamiento preferencial a aquellos paquetes con mayor prioridad. Pero los paquetes transmitidos por esta aplicación tienen la **Clase de tráfico** en **cero,** o sea que no les asigna ninguna prioridad por sobre otros tráficos como e-mail o http. Tampoco les asigna una **etiqueta de flujo,** que serviría para un procesamiento más eficiente de parte de los routers, aunque la etiqueta de flujo sólo sirve en combinación con otros elementos de QoS como la clase de tráfico o reserva de recursos. De esto se deduce que la aplicación no hace ningún aprovechamiento de las facilidades que ofrece IPv6 para el manejo de QoS de acuerdo al paradigma de servicios diferenciados (Differentiated Services).

El otro paradigma de QoS, visto en 1.5.1, tiene que ver con la reserva de recursos mediante RSVP. IPv6 hace más eficiente la operación de RSVP con el uso del encabezado de extensión Hop-by-Hop. El Bonephone no hace ningún tipo de reserva de recursos, y por tanto desaprovecha también la ventaja que otorga IPv6 en este tópico. Si bien la aplicación corrió en dos computadoras conectadas al mismo segmento de red, esto es, no participaron routers intermedios donde el RSVP es usado, no es usual que una aplicación verifique cuál es el alcance del paquete que va a enviar para elegir qué tipo de paquete enviar. Esto significa que el comportamiento va a ser el mismo si se encuentran presentes routers en el camino, y por lo tanto los paquetes serán iguales, esto es, no harán ningún tipo de reserva de recursos.

No se hicieron pruebas de multicast ni de movilidad en IPv6 para analizar el comportamiento de la aplicación. Un breve repaso por el código fuente permite deducir que tampoco en estos casos el Bonephone utiliza las ventajas de IPv6; más aún, es poco probable que al menos funcione. En desarrollos posteriores puede analizarse más en detalle multicast y movilidad.

El análisis hecho hasta aquí se ha centrado en los tópicos más importantes que pueden relacionar a las aplicaciones de Voz sobre IP con el IP versión 6: calidad de servicio orientada a la clase de tráfico y calidad de servicio orientada reserva de recursos. Otro tema que adquirirá mucha importancia a medida que avance el desarrollo de las tecnologías de VoIP es el de la seguridad, dado que los datos se transmiten por redes públicas abiertas. IPv6

incorpora la seguridad en forma estandar lo que provocará que se le sumen más adeptos ante la necesidad de resolver esta cuestión.

# Conclusión

En este trabajo se ha presentado un caso del comportamiento de una aplicación en IPv6, en el que resulta visible que no hace uso de la mayor parte de las facilidades que el protocolo provee. Aunque no puede generalizarse -no todas las aplicaciones serán iguales- es factible que otros sistemas adaptados a IPv6 trabajen en forma similar.

El Bonephone se eligió como aplicación IPv6 a analizar por sus necesidades intrínsecas de calidad de servicio. Al verificar su funcionamiento en un entorno IPv6 se descubrió que no utiliza las herramientas que el nuevo protocolo provee para brindar calidad y que se limita a soportar un mayor tamaño de direcciones, en el afán de ser apto para múltiples plataformas.

IPv6 pretendió dar un salto tecnológico respecto de su antecesor, no sólo resolver las cuestiones que con IPv4 no se pueden resolver. Tan es así que el esquema de direcciones globales agregables, pensado y definido inicialmente en la RFC 2373, fue cambiado por otro menos innovador que se parece mucho al IPv4, sólo que amplía el espacio direccionable. En la definición original, la ruta que debe tomar un paquete puede deducirse de la dirección de destino usando pequeñas tablas de ruteo. Pero según la RFC 3587 -la que define el nuevo esquema de direcciones- este cambio "no es el mejor en esta etapa del desarrollo de IPv6" (Hinden y otros, 2003, p.1). Con el nuevo formato de direcciones IPv6, el ruteo de un paquete se hace en forma muy similar a la del ruteo en IPv4. La definición original planteó un salto muy amplio que no pudo ser asumido por la comunidad de Internet.

Este cambio tecnológico obliga a que, para que IPv6 realmente aporte una mejora en la comunicación de red, debe ser usado en toda su amplitud y funcionalidad. Si sólo es usado soportando el tamaño de sus direcciones, lo único que se obtendrá es que se necesitará mayor ancho de banda para transmitir sus paquetes, habida cuenta del mayor tamaño de encabezado. Las aplicaciones no deben limitarse a adaptar las funciones de red al nuevo tamaño de direcciones, sino que deberán **rediseñarse** para incorporar las ventajas de IPv6.

Una buena forma de asegurarse que una aplicación aproveche a IPv6 es diseñándola para ese protocolo y luego recorrer el camino inverso, esto es, adaptarla para que soporte IPv4.

Más allá de algunas soluciones de compromiso que se han debido adoptar -por ejemplo NAT para remediar la escasez de direcciones- la red IPv4 funciona correctamente en la actualidad. Entonces ¿cuál es la conveniencia del diseño de aplicaciones para IPv6? ¿Por qué deberíamos migrar a IPv6?. Si tomamos en cuenta que los dispositivos móviles, por dar un ejemplo, se verán muy beneficiados con el nuevo protocolo, es muy probable que lo incorporen rápidamente. El resto de la red deberá entonces acompañar ese cambio y se verá obligada a adaptarse. Por otra parte, uno de los fuertes motivos por los que IPv6 no se usa en forma masiva es el soporte de software. Si bien hace varios años está disponible para Linux, Sun SCO o Solaris, la mayoría de los routers comerciales lo soportan desde hace poco tiempo y sólo las versiones recientes de Windows XP lo incorpora en forma estándar. Una vez que se

produzca la renovación total de hardware que ejecute XP sin inconvenientes y la renovación de la mayoría de routers y server de la red, la migración a IPv6 será más factible y por lo tanto, más rápidamente realizada.

El escenario de la Internet del futuro contiene a IPv6 como protocolo estándar, con gran cantidad de dispositivos móviles, con aplicaciones multimedia comunicándose mediante VoIP, llevándose la mayor porción del tráfico de la red. Resulta beneficioso entonces comenzar a pensar y construir esa nueva Internet.

# Anexo A: Indice de H.323 comentado

1 SCOPE
2 Normative Reference
3 Definitions
4 Symbols
5 Conventions
6 SYSTEM DESCRIPTION
- 6.1 Information Streams
- 6.2 Terminals caracteristics:
  Video codecs, audio codecs, receive path delay, H.225.0 Layer, H.245 Control, Call Control H.225.0, RAS Control H.225.0
  - 6.2.1 Terminal elements outside the scope of this Recommendation
  - 6.2.2 Terminal elements within the scope of this Recommendation
    Video Codec, Audio Codec, Data Channel, System Control Unit (Call Control, RAS Control, H.245 Control), H.225.0 Layer.
  - 6.2.3 Packet based network interface
    Es específico de la implementación, pero debe ajustarse a H.225.0
  - 6.2.4 Video Codec
    Especifica qué codecs de video debe soportar
  - 6.2.5 Audio Codec
    audio mixing, low bit rate operation, maximun audio-video transmit skew
    Especifica qué codecs de audio debe soportar y cómo tratar otros temas de audio
  - 6.2.6 Receive Path Delay
    Paquetes de medios se demoran para sincronizarse.
  - 6.2.7 Data Channel
    Características de los canales de datos opcionales (fax, mensajes de texto, etc.)
  - 6.2.8 H.245 control function
    Cómo se establece un canal de control entre entidades H.323 extremo a extremo. Incluye:
    Capability exchange
    Logical channel signalling
    Mode preferences
    Master-slave determination
    Multiplexed stream transmission over a single logical channel
  - 6.2.9 RAS signalling function (H.225.0)
    Registration, admision and status, se transmite con mensajes H.225.0, entre endpoints y gatekeepers.
  - 6.2.10 Call signalling function (H.225.0)
    Establece mediante mensajes H.225.0 una llamada entre dos endpoints, antes de abrir un canal lógico con H.245
  - 6.2.11 H.225.0 layer
    Mensajes de control con los gatekeepers
- 6.3 Gataway caracteristics
  Describe qué es y cómo se compone un gateway. Establece que el gateway deberá traducir formato de transmisión y procedimientos de comunicación entre un terminal de red y un terminal SCN.
  - 6.3.1 Gateway decomposition
    Cómo es físicamente un gateway
  - 6.3.2 Gateway applications
    Trunking gateways y access gateways, aclaración de terminología
- 6.4 Gatekeeper characteristics
  Describe qué funciones tiene un gatekeeper de control de llamadas
- 6.5 Multipoint controller characteristics
  Describe qué hace el MC
- 6.6 Multipoint processor characteristics
  Describe qué hace el multipoint processor
- 6.7 Multipoint control unit characteristics
- 6.8 Multipoint capability
  - 6.8.1 Centralized multipoint capability
  - 6.8.2 Decentralized multipoint capability
  - 6.8.3 Hybrid multipoint - Centralized audio
  - 6.8.4 Hubrid multipoint - Centralized video
  - 6.8.5 Establishment of common mode
  - 6.8.6 Multipoint rate matching
  - 6.8.7 Multipoint lip synchronization
  - 6.8.8 Multipoint encryption
  - 6.8.9 Cascading multipoint control units
- 6.9 Models for supplementary services

7 Call signalling
Mensajes y procedimientos para establecer, requerir cambios de bandwidth, obtener el estado de los extremos y desconectar una llamada
7.1 Addresses
7.1.1 Network address
7.1.2 TSAP identifier
7.1.3 Alias address
7.1.4 H.323 URL scheme
7.2 Registration, Admission and Status (RAS) channel
7.2.1 Gatekeeper discovery
Explica el proceso de encontrar el Gatekeeper
7.2.2 Endpoint registration
Explica el proceso de unirse a una Zona administrada por un Gatekeeper. RRQ, RCF y RRJ. Lightweight RRQ, Additive Registrations.
7.2.3 Endpoint location
7.2.4 Asmissions, bandwidth change, status and disengage
7.2.5 Access tokens
7.2.6 Alternate gatekeeper procedures
7.2.7 Usage information reporting
Para accounting y billing
7.2.8 Call credit-related capabilities
Para avisar al usuario el crédito para hablar disponible
7.2.9 Alternate transport addresses
Indica soporte de protocolo de transporte alternativo
7.3 Call signalling channel
Explica cómo se transfieren mensajes de control por el canal de señalización de la llamada.
7.3.1 Call signalling channel routing
7.3.2 Control channel routing
7.3.3 Call Signalling and Control protocol revisions
7.4 Call reference value
Explica qué es el CRV.
7.5 Call ID
Establece qué es el call ID.
7.6 Conference ID and Conference Goal
7.7 Endpoint call capacity
7.8 Caller identification services
Explica cómo deben identificarse cada parte en una llamada, qué restricciones tiene que respetar y qué procedimientos seguir para la presentación.
7.8.1 Description of services
7.8.2 Messages and information elements
7.8.3 Actions at the originating endpoint
7.8.4 Actions at the terminating endpoint
7.8.5 Actions at a gatekeeper
7.9 Generic extensible framework
Determina el uso de H.225.0 para adicionar nuevas funciones a las existentes.
7.9.1 Format of a GenericData structure
7.9.2 Negotiation using the extensible framework-general
7.9.3 Negotiation using the extensible framework-RAS
7.9.4 Negotiation using the extensible framework-call signalling
8 Call signalling procedures
Explica paso por paso cómo se establece una comunicación hasta la finalización.
8.1 Phase A - Call setup
Explica el intercambio de mensajes para el establecimiento, de acuerdo a si hay o no Gatekeepers , Gateways o MCUs presentes.
8.1.1 Basic call setup - neither endpoint registered
8.1.2 Both endpoints registered to the same gatekeeper
8.1.3 Only calling endpoint has gatekeeper
8.1.4 Only called endpoint has garekeeper
8.1.5 Both endpoints registered to diferents gatekeepers
8.1.6 Optional called endpoint signalling
8.1.7 Fast connect procedure
8.1.8 Call setup via gateways
8.1.9 Call setup with an MCU
8.1.10 Call forwarding
8.1.11 Broadcast call setup
8.1.12 Overlapped sending
8.1.13 Call setup to conference alias
8.1.14 Gatekeeper modification of destination addresses
8.1.15 Indicating desired protocols
8.1.16 Gatekeeper requested tones and announcements
8.2 Phase B - Initial communication and capability exchange
Explica cómo negocian ambas partes las facilidades con las que cuentan, mediante H.245.
8.2.1 Encapsulation of H.245 messages within Q.931 messages
8.2.2 Tunneling through intermediate signalling entities

8.2.3 Switching to a separate H.245 connection
8.2.4 Initiating H.245 tunneling in parallel with fast connect
8.3 Phase C - Establisment of audiovisual communication
Explica cómo abrir los canales lógicos de control y establecer el intercambio de datos.
8.3.1 Mode changes
8.3.2 Exchange of video by mutual agreement
8.3.3 Media stream address distribution
8.3.4 Correlation of media streams in multipoint conferences
8.3.5 Communication mode command procedures
8.4 Phase D - Call services
Trata los cambios que pueden suceder durante una comunicación, incorporación de integrentes, etc.
8.4.1 Bandwidth changes
8.4.2 Status
8.4.3 Ad hoc conference expansion
8.4.4 Supplementary services
8.4.5 Multipoint cascading
8.4.6 Third party initiated pause and re-routing
8.5 Phase D - Call termination
Explica los procedimientos necesarios para finalizar una comunicación.
8.5.1 Call clearing without a gatekeeper
8.5.2 Call clearing with a gatekeeper
8.5.3 Call clearing by gatekeeper
8.6 Protocol failure handling
9 Interoperation wiht other terminal types
Menciona cómo un terminal debe interactuar con otras redes.
9.1 Speech-only terminals
9.2 Visual telephone terminals over the ISDN
9.3 Visual telephone terminals over GSTN
9.4 Visual telephone terminals over mobile radio
9.5 Visual telephone terminals over ATM
9.6 Visual telephone terminals over guaranteed quality of service LANs
9.7 Simultaneous voice and data terminals over GSTN
9.8 T.120 terminals on the packet based network
9.9 Gateway for H.323 media transport over ATM
10 Optional enhancements
10.1 Encryption
10.2 Multipoint operation
10.2.1 H.243 control and indication
10.3 Call Linkage in H.323
10.3.1 Description
10.3.2 Invocation and operation
10.3.3 Interaction with H.450 supplementary services
10.4 Tunneling of non-G.323 signalling messages
10.4.1 Indicating support of tunneled protocols
10.4.2 Requesting a specific protocol tunnel to a gatekeeper
10.4.3 Tunneling a signalling protocol in H.225.0 call signalling messages
10.4.4 Gatekeeper considerations
10.5 Use of RTP payload for DTMF digits, telephony tones and telephony signals
11 Maintenance
11.1 Loopbacks for maintenance purposes
11.2 Monitoring methods

# Anexo B: Algunos tópicos sobre la instalación del Bonephone

En este anexo se mencionan algunas características del software Bonephone y también los problemas encontrados durante la instalación y puesta en funcionamiento.

- Mbus, Rat y NIST SIP stack: el programa combina estos tres elementos para realizar su función. Mbus es un protocolo para desarrollar aplicaciones distribuidas en múltiples hosts de una red. Rat es una aplicación de sonido en conferencia de open source para Linux. El NIST SIP stack es un framework que incluye componentes para realizar una sesión SIP.
- Video: Se experimentaron problemas con drivers video. El Xserver debe tener configurado el driver adecuado para la tarjeta que se dispone. A pesar otras aplicaciones funcionan correctamente con drivers genéricos, Bonephone da errores.
- Direcciones IPv6 en Mbus: hay un archivo de configuración llamado .mbus que incluye una dirección multicast IPv4. Esa dirección debe traducirse a IPv6.
- Versión de Linux: los escasos documentos de Bonephone (ver *Bonephone Project, 2003*) hablan de instalaciones exitosas en RedHat 7.2. En este caso se instaló con éxito en Debian 3.0 con el kernel 2.4.18. La instalación y configuración de Debian para IPv6 se realizó siguiendo las instrucciones de *Debian IPv6 Project (2004)* y el *Manual de referencia Debian* (Aoki y otros).
- Usuario: la ejecución del software presentó problemas cuando se intentó correr con un usuario sin privilegios de administrador. Por eso el único usuario que pudo ser usado fue root.
- Direcciones IPv6s asignadas: las IPs asignadas surgieron de aplicar el protocolo 6to4 a las direcciones IPv4 200.80.34.16 y 11.
- Nombres de hosts: La aplicación presentó problemas para obtener la dirección IPv6 a partir del nombre. Por ese motivo, hubo que reemplazar en todas las configuraciones y en la operación el nombre por la dirección correspondiente.
- Problema de interacción con otras aplicaciones SIP: La aplicación funciona correctamente mientras se conecte con otras aplicaciones Bonephone. Se detectaron problemas al intentar conectar con otra aplicación SIP para Windows llamada EZ-Phone.
- Ring: Cuando una de las aplicaciones entra en estado RINGING, se muestra una campana en pantalla pero no emite ningún sonido.
- Aunque la documentación no lo especifica, Bonephone requiere XWindows instalado para operar.
- Se encontraron dificultades para transcribir la información de paquetes capturados y hacer funcionar filtros. En *Ethereal Documentation* se encontraron algunas explicaciones para llegar a una solución.

# Anexo C: Detalle de paquetes capturados

```
Frame 11 (641 bytes on wire, 641 bytes captured)
    Arrival Time: Jan 28, 2004 18:54:11.793974000
    Time delta from previous packet: 2.673559000 seconds
    Time since reference or first frame: 3.758929000 seconds
    Frame Number: 11
    Packet Length: 641 bytes
    Capture Length: 641 bytes
Ethernet II, Src: 00:d0:09:f6:ed:ed, Dst: 00:d0:09:05:37:8b
    Destination: 00:d0:09:05:37:8b (200.80.34.16)
    Source: 00:d0:09:f6:ed:ed (200.80.34.19)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 587
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:220b::
    Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 32773 (32773), Dst Port: 5060
(5060)
    Source port: 32773 (32773)
    Destination port: 5060 (5060)
    Length: 587
    Checksum: 0x55b2 (correct)
Session Initiation Protocol
    Request line: INVITE sip:root@[2002:c850:2220::] SIP/2.0
        Method: INVITE
    Message Header
        CSeq: 12 INVITE
        Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
        From: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
            SIP from address: "Carlos Espositov6" <sip:root@voipv6>
            SIP tag: dd293413
        To: "" <sip:root@[2002:c850:2220::]>
        Via: SIP/2.0/UDP [2002:c850:220b::]:5060
        Contact: "Carlos Espositov6" <sip:root@[2002:c850:220b::]:5060;transport=udp>
        Subject: empty
        Content-Type: application/sdp
        Content-Length: 178
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): root 1075326941736
1075326941737 IN IP6 [2002:c850:220b:0:0:0:0:0]
        Owner Username: root
        Session ID: 1075326941736
        Session Version: 1075326941737
        Owner Network Type: IN
        Owner Address Type: IP6
        Owner Address: [2002:c850:220b:0:0:0:0:0]
    Session Name (s): BonePhone
    Connection Information (c): IN IP6 [2002:c850:220b:0:0:0:0:0]
        Connection Network Type: IN
        Connection Address Type: IP6
        Connection Address: [2002:c850:220b:0:0:0:0:0]
    Time Description, active time (t): 0 0
        Session Start Time: 0
        Session Start Time: 0
    Media Description, name and address (m): audio 9032 RTP/AVP 0
        Media Type: audio
        Media Port: 9032
        Media Proto: RTP/AVP
        Media Format: 0
    Media Attribute (a): rtpmap:0 PCMU/8000
        Media Attribute Fieldname: rtpmap
        Media Attribute Value: 0 PCMU/8000

Frame 16 (424 bytes on wire, 424 bytes captured)
    Arrival Time: Jan 28, 2004 18:54:12.714540000
    Time delta from previous packet: 0.920566000 seconds
    Time since reference or first frame: 4.679495000 seconds
    Frame Number: 16
    Packet Length: 424 bytes
    Capture Length: 424 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 370
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 1029 (1029), Dst Port: 5060
(5060)
    Source port: 1029 (1029)
    Destination port: 5060 (5060)
    Length: 370
    Checksum: 0xd3bb (correct)
Session Initiation Protocol
    Status line: SIP/2.0 180 Ringing
        Status-Code: 180
    Message Header
        Via: SIP/2.0/UDP [2002:c850:220b::]:5060
        From: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
            SIP from address: "Carlos Espositov6" <sip:root@voipv6>
            SIP tag: dd293413
        To: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
            SIP to address: "" <sip:root@[2002:c850:2220::]>
            SIP tag: 7587d6da
        Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
        CSeq: 12 INVITE
        Date: Wed, 28 Jan 2004 06:55:53 GMT
        Contact: ""
<sip:root@[2002:c850:2220::]:5060;transport=udp>
        Content-Length: 0

Frame 45 (86 bytes on wire, 86 bytes captured)
    Arrival Time: Jan 28, 2004 18:54:16.784400000
    Time delta from previous packet: 4.069860000 seconds
    Time since reference or first frame: 8.749355000 seconds
    Frame Number: 45
    Packet Length: 86 bytes
    Capture Length: 86 bytes
Ethernet II, Src: 00:d0:09:f6:ed:ed, Dst: 00:d0:09:05:37:8b
    Destination: 00:d0:09:05:37:8b (200.80.34.16)
    Source: 00:d0:09:f6:ed:ed (200.80.34.19)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 32
    Next header: ICMPv6 (0x3a)
    Hop limit: 255
    Source address: fe80::2d0:9ff:fef6:eded
    Destination address: 2002:c850:2220::
Internet Control Message Protocol v6
    Type: 135 (Neighbor solicitation)
    Code: 0
    Checksum: 0x71d5 (correct)
    Target: 2002:c850:2220::
    ICMPv6 options
        Type: 1 (Source link-layer address)
        Length: 8 bytes (1)
        Link-layer address: 00:d0:09:f6:ed:ed

Frame 46 (86 bytes on wire, 86 bytes captured)
    Arrival Time: Jan 28, 2004 18:54:16.784672000
    Time delta from previous packet: 0.000272000 seconds
    Time since reference or first frame: 8.749627000 seconds
    Frame Number: 46
    Packet Length: 86 bytes
    Capture Length: 86 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 32
    Next header: ICMPv6 (0x3a)
    Hop limit: 255
    Source address: 2002:c850:2220::
    Destination address: fe80::2d0:9ff:fef6:eded
```

```
Internet Control Message Protocol v6
    Type: 136 (Neighbor advertisement)
    Code: 0
    Checksum: 0x4728 (correct)
    Flags: 0xe0000000
      1... .... .... .... .... .... .... .... = Router
      .1.. .... .... .... .... .... .... .... = Solicited
      ..1. .... .... .... .... .... .... .... = Override
    Target: 2002:c850:2220::
    ICMPv6 options
      Type: 2 (Target link-layer address)
      Length: 8 bytes (1)
      Link-layer address: 00:d0:09:05:37:8b

Frame 50 (630 bytes on wire, 630 bytes captured)
    Arrival Time: Jan 28, 2004 18:54:17.244161000
    Time delta from previous packet: 0.459489000 seconds
    Time since reference or first frame: 9.209116000 seconds
    Frame Number: 50
    Packet Length: 630 bytes
    Capture Length: 630 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 576
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 1029 (1029), Dst Port: 5060
(5060)
    Source port: 1029 (1029)
    Destination port: 5060 (5060)
    Length: 576
    Checksum: 0xcd16 (correct)
Session Initiation Protocol
    Status line: SIP/2.0 200 OK
      Status-Code: 200
    Message Header
      Via: SIP/2.0/UDP [2002:c850:220b::]:5060
      From: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
        SIP from address: "Carlos Espositov6" <sip:root@voipv6>
        SIP tag: dd293413
      To: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
        SIP to address: "" <sip:root@[2002:c850:2220::]>
        SIP tag: 7587d6da
      Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
      CSeq: 12 INVITE
      Content-Type: application/sdp
      Date: Wed, 28 Jan 2004 06:55:58 GMT
      Contact: ""
<sip:root@[2002:c850:2220::]:5060;transport=udp>
      Content-Length: 178
Session Description Protocol
    Session Description Protocol Version (v): 0
    Owner/Creator, Session Id (o): root 1075316158226
1075316158227 IN IP6 [2002:c850:2220:0:0:0:0:0]
      Owner Username: root
      Session ID: 1075316158226
      Session Version: 1075316158227
      Owner Network Type: IN
      Owner Address Type: IP6
      Owner Address: [2002:c850:2220:0:0:0:0:0]
    Session Name (s): BonePhone
    Connection Information (c): IN IP6 [2002:c850:2220:0:0:0:0:0]
      Connection Network Type: IN
      Connection Address Type: IP6
      Connection Address: [2002:c850:2220:0:0:0:0:0]
    Time Description, active time (t): 0 0
      Session Start Time: 0
      Session Start Time: 0
    Media Description, name and address (m): audio 9022 RTP/AVP 0
      Media Type: audio
      Media Port: 9022
      Media Proto: RTP/AVP
      Media Format: 0
    Media Attribute (a): rtpmap:0 PCMU/8000
      Media Attribute Fieldname: rtpmap
      Media Attribute Value: 0 PCMU/8000

Frame 52 (434 bytes on wire, 434 bytes captured)
    Arrival Time: Jan 28, 2004 18:54:17.653179000
    Time delta from previous packet: 0.409018000 seconds
    Time since reference or first frame: 9.618134000 seconds
    Frame Number: 52
    Packet Length: 434 bytes
    Capture Length: 434 bytes
Ethernet II, Src: 00:d0:09:f6:ed:ed, Dst: 00:d0:09:05:37:8b
    Destination: 00:d0:09:05:37:8b (200.80.34.16)
    Source: 00:d0:09:f6:ed:ed (200.80.34.19)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 380
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:220b::
    Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 32773 (32773), Dst Port: 5060
(5060)
    Source port: 32773 (32773)
    Destination port: 5060 (5060)
    Length: 380
    Checksum: 0xcd2f (correct)
Session Initiation Protocol
    Request line: ACK sip:root@[2002:c850:2220::] SIP/2.0
      Method: ACK
    Message Header
      CSeq: 55 ACK
      Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
      From: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
        SIP from address: "Carlos Espositov6" <sip:root@voipv6>
        SIP tag: dd293413
      To: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
        SIP to address: "" <sip:root@[2002:c850:2220::]>
        SIP tag: 7587d6da
      Via: SIP/2.0/UDP [2002:c850:220b::]:5060
      Contact: "Carlos Espositov6"
<sip:root@[2002:c850:220b::]:5060;transport=udp>
      Subject: OK
      Content-Length: 0

Frame 155 (110 bytes on wire, 110 bytes captured)
    Arrival Time: Jan 28, 2004 18:55:04.089894000
    Time delta from previous packet: 2.608945000 seconds
    Time since reference or first frame: 56.054849000 seconds
    Frame Number: 155
    Packet Length: 110 bytes
    Capture Length: 110 bytes
Ethernet II, Src: 00:d0:09:f6:ed:ed, Dst: 00:d0:09:05:37:8b
    Destination: 00:d0:09:05:37:8b (200.80.34.16)
    Source: 00:d0:09:f6:ed:ed (200.80.34.19)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 56
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:220b::
    Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 9033 (9033), Dst Port: 9023
(9023)
    Source port: 9033 (9033)
    Destination port: 9023 (9023)
    Length: 56
    Checksum: 0xa2e4 (correct)
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Reception report count: 0
    Packet type: Receiver Report (201)
    Length: 1
    Sender SSRC: 687951629
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Source count: 1
    Packet type: Source description (202)
    Length: 9
    Chunk 1, SSRC/CSRC 687951629
      Identifier: 687951629
      SDES items
        Type: CNAME (user and domain) (1)
        Length: 21
        Text: root@2002:c850:220b::
        Type: NOTE (note about source) (7)
        Length: 0
        Text:
        Type: NAME (common name) (2)
        Length: 4
        Text: root

Frame 170 (110 bytes on wire, 110 bytes captured)
    Arrival Time: Jan 28, 2004 18:55:12.207417000
    Time delta from previous packet: 2.945145000 seconds
    Time since reference or first frame: 64.172372000 seconds
    Frame Number: 170
    Packet Length: 110 bytes
    Capture Length: 110 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 56
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9023 (9023), Dst Port: 9033
(9033)
    Source port: 9023 (9023)
```

```
    Destination port: 9033 (9033)
    Length: 56
    Checksum: 0x7d8d (correct)
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Reception report count: 0
    Packet type: Receiver Report (201)
    Length: 1
    Sender SSRC: 808055695
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Source count: 1
    Packet type: Source description (202)
    Length: 9
    Chunk 1, SSRC/CSRC 808055695
      Identifier: 808055695
      SDES items
        Type: CNAME (user and domain) (1)
        Length: 21
        Text: root@2002:c850:2220::
        Type: NOTE (note about source) (7)
        Length: 0
        Text:
        Type: NAME (common name) (2)
        Length: 4
        Text: root

Frame 176 (234 bytes on wire, 234 bytes captured)
    Arrival Time: Jan 28, 2004 18:55:17.451353000
    Time delta from previous packet: 3.193380000 seconds
    Time since reference or first frame: 69.416308000 seconds
    Frame Number: 176
    Packet Length: 234 bytes
    Capture Length: 234 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 180
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9022 (9022), Dst Port: 9032
(9032)
    Source port: 9022 (9022)
    Destination port: 9032 (9032)
    Length: 180
    Checksum: 0x4ba0 (correct)
Real-Time Transport Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Extension: False
    Contributing source identifiers count: 0
    Marker: True
    Payload type: ITU-T G.711 PCMU (0)
    Sequence number: 2820
    Timestamp: 986469228
    Synchronization Source identifier: 808055695
    Payload: F3F3F4F4F6F6F4F3F4F4F4F6F5F6F5F5...

Frame 177 (234 bytes on wire, 234 bytes captured)
    Arrival Time: Jan 28, 2004 18:55:17.467458000
    Time delta from previous packet: 0.016105000 seconds
    Time since reference or first frame: 69.432413000 seconds
    Frame Number: 177
    Packet Length: 234 bytes
    Capture Length: 234 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 180
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9022 (9022), Dst Port: 9032
(9032)
    Source port: 9022 (9022)
    Destination port: 9032 (9032)
    Length: 180
    Checksum: 0xa2d2 (correct)
Real-Time Transport Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Extension: False
    Contributing source identifiers count: 0
    Marker: False
    Payload type: ITU-T G.711 PCMU (0)
    Sequence number: 2821
    Timestamp: 986469388
    Synchronization Source identifier: 808055695
    Payload: 7C7E7E7EFFFE7E7EFEFFFF7DFF7D7B7C...

Frame 291 (234 bytes on wire, 234 bytes captured)
    Arrival Time: Jan 28, 2004 18:56:00.652244000
    Time delta from previous packet: 0.000296000 seconds
    Time since reference or first frame: 112.617199000 seconds
    Frame Number: 291
    Packet Length: 234 bytes
    Capture Length: 234 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 180
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9022 (9022), Dst Port: 9032
(9032)
    Source port: 9022 (9022)
    Destination port: 9032 (9032)
    Length: 180
    Checksum: 0x199e (correct)
Real-Time Transport Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Extension: False
    Contributing source identifiers count: 0
    Marker: False
    Payload type: ITU-T G.711 PCMU (0)
    Sequence number: 2857
    Timestamp: 987108428
    Synchronization Source identifier: 808055695
    Payload: 7D7D7C7D7B7D7E7E7E7D7E7E7E7E7E...

Frame 292 (134 bytes on wire, 134 bytes captured)
    Arrival Time: Jan 28, 2004 18:56:00.748444000
    Time delta from previous packet: 0.096200000 seconds
    Time since reference or first frame: 112.713399000 seconds
    Frame Number: 292
    Packet Length: 134 bytes
    Capture Length: 134 bytes
Ethernet II, Src: 00:d0:09:f6:ed:ed, Dst: 00:d0:09:05:37:8b
    Destination: 00:d0:09:05:37:8b (200.80.34.16)
    Source: 00:d0:09:f6:ed:ed (200.80.34.19)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 80
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:220b::
    Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 9033 (9033), Dst Port: 9023
(9023)
    Source port: 9033 (9033)
    Destination port: 9023 (9023)
    Length: 80
    Checksum: 0x473a (correct)
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Reception report count: 1
    Packet type: Receiver Report (201)
    Length: 7
    Sender SSRC: 687951629
    Source 1
      Identifier: 808055695
      SSRC contents
        Fraction lost: 4 / 256
        Cumulative number of packets lost: 720896
      Extended highest sequence number received: 2857
        Sequence number cycles count: 0
        Highest sequence number received: 2857
      Interarrival jitter: 2004631
      Last SR timestamp: 2223094121
      Delay since last SR timestamp: 2380482
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Source count: 1
    Packet type: Source description (202)
    Length: 9
    Chunk 1, SSRC/CSRC 687951629
      Identifier: 687951629
      SDES items
        Type: CNAME (user and domain) (1)
        Length: 21
        Text: root@2002:c850:220b::
        Type: NOTE (note about source) (7)
        Length: 0
        Text:
        Type: NAME (common name) (2)
        Length: 4
```

```
    Text: root

Frame 294 (130 bytes on wire, 130 bytes captured)
  Arrival Time: Jan 28, 2004 18:56:03.090535000
  Time delta from previous packet: 2.342091000 seconds
  Time since reference or first frame: 115.055490000 seconds
  Frame Number: 294
  Packet Length: 130 bytes
  Capture Length: 130 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
  Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
  Source: 00:d0:09:05:37:8b (200.80.34.16)
  Type: IPv6 (0x86dd)
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 76
  Next header: UDP (0x11)
  Hop limit: 64
  Source address: 2002:c850:2220::
  Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9023 (9023), Dst Port: 9033
(9033)
  Source port: 9023 (9023)
  Destination port: 9033 (9033)
  Length: 76
  Checksum: 0xc8e2 (correct)
Real-time Transport Control Protocol
  Version: RFC 1889 Version (2)
  Padding: False
  Reception report count: 0
  Packet type: Sender Report (200)
  Length: 6
  Sender SSRC: 808055695
  Timestamp, MSW: 3284305064
  Timestamp, LSW: 1721552325
  RTP timestamp: 987161388
  Sender's packet count: 38
  Sender's octet count: 6536
Real-time Transport Control Protocol
  Version: RFC 1889 Version (2)
  Padding: False
  Source count: 1
  Packet type: Source description (202)
  Length: 9
  Chunk 1, SSRC/CSRC: 808055695
    Identifier: 808055695
    SDES items
      Type: CNAME (user and domain) (1)
      Length: 21
      Text: root@2002:c850:2220::
      Type: NOTE (note about source) (7)
      Length: 0
      Text:
      Type: NAME (common name) (2)
      Length: 4
      Text: root

Frame 296 (234 bytes on wire, 234 bytes captured)
  Arrival Time: Jan 28, 2004 18:56:04.818517000
  Time delta from previous packet: 1.727982000 seconds
  Time since reference or first frame: 116.783472000 seconds
  Frame Number: 296
  Packet Length: 234 bytes
  Capture Length: 234 bytes
Ethernet II, Src: 00:d0:09:f6:ed:ed, Dst: 00:d0:09:05:37:8b
  Destination: 00:d0:09:05:37:8b (200.80.34.16)
  Source: 00:d0:09:f6:ed:ed (200.80.34.19)
  Type: IPv6 (0x86dd)
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 180
  Next header: UDP (0x11)
  Hop limit: 64
  Source address: 2002:c850:220b::
  Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 9032 (9032), Dst Port: 9022
(9022)
  Source port: 9032 (9032)
  Destination port: 9022 (9022)
  Length: 180
  Checksum: 0xcb11 (correct)
Real-Time Transport Protocol
  Version: RFC 1889 Version (2)
  Padding: False
  Extension: False
  Contributing source identifiers count: 0
  Marker: True
  Payload type: ITU-T G.711 PCMU (0)
  Sequence number: 13226
  Timestamp: 638434027
  Synchronization Source identifier: 687951629
  Payload: 282522201D18223BCC9C969497A1CC25...

Frame 1690 (234 bytes on wire, 234 bytes captured)
  Arrival Time: Jan 28, 2004 18:56:43.400741000
  Time delta from previous packet: 0.000390000 seconds
  Time since reference or first frame: 155.365696000 seconds
  Frame Number: 1690
  Packet Length: 234 bytes
  Capture Length: 234 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
  Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
  Source: 00:d0:09:05:37:8b (200.80.34.16)
  Type: IPv6 (0x86dd)
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 180
  Next header: UDP (0x11)
  Hop limit: 64
  Source address: 2002:c850:2220::
  Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9022 (9022), Dst Port: 9032
(9032)
  Source port: 9022 (9022)
  Destination port: 9032 (9032)
  Length: 180
  Checksum: 0xf2a4 (correct)
Real-Time Transport Protocol
  Version: RFC 1889 Version (2)
  Padding: False
  Extension: False
  Contributing source identifiers count: 0
  Marker: False
  Payload type: ITU-T G.711 PCMU (0)
  Sequence number: 3306
  Timestamp: 987614348
  Synchronization Source identifier: 808055695
  Payload: 7EFEFDFDFEFDFE7D7D7E7D7D7EFE7EFF...

Frame 1700 (150 bytes on wire, 150 bytes captured)
  Arrival Time: Jan 28, 2004 18:56:46.381910000
  Time delta from previous packet: 2.981169000 seconds
  Time since reference or first frame: 158.346865000 seconds
  Frame Number: 1700
  Packet Length: 150 bytes
  Capture Length: 150 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
  Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
  Source: 00:d0:09:05:37:8b (200.80.34.16)
  Type: IPv6 (0x86dd)
Internet Protocol Version 6
  Version: 6
  Traffic class: 0x00
  Flowlabel: 0x00000
  Payload length: 96
  Next header: UDP (0x11)
  Hop limit: 64
  Source address: 2002:c850:2220::
  Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9023 (9023), Dst Port: 9033
(9033)
  Source port: 9023 (9023)
  Destination port: 9033 (9033)
  Length: 96
  Checksum: 0x0d29 (correct)
Real-time Transport Control Protocol
  Version: RFC 1889 Version (2)
  Padding: False
  Reception report count: 1
  Packet type: Sender Report (200)
  Length: 12
  Sender SSRC: 808055695
  Timestamp, MSW: 3284305107
  Timestamp, LSW: 3063004186
  RTP timestamp: 987638668
  Sender's packet count: 487
  Sender's octet count: 83764
  Source 1
    Identifier: 687951629
    SSRC contents
      Fraction lost: 0 / 256
      Cumulative number of packets lost: 0
    Extended highest sequence number received: 14042
      Sequence number cycles count: 0
      Highest sequence number received: 14042
    Interarrival jitter: 436
    Last SR timestamp: 2935192290
    Delay since last SR timestamp: 332206
Real-time Transport Control Protocol
  Version: RFC 1889 Version (2)
  Padding: False
  Source count: 1
  Packet type: Source description (202)
  Length: 8
  Chunk 1, SSRC/CSRC: 808055695
    Identifier: 808055695
    SDES items
      Type: CNAME (user and domain) (1)
      Length: 21
      Text: root@2002:c850:2220::
      Type: NOTE (note about source) (7)
      Length: 0
      Text:

Frame 1701 (78 bytes on wire, 78 bytes captured)
  Arrival Time: Jan 28, 2004 18:56:46.628495000
  Time delta from previous packet: 0.246585000 seconds
  Time since reference or first frame: 158.593450000 seconds
```

```
    Frame Number: 1701
    Packet Length: 78 bytes
    Capture Length: 78 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 24
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 9023 (9023), Dst Port: 9033
(9033)
    Source port: 9023 (9023)
    Destination port: 9033 (9033)
    Length: 24
    Checksum: 0x5a5c (correct)
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Reception report count: 0
    Packet type: Receiver Report (201)
    Length: 1
    Sender SSRC: 808055695
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Source count: 1
    Packet type: Goodbye (203)
    Length: 1
    Identifier: 808055695

Frame 1702 (186 bytes on wire, 186 bytes captured)
    Arrival Time: Jan 28, 2004 18:56:46.643941000
    Time delta from previous packet: 0.015446000 seconds
    Time since reference or first frame: 158.608896000 seconds
    Frame Number: 1702
    Packet Length: 186 bytes
    Capture Length: 186 bytes
Ethernet II, Src: 00:d0:09:f6:ed:ed, Dst: 00:d0:09:05:37:8b
    Destination: 00:d0:09:05:37:8b (200.80.34.16)
    Source: 00:d0:09:f6:ed:ed (200.80.34.19)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 132
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:220b::
    Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 9033 (9033), Dst Port: 9023
(9023)
    Source port: 9033 (9033)
    Destination port: 9023 (9023)
    Length: 132
    Checksum: 0x799a (correct)
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Reception report count: 1
    Packet type: Sender Report (200)
    Length: 12
    Sender SSRC: 687951629
    Timestamp, MSW: 3284315896
    Timestamp, LSW: 3605677783
    RTP timestamp: 638932907
    Sender's packet count: 817
    Sender's octet count: 140524
    Source 1
      Identifier: 808055695
      SSRC contents
        Fraction lost: 4 / 256
        Cumulative number of packets lost: 720896
      Extended highest sequence number received: 3306
        Sequence number cycles count: 0
        Highest sequence number received: 3306
      Interarrival jitter: 257
      Last SR timestamp: 2228467345
      Delay since last SR timestamp: 16799
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Source count: 1
    Packet type: Source description (202)
    Length: 17
    Chunk 1, SSRC/CSRC: 687951629
      Identifier: 687951629
      SDES items
        Type: CNAME (user and domain) (1)
        Length: 21
        Text: root@2002:c850:220b::
        Type: NOTE (note about source) (7)
        Length: 0
        Text:
        Type: TOOL (name/version of source app) (6)
        Length: 34
        Text: RAT v4.2.22 Linux 2.4.18-k6 (i586)

Frame 1707 (440 bytes on wire, 440 bytes captured)
    Arrival Time: Jan 28, 2004 18:56:49.671784000
    Time delta from previous packet: 3.027843000 seconds
    Time since reference or first frame: 161.636739000 seconds
    Frame Number: 1707
    Packet Length: 440 bytes
    Capture Length: 440 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 386
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
User Datagram Protocol, Src Port: 1030 (1030), Dst Port: 5060
(5060)
    Source port: 1030 (1030)
    Destination port: 5060 (5060)
    Length: 386
    Checksum: 0xda2b (correct)
Session Initiation Protocol
    Request line: BYE
sip:root@[2002:c850:220b::] 5060;transport=UDP SIP/2.0
      Method: BYE
    Message Header
      CSeq: 13 BYE
      Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
      From: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
        SIP from address: "" <sip:root@[2002:c850:2220::]>
        SIP tag: 7587d6da
      To: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
        SIP to address: "Carlos Espositov6" <sip:root@voipv6>
        SIP tag: dd293413
      Via: SIP/2.0/UDP [2002:c850:2220::]:5060
      Contact: "Carlos Espositov6"
<sip:root@[2002:c850:2220::]:5060;transport=udp>
      Content-Length: 0

Frame 1711 (78 bytes on wire, 78 bytes captured)
    Arrival Time: Jan 28, 2004 18:56:52.518663000
    Time delta from previous packet: 1.139983000 seconds
    Time since reference or first frame: 164.483618000 seconds
    Frame Number: 1711
    Packet Length: 78 bytes
    Capture Length: 78 bytes
Ethernet II, Src: 00:d0:09:f6:ed:ed, Dst: 00:d0:09:05:37:8b
    Destination: 00:d0:09:05:37:8b (200.80.34.16)
    Source: 00:d0:09:f6:ed:ed (200.80.34.19)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 24
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:220b::
    Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 9033 (9033), Dst Port: 9023
(9023)
    Source port: 9033 (9033)
    Destination port: 9023 (9023)
    Length: 24
    Checksum: 0xb1b1 (correct)
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Reception report count: 0
    Packet type: Receiver Report (201)
    Length: 1
    Sender SSRC: 687951629
Real-time Transport Control Protocol
    Version: RFC 1889 Version (2)
    Padding: False
    Source count: 1
    Packet type: Goodbye (203)
    Length: 1
    Identifier: 687951629

Frame 1712 (126 bytes on wire, 126 bytes captured)
    Arrival Time: Jan 28, 2004 18:56:52.519222000
    Time delta from previous packet: 0.000559000 seconds
    Time since reference or first frame: 164.484177000 seconds
    Frame Number: 1712
    Packet Length: 126 bytes
    Capture Length: 126 bytes
Ethernet II, Src: 00:d0:09:05:37:8b, Dst: 00:d0:09:f6:ed:ed
    Destination: 00:d0:09:f6:ed:ed (200.80.34.19)
    Source: 00:d0:09:05:37:8b (200.80.34.16)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
```

```
    Flowlabel: 0x00000
    Payload length: 72
    Next header: ICMPv6 (0x3a)
    Hop limit: 64
    Source address: 2002:c850:2220::
    Destination address: 2002:c850:220b::
Internet Control Message Protocol v6
    Type: 1 (Unreachable)
    Code: 4 (Port unreachable)
    Checksum: 0x7879 (correct)
    Internet Protocol Version 6
        Version: 6
        Traffic class: 0x00
        Flowlabel: 0x00000
        Payload length: 24
        Next header: UDP (0x11)
        Hop limit: 64
        Source address: 2002:c850:220b::
        Destination address: 2002:c850:2220::
    User Datagram Protocol, Src Port: 9033 (9033), Dst Port: 9023
(9023)
        Source port: 9033 (9033)
        Destination port: 9023 (9023)
        Length: 24
        Checksum: 0xb1b1 (correct)
    Real-time Transport Control Protocol
        Version: RFC 1889 Version (2)
        Padding: False
        Reception report count: 0
        Packet type: Receiver Report (201)
        Length: 1
        Sender SSRC: 687951629
    Real-time Transport Control Protocol
        Version: RFC 1889 Version (2)
        Padding: False
        Source count: 1
        Packet type: Goodbye (203)
        Length: 1
        Identifier: 687951629

Frame 1713 (433 bytes on wire, 433 bytes captured)
    Arrival Time: Jan 28, 2004 18:56:53.008234000
    Time delta from previous packet: 0.489012000 seconds
    Time since reference or first frame: 164.973189000 seconds
    Frame Number: 1713
    Packet Length: 433 bytes
    Capture Length: 433 bytes
Ethernet II, Src: 00:d0:09:f6:ed:ed, Dst: 00:d0:09:05:37:8b
    Destination: 00:d0:09:05:37:8b (200.80.34.16)
    Source: 00:d0:09:f6:ed:ed (200.80.34.19)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 379
    Next header: UDP (0x11)
    Hop limit: 64
    Source address: 2002:c850:220b::
    Destination address: 2002:c850:2220::
User Datagram Protocol, Src Port: 32774 (32774), Dst Port: 5060
(5060)
    Source port: 32774 (32774)
    Destination port: 5060 (5060)
    Length: 379
    Checksum: 0xf8a4 (correct)
Session Initiation Protocol
    Status line: SIP/2.0 200 OK
        Status-Code: 200
    Message Header
        Via: SIP/2.0/UDP [2002:c850:2220::]:5060
        From: "" <sip:root@[2002:c850:2220::]>;tag=7587d6da
            SIP from address: "" <sip:root@[2002:c850:2220::]>
            SIP tag: 7587d6da
        To: "Carlos Espositov6" <sip:root@voipv6>;tag=dd293413
            SIP to address: "Carlos Espositov6" <sip:root@voipv6>
            SIP tag: dd293413
        Call-Id: 61a9d99021502f5022c506bb071ec638@localhost
        CSeq: 13 BYE
        Date: Wed, 28 Jan 2004 09:58:22 GMT
        Contact: "Carlos Espositov6"
<sip:root@[2002:c850:220b::]:5060;transport=udp>
```

# Bibliografía


Aoki, Osamu y otros. *Debian Reference Manual.* Usado en la instalación de Linux y XWindows.

*Bonephone Project.* Visitada diciembre 2003. http://www.iptel.org/products/bonephone/.

Braden, R., Zhang, L., Berson, S., Herzog, S. y Jamin, S. *Resource Reservation Protocol (RSVP) (RFC 2205).* Septiembre 1997. Definición del protocolo, funciones provistas.

*Debian IPv6 Project - Setup instructions.* Visitada enero 2004. http://people.debian.org/~csmall/ipv6/setup.html.

Deering, S. y Hinden, R. *IP version 6 Adressing Architecture (RFC 2373).* Julio 1998. Explicación del formato de direcciones IPv6, quedó OBSOLETA.

Deering, S. y Hinden, R. *Internet Protocol version 6 (RFC 2460).* Diciembre 1998. Definición del protocolo, para consulta permanente.

*Ethereal Documentation.* Visitada enero 2004. http://www.ethereal.com/docs/

Hagen, Silvia. *IPv6 Essentials.* 2002. Explica en detalle el protocolo, muy útil en temas como QoS, Movilidad y Multicast.

Handley, M. y Jacobson, V. *SDP: Session Description Protocol (RFC 2327).* Abril 1998. De consulta para estudiar los campos SDP.

Hinden, R. y Deering, S. *IPv6 Multicast Address Assignments (RFC 2375).* Julio 1998. Tabla de asignación inicial transcripta.

Hinden, R., Deering, S. y Nordmark, E. *IPv6 Global Unicast Address Format (RFC 3587),* Agosto 2003. Muestra el formato de direcciones unicast actual, de donde se extrae la explicación de cada campo.

Hinden, R., O'Dell, M. y Deering, S. *An IPv6 Aggregatable Global Unicast Address Format (RFC 2374).* Julio 1998. Muestra el complejo sistema de direcciones agregable pensando inicialmente para IPv6. OBSOLETA.

International Telecommunication Union - Telecommunication Standardization Sector. *Packet-Based Multimedia Communications Systems (H.323).* Noviembre 2000a. Definición completa de los elementos de un sistema H.323. La mayor parte de lo mostrado en el trabajo fue tomado de esta.

International Telecommunication Union - Telecommunication Standardization Sector. *Protocolos de señalización de llamada y paquetización de trenes de medios para sistemas de comunicación multimedios por paquetes (H.225.0).* Noviembre 2000b. Sólo para consulta de algunos formatos de mensajes, organizacion jerárquica y tratamiento de RTP/RTCP.

Malkin, G. y Minnear, R. *RIPng for IPv6 (RFC 2080).* Enero 1997. Especifica los cambios al protocolo original para funcionar en IPv6.

Miller, Mark A. *Voice over IP Technologies. Building the Converged Network.* 2002. Guía completa de protocolos, implementaciones y explicación de cómo funciona un sistema VoIP.

Morton, David. *Understanding IPv6.* PC Network Advisor (Issue 83). Mayo 1997. Artículo que resume las principales características de IPv6.

Nichols, K., Blake, S., Baker, F., Black, D. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (RFC 2474).* Diciembre 1998.

Palet Martinez, Jordi. *Tutorial de IPv6.* Consulintel e IPv6 Forum. http://www.consulintel.es/Html/ForoIPv6/Documentos/Tutorial de IPv6.pdf. Completa descripción del protocolo IPv6, sus características e historia.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. y E. Schooler, E. *SIP: Session Initiation Protocol (RFC 3261).* Junio 2002. Descripción completa del protocolo, estructura, funciones, comportamiento, operación, etc.

Schulzrinne, H., Casner, S., Frederick, R. y Jacobson, V. *RTP: A Transport Protocol for Real-Time Applications (RFC 3550).* Julio 2003. Se tomaron conceptos generales, formato de encabezado y algunos datos complementarios.

Sisalem, Dorghan y Kuthan, Jiri. *Understanding SIP.* http://www.iptel.org/sip/siptutorial.pdf. Completo resumen de SIP, con ejemplos y gráficos de funcionamiento.

Teitelbaum, Ben. *VoIPv6.* ESCC Internet 2 Techs Workshop, Febrero 2003. http://people.internet2.edu/~ben/talks/20030204-VoIPv6.pdf. Describe ventajas de VoIPv6 y aporta un listado de aplicaciones, de donde surgió Bonephone.

**Otra bibliografía de consulta para la elaboración de este trabajo:**

Barlow, John y Kinham, Stephen. *Presentación de VoIP Overview and Futures.* Enero 2003. Presenta un sistema de VoIP comercial. No aporta nada específico a este trabajo.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. y Weiss, W. *An Architecture for Differentiated Services (RFC 2475).* Diciembre 1998. Funcionamiento de la arquitectura DS. Consulta.

Bonfiglio, S., Bortolotto, H. y Tolosa, D. *Trabajo de Grado: Encapsulamiento de IPv6 en Redes ATM.* 1999. http://www.linti.unlp.edu.ar/trabajos/tesisDeGrado/Redes ATM/Tesis.doc. Para referencia de ATM y cuestiones formales.

Comer, Douglas. *TCP/IP. Principios básicos, protocolos y arquitectura.* 1996. Libro básico de consulta de TCP, UDP, IPv4, ICMP, etc.

Deering, S. y Hinden, R. *IP version 6 Adressing Architecture (RFC 3513).* Abril 2003. Explicación del nuevo formato de direcciones IPv6, útil para entender cómo está dividida jerárquicamente la red, y para ver qué diferencias con el anterior esquema se plantean.

Ericson Telebit. *Voice over IPv6 an IP Telephony Prototype.* Describe VoIPv6 y acompaña interesantes referencias.

Eurescom. *IPv6 - New Opportunities for European Public Network Operators. Descripción de un sistema comercial de VoIPv6.* No aporta nada especifico a este trabajo.

Hagino, Jun-ichiro itojun, Ettikan, K. *An analysis of IPv6 anycast.* Junio 28, 2003. http://www.ietf.org/internet-drafts/draft-ietf-ipngwg-ipv6-anycast-analysis-02.txt.

ITU. *Suggested list of Questions for Discussion.* IP Telephony Workshop. Mayo 2000. http://www.itu.int/osg/spu/ni/iptel/workshop/iptel-questions.pdf. Aporta algunos datos respecto de voip.

Olson, S., Camarillo, G. y Roach, A. B. *Support for IPv6 in Session Description Protocol (RFC 3266).* Junio 2002. Describe pequeños agregados a SDP para soportar IPv6.

Palet Martínez, Jordi. *Situación Mundial de IPv6.* Consulintel e IPv6 Forum. http://www.consulintel.es/Html/ForoIPv6/Documentos/Situación Mundial de IPv6.pdf Describe organismos, implementaciones y problemática de IPv6. No aporta nada específico a este trabajo.

Rekhter, Y. y Li, T. *A Border Gateway Protocol 4 (BGP-4) (RFC 1771).* Marzo 1995. Descripción del protocolo visto en 1.4.3.

Tanenbaum, Andrew. *Redes de ordenadores.* 1991. Modelo de arquitectura de red OSI.

Vovida Open Communication Aplication Library. *Manual de VOCAL versión 1.4.0.* Software similar al Bonephone, descartado por la plataforma que requiere.

Woclawsky, J. *The use of RSVP with IETF Integrated Services (RFC 2210).* Septiembre 1997. Explica cómo usar el protocolo RSVP dentro del framework de QoS Integrated Services. Consulta.

DONACION Recibido
$
Fecha 16-10-07
Inv. E.....Inv. B. 0029399

TES
04/13
DIF-02939
SALA

UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMATICA
Biblioteca
50 y 120 La Plata
catalogo.info.unlp.edu.ar
biblioteca@info.unlp.edu.ar


DIF-02939