

TRABAJO ESPECIAL DE GRADO

IMPLEMENTACION DE UNA UNIDAD DE CONTROL MULTIPUNTO (MCU) BASADO EN CODIGO ABIERTO SOBRE LA RED DE DATOS DE LA UNIVERSIDAD CENTRAL DE VENEZUELA

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Ruiz, Maximiliano A.
para optar al título de
Ingeniero Electricista

Caracas, 2011

TRABAJO ESPECIAL DE GRADO

IMPLEMENTACION DE UNA UNIDAD DE CONTROL MULTIPUNTO (MCU) BASADO EN CODIGO ABIERTO SOBRE LA RED DE DATOS DE LA UNIVERSIDAD CENTRAL DE VENEZUELA

Tutor Académico: Prof. Luis J. Fernández

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Ruiz, Maximiliano A.
para optar al título de
Ingeniero Electricista

Caracas, 2011

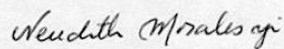
CONSTANCIA DE APROBACIÓN

Caracas, 04 de noviembre de 2011

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Maximiliano A. Ruiz titulado:

**“IMPLEMENTACIÓN DE UNA UNIDAD DE CONTROL MULTIPUNTO (MCU)
BASADO EN CODIGO ABIERTO SOBRE LA RED DE DATOS DE LA
UNIVERSIDAD CENTRAL DE VENEZUELA”**

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Comunicaciones, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.


Prof. Neudith Morales
Jurado


Prof. Carlos Moreno
Jurado


Prof. Luis Fernández
Tutor Académico

DEDICATORIA

A Dios...
A mi abuela...
A mi mama y hermano...

RECONOCIMIENTOS Y AGRADECIMIENTOS

Primeramente, quiero agradecer a mi **Dios** y Señor, por todas las bendiciones que ha dado a mi vida, y permitir llegar a este momento clave de mi vida, en la cual ha sido mi socio, compañero, amigo, padre. La gloria sea para ti por siempre.

A mis amigos, que compartieron conmigo estos años de penurias y alegrías:

Sergio, Roberto, Jorge, Alfonso, Juan Alejandro, Edgardo, Ángel, Manuel, “brothers in arms”, los cuales en los momentos difíciles estuvieron aupándome y ayudando a revelar lo mejor de mí.

A **María Auxiliadora**, por la mística de trabajo que proyecta en el departamento de Comunicaciones, y su soporte incondicional con los alumnos de la Escuela de Ingeniería Eléctrica.

Agradezco a **Geraldine**, por haberme sacado de ese bache en donde estaba y darme el empujón necesario en estos últimos años de carrera y toda su colaboración prestada (gracias por lo dado y lo prestado).

Agradezco a todos los profesores que me impartieron clase, en especial a mi profesor tutor, **Luis Jacinto**, me ha mostrado que el mundo puede cambiar sin uno olvidarse de sus principios.

En especial quiero agradecerle a mi amiga y compañera, **Sarith**, y a su familia, pues esta tesis no hubiese sido concluida a no ser por ella. (mani te quiero y te debo una).

Abuelos, **Luis Fernando y Clara**, mama **Clara** y **Jonathan**, a Uds. les va todo mi reconocimiento, agradecimiento y amor. Dios no me pudo dar mejor familia que Uds.(los quiero).

Ruiz, Maximiliano A.

**IMPLEMENTACION DE UNA UNIDAD DE CONTROL
MULTIPUNTO (MCU) BASADO EN CODIGO ABIERTO SOBRE
LA RED DE DATOS DE LA UNIVERSIDAD CENTRAL DE
VENEZUELA**

Profesor guía: Luis Jacinto Fernández. Tesis. Caracas. UCV. Facultad de Ingeniería Eléctrica. Ingeniero Electricista, Opción Comunicaciones. Trabajo de Grado. 2011. 92h. + anexos.

Palabras Clave: MCU; Estándar H.323; VOIP; Videoconferencia; software libre; código abierto.

Resumen. El presente Trabajo Especial de Grado, tiene como objetivo principal implementar una Unidad de Control Multipunto (MCU) la cual fungirá como controladora y gestionadora de salas de videoconferencias, manejando el flujo de datos desde y hacia los terminales y preparando este mismo flujo para su entrega al Gateway el cual sería la parte restante para la solución total a los requerimientos de interoperabilidad entre esta red y red de datos de la Universidad Central de Venezuela. La implementación de esta unidad de control multipunto, habilitará cosas tales como reducción del presupuesto (costos de los equipos terminales, así como de los licenciamientos de protocolos) debido al desarrollo del software libre basado en una codificación abierta. También proveerá comunicación instantánea en tiempo real entre varios participantes como pueden ser profesores, grupos de estudiantes, autoridades y/o cualquier miembro de la comunidad universitaria. El desarrollo de este MCU se realizó dentro del marco del estándar H.323 de la ITU y consta de las siguientes partes: en principio, a partir de los requisitos planteados, se realizará una propuesta de un sistema que pueda ser instalable modularmente y escalable en el tiempo. Luego, se establecerá el método de implementación, y por último, serán presentadas las pruebas de comprobación y validación de la funcionalidad del software.

CONTENIDO

	Pag.
CONSTANCIA DE APROBACIÓN	iii
DEDICATORIA	iv
RECONOCIMIENTOS Y AGRADECIMIENTOS	v
RESUMEN	vi
INDICE GENERAL	vii
LISTA DE FIGURAS	xi
LISTA DE TABLAS	xiv
ACRÓNIMOS	xv
INTRODUCCIÓN	1
CAPÍTULO I	
1. DESCRIPCIÓN DEL PROYECTO	3
1.1 Planteamiento Del Problema.....	3
1.2 Justificación.....	3
1.3 Objetivos.....	5
1.3.1 Objetivo General.....	5
1.3.2 Objetivos Específicos.....	5
CAPITULO II	
2. MARCO TEÓRICO	6

2.1	Videoconferencia.....	6
2.1.1.	Tipos de Videoconferencia.....	7
2.1.2.	Modalidades de Videoconferencia.....	7
2.1.3	Elementos Básicos de un Sistema de Videoconferencia.....	8
2.1.4	Codificación-Decodificación de audio y video.....	10
2.2	TCP.....	14
2.3	UDP.....	16
2.4	Protocolo De Internet (IP).....	16
2.5	Recomendación H.323.....	17
2.5.1	Protocolos.....	17
2.5.2	Modelos de Señalización.....	19
2.5.3	Componentes.....	21
2.5.4	Flujo de comunicación en H.323.....	23
2.6	OPENH.323.....	26
2.6.1	OpenH.323plus.....	27
2.7	Calidad de Servicio en Redes IP.....	35
2.7.1	Latencia.....	36
2.7.2	Jitter.....	37
2.7.3	Eco.....	39
2.7.4	Packet Loss.....	39

2.7.5 Throughput.....	40
2.8 Ekiga.....	40
2.9 Ubuntu.....	42
2.10 Iperf.....	44
CAPÍTULO III	
3. METODOLOGÍA.....	46
3.1 Fase 1. Estudio Documental.....	46
3.2 Fase 2. Análisis de Arquitectura.....	48
3.2.1 Entorno de Desarrollo de Pruebas.....	48
3.2.2 Selección de Software	50
3.3 Fase 3. Implementación de OpenMCU.....	53
3.4 Fase 4. Elaboración del Informe Final	72
CAPÍTULO IV	
4. RESULTADOS.....	73
4.1 Pruebas Iniciales	73
4.2 Interconexión con otros Equipos.....	79
4.3 Pruebas de Rendimiento con Iperf	82
CAPÍTULO V	
5. CONCLUSIONES.....	85

CAPÍTULO VI

6. RECOMENDACIONES.....	88
REFERENCIAS BIBLIOGRÁFICAS.....	90
BIBLIOGRAFÍA.....	92
ANEXOS.....	93

LISTA DE FIGURAS

	Pag.
Figura 1. Pila de protocolos de H.323.....	18
Figura 2. Señalización directamente a través del Terminal.....	19
Figura 3. Señalización encaminada por el gatekeeper excepto los mensajes de control de conferencia.....	20
Figura 4. Señalización Encaminada por el Gatekeeper.....	20
Figura 5. Zona de control H.323.....	21
Figura 6. Establecimiento y finalización de una llamada H.323.....	25
Figura 7. Esquema de clases de la librería OpenH323.....	30
Figura 8. Logo OpenH323.....	33
Figura 9. Logo OpenMCU ..	34
Figura 10. Software Ekiga ..	41
Figura 11. Logo de la Distribución GNU/Linux Ubuntu ..	42
Figura 12. Organigrama de jerarquía lógica de red de datos UCV.....	49
Figura 13. Configuraciones típicas de ISABEL.....	51
Figura 14. Topología de red tipo estrella.....	53
Figura 15. Instalación de Ubuntu.....	54
Figura 16. Consola OpenMCU.....	55
Figura 17. Orígenes del Software.....	56

Figura 18. Menú de selección de códecs de audio.....	67
Figura 19. Menú de selección de códecs de video.....	69
Figura 20. Parámetros de Openmcu	71
Figura 21. Primera conexión entre OpenMCU y NetMeeting	73
Figura 22. NetMeeting conectado a OpenMCU.....	74
Figura 23. OpenMCU con errores de compilado.....	75
Figura 24. Prueba de videoconferencia OpenMCU con dos participantes.....	76
Figura 25. Estatus de OpenMCU en videoconferencia de prueba con dos participantes.....	76
Figura 26. Prueba de videoconferencia OpenMCU con tres participantes.....	77
Figura 27. Estatus de videoconferencia OpenMCU con tres participantes.....	77
Figura 28. Prueba de videoconferencia OpenMCU con cuatro participantes.....	78
Figura 29. Estatus de videoconferencia OpenMCU con cuatro participantes.....	78
Figura 30. Estatus de la OpenMCU en la comunicación entre equipos SIP y terminales H.323 (Audio) interconectando el Gateway con la OpenMCU de la UCV.....	79
Figura 31. Prueba de llamada entre equipos SIP y terminales H.323 (Audio y Video) interconectando el Gateway con la MCU de la UCV con 2 participantes.....	80
Figura 32. Estatus de llamada entre equipos SIP y terminales H.323	

(Audio y Video) interconectando el Gateway con la MCU de la
UCV con 2 participantes.....80

Figura 33. Prueba de llamada entre OpenMCU y Equipo Polycom81

LISTA DE TABLAS

	Pag.
Tabla 1. Tipos de Videoconferencia de acuerdo a su Organización.....	9
Tabla 2. Códecs de Audio.....	12
Tabla 3. Tabla para la determinación principal-subordinado H.245.....	33
Tabla 4. Características de los códecs de audio seleccionados.....	67
Tabla 5. Ancho de Banda por llamada G.711.....	68
Tabla 6. Ancho de Banda por llamada GSM.....	68
Tabla 7. Características de los códecs de audio seleccionados.....	69
Tabla 8. Medición de latencia, jitter y perdidas entre OpenMCU y Cliente Ekiga...	82

ACRÓNIMOS

ATM: Modo de Transferencia Asíncrona
CIF: Common Intermediate Format
Códec: abreviación de codificador-decodificador
DNS: Domain Name System
EP: End Point
FTP: Protocolo de Transferencia de Archivos
GCF: Gatekeeper ConFirmation
GKRCS: Gatekeeper Routed Call Signaling
GRJ: Gatekeeper ReJect
GRQ: Gatekeeper ReQuest
ILS: Internet Locator Service
IP: Internet Protocol
ISDN: Integrated Services Digital Network
ITU: Unión Internacional de Telecomunicaciones
ITU: Unión Internacional de Telecomunicaciones
LAN: Local Area Network
MBone: Multicast Backbone
MC: Controlador Multipunto
MCU: Unidad Multipunto de Control
MEGACO: Media Gateway Control Protocol
MP: Procesador Multipunto
NAT: Network Address Translation
OPAL: Open Phone Abstration Library
OSI: Organización Internacional para la Estandarización
PDU: Packet Data Unit - Unidades de Datos de Protocolo
POSTS: Plain Old Telephone Service
PSTN: Red Pública de Telefonía Conmutada

QCIF: Quarter Common Intermediate Format

QoS: Quality of Service

RAM: Random Access Memory

RAS: Registration, Admission and Status

RDSI: Red Digital de Servicios Integrados

RIP: Registration In Progress

RRJ: Registrarion ReJect

RRQ: Registration ReQuest

RSVP: Resource Reservation Protocol

RTCP: Real Time Control Protocol

RTP: Real Time Protocol

SIP: Session Initiation Protocol

SS7: Signalling System number 7

VoIP: Voice Over IP

INTRODUCCIÓN

Es un hecho bastante conocido que la mayoría de la información procesada por el cerebro es de carácter visual. De hecho casi 2/3 de la actividad neuronal del cerebro, bien sea memoria, procesamiento y/o adquisición de datos están referidos al área visual. Es por esto, que la comunicación cara a cara, es la más efectiva, pues obtenemos mayor información de las expresiones faciales, más que de sus palabras o calidad de voz combinadas. Se ha determinado que cuando se establece la comunicación en donde los interlocutores puedan observarse el rostro, sólo el siete por ciento de lo que es comunicado es transferido por el significado de las palabras. Otro treinta y ocho por ciento proviene de cómo las palabras son dichas. Eso deja al cincuenta y cinco por ciento restante de la comunicación, tomar la forma de señales visuales.

Es debido a esto que a nivel de enseñanza, la transmisión de conocimiento sea de manera presencial. En lugares donde el acceso sea restringido por diferentes causas, la única manera de presencia es la telepresencia. Pues bien, la forma más conocida y utilizada de telepresencia es la videoconferencia, la cual permite llevar a cabo el encuentro de varias personas ubicadas en sitios distantes, y establecer una conversación como lo harían si todas se encontraran reunidas en una sala.

El aumento de población, la creciente escasez de recursos, la constante especialización, hace de la videoconferencia una alternativa óptima en el sector educativo para cubrir las progresivas demandas del mundo actual. La videoconferencia ofrece hoy en día una solución accesible a esta necesidad de comunicación, con sistemas que permiten el transmitir y recibir información visual y sonora entre puntos o zonas diferentes evitando así los gastos y pérdida de tiempo que implican el traslado físico de la persona, todo esto a costos cada vez más bajos y con señales de mejor calidad.

Unido a estos hechos, debido a la gran expansión de Internet, las clases por videoconferencia están teniendo un gran auge en los últimos años y ya se están volviendo imprescindibles en facultades de todo el mundo, especialmente en el área de la salud. La Universidad Central de Venezuela cuenta con una plataforma de videoconferencias basadas en equipos Polycom, que si bien está apoyada en estándares internacionales, hace uso de protocolos propietarios que limitan su expansión a la totalidad de la UCV debido a los altos costos de los equipos terminales.

En la búsqueda de alternativas que permitan solventar el problema antes mencionado, se pretende estudiar la posibilidad de la implementación de una plataforma de videoconferencias basada en código abierto, mediante la cual sea posible extender y sobrepasar los límites de los protocolos propietarios y equipos terminales haciendo accesible a la población estudiantil de la UCV un sistema de videoconferencias de bajo costo y de suficiente capacidad.

Parte de esta plataforma es una Unidad de Control Multipunto (MCU), la cual se encargará de la administración y gestionar la comunicación entre diferentes terminales, y esta parte es la primera en estudio de otros fragmentos necesarios para la implementación total de la plataforma de videoconferencias necesarias para nuestra casa de estudios.

CAPÍTULO I

1. DESCRIPCIÓN DEL PROYECTO

1.1 Planteamiento Del Problema

Actualmente, la Universidad Central de Venezuela cuenta con un sistema de videoconferencias basado en equipos de la marca Polycom. Son equipos dedicados, es decir, su uso está destinado solo al apartado de videoconferencia. Esto imposibilita su aprovechamiento en otros ámbitos, a pesar de las capacidades de procesamiento y control. Por tanto, esto se convierte en un factor de peso a la hora de invertir en equipos con una especialización de esta naturaleza.

Adicional a esto, la cancelación periódica de las licencias por concepto de uso de software propietario y mantenimiento anual del equipo, debido al uso por parte de la universidad de equipos terminales de la marca Polycom, hace que la expansión del sistema de videoconferencias de la Universidad Central de Venezuela se vea severamente restringido.

Estas son razones por las cuales surge la necesidad de integrar al sistema existente de videoconferencias otras plataformas existentes basadas en software de código abierto, eliminando los gastos por cobro de licencias y facilitando la expansión de la red de videoconferencias a todo el rango de la universidad.

1.2 Justificación

El software que se ha desarrollado en los últimos años (propietario o código abierto) ha hecho posible el uso de computadores personales como equipos terminales o equipos servidores. Este hecho da un gran radio de acción pues la consecución de un ordenador personal representa poco problema. La red de videoconferencias existente en la universidad se basa en equipos de varias marcas propietarias como ya se

expresó anteriormente. Atendiendo este hecho, estos equipos a pesar de tener software privativo de índole cerrado, debe cumplir con estándares internacionales.

Basándose en este criterio y posterior estudio del equipo, el sistema usa el estándar H.323 para transmisión de video y siguiendo los lineamientos de este estándar, la red alternativa debe poseer al menos un Gateway que permita la adaptación de la red Polycom a nuestra red, la Unidad de Control Multipunto (MCU), la cual se encargará de la gestión de los flujos de audio y vídeo. También permite la creación de salas de multiconferencia en las que un administrador puede estar prestando servicio a varios terminales al mismo tiempo, y los equipos terminales, todos estos con un software específico para cada tarea sobre una PC como hardware.

Como parte de la solución al problema se propone la implementación de la MCU, la cual fungiría como controladora y gestiona de salas de videoconferencias, controlando el flujo de datos desde y hacia los terminales y preparando este mismo flujo para su entrega al Gateway el cual sería la parte restante para la solución total a los requerimientos de interoperabilidad entre esta red y el sistema Polycom.

La implementación de esta unidad de control multipunto, habilitará cosas tales como:

- Reducción del presupuesto (costos de los equipos terminales, así como de los licenciamientos de protocolos y contratos de mantenimiento) debido al desarrollo del software libre basado en una codificación abierta.
- Comunicación instantánea en tiempo real entre varios participantes como pueden ser profesores, grupos de estudiantes, autoridades y/o cualquier miembro de la comunidad universitaria.
- Eliminación de gastos de desplazamiento (pasajes de avión, taxis, autos de alquiler y/o kilometraje).
- Eliminación de gastos de hotel y alimentación.

1.3 Objetivos

1.3.1 Objetivo General

Implementación de una unidad de control multipunto (MCU) basado en código abierto sobre la red de datos de la Universidad Central de Venezuela.

1.3.2 Objetivos Específicos

- Estudiar el funcionamiento de una unidad de control multipunto para efectuar su implementación en una red.
- Analizar y estudiar el estándar H.323 junto con sus códecs complementarios.
- Implementación de la unidad de control multipunto (MCU).
- Realizar pruebas de validación, para la evaluación del desempeño y funcionalidad de la unidad de control multipunto (MCU).
- Recomendar, mediante las pruebas de validación, el uso de un software cliente H.323 que presente la mayor compatibilidad con la MCU planteada.

CAPITULO II

2. MARCO TEÓRICO

2.1 Videoconferencia

La videoconferencia no es más que una comunicación digital, en donde existe intercambio de audio y video entre 2 o más personas, las cuales están en locaciones remotas, sucediendo todo esto en tiempo real y full dúplex. Además de poder verse y oírse, los participantes tienen acceso a intercambio de documentos e interacción personal, tal cual una conferencia en persona.

Los sistemas de videoconferencia operan sobre los mismos principios. Las características esenciales con las que cuentan, es la transmisión y recepción digital. Estos enlaces pueden establecerse sobre satélite, cable, fibra óptica etc., y sus velocidades de conexión pueden ir desde los 64 Kbps, hasta 2 Mbps de acuerdo con el ancho de banda que se tenga.

A principios de la década pasada se necesitaban 6 Mbps para transmitir la Videoconferencia, requerimiento que a finales de los años 80 pasó a 2 Mbps. Desde principios de la década actual, es posible la transmisión de videoconferencia sobre líneas digitales conmutadas de menor velocidad. Los datos se comprimen en el equipo origen, viajan comprimidos a través del circuito de comunicación y se descomprimen en el destino. La calidad de las imágenes que percibimos está en función del nivel de compresión y de la capacidad de transmisión de datos. [1]

2.1.1. Tipos de Videoconferencia

En realidad existen dos tipos según su enlace:

- Punto a Punto: la videoconferencia se realiza entre dos únicos terminales. La negociación debe ser bilateral.
- Multipunto: esta videoconferencia se realiza entre más de 2 terminales, es necesario entonces un equipo capaz de gestionar la comunicación entre los terminales; este equipo recibe el nombre de MCU (Unidad de Control Multipunto). La unidad MCU es la encargada de recibir la señal de todos los equipos, controlando el flujo de datos desde y hacia los terminales, distribuyéndolos este flujo a todos los terminales de manera que todos los terminales puedan participar al mismo tiempo.

2.1.2. Modalidades de Videoconferencia

Se pueden encontrar diferentes clasificaciones en función de la variable que se tenga en cuenta. Se exponen algunas de ellas:

Atendiendo al tipo de equipo con el que se realiza la videoconferencia, se pueden encontrar tres tipos:

- Sistemas para PC: es el supuesto en que dos personas se comunican transmitiendo señales de audio y video, y en función de la tecnología utilizada, también se pueden transmitir archivos.
- Sistemas de Sobremesa: como en el caso anterior, pero diseñados para grupos medianos o reducidos
- Sistemas de Sala: es una multivideoconferencia donde pueden participar un alto número de personas activamente, transmitiendo señales de audio, video y archivos.

En función del número de sitios enlazados:

- Cuando son dos los sitios enlazados se denomina conferencia punto a punto. La videoconferencia se realiza entre dos únicos usuarios.
- Cuando son más de dos los sitios enlazados, se denomina multipunto. Es necesario un equipo que sea capaz de unir todos los terminales que participen en la multiconferencia.

Según el tipo de participación:

- Uno a uno: Videoconferencia con dos participantes
- Uno a varios: situación en la que uno de los participantes difunde su información al resto, pero el resto no produce ninguna retroalimentación.
- Varios a Varios: Videoconferencia con tres o más participantes en las que todos difunden su información al resto.

2.1.3 Elementos Básicos de un Sistema de Videoconferencia

Por lo general, para abordar el estudio de la videoconferencia, se suele subdividir su entorno en tres bloques que constituyen los elementos básicos de un sistema de videoconferencia. Estos elementos son:

- La Red de comunicaciones: para poder realizar cualquier tipo de comunicación es necesario contar primero con un medio que transporte la información del transmisor al receptor y viceversa o paralelamente (en dos direcciones). En los sistemas de videoconferencia se requiere que este medio proporcione una conexión digital bidireccional y de alta velocidad entre los dos puntos a conectar. El número de posibilidades existentes de redes de comunicaciones es muy grande, pero se debe

señalar que la opción particular depende enteramente de los requerimientos de los usuarios.

- **Sala de Videoconferencia:** la sala de videoconferencia es el área especialmente acondicionada en la cual se alojará el personal de videoconferencia, así como también, el equipo de control, de audio y de video, que permitirá capturar y controlar las imágenes y los sonidos que habrán de transmitirse. En función del tipo de videoconferencia a realizar, el esquema organizativo de la sala cambiará, en la siguiente tabla se muestran tres ejemplos de organización de una sala de videoconferencia.

Tabla 1. Tipos de Videoconferencia de acuerdo a su Organización

Fuente: Elaboración Propia

Configuración	Esquema físico	Dispositivo de visualización típico	Cámaras	Cantidad de participantes	Micrófonos
Sala preparada	Salón de actos	Proyectores digitales o monitores grandes	Generalmente varias	Gran grupo	Varios
Rollabout	Módulo portátil en una sala preparada	Proyector o TV (según tipo de rollabout).	Una o dos	Pequeño grupo	Normalmente uno
Escritorio PC	Sistema en un ordenador	Monitor	WebCam	Individual	Uno

- **El Códec (Codificador/Decodificador):** es el dispositivo que se encarga de codificar las señales analógicas en digitales para que se transmitan a través de la red, o de decodificarlas cuando llegan, para poder verlas y oírlas. En el mercado existen equipos modulares que junto con el códec, incluyen los equipos de video, de audio y de control, así como también los periféricos que pueden servir de ayuda en la comunicación, como: cámara para documentos, magnetoscopio, tableta gráfica, pizarra electrónica, etc.

2.1.4 Codificación-Decodificación de audio y video

Un Codificador-Decodificador (CODEC), es aquel conjunto de algoritmos que captura las señales de audio y video y las comprime para ser transmitidas a un sitio remoto. Existen diversos métodos para transformar una señal analógica en digital, todos estos métodos están regidos por estándares. Además, el códec codifica la secuencia de datos. Esto hace que exista un ahorro de ancho de banda. Particularmente es importante en enlaces de poca capacidad. Al ser procesos de compresión dedicados existen específicamente códecs para audio y para video, de los cuales se hará una reseña a continuación.

Códec de audio

El algoritmo que permite comprimir y digitalizar una señal de audio y a su vez permite recuperar la información codificada en audio digital para convertirla en una sucesión de sonidos es denominado códec de audio. Dentro de los códecs de audio están los códecs enfocados para la transmisión de voz, entre ellos tenemos:

- **G.711:** Es un estándar de la ITU-T para la compresión de audio. Este estándar es usado principalmente en telefonía, y fue liberado para su uso en el año 1972. G.711 es un estándar para representar señales de audio con frecuencias de la voz humana, mediante muestras comprimidas de una señal de audio digital con una tasa de muestreo de 8000 muestras por segundo. El códec G.711 proporcionará un flujo de datos de 64 kbit/s. Para este estándar existen dos algoritmos principales, el μ -law (usado en Norte América y Japón) y el A-law (usado en Europa y el resto del mundo).
- **G.728:** Estándar ITU-T utilizado en VoIP que codifica una señal de audio de con un ancho de banda de 3.4 KHz para transmitir a 16 Kbps. Es utilizado en sistemas de videoconferencia que funcionan a 56 Kbps o 64 Kbps. [3]
- **G.729:** Es un algoritmo de compresión de datos de audio para voz que comprime audio de voz en trozos de 10 milisegundos. La música o los tonos tales como los

tonos de DTMF o de fax no pueden ser transportados confiablemente con este códec, y utilizar así G.711 o métodos de señalización fuera de banda para transportar esas señales. Se usa mayoritariamente en aplicaciones de Voz sobre IP por sus bajos requerimientos en ancho de banda. El estándar G.729 opera a una tasa de bits de 8 kbit/s.[4]

- **GSM:** El GSM (sistema global para las comunicaciones móviles) es un popular estándar del sistema del teléfono portátil fuera de los E.E.U.U. El GSM incluye un códec, a menudo apenas designado el GSM al discutir códecs. El códec “de exploración completa” del GSM se nombra RPE-LTP (predicción a largo plazo de la excitación regular del pulso). Este códec utiliza la información de muestras anteriores (esta información no cambia muy rápidamente) para predecir la muestra actual. La señal de entrada se divide en bloques de 20ms. Cada bloque contiene 260 bits ($188 + 36 + 36 = 260$). Esto es razonable, ya que $260 \text{ bits} / 20 \text{ ms} = 13\,000 \text{ bits/s} = 13 \text{ kbits/s}$, es decir, opera a 13 kbits / s. Como comparación, las redes de viejos teléfonos públicos usan una codificación de señal de voz con una tasa de bits de 64 kbit/s. A pesar de esto no hay diferencia significativa en la calidad de la misma. Esto se debe a que las frecuencias de la voz humana permiten filtrar y comprimir mucha información, ya que al comunicarse con otra persona no se necesita mucha calidad para entender lo que se habla. Por el contrario, si escuchamos una canción o algo similar por un teléfono, escucharemos bastante mal. Pero en cuanto a la voz, el códec GSM es una gran opción para comprimir la información, ya que no existe una gran diferencia, como en el caso descrito anteriormente de una tasa de bits de 64 kbps a una de 13 Kbps. [5][6]

- **SPEEX:** El proyecto Speex fué iniciado para atender la necesidad de un códec de voz que fuera de código libre y libre de patentes de software. Éstas son condiciones esenciales para ser usado por cualquier software de código libre. Vorbis ya hace de audio general, pero no está realmente optimizado para la voz. También a diferencia de muchos otros códecs de voz, Speex no está diseñado para teléfonos celulares, pero si para Voz sobre IP (VoIP) y compresión basada en archivos.

Las metas en el diseño eran permitir buena calidad en la voz y bajo “bit-rate” (desafortunadamente no al mismo tiempo). Buena calidad también significaba tener soporte para wideband (frecuencia de muestreo de 16 kHz) además de narrowband (calidad de teléfono, frecuencia de muestreo de 8 kHz). El diseño orientado a VoIP en lugar de ser diseñado para teléfonos celulares obliga a Speex a ser robusto a la pérdida de paquetes, sin corromperlos, de esta manera los paquetes que llegan, llegan correctos al terminal.[7]

Tabla 2. Códecs de Audio

Fuente: Elaboración Propia

Nombre	Estandarizado	Descripción	Bit rate (kb/s)	Sampling rate (kHz)	Frame size (ms)	Observaciones
G.711	ITU-T	Pulse code modulation (PCM)	64	8	Muestreada	Dos versiones u-law (US, Japan) y a-law (Europa) para muestrear la señal
GSM 06.10	ETSI	RegularPulse Excitation LongTerm Predictor (RPE-LTP)	13	8	22.5	Usado por la tecnología celular GSM
G.728	ITU-T	16 kbit/s low-delay code excited linear prediction	16	8	2.5	CELP.
G.729	ITU-T	8 kbit/s conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)	8	8	10	Bajo retardo (15 ms)
Speex			8, 16, 32	2.15-24.6 (NB) 4-44.2 (WB)	30 (NB) 34 (WB)	

Códec de vídeo

Un códec de vídeo es un programa que permite comprimir y descomprimir vídeo digital. Normalmente los algoritmos de compresión empleados conllevan una pérdida de información. Debido a la gran cantidad de información que conlleva

apenas unos minutos de video, se comprime en el momento de guardar la información hacia un archivo y se descomprime, en tiempo real, en el momento de la visualización. Dentro de los códec orientados para videoconferencia están:

- **H.261:** Es un estándar de la Unión Internacional de Telecomunicaciones (ITU), ratificado en noviembre de 1988. Originalmente diseñado para la transmisión a través de líneas RDSI en el que las tasas de datos son múltiplos de 64 kbit/s. Forma parte de la familia H.26x, estándares de codificación de vídeo del ITU-T Video Coding Experts Group (VCEG). El algoritmo de codificación fue diseñado para ser capaz de operar a tasas de bits de 40 kbit / s, 2 Mbit / s. El estándar soporta dos tamaños de fotograma de vídeo: CIF (352x288 luma con croma) y QCIF (176x144 88x72 con croma) utilizando un esquema de muestreo de 4:2:0.[8]

- **H.263:** H.263 es un estándar de la Unión Internacional de Telecomunicaciones (ITU) para la codificación de vídeos con compresión que data de 1995/1996, documento en el cual también se detalla, en su segunda parte, el estándar MPEG-4 de la Moving Picture Experts Group. H.263 describe un Códec, que en primera línea se concibió para videoconferencias. Está optimizado para una tasa de bits baja (bajo 64 kbit/s, es decir, velocidad ISDN) y un movimiento relativo reducido. Si bien, el propio estándar no define una tasa de bits concreta. H.263 se desarrolló a partir del estándar H.261, el cual sirve de base para el desarrollo de MPEG optimizado para tasas de bits más elevadas. Contiene una componente para una compresión temporal más intensa y funciona preferiblemente con secuencias de vídeos, que presentan cambios reducidos entre imagen e imagen.

- **H.264** o MPEG-4 es una norma que define un códec de vídeo de alta compresión, desarrollada conjuntamente por el ITU-T Video Coding Experts Group (VCEG) y el ISO/IEC Moving Picture Experts Group (MPEG). La intención del proyecto H.264/AVC fue la de crear un estándar capaz de proporcionar una buena calidad de imagen con tasas binarias notablemente inferiores a los estándares previos (MPEG-2, H.263 o MPEG-4 parte 2), además de no incrementar la complejidad de su diseño. Para garantizar un ágil desarrollo de la misma, la ITU-T y

la ISO/IEC acordaron unirse para desarrollar conjuntamente la siguiente generación de códecs de vídeo. El Joint Video Team (JVT) estaba formado por expertos del VCEG y MPEG y nació en diciembre de 2001 con el objetivo de completar el desarrollo técnico del estándar hacia el 2003. La ITU-T planeó adoptar el estándar bajo el nombre de ITU-T H.264 y ISO/IEC bajo el nombre de MPEG-4 Parte 10 Códec de Vídeo Avanzado (AVC) y de aquí surgió el nombre híbrido de H.264/MPEG-4 AVC. Para empezar a programar el código del nuevo estándar adoptaron las siguientes premisas:

La estructura DCT + Compensación de Movimiento de las versiones anteriores era superior a otros estándares y por esto no había ninguna necesidad de hacer cambios fundamentales en la estructura. Algunas formas de codificación de vídeo que habían sido excluidas en el pasado debido a su complejidad y su alto costo de implementación se volverían a examinar para su inclusión puesto que la tecnología VLSI había sufrido un adelanto considerable y una bajada de costos de implementación. Para permitir una libertad máxima en la codificación y evitar restricciones que comprometan la eficiencia, no se contempla mantener la compatibilidad con normas anteriores.

2.2 TCP

TCP (que significa *Protocolo de Control de Transmisión*) es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. En el nivel de aplicación, posibilita la administración de datos que vienen del nivel más bajo del modelo, o van hacia él, (es decir, el protocolo IP). Cuando se proporcionan los datos al protocolo IP, los agrupa en datagramas IP, fijando el campo del protocolo en 6 (para que sepa con anticipación que el protocolo es TCP). TCP es un protocolo orientado a conexión, es decir, que permite que dos máquinas que están comunicadas controlen el estado de la transmisión.

Las principales características del protocolo TCP son las siguientes:

- TCP permite colocar los datagramas nuevamente en orden cuando vienen del protocolo IP.
- TCP permite que el monitoreo del flujo de los datos y así evita la saturación de la red.
- TCP permite que los datos se formen en segmentos de longitud variada para "entregarlos" al protocolo IP.
- TCP permite multiplexar los datos, es decir, que la información que viene de diferentes fuentes (por ejemplo, aplicaciones) en la misma línea pueda circular simultáneamente.
- Por último, TCP permite comenzar y finalizar la comunicación fácilmente[9]

Con el uso del protocolo TCP, las aplicaciones pueden comunicarse en forma segura (gracias al sistema de acuse de recibo del protocolo TCP) independientemente de las capas inferiores. Esto significa que los routers (que funcionan en la capa de Internet) sólo tienen que enviar los datos en forma de datagramas, sin preocuparse con el monitoreo de datos porque esta función la cumple la capa de transporte (o más específicamente el protocolo TCP). Durante una comunicación usando el protocolo TCP, las dos máquinas deben establecer una conexión. La máquina emisora (la que solicita la conexión) se llama cliente, y la máquina receptora se llama servidor. Por eso es que decimos que estamos en un entorno Cliente-Servidor.

Las máquinas de dicho entorno se comunican en modo en línea, es decir, que la comunicación se realiza en ambas direcciones. Para posibilitar la comunicación y que funcionen bien todos los controles que la acompañan, los datos se agrupan; es decir, que se agrega un encabezado a los paquetes de datos que permitirán sincronizar las transmisiones y garantizar su recepción. Otra función del TCP es la capacidad de controlar la velocidad de los datos usando su capacidad para emitir mensajes de tamaño variable. Estos mensajes se llaman segmentos.

2.3 UDP

UDP son las siglas de Protocolo de Datagrama de Usuario (en inglés *User Datagram Protocol*) un protocolo sin conexión que, como TCP, funciona en redes IP.UDP/IP proporciona muy pocos servicios de recuperación de errores, ofreciendo en su lugar una manera directa de enviar y recibir datagramas a través una red IP. Se utiliza sobre todo cuando la velocidad es un factor importante en la transmisión de la información, su uso principal es para protocolos como DHCP, BOOTP, DNS y demás protocolos en los que el intercambio de paquetes de la conexión/desconexión son mayores, o no son rentables con respecto a la información transmitida, así como para la transmisión de audio y vídeo en tiempo real, donde no es posible realizar retransmisiones por los estrictos requisitos de retardo que se tiene en estos casos. [10].

2.4 Protocolo De Internet (IP)

Protocolo para la comunicación en una red a través de paquetes conmutados, es principalmente usado en Internet. Los datos se envían en bloques conocidos como paquetes (datagramas) de un determinado tamaño (MTU). El envío es no fiable (conocido también como “best effort” o mejor esfuerzo); se llama así porque el protocolo IP no garantiza si un paquete alcanza o no su destino correctamente. Un paquete puede llegar dañado, repetido, en otro orden o no llegar. Para la fiabilidad se utiliza el protocolo TCP de la capa de transporte.[11]

La unidad de información intercambiada por IP es denominada datagrama. Tomando como analogía los marcos intercambiados por una red física los datagramas contienen un encabezado y un área de datos. IP no especifica el contenido del área de

datos, ésta será utilizada arbitrariamente por el protocolo de transporte, mencionado anteriormente.[12]

2.5 Recomendación H.323

H.323 es una recomendación del UIT-T, la cual especifica cómo deben ser los sistemas de comunicaciones en los que está envuelto video, audio y trabajo en conjunto de archivos, en donde la red de transporte sea por conmutación de paquetes. En esta recomendación se enumera los componentes la cual incluye endpoints, Gateway, gatekeeper y MCU. Además se enumeran los procedimientos y mensajes para que estos elementos se comuniquen entre sí. Así, los usuarios no se tienen de que preocupar de cómo el equipo receptor actúa, siempre y cuando cumpla este estándar. Por ejemplo, la gestión del ancho de banda disponible para evitar que la LAN se colapse con la comunicación de audio y vídeo también está contemplada en el estándar, esto se realiza limitando el número de conexiones simultáneas.

Este estándar no es un protocolo en sí, más bien es una especificación que engloba a varios protocolos y especifica cómo deben actuar entre sí.

2.5.1 Protocolos

Como se menciona anterior mente la recomendación H.323 es la especificación de interacción de un conjunto de protocolos. En la siguiente figura se muestra la pila de los protocolos más relevantes en H.323.

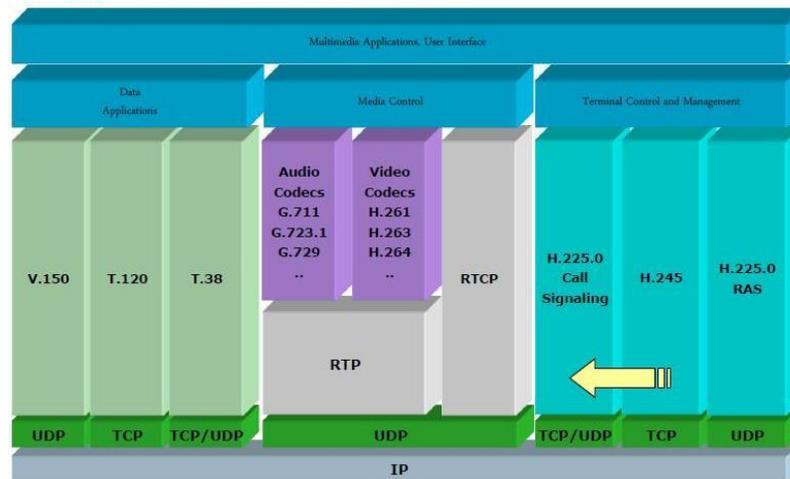


Figura 1. Pila de protocolos de H.323

Fuente: Packetizer.com

• **H.225:** Protocolo utilizado para describir la señalización de llamada, el medio (audio y video), el empaquetamiento de las tramas, la sincronización de tramas de medio y los formatos de los mensajes de control. Se puede dividir en tres partes como lo son:

- **Registro, admisiones y situación (RAS):** Utiliza mensajes H.225.0 para efectuar el registro, las admisiones, cambios en ancho de banda, situación y desconexión entre endpoints. Los mensajes son codificados de acuerdo a las Normas de Codificación de Paquetes (*del inglés Packed Encoding Rules-PER*) de la norma ASN.1.
- **Señalización de la llamada:** Establecimiento, control y finalización de una llamada H.323. La señalización H.225.0 está basada en los procedimientos de establecimiento de llamada de RDSI, Para ello se utilizan mensajes codificados según la recomendación Q.931.
- **Empaquetamiento:** Esta parte de el protocolo H.225. se encarga de la conformación de los paquetes de audio, video y control, así como detección y corrección de errores.

- **H.245:** Protocolo de control para comunicaciones multimedia. Describe los mensajes y procedimientos utilizados para abrir y cerrar canales lógicos para audio, video y datos, capacidad de intercambio, control e indicaciones.

- **Q.931:** Este protocolo fue definido originalmente para señalización en accesos ISDN básico. Es equivalente al ISUP utilizado desde el GW hacia la red PSTN.

- **H.235:** Provee una mejora sobre H.323 mediante el agregado de servicios de seguridad como autenticación y privacidad (criptografía). El H.235 trabaja soportado en H.245 como capa de transporte. Todos los mensajes son con sintaxis ASN.1.

2.5.2 Modelos de Señalización

Hay tres métodos para transmitir la señalización de las llamadas:

- Directamente a través del terminal. Los mensajes de señalización son enviados directamente entre los dos terminales.

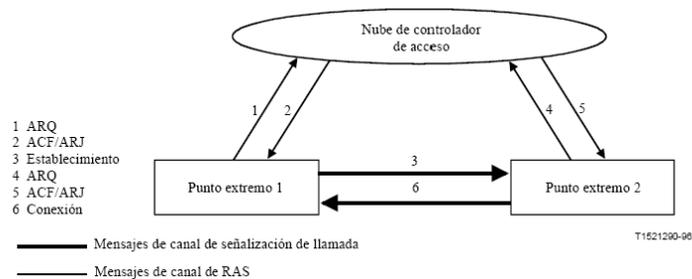


Figura 2. Señalización directamente a través del Terminal

Fuente: Elaboración Propia

- Señalización encaminada por el gatekeeper. Los mensajes H.225.0 de RAS y los H.225.0 de señalización de llamada se encaminan a través del Gatekeeper, mientras que los mensajes de control de conferencia H.245 son encaminados directamente entre los dos terminales. Este método permite al gatekeeper reencaminar el canal de control H.245 a una MCU cuando una conferencia multipunto ad hoc pasa de

conferencia punto a punto a conferencia multipunto. El gatekeeper realiza esta elección. Cuando se utiliza la señalización de llamada de punto extremo directa, el canal de control H.245 sólo puede ser conectado directamente entre los puntos extremos.

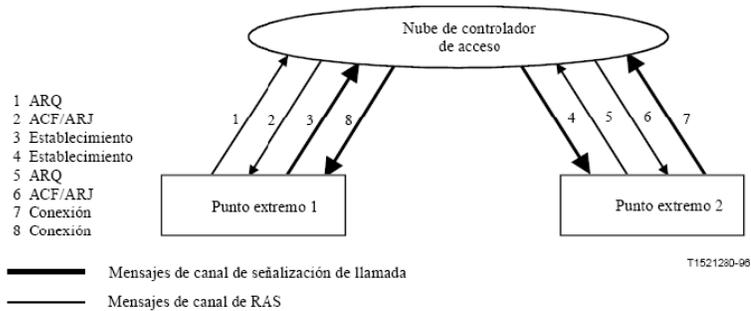


Figura 3. Señalización encaminada por el gatekeeper excepto los mensajes de control de conferencia

Fuente: Elaboración Propia

- Señalización encaminada por el gatekeeper. Los mensajes de señalización entre los terminales son enrutados a través del gatekeeper

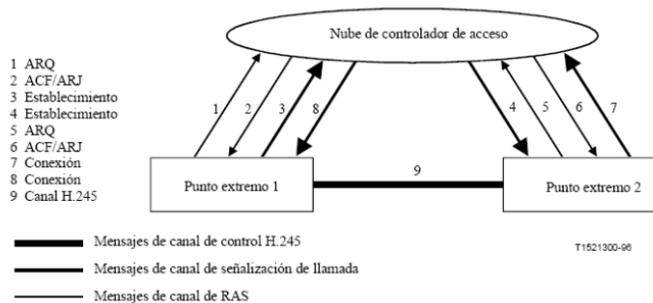


Figura 4. Señalización Encaminada por el Gatekeeper

Fuente: Elaboración Propia

2.5.3 Componentes

Terminales o Endpoints: Un terminal es un extremo de la red que proporciona comunicaciones bidireccionales en tiempo real con otro terminal, bien sea a través de un gatekeeper o una unidad de control multipunto (MCU). Todos los terminales deben soportar la comunicación de voz, mientras que la de vídeo y datos son opcionales. Además, deben soportar la norma H.245 que se emplea para la negociación del uso del canal y sus prestaciones; Q.931 para el establecimiento de la llamada y la señalización; RAS (Registration/ Admission/Status), un protocolo utilizado para la comunicación con el Gatekeeper y sólo si éste está presente en la red; soporte para RTP/RTCP (Real-time Transport Protocol/Real-time Transport Control Protocol) que fija la secuencia de los paquetes de audio y vídeo.

Gatekeeper o guardián de puerta: La función del guardián es gestionar una "zona de control" que consiste en un conjunto de equipos registrados (terminales, pasarelas y MCUs). Para las comunicaciones entre el guardián y los equipos de su zona se utiliza el protocolo RAS (Registro, Admisión, Situación). El gatekeeper puede también ofrecer otros servicios a los terminales, Gateway y MCUs, tales como gestión del ancho de banda y localización de los Gateway.

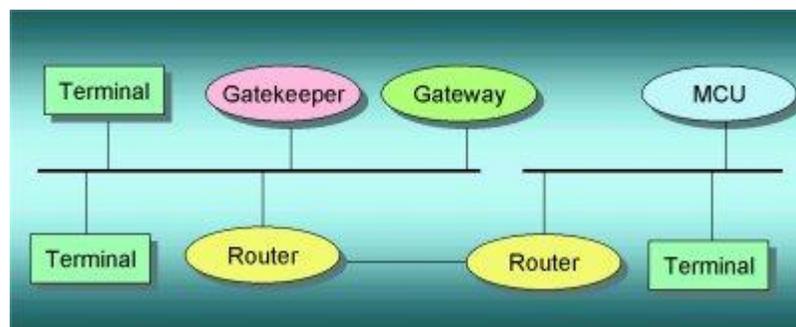


Figura 5. Zona de control H.323

Fuente: José Manuel Huidobro

Gateway: El Gateway es un elemento opcional en una conferencia H.323, que proporciona muchos servicios incluida la adaptación con otras normas del UIT. En general, su misión es establecer un enlace con otros terminales ubicados en la RTB o RDSI. Además de realizar la conversión de protocolo puede realizar opcionalmente una conversión de formatos de audio y vídeo (transcodificación). Un ejemplo de pasarela (y guardián) es Asterisk (es tanto pasarela como PBX completo tanto para H.323 como SIP).

Unidad de Control Multipunto (MCU) La Unidad de Control Multipunto es la entidad que soporta la conferencia entre tres o más puntos, bajo el estándar H.323, llevando la negociación entre terminales para determinar las capacidades comunes para el proceso de audio y vídeo y controlar la multidifusión. Es decir, si en una videoconferencia se necesitan interconectar tres o más participantes, hace falta una Unidad de Control Multipunto (MCU).

Una MCU comprende dos unidades lógicas:

- **Controlador Multipunto (MC: *Multipunto Controlar*):** gestiona las conexiones y se encarga de realizar la negociación entre los terminales para determinar las capacidades comunes para el proceso de audio y vídeo.
- **Procesador Multipunto (MP: *Multipunto Proceso*):** mezcla, conmuta y procesa los diferentes canales de audio, vídeo y/o datos y los envía a los participantes.

Como se menciona anteriormente es el MC el que tramita las negociaciones H.245 entre todos los terminales para determinar las capacidades comunes para el procesado de audio y video. El MC también controla los recursos de la conferencia para determinar cuáles de los flujos, si hay alguno, serán multicast. Las capacidades son enviadas por el MC a todos los extremos en la conferencia indicando los modos en los que pueden transmitir. El conjunto de capacidades puede variar como resultado de la incorporación o salida de terminales de la conferencia.

El MC no trata directamente con ningún flujo de datos, audio o video. Esto se lo deja al MP, este mezcla, conmuta y procesa audio, video y/o bits de datos. Las capacidades del MC y MP pueden estar implementadas en un componente dedicado o ser parte de otros componentes H.323, en concreto puede ser parte de un Gatekeeper, un Gateway, un terminal o una MCU. El MP recibe flujos de audio, video o datos desde los extremos, estos pueden estar involucrados en una conferencia centralizada, descentralizada o híbrida. El MP procesa esos flujos y los devuelve a los extremos. La comunicación bajo H.323 contempla las señales de audio y vídeo. La señal de audio se digitaliza y se comprime bajo uno de los algoritmos soportados, tales como el G.711 o G.723, y la señal de vídeo (opcional) se trata con la norma H.261 o H.263. Los datos (opcional) se manejan bajo el estándar T.120 que permite la compartición de aplicaciones en conferencias punto a punto y multipunto. La comunicación entre el MC y el MP no es asunto de estandarización.

2.5.4 Flujo de comunicación en H.323

A continuación se analizará detalladamente el establecimiento de una videoconferencia o llamada en H.323. En la figura 6, se expondrá un ejemplo de establecimiento entre un endpoint y una MCU, teniendo de intermediario un gatekeeper. En el estándar H.323 se le menciona como opcional, y la MCU es capaz de reemplazar en funciones de control y señalización al gatekeeper. Por tanto la siguiente figura es válida como ejemplo de funcionamiento tanto para un Gatekeeper como para una MCU. En una videoconferencia o llamada H.323 existen varias fases, a saber:

Fase de Establecimiento: Antes que nada una MCU está en modo “listening” es decir en modo escucha a la espera de captar mensajes H.225. Generalmente la MCU deja el puerto 1720 abierto a estos mensajes. Cuando un terminal desea entrar a una

videoconferencia debe solicitar acceso a la red al gatekeeper. Esto se hace por medio de un mensaje de petición de admisión (ARQ, Admission Request). El gatekeeper concede la petición con un mensaje de confirmación de admisión (ACF, Admission Confirm). La señalización puede ir encaminada a través del gatekeeper o entre la MCU y los endpoints directamente, sin embargo, en este ejemplo haremos uso del gatekeeper. Posteriormente, utilizando el protocolo H.225 el terminal llamante manda un mensaje de SETUP o establecimiento para iniciar una videoconferencia H.323. A su vez la MCU involucrada en la conferencia realiza los mismos pasos. La MCU contesta a todos los participantes con un CALL PROCEEDING advirtiendo del intento de establecer una videoconferencia. Luego la MCU genera un mensaje de ALERTING para todos los terminales indicando la preparación de los mismos para el inicio de conferencia. Por último CONNECT indica el comienzo de la conexión

Fase de Señalización de Control:

En esta parte se procede a la negociación mediante el protocolo H.245 (control de conferencia), el intercambio de los mensajes (petición y respuesta) el terminal y la MCU, se establecen las capacidades de los participantes y códecs de audio y video a utilizar. Como punto final de esta negociación se abre el canal de comunicación (direcciones IP, puerto).

Los principales mensajes H.245 que se utilizan en esta fase son:

- TerminalCapabilitySet (TCS)**. Mensaje de intercambio de capacidades soportadas por los terminales que intervienen en una llamada.
- OpenLogicalChannel (OLC)**. Mensaje para abrir el canal lógico de información que contiene información para permitir la recepción y codificación de los datos. Contiene la información del tipo de datos que serán transportados.

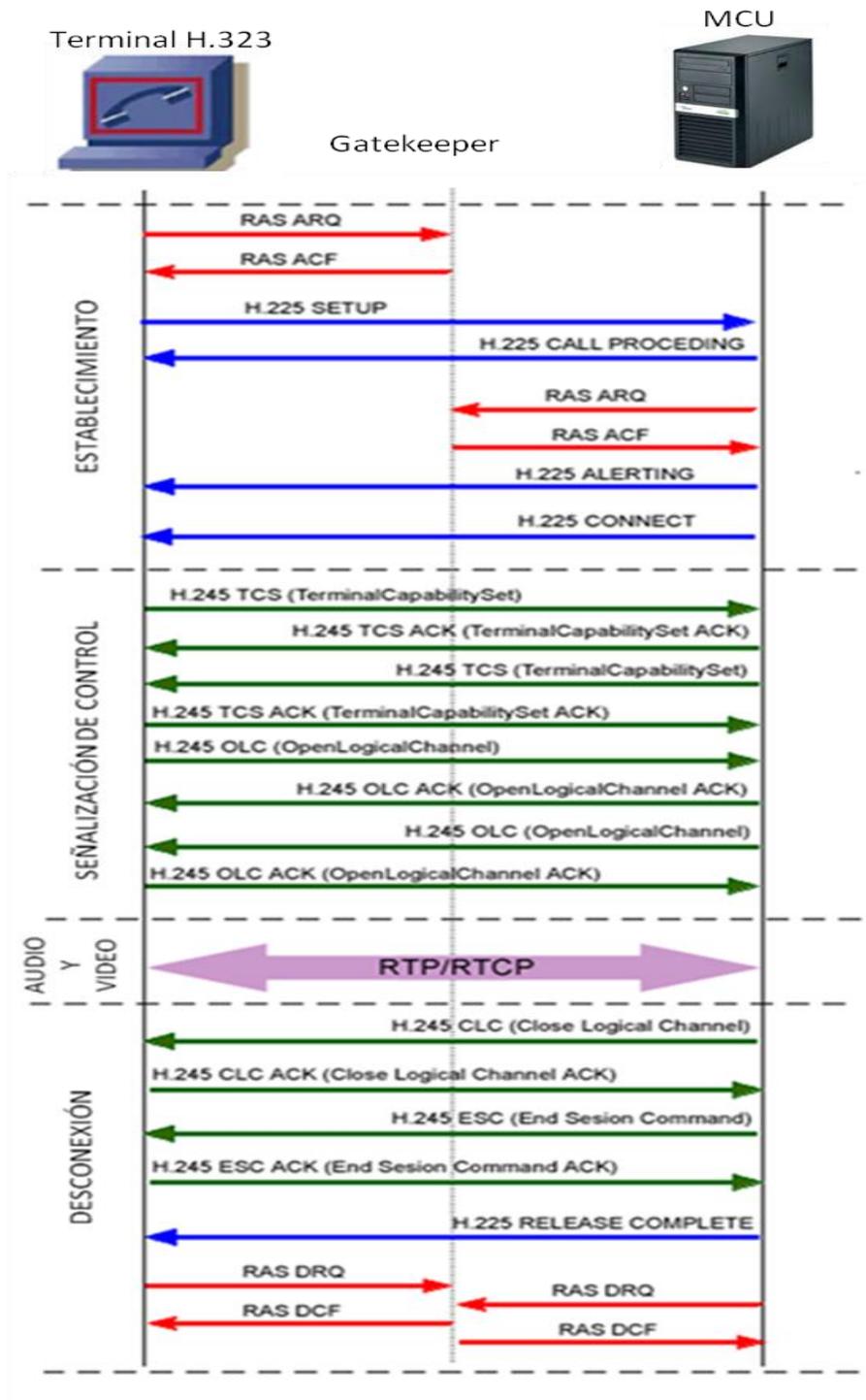


Figura 6. Establecimiento y finalización de una llamada H.323.

Fuente: Elaboración propia.

Fase de intercambio de Audio y Video:

Los terminales inician la comunicación y el intercambio de audio y video, a través de la MCU, donde ella recibe el flujo de datos de cada participante lo mezcla y reenvía a cada uno, todo esto mediante el protocolo RTP/RTCP.

Fase de Desconexión:

En este ejemplo se observa cómo se realiza la desconexión de un terminal efectuado por la MCU, la cual no afecta a los demás participantes. Se inicia con los mensajes de cierre de canal lógico y comando de cierre de sesión las cuales son respondidas por el terminal con ACK. Posteriormente utilizando H.225 se cierra la conexión con el mensaje RELEASE COMPLETE. Por último se liberan los registros con el gatekeeper utilizando mensajes del protocolo RAS.[14]

2.6 OPENH.323

El proyecto OpenH323 tuvo como objetivo el desarrollo de una completa implementación en código abierto del protocolo H.323. El código fue escrito en C++, y, a través del esfuerzo de desarrollo de numerosas personas en todo el mundo, se ha apoyado un amplio subconjunto del protocolo H.323. El proyecto OpenH.323 nació en septiembre de 1998 por parte de la compañía australiana *Equivalent Pty Ltd.* y está actualmente coordinado por el operador de telefonía local norteamericano *QuickNet Technologies Inc.* La iniciativa de OpenH.323 de crear una implementación de la recomendación H.323 libre, favorece el uso y el fomento de H.323 por parte de usuarios domésticos y pequeñas empresas, incapaces de pagar las licencias atribuidas a implementaciones comerciales. El software ha sido integrado a varios proyectos de código abierto y de productos de software comercial. El proyecto se bifurca en dos nuevos proyectos en octubre de 2007. Cada

uno de estos proyectos tiene un enfoque diferente. La Open Phone Abstraction Library (OPAL) continuó el desarrollo arquitectónico de OpenH323 con el objetivo de integrar otros protocolos VoIP como SIP y IAX2. El proyecto H323Plus continuó la expansión del soporte para H.323, así como un apoyo a las aplicaciones OpenH323.

2.6.1 OpenH.323plus

H.323 Plus (o, "H323plus") es el nuevo desarrollo de código abierto H.323 anteriormente conocido como OpenH323. El nuevo nombre fue elegido para reflejar el hecho de que el proyecto de código abierto H.323 contiene aún más características.

Esta versión posee librerías adicionales, además de las conocidas de la versión anterior como lo son:

- Ohphone : Terminal H.323 basado en línea de comandos
- Open Phone: Terminal H.323 con una interfaz gráfica
- Open GK: *Gatekeeper* H.323
- Open MCU: MCU H.323
- PSTN *Gateway*: Pasarela H.323-RTB

Teniendo como librerías nuevas a las siguientes:

- Callgen323 : Generador de llamadas de prueba.
- My phone: Terminal H.323 con una interfaz gráfica(antiguo ohphone).
- Openam: Contestadora automática H.323
- T38modem: Modem que cumple con la recomendación t.38 de fax sobre IP.

Todas estas aplicaciones están desarrolladas haciendo uso de dos librerías creadas por el proyecto OpenH.323; estas librerías son PWLib (que luego paso a llamarse

PTLIB) y OpenH.323plus. Para comprender mejor el funcionamiento de estas aplicaciones y conocer cómo están desarrolladas es interesante realizar una descripción de ambas librerías.[15]

Librería PWLIB/PTLIB

PTLib es quizás la más grande biblioteca de C ++ que se originó hace muchos años como un método para producir aplicaciones que se ejecutan en Microsoft Windows y sistemas Unix X-Windows. También iba a tener un ejecutable en Macintosh también, pero esto no ha sido viable. En aquellos días se llamaba la PWLib (Portable Windows Library) la biblioteca portátil Windows.

Desde entonces, la disponibilidad de kits de herramientas multi-plataforma de interfaz gráfica de usuario como KDE y Windows, y el desarrollo de los proyectos OpenH323 y OPAL como proyectos principales usuarios de la biblioteca, los desarrolladores de esta librería han hecho hincapié en el enfoque de redes y la portabilidad de protocolo. Sobre todo, la biblioteca se utiliza para crear un alto rendimiento y aplicaciones centradas en redes, siendo, altamente portátil. Así que todas las abstracciones GUI que se han imaginado y se han necesitado han sido puestas juntas y es lo que pasó a llamarse PTLIB hoy en día.

Las clases o elementos ejecutables en PTLib se dividen en dos tipos: las clases base y componentes de consola. Las clases base contiene todo el apoyo esencial para la infraestructura de programación tales como contenedores, eventos multi-tarea y puertos de uso que dependen de las características específicas de la plataforma. Todos los programas basados en PTLib requerirán de las clases base.

Los componentes de consola amplían la funcionalidad y por lo general son independientes de la plataforma, y, puede no ser necesaria para todos los programas.

En algunas plataformas (sobre todo Windows) las clases base y componentes de consola se pueden dividir en archivos de biblioteca discretos. En otras plataformas (sobre todo las plataformas Unix) se combinan todo el código en una sola biblioteca, y se basan en un enlazador para omitir el código que no es necesario. Las versiones anteriores de PTLib también contenían clases de interfaz gráfica de usuario y componentes de interfaz gráfica de usuario, pero el apoyo a estas clases se ha interrumpido.

Librería OPENH323/H323PLUS

La librería OpenH323 es la librería que realmente implementa la torre de protocolos H.323. Para ello, modela cada uno de los elementos necesarios de cada nivel de red que intervienen en la comunicación H.323, así como los parámetros de un punto extremo de red. La clase fundamental de la librería OpenH323 es el endpoint o terminal. Aquí en esta parte la acepción de terminal es para cualquier unidad H.323 haciéndola lo más isomorfa con respecto a las demás unidades.

Como OpenH323 está escrita en lenguaje C++, el cual es un lenguaje orientado a objetos facilita el modelado de los elementos involucrados en una comunicación H.323. De esta manera, es posible programar a los elementos, parámetros y unidades que son indispensables en una videoconferencia. En la programación orientada a objetos, los pertenecientes a una clase son pedazos separados que cooperan entre sí para dar sentido a una clase, en este caso a una clase H.323. A su vez estos objetos poseen características, las cuales se denominan atributos, y al ser estos variables, tiene como consecuencia que los objetos son configurables a través de los parámetros. Además, los objetos establecen relaciones entre sí, estas relaciones son denominadas métodos. OpenH323 también aprovecha la herencia entre clases para facilitar una programación más estructurada y evitar la aparición de código redundante. Existe una característica intrínseca a las clases llamada polimorfismo, el cual se aprovecha para que esta misma clase permita redefinir

métodos bien sea para reducir, ampliar o modificar la funcionalidad y el comportamiento de una determinada clase H.323. En la siguiente figura se muestra un esquema de clases con sus relaciones (métodos) directas:

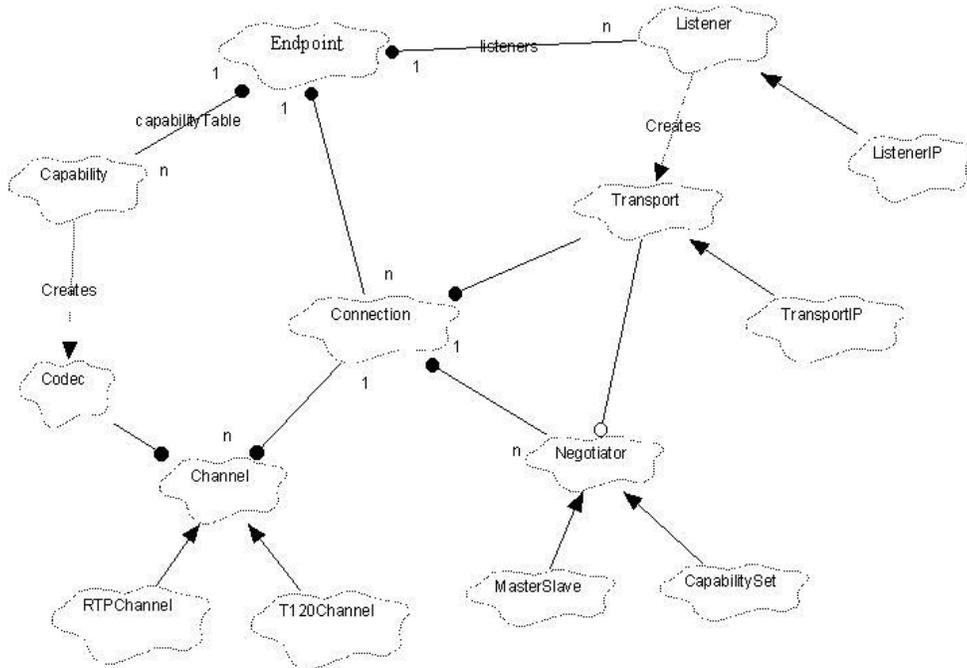


Figura 7. Esquema de clases de la librería OpenH.323

Fuente: OpenH.323.org

Anteriormente se expresó que la unidad fundamental de la librería OpenH.323 es la clase Endpoint. Cualquier aplicación implementada sobre OpenH323 sólo tiene un objeto de esta clase, la cual hará el papel bien sea de un terminal, Gatekeeper, una MCU, o cualquier componente H.323. Además, el objeto de más alto nivel dentro de la jerarquía OpenH323, es un endpoint; a él se atribuyen funciones genéricas como realizar una llamada, terminar una llamada, localizar un gatekeeper, registrarse en un gatekeeper, añadir información relacionada con el terminal, configurar la tabla de capacidades. Dependiendo de los códecs asignados en la tabla de capacidades del endpoint, éste podrá comunicarse con unos u otros terminales. Esta última acción es quizás la más importante a la hora de configurar el endpoint,

puesto que permite decidir qué códecs de audio y/o vídeo el endpoint usará en la videoconferencia. Existen una gran variedad de Códecs disponibles para seleccionar en una comunicación H.323, como se mencionado en párrafos anteriores. Sin embargo la ITU tan sólo obliga a los terminales H.323 a disponer de los Códecs de audio G.711 ley A y G.711 ley μ .

Todos los objetos endpoint poseen métodos de configuración de la tabla de capacidades disponibles y la prioridad en cuanto a uso se refiere. El endpoint asimismo puede generar negociaciones en cuanto al uso de capacidades o Códecs con otros terminales, de tal manera que incluyan o excluyan el uso de Códecs de forma simultánea. La negociación para la selección de Códecs está basada en la norma UIT-T H.245. Esta negociación se conoce como “Intercambio de capacidades”. Como métodos o funciones de más bajo nivel, la clase EP también ofrece el manejo de canales lógicos, determinación del tipo de terminal, configuración de temporizadores y contadores, etc.

Una clase de jerarquía más baja que el endpoint es la clase H323Listener. Esta clase está al tanto de las comunicaciones primigenias entrantes. El tipo de listener depende del tipo de protocolo. Por ejemplo, para el caso de una comunicación VoIP basada en H.323 se empleará un listenerIP lo que significa tener un tester de línea para cualquier petición de llamada entrante. Una vez que la petición llega al listener, éste creará un objeto perteneciente a la clase H323Transport. Un objeto de tipo H323Transport encarna a una forma equivalente al nivel de transporte del modelo OSI. El objeto del nivel de transporte recibirá y enviará las PDU (Packet Data Unit) relacionadas con la llamada H.323. Una vez que una llamada es admitida, a la llamada se le fija un identificador único, lo que equivale identificar la conexión H.323 realizada. La conexión H.323 se establece mediante objetos de la clase H323Connection; al crearse un objeto de esta clase se crea un canal ITU-T Q.931 sobre el que comienza el intercambio de señalización ITU-T H.225.0. Con la finalización de la señalización H.225.0 se habrá establecido un canal de control por

el que se llevará a cabo la señalización de llamada. En una aplicación basada en OpenH323 normalmente se redefine la clase H323Connection debido a que ofrece más funcionalidad de la que necesita la aplicación. Además, la redefinición de los métodos de la clase H323Connection permite alterar el comportamiento de los elementos de red durante una conexión. Por ejemplo, se podría redefinir el código del método que se ejecuta al recibir una petición de llamada entrante; de esta forma se podría responder aceptando la llamada, manteniéndola en espera o rechazándola directamente. Usando la clase H323Connection, o cualquier modificación de la misma, al finalizar el proceso de establecimiento de llamada se creará un objeto de la clase H323Negotiator encargado de establecer la negociación H.245. En esta negociación se determinará el códec a usar de la tabla de capacidades del endpoint. Es evidente que el códec seleccionado estará presente en la tabla de capacidades de los dos o más terminales que intervienen en la llamada. Además del intercambio de capacidades, se negocia la determinación de maestro y esclavo. Esta negociación tiene como objetivo determinar que terminal es maestro y cuál es esclavo; la adjudicación de roles se hace según la tabla 3, y será la entidad H.323 con el valor numérico más alto la que haga las veces de maestro. La selección de un maestro y un esclavo tiene como objetivo resolver posibles conflictos entre puntos extremos que pueden ser MC de una conferencia o entre dos endpoint que estén intentando establecer un canal bidireccional. Una vez fijado el tipo de terminal, se intercambian números aleatorios pertenecientes al intervalo $[0 - (2^{24}-1)]$.

Un terminal al que puedan asociarse varios tipos utilizará el valor más elevado para realizar el intercambio. Durante las negociaciones H.245, y una vez decidido el códec a utilizar, se abrirán canales lógicos para el códec seleccionado permitiéndose así el intercambio de flujos de información de audio o vídeo. También pueden abrirse canales de datos T.120. Los canales se modelan como instancias de la clase H323Channel haciendo especialización según RTPChannel para voz y video o T120Channel para datos.

Tabla 3. Tabla para la determinación principal-subordinado H.245

Fuente: Elaboración Propia

Tabla de Valores por tipo de Terminal	Entidad H.323			
	Terminal	Gateway	Gatekeeper	MCU
Entidad sin MC	50	60	NA	NA
Entidad con MC sin MP	70	80	120	160
Entidad con MC y datos de MP	NA	90	130	170
Entidad con MC y datos, audio de MP	NA	100	140	180
Entidad con MC y datos, audio, video de MP	NA	110	150	190

Todas estas clases son fundamentales para comprender el establecimiento, mantenimiento y finalización de una conexión H.323 están creadas a partir de otras clases de más bajo nivel y no por ello menos importantes como son clases: H323AudioCodec, H323LogicalChannel, H255PDU, H245PDU, etc. También se modelan de forma especial las clases relacionadas con los Gatekeepers y MCU por esto OpenH323 dispone de clases exclusivas a cada entidad. Todas estas clases permiten la implementación de aplicaciones software que funcionen como un completo Gatekeeper o MCU.



Figura 8. Logo OpenH323

Fuente: <http://www.dse.nl>

OpenMCU

OpenMCU fue creada en mayo de 2000 por Craig Southeren como un experimento para investigar algunas ideas sobre cómo una MCU de audio podría funcionar. Durante los próximos dos años, el código fue modificado por Derek Smithies y Roger Hardiman para incluir capacidades de vídeo y añadir funciones adicionales, como soporte para múltiples salas de conferencia. El resultado final de este esfuerzo fue OpenMCU v1.7.1 que apoyó el vídeo y el audio, pero era inestable y difícil de entender.

OpenMCU versión 2 se inició en marzo de 2004 con el apoyo de las redes Citron con la intención de arreglar los problemas conocidos con la versión 1. El código fue reescrito casi por completo a ser más modular, y los temas abordados fueron la estabilidad. Este trabajo terminó alrededor de agosto de 2004 con una versión que sólo se admite audio. Un nuevo proyecto se inició en marzo de 2006 con el apoyo de Stonevoice para restaurar la funcionalidad de video a OpenMCU. Luego Openh323 se convirtió en Openh323plus pasando de la versión Openh323 v1.19.0.1 (titán release) a la versión Openh323 plus v 1.21.0 generando una versión de OpenMCU nueva, la v2.2.5. Actualmente está la versión Openh323 v1.23.0 con la versión mejorada OpenMCU v2.2.5.1.



Figura 9. Logo OpenMCU

Fuente: <http://www.en.pudn.com>

El modo de funcionamiento del OpenMCU es como el indicado en párrafos anteriores, para cualquier entidad OpenH323. Primero, al inicio de comunicaciones,

El OpenMCU comienza sus funciones como un endpoint más. Luego pasa a modo escucha por el puerto 1720 con el método H323Listener. Al instante de recibir una petición crea un objeto de la clase H323Transport. El H323Transport recibirá y enviará las PDU (Packet Data Unit) relacionadas con la llamada H.323. Una vez que una llamada es aceptada, a la llamada se le asigna un identificador único, lo que equivale a identificar la conexión H.323 realizada. Con los objetos de la clase H323Connection crea un canal ITU-T Q.931 sobre el que comienza la señalización ITU-T H.225.0. Con la finalización de la señalización H.225.0 se establece un canal de control por el que se llevará a cabo la señalización de llamada. Al finalizar el proceso de establecimiento de llamada se creará un objeto de la clase H323Negotiator encargado de establecer la negociación H.245. Aquí se determina el códec a usar de la tabla de capacidades de cada endpoint, y se determina quién va a ser maestro y quien esclavo. En la tabla 3 se observa que siempre la unidad MCU será maestra y cualquier otra entidad será esclava. Al mismo tiempo se abren los canales lógicos para el códec seleccionado donde ocurre así el intercambio de flujos de información de audio o vídeo. Estos canales se especializan según su función que en nuestro caso concreto sería el RTPChannel para voz y video. Esto es a grandes rasgos el funcionamiento de la unidad OpenMCU.

2.7 Calidad de Servicio en Redes IP

Dos son los mitos que involucran a la Telefonía-IP. Uno se refiere a la baja calidad de Internet. Se confunden las prestaciones de los accesos dial-up con el uso de canales de transporte punto-a-punto con calidad contratada. Otro se refiere al medio de transportar a los paquetes IP (sólo ATM está en condiciones de garantizar la calidad de servicio). Nuevamente se ignora la serie de herramientas que posee una red IP y Gigabit-Ethernet para garantizar una calidad de servicio. Los problemas que son evidentes en una red de VoIP, son la Latencia, el Jitter, el Eco y la pérdida de paquetes. En Telefonía-IP estos problemas son resueltos mediante diversas técnicas.

2.7.1 Latencia

Se define así al gap en la conversación debido a los retardos acumulados. El primer retardo es en la matriz de switch (el retardo producido por el proceso store-and-forward) y el retardo de procesamiento (cambio de encabezado de paquetes, por ejemplo). A esto se suman los retardos propios del proceso de compresión vocal (insignificante en codificación G.711 y más elevado en aplicaciones con G.729).

Los retardos en la red pueden ser reducidos mediante el protocolo de reservación RSVP. El retardo debido a la compresión vocal se puede eliminar usando la velocidad de 64 kbps sin compresión (G.711). Este último aspecto es muy interesante. Inicialmente VoIP se desarrolló para reducir costos con menor velocidad y usando la infraestructura de Internet. Actualmente, con el modelo de una red IP de alta velocidad, la compresión vocal no es obligatoria en una red local. En este caso, Telefonía-IP se desarrolla para brindar una red de servicios integrados soportada en protocolo IP, sin límites de ancho de banda.

Cuando se trabaja con señales en Internet en cambio, el ancho de banda es limitado y por ello se requiere compresión vocal. Por ejemplo, el tamaño de un paquete RTP incluye 66 Bytes de encabezado (26 de MAC, 20 de IP, 8 de UDP y 12 de RTP) y 71 de carga útil. El overhead puede ser comprimido. La información vocal puede ser reducida. Por ejemplo: para G.723 trabajando a 6,3 kbps (trama de 30 mseg) sin supresión de silencios se requieren 11 paquetes/seg y 71 Bytes/paquete. Si integramos la supresión de silencios (técnica VAD) esta velocidad se reduce sustancialmente.

Causas: A la latencia también se la llama retardo. No es un problema específico de las redes no orientadas a conexión y por tanto de la VoIP. Es un problema general de las redes de telecomunicación. Por ejemplo, la latencia en los enlaces vía

satélite es muy elevada por las distancias que debe recorrer la información. La latencia se define técnicamente en VoIP como el tiempo que tarda un paquete en llegar desde la fuente al destino. Las comunicaciones en tiempo real (como VoIP) y full-dúplex son sensibles a este efecto. Al igual que el jitter, es un problema frecuente en enlaces lentos o congestionados

Valores Recomendados: La latencia o retardo entre el punto inicial y final de la comunicación debiera ser inferior a 150 ms. El oído humano es capaz de detectar latencias de unos 250 ms, 200 ms en el caso de personas bastante sensibles. Si se supera ese umbral la comunicación se vuelve molesta.

Posibles Soluciones: No hay una solución que se pueda implementar de manera sencilla. Muchas veces depende de los equipos por los que pasan los paquetes, es decir, de la red misma. Se puede intentar reservar un ancho de banda de origen a destino o señalar los paquetes con valores de TOS para intentar que los equipos sepan que se trata de tráfico en tiempo real y lo traten con mayor prioridad pero actualmente no suelen ser medidas muy eficaces ya que no disponemos del control de la red. Si el problema de la latencia está en la propia red interna se puede aumentar el ancho de banda, la velocidad del enlace o priorizar esos paquetes en la red.

2.7.2 Jitter

Es el efecto por el cual el retardo entre paquetes no es constante. Se trata de una latencia variable producida por la congestión de tráfico en el “backbone” de red, por distinto tiempo de tránsito de paquetes debido al “connectionless”, etc. Se puede utilizar un buffer para distribuir los paquetes y reducir el jitter, pero introduce un retardo adicional. Lo correcto es incrementar el ancho de banda del enlace; solución posible en un backbone pero de menor posibilidad en los enlaces WAN. Otra

posibilidad es la formación de colas para prioridad de tráfico de telefonía sobre los de datos.

Causas: El jitter es un efecto de las redes de datos no orientadas a conexión y basadas en conmutación de paquetes. Como la información se discretiza en paquetes cada uno de los paquetes puede seguir una ruta distinta para llegar al destino. Se define, como la variación en el tiempo en la llegada de los paquetes, causada por congestión de red, pérdida de sincronización o por las diferentes rutas seguidas por los paquetes. Las comunicaciones en tiempo real (como VoIP) son especialmente sensibles a este efecto. En general, es un problema frecuente en enlaces lentos o congestionados. Se espera que el aumento de mecanismos de QoS (calidad del servicio) como prioridad en las colas, reserva de ancho de banda o enlaces de mayor velocidad (100Mb Ethernet, E3/T3, SDH) puedan reducir los problemas del jitter en el futuro.

Valores Recomendados: El jitter entre el punto inicial y final de la comunicación debiera ser inferior a 100 ms. Si el valor es menor a 100 ms el jitter puede ser compensado de manera apropiada. En caso contrario debiera ser minimizado.

Posibles Soluciones: La solución más ampliamente adoptada es la utilización del jitter buffer. El jitter buffer consiste básicamente en asignar una pequeña cola o almacén para ir recibiendo los paquetes y sirviéndolos con un pequeño retraso. Si algún paquete no está en el buffer (se perdió o no ha llegado todavía) cuando sea necesario se descarta. Normalmente en los teléfonos IP (hardware y software) se pueden modificar los buffers. Un aumento del buffer implica menos pérdida de paquetes pero más retraso. Una disminución implica menos retardo pero más pérdida de paquetes.

2.7.3 Eco

Las características anteriores (latencia y jitter) pueden producir eco sobre la señal telefónica, lo cual hace necesario el uso de canceladores de eco (ITU G.168). Se tienen 2 tipos de eco. Uno tiene alto nivel y poco retardo y se produce en el circuito híbrido de 2 a 4 hilos local; mientras que otro es de bajo nivel y gran retardo y se produce en el circuito separador híbrido remoto. El cancelador de eco se construye mediante la técnica de ecualización transversal autoadaptativa. Consiste en usar una parte de la señal de transmisión para cancelar el eco producido por la desadaptación de impedancias en el circuito híbrido que convierte de 4 a 2 hilos.

2.7.4 Packet Loss

Es la tasa de pérdida de paquetes. Representa el porcentaje de paquetes transmitidos que se descartan en la red. Estos descartes pueden ser producto de alta tasa de error en alguno de los medios de enlace o por sobrepasarse la capacidad de un buffer de una interfaz en momentos de congestión. Los paquetes perdidos son retransmitidos en aplicaciones que no son de Tiempo Real; en cambio para telefonía, no pueden ser recuperados y se produce una distorsión vocal. El delay afecta al rendimiento de aplicaciones interactivas (por ejemplo, Telnet). El throughput afecta a la performance de aplicaciones que mueven grandes volúmenes de información (por ejemplo, Mail y FTP). El packet loss afecta a ambos tipos de aplicaciones. El jitter afecta a aplicaciones de tiempo real como la voz y el video por IP.

Causas: Las comunicaciones en tiempo real están basadas en el protocolo UDP. Este protocolo no está orientado a conexión y si se produce una pérdida de paquetes no se reenvían. Además la pérdida de paquetes también se produce por descartes de paquetes que no llegan a tiempo al receptor. Sin embargo la voz es bastante

predictiva y si se pierden paquetes aislados se puede recomponer la voz de una manera bastante aceptable. El problema es mayor cuando se producen pérdidas de paquetes en ráfagas.

Valores Recomendados: La pérdida de paquetes máxima admitida para que no se degrade la comunicación deber ser inferior al 1%. Pero es bastante dependiente del códec que se utiliza. Cuanto mayor sea la compresión del códec más pernicioso es el efecto de la pérdida de paquetes.

Posibles soluciones: Para evitar la pérdida de paquetes una técnica muy eficaz en redes con congestión o de baja velocidad es no transmitir los silencios. Gran parte de las conversaciones están llenas de momentos de silencio. Si solo transmitimos cuando haya información audible liberamos bastante los enlaces y evitamos fenómenos de congestión. De todos modos este fenómeno puede estar también bastante relacionado con el jitter y el jitter buffer.

2.7.5 Throughput

Es la capacidad de un enlace de transportar información útil. Representa a la cantidad de información útil que puede transmitirse por unidad de tiempo. No tiene relación directa con el delay. (Por ejemplo, se puede tener un enlace de alto throughput y alto delay o viceversa, como sería por ejemplo un enlace satelital de 2Mbps y 500 mseg de delay).

2.8 Ekiga

Ekiga, anteriormente llamado Gnomemeeting, es una aplicación de software libre que se encarga de realizar videoconferencias y telefonía IP (como Microsoft NetMeeting) y se libera bajo licencia GPL. Además está disponible para sistemas

Unix y Windows. Permite todas las características modernas de una videoconferencia como soporte de proveedor inteligente o llamadas de telefonía desde el ordenador a un teléfono.



Figura 10. Software Ekiga

Fuente: www.ekiga.org

Para su correcto funcionamiento debe disponerse de una cuenta SIP, que puede crearse gratuitamente desde el servidor de Ekiga.net. También trabaja con otros protocolos de comunicaciones como son H.323, en donde permite la comunicación H.323-H.323 y establece a su vez sesiones de multiconferencia bajo este mismo protocolo.

Por otro lado, Ekiga posibilita la realización de llamadas a teléfonos convencionales desde el PC. Para ello, se debe disponer de una cuenta con algún servidor de telefonía por internet. El mismo programa recomienda el proveedor Diamondcard Worldwide Communication Service, si bien existen muchos otros como VoIPBuster. Estos servicios no son gratuitos sino que se paga al proveedor del servicio en función del teléfono de destino según sus tarifas.

2.9 Ubuntu

Ubuntu es una filosofía sudafricana vinculada a la lealtad y la solidaridad. El término proviene de las lenguas zulúes y xhosa y puede traducirse como “humanidad hacia otros” o “soy porque nosotros somos”.



Figura 11. Logo de la Distribución GNU/Linux Ubuntu

Fuente: www.ubuntu.es

Esta noción se hizo popular en el ámbito de la tecnología ya que Ubuntu es el nombre elegido por la compañía británica Canonical Ltd. para denominar a una distribución GNU/Linux que se basa en Debian GNU/Linux. En este sentido, Ubuntu es un sistema operativo enfocado a la facilidad de uso e instalación, pensado para el usuario promedio. Por eso su lema es “Ubuntu: Linux para seres humanos”.

Ubuntu está compuesto por diversos paquetes de software que, en su mayoría, son distribuidos bajo código abierto y licencia libre. Este sistema operativo no tiene fines lucrativos (se consigue de manera gratuita) y aprovecha las capacidades de los desarrolladores de la comunidad para mejorar sus prestaciones. El sistema se financia a través de la venta de soporte técnico y de otros servicios vinculados al sistema operativo. Esta distribución, tiene nuevas versiones cada seis meses, que cuentan con el soporte de Canonical.

El aspecto colaborativo de Ubuntu se refleja en la posibilidad disponible para cualquier usuario de realizar sugerencias y presentar ideas para futuras versiones del sistema operativo. Para esto simplemente hay que ingresar a la página web oficial de la comunidad y publicar las propuestas o votar otras realizadas por el resto de los usuarios. Ubuntu tiene versiones en más de 130 idiomas, incluyendo el español. Puede descargarse de Internet o instalarse a través de un CD.

Características

- ✓ Basada en la distribución Debian.
- ✓ Disponible en varias arquitecturas: Intel x86, AMD64, SPARC (para esta última sólo existe la versión servidor).
- ✓ Los desarrolladores de Ubuntu se basan en gran medida en el trabajo de las comunidades de Debian y GNOME.
- ✓ Las versiones estables se liberan cada 6 meses y se mantienen actualizadas en materia de seguridad hasta 18 meses después de su lanzamiento.
- ✓ La nomenclatura de las versiones no obedece principalmente a un orden de desarrollo, se compone del dígito del año de emisión y del mes en que esto ocurre. La versión 4.10 es de octubre de 2004, la 5.04 es de abril de 2005, la 5.10 de octubre de 2005, la 6.06 es de junio de 2006, la 6.10 es de octubre de 2006 y la 7.04 es de abril de 2007.
- ✓ El entorno de escritorio oficial es Gnome y se sincronizan con sus liberaciones.
- ✓ Para centrarse en solucionar rápidamente los bugs, conflictos de paquetes, etc. se decidió eliminar ciertos paquetes del componente main, ya que no son populares o simplemente se escogieron de forma arbitraria por gusto o sus bases de apoyo al software libre. Por tales motivos inicialmente KDE no se encontraba con más soporte de lo que entregaban los mantenedores de Debian en sus repositorios, razón por la que se sumó la comunidad de KDE distribuyendo la distro llamada Kubuntu.
- ✓ De forma sincronizada a la versión 6.06 de Ubuntu, apareció por primera vez la distribución Xubuntu, basada en el entorno de escritorio XFce.

- ✓ El navegador web oficial es Mozilla Firefox.
- ✓ El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar, de forma predeterminada, procesos latentes al momento de instalarse. Por eso mismo, no hay un firewall predeterminado, ya que no existen servicios que puedan atentar a la seguridad del sistema.
 - ✓ Para labores/tareas administrativas en terminal incluye una herramienta llamada sudo (similar al Mac OS X), con la que se evita el uso del usuario root (administrador).
 - ✓ Mejora la accesibilidad y la internacionalización, de modo que el software está disponible para tanta gente como sea posible. En la versión 5.04, el UTF-8 es la codificación de caracteres en forma predeterminada.
 - ✓ No sólo se relaciona con Debian por el uso del mismo formato de paquetes deb, también tiene uniones muy fuertes con esa comunidad, contribuyendo con cualquier cambio directa e inmediatamente, y no solo anunciándolos. Esto sucede en los tiempos de lanzamiento. Muchos de los desarrolladores de Ubuntu son también responsables de los paquetes importantes dentro de la distribución de Debian.
 - ✓ Todos los lanzamientos de Ubuntu se proporcionan sin costo alguno. Los CDs de la distribución se envían de forma gratuita a cualquier persona que los solicite mediante el servicio ShipIt (la versión 6.10 no se llegó a distribuir de forma gratuita en CD, pero la versión 7.04 sí). También es posible descargar las imágenes ISO de los discos por transferencia directa o bajo la tecnología Bittorrent.
 - ✓ Ubuntu no cobra honorarios por la suscripción de mejoras de la “Edición Enterprise”.

2.10 Iperf

IPerf es un software que permite la evaluación de rendimientos en las comunicaciones de una red local, para posteriormente optimizar los parámetros de la misma. Con IPerf se puede medir el ancho de banda y rendimiento de una conexión entre dos host. Se trata de una herramienta cliente-servidor. Lo que se traduce en que

para ejecutarse Iperf se necesiten dos máquinas. Una hará de Servidor y otra de Cliente.

Funcionamiento

IPerf establece una conexión cliente-servidor entre dos hosts, es decir, en cada extremo del canal se coloca un host. Uno de estos actuará como servidor IPerf y el otro, en consecuencia, será el cliente. Esto es útil si queremos ver la velocidad máxima de nuestro switch o router y si cambiando ciertos parámetros como el MTU de la red, conseguimos más velocidad. Este programa es compatible para las plataformas Windows y también para las distribuciones GNU/Linux. En general, este programa arrojará datos como: ancho de banda, pérdida de paquetes, velocidad de conexión, jitter, latencia, entre otros parámetros de conexión.

CAPÍTULO III

3. METODOLOGÍA

3.1 Fase 1. Estudio Documental

La exploración sobre los diversos conceptos teóricos envueltos en el ámbito de los sistemas de videoconferencias, se enfocó en el desarrollo de proyectos basados en código abierto que dieran soporte a videoconferencia, y todo lo relacionado e involucrado con esta, como lo son los protocolos y estándares de comunicación, su implementación, recomendaciones de la ITU respecto al tema, así como también las bases de funcionamiento de programas y soluciones que dan respuesta a los problemas actuales que se plantean sobre videoconferencia. Esto ha dado sustento al capítulo II y fundamentado la plataforma conceptual sobre la que se asienta el presente Trabajo Especial de Grado.

Esta fase se dividió en cuatro subfases a saber:

- Documentación técnica acerca de la recomendación H.323.
- Documentación técnica acerca de los protocolos IP, TCP y UDP.
- Documentación técnica sobre OpenH.323, OpenMCU.
- Documentación relacionada con los lenguajes de programación utilizados en software de código abierto.

En la primera subfase se concentró en la adquisición y procesamiento de información referente a la recomendación H.323 de la ITU, lo pertinente a sus recomendaciones predecesoras, restricciones, alcances, componentes relacionados con el estándar, y aplicaciones.

En la segunda subfase se encauza a los protocolos IP, TCP y UDP donde en la consecución de estas unidades de conocimiento se hizo necesario el estudio y ampliación de esta recopilación de información hacia otras áreas de las redes por conmutación de paquetes, indagando sobre nuevo modos de transferencia de datos. En la siguiente subfase, se recabó toda la información posible respecto a los proyectos de código abierto OpenH323 y OpenMCU donde se profundizó en el funcionamiento detallado de estas entidades y otras como la librerías PWLib/PTLib, además se amplió también la investigación hacia los códecs de audio y video, señalización H323, componentes H.323, flujo de comunicación en H.323, los protocolos H.225, H.235, H.245, de donde provienen los conceptos de RAS(Registro, admisiones y situación) y el protocolo Q.931.

Por último, la investigación se basó en la búsqueda de documentación formativa con relación a los lenguajes de programación utilizados en software de de proyectos establecidos en código abierto., plataformas más utilizadas y distribuciones que una vez instalados, pudieran ser modificados para ser adaptados al entorno de estudio de este trabajo. Estos lenguajes de programación serian tales como el proyecto Python el cual es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia, el lenguaje Java y C++, los cuales son lenguajes de programación orientados a objetos, puestos estos facilitan el modelado de los elementos de comunicación H.323. También se documentó sobre proyectos de broadcasting sobre ip como lo es Xiph.org, y proyectos hermanos de OpenH323, tales como Ekiga, Opal, Gnugk, los cuales están basados en las mismas clases y métodos que este, de manera de entender el método raíz, con el fin de auditar errores de implementación Además, se estudiaron varios softphones H.323, tanto libres como privativos para la verificación de compatibilidad de software.

3.2 Fase 2. Análisis de Arquitectura

En esta fase, tomando como base la información recopilada en la fase 1, y los requerimientos del sistema de videoconferencia existente, se plantea un esquema prototipo de arquitectura de red, que cumpla con los objetivos propuestos.

Para esto es necesario conocer el entorno lógico actual de implementación, los equipos con los cuales se iban a hacer las pruebas de desempeño, sus capacidades y su compatibilidad.

3.2.1 Entorno de Desarrollo de Pruebas

En Venezuela se tiene una red académica con cobertura nacional y acceso a Internet, que tiene por nombre REACCCIUN (Red Académica Cooperativa entre Centros de Investigación y Universidades Nacionales), el cual es un proyecto del Centro Nacional de Innovación Tecnológica (CENIT) que interconecta a laboratorios de universidades nacionales y centros de investigación con redes internacionales experimentales de Internet de alta velocidad. Nuestra casa de estudios es miembro de esta red, la cual le provee aproximadamente el 40% de ancho de banda. El otro 60% restante es provisto por la empresa telefónica estatal CANTV y otros proveedores.

La topología de red es la que se explica a continuación: Este conjunto de servidores proveen al servidor perteneciente a la UCV. De allí sirve a tres dependencias, Ciencias, Medicina e Ingeniería, conectadas en estrella con rectorado, a través de fibra óptica, lo cual permite velocidades de hasta 10 Gbps. A su vez la Escuela de Ingeniería Eléctrica de la UCV se conecta al nodo de Ingeniería, con velocidades de 1 Gbps. Esto garantiza que el ancho de banda por conexión de usuario final sea al menos de 32 Mbps.

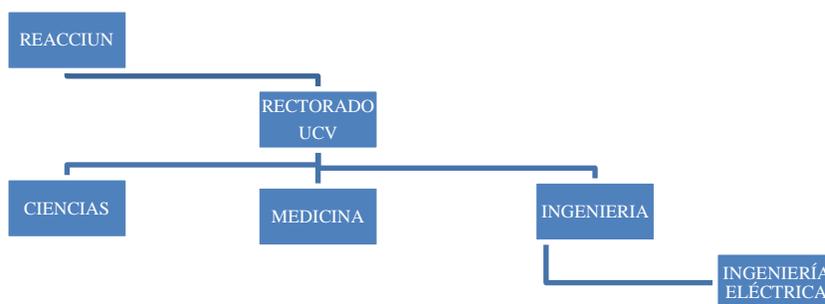


Figura 12. Organigrama de jerarquía lógica de red de datos UCV

Fuente: Elaboración Propia

En la actualidad existe en la plataforma de datos de la UCV un único equipo autónomo que proporciona funciones como MCU. El equipo en cuestión es MGC-100, el cual es el único equipo existente de esta marca que presta servicios de videoconferencia en forma autónoma dentro de la red de la UCV, sin embargo, algunos terminales como los IPELA (equipos de la marca SONY) y algunos modelos de Polycom y Lifesize, también proveen funciones de MCU a pesar de carecer de autonomía en su funcionamiento.

Teniendo esta información se determinó que el laboratorio de comunicaciones de la Facultad de Ingeniería, era sede ideal para el inicio de pruebas de conexión. En cuanto a los equipos disponibles en el laboratorio de comunicaciones para las pruebas se tienen:

- Dos (2) Máquinas PC Intel Pentium IV con 512Mb de Memoria RAM y 80Gb de Disco Duro en cada una (con su monitor, teclado, y mouse correspondiente).
- Una (1) Máquina PC Intel Pentium II con 398Mb de Memoria RAM y 40Gb de Disco Duro (con su monitor, teclado, y mouse correspondiente).
- Una (1) Máquina PC Intel Pentium II con 256Mb de Memoria RAM y 30Gb de Disco Duro (con su monitor, teclado, y mouse correspondiente).
- Teléfonos IP basados en el protocolo SIP y protocolo H.323.

- Dos (2) Switches Netgear ProSafe Ethernet Gigabit 10/100 Mbps de 16 puertos cada uno.
- Dos (2) Routers Cisco AS5300 de 150Mbps.

Adicionalmente se cuenta con dos computadoras más (propiedad del tesista), que poseen las siguientes características:

PC Intel Core 2 Dúo 1.8GHz:

- 1GB de Memoria RAM
- 160GB de Disco Duro
- Ethernet 10/100 PCI, LAN y Bluetooth
- Cámara y Micrófono Logibyte Periféricos

Laptop Dell D620

- Procesador Intel Core 2 Dúo 1.8GHz
- 1GB de Memoria RAM
- 80GB de Disco Duro
- Ethernet 10/100 PCI, LAN y Bluetooth
- Cámara y Micrófono Genius Periféricos

Una vez establecido este punto se prosiguió con la escogencia de los programas a ser usados en el prototipo inicial.

3.2.2 Selección de Software

Tomando como base la investigación expuesta en el capítulo II “Marco Teórico” se procedió con la escogencia del software que cumpliera con los requerimientos del problema. En primer lugar, se debía elegir como sistema operativo uno que fuese

de código abierto. Entre las opciones disponibles, como lo eran Debian, Mandriva, Red Hat, Icepack y Ubuntu, en primera instancia se escogieron, Debian v6.0.1 con netinst la cual tenía muchas características importantes de redes, Redhat 9 (Shrike) de difícil manejo e instalación de paquetes pero con pruebas reconocidas de voz sobre IP, y por último, Ubuntu 9.04(Jaunty Jackalope). De esto tres sistemas operativos se eligió el Ubuntu debido a su interfaz amigable, fácil instalación y de constantes mejoras en el desarrollo de sus paquetes. Una vez escogido el sistema operativo sobre el cual se iba a trabajar solo restó la elección del MCU. La decisión fluctuaba entre el programa ISABEL y OpenMCU. Isabel v3.10 es un programa de videoconferencias multipunto el cual puede operar bien sea como un terminal, un Gateway, gatekeeper o MCU. Este programa usa la combinación de técnicas de producción de TV que proporcionan un mensaje visual uniforme a todos los participantes, con un mecanismo de control que permite una intervención ordenada de los participantes. La versión estudiada posee la capacidad además de compartir presentaciones y de usar el pizarrón. La siguiente figura muestra las configuraciones típicas:

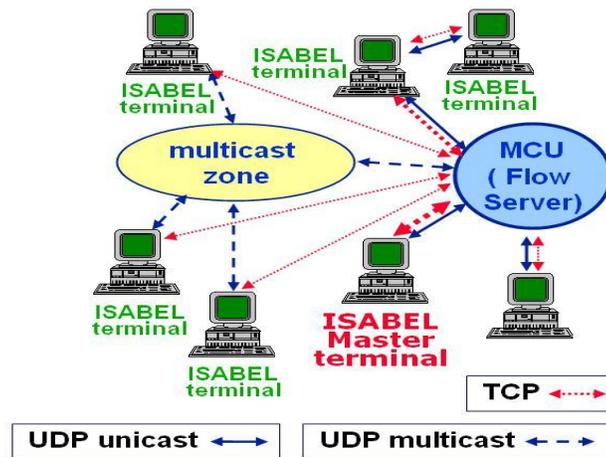


Figura 13. Configuraciones típicas de ISABEL

Fuente: <http://isabel.dit.upm.es>

Este programa es de fácil instalación y posee un lanzador propio para el sistema operativo Ubuntu. Sin embargo, la siguiente versión el programa si bien seguía siendo libre dejó de ser de código abierto. En una versión posterior la versión era libre bajo ciertas restricciones. Por esta razón, puesto que la intención del presente trabajo era la instalación de software código abierto, se desechó a Isabel como una solución viable. Esto deja camino al OpenMCU, que siendo un proyecto basado en OpenH.323 garantizaba que fuese de código abierto y además libre de cualquier tipo de licencia. También, el OpenMCU garantiza varias salas de videoconferencia (hasta 16) y varios usuarios (hasta 8).

Una vez escogidos la plataforma y el MCU queda la elección del software cliente que tuviese un buen desempeño con el MCU. En la actualidad existen varios software clientes H.323 con video, se analizaron varios software entre los cuales estaban Isabel (el cual es un MCU y Cliente), Openphone, Myphone, Ekiga, (estos últimos tres actuando como clientes), el criterio principal para realizar la elección fue que el software fuese netamente basado en código abierto y libre, es decir que no requiriera de pagos por licencia, otros factores importantes fueron el comportamiento libre de problemas al atravesar un NAT, estabilidad y buen desempeño, teniendo en cuenta estos aspectos se seleccionó Ekiga como el cliente que mejor se adaptaba a las necesidades del proyecto. Ekiga se consideró como el más apropiado ya que se encontraba basado en protocolos abiertos y muy difundidos hoy en día como lo son SIP y H323.

Aparte de esta selección, está el software propietario conocido como Microsoft NetMeeting el cual es un cliente de videoconferencia VoIP y multipunto incluido en muchas versiones de Microsoft Windows (desde Windows 95 hasta Windows XP). Este software cliente utiliza también el protocolo H.323 para realizar las conferencias, por lo que es interoperable con clientes basados en OpenH323 como Ekiga. Cualquier usuario de Windows XP normalmente tiene el Microsoft NetMeeting ya instalado en la carpeta *archivos de programa\netmeeting\conf.exe.*, y

puede ejecutarlo desde INICIO, luego pulsa el botón de ejecutar y luego escribe conf.exe, para después pulsar Enter. Respecto al Microsoft NetMeeting, el tesista solo hace mención, pues no es parte de los objetivos de este trabajo, sin embargo, es un recurso útil para realizar pruebas de compatibilidad.

Habiéndose hecho la selección del software se inició el diseño de la arquitectura lógica. Según la recomendación H.323 el uso de un gatekeeper es opcional en este conjunto de protocolos y como se expresó en el marco teórico, el proyecto OpenH.323 hace que todas sus entidades comiencen como endpoints y luego continúen hacia el rol que deben jugar en la comunicación H.323. Además, el MCU establece comunicaciones unicast con cada uno de los terminales por lo que el uso de un gatekeeper es superfluo.

Es así, que se obtuvo como resultado la topología lógica de estrella, donde la corona es el grupo de terminales participantes en la(s) videoconferencia(s), y el núcleo es el OpenMCU, siendo este interconectable a otros MCU's.

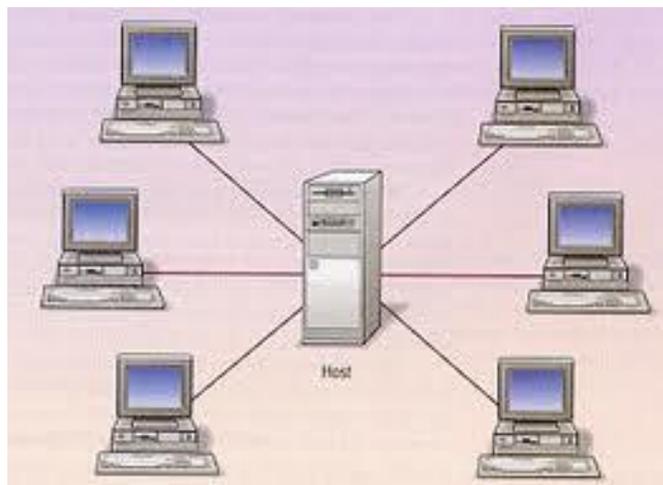


Figura 14. Topología de red tipo estrella

Fuente: Elaboración Propia

3.3 Fase 3. Implementación de OpenMCU

Una vez con la arquitectura ya definida, obtenida de la fase 2 se procede a implementar una MCU funcional. Para esto, se procede a revisar toda la documentación recabada, respecto a cómo se hace una instalación en Ubuntu. Se procede a instalar Ubuntu 9.04 (Jaunty Jackalope). Esta se hizo sobre las maquinas que posee el Laboratorio de Comunicaciones la cual sus características fundamentales son: Intel Pentium IV con 512Mb de Memoria RAM y 80Gb de disco duro.

Una vez realizada la instalación, se procedió a instalar el OpenMCU por el gestor de paquetes Synaptic, el cual es una herramienta de *Linux* de búsqueda e instalación de librerías. El gestor revisó las dependencias propias e instaló los paquetes siguientes:

- Libgsm 1
- Libopenh323-1.19.1
- Libpt-1.11.2
- Libpt-1.11.2-plugins-alsa
- Libpt-1.11.2-plugins-v4l
- Openmcu 2.2.1-2

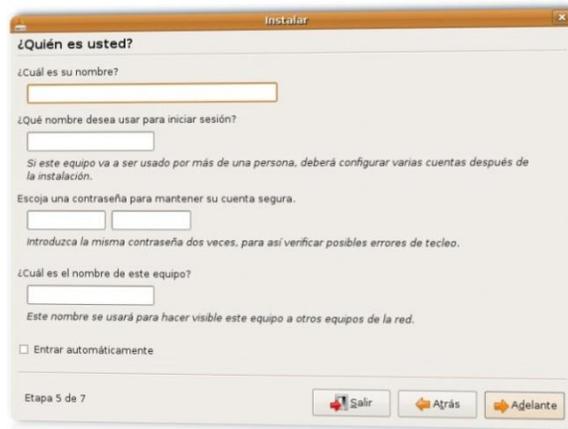


Figura 15. Instalación de Ubuntu

Fuente: Elaboración Propia

Luego se verifico que el programa funcionase con la acción *Aplicaciones* →*Terminal* →*openmcu* el cual muestra el menú:

```
maximiliano@maximiliano-laptop:~$ sudo su
[sudo] password for maximiliano:
root@maximiliano-laptop:/home/maximiliano# openmcu
error: must specify one of -v, -h, -t, -k, -d or -x
usage: [-c] -v|-d|-h|-x
-h --help          output this help message and exit
-v --version       display version information and exit
-d --daemon        run as a daemon
-u --uid uid       set user id to run as
-g --gid gid       set group id to run as
-p --pid-file      name or directory for pid file
-t --terminate     orderly terminate process in pid file
-k --kill          preemptively kill process in pid file
-s --status        check to see if daemon is running
-c --console       output messages to stdout rather than syslog
-l --log-file file output messages to file or directory instead of syslog
-x --execute       execute as a normal program
-i --ini-file      set the ini file to use, may be explicit file or
                  a ':' separated set of directories to search.
-H --handlemax n   set maximum number of file handles (set before uid/gid)
-C --core-size     set the maximum core file size

root@maximiliano-laptop:/home/maximiliano#
```

Figura 16.Consola OpenMCU

Fuente: Elaboración Propia

Una vez verificado su funcionamiento se ejecutó *openmcu-xc* el cual inicializa al MCU como un programa normal, además que permite la salida de mensajes de aviso a la consola. El programa OpenMCU cuenta con una interfaz gráfica a la cual se accede por el puerto 1420. Esta interfaz es muy pobre y presenta solo lo justo para el manejo a distancia del programa. Para ingresar a esta GUI (interfaz gráfica de usuario), se abre cualquier explorador de internet, se escribe en la barra de direcciones la dirección IP correspondiente del equipo (la cual se puede visualizar ejecutando el comando *ifconfig* en el terminal) seguido de dos puntos y el número de puerto 1420 (http).

Al punto de observarse la interfaz se procedió a hacer una prueba con el NetMeeting que viene incluido en la instalación de Windows XP. Esto se hizo solo para pruebas de compatibilidad entre equipos H.323, pues como ya se hablo anteriormente el

software cliente a usar sería Ekiga. Los resultados de esta prueba se encuentran en el capítulo siguiente.

En esta parte se observó problemas con el video, confirmando en investigaciones posteriores, que los desarrolladores en su proyecto, en la parte de video, era muy inestable en la negociación. Luego se proyectó el uso del reciente Ubuntu 10.04.LTS (Lucid Lynx) con interfaz gráfica. Después de realizar la nueva instalación, y completada la investigación sobre el OpenMCU sobre una nueva distribución OpenH.323+ (Octubre 2010), se procedió a la descarga y posterior instalación de todas las librerías necesarias para el proyecto. Para esto fue necesario, mediante el cambio de repositorios ejecutando la acción: *Sistema*→ *Administración*→ *Orígenes del software*, siendo el *archive.linux.duke.edu* de los Estados Unidos el que posee el conjunto completo de librerías. Sin embargo, en caso de que no fuese posible conectarse a este repositorio, la acción a tomar es la edición del archivo */etc/apt/sources.list*, reemplazando el archivo de fuentes de descarga completamente con el archivo de edición presente en anexos mediante la acción *Aplicaciones* → *Terminal* → *nano /etc/apt/sources.list*.

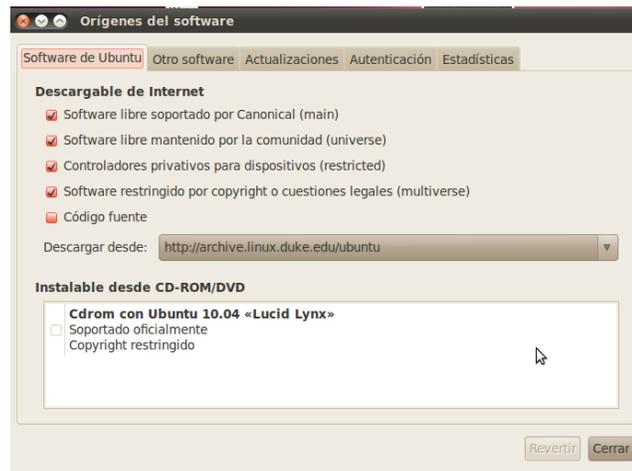


Figura 17. Orígenes del Software

Fuente: Elaboración Propia

Una vez hecho esto se actualizaron los cambios del archivo antes mencionado con el comando *apt-get update* en el terminal. Después se deben buscar todas las librerías necesarias, siendo estas en su mayoría compiladores y mini proyectos de código abierto los cuales fueron indispensables para la instalación del OpenMCU en este Trabajo Especial de Grado. Las librerías fueron las siguientes:

```
autoconf2.59, cpp, cpp-4.4, dpkg-dev, gcc, gcc-4.4, gcc-4.4-base, gcc-4.4-doc, gcc-4.4-multilib, gcc-4.4-plugin-dev, gcc-4.4-source, gcc-doc, gcc-multilib, gccxml, gcj, gcj-4.4-base, gcj-4.4-jdk, gcj-4.4-jre-lib, gcj-jdk, gfortran-doc, gfortran, gfortran-4.4-doc, gfortran-4.4, gfortran-4.4-multilib, gfortran-multilib, gobjc, gobjc-4.4, hardening-wrapper, lib64gcc1, lib64gcc1-dbg, lib64gomp1, lib64gomp1-dbg, libeigen2-dev, libgcc1, libgcc1-dbg, libgcj10, libgcj10-dbg, libgcj10-dev, libgomp1, libgomp1-dbg, llvm, uno-libs3, uno-libs3-dbg. install flex bison //
```

Es importante hacer notar que estas librerías se instalan por medio del comando *apt-get install* escrito en el terminal unas de ellas y otras por el comando *aptitude install* dependiendo de la imagen de Ubuntu 10.04.LTS (Lucid Lynx) que se instale y del repositorio de donde se haya descargado.

Con una velocidad de descarga de 92 kbps la instalación tarda alrededor de 40 min. Inmediatamente de ser instaladas, se colocaron en el escritorio las carpetas contentivas de todos los códecs, plugins e identidades compatibles al proyecto H.323 v.1.21.0 para la instalación correcta del OpenMCU v.2.2.5. Estos paquetes fueron *ptlib-2.4.5.tar.bz2*, *plugins-v1_21_0.tar.gz*, *h323plus-v1_21_0.tar.gz*, *openmcu-v2_2_5.tar.gz*. Es necesario acotar, que existe una sola librería PTLib compatible con una sola librería H.323.

Una vez en el escritorio se procedió mediante forma gráfica a descomprimir el paquete *ptlib-2.4.5.tar.bz2* y recomprimirlo en la forma *ptlib-2.4.5.tar.gz*, por comodidad a la hora de descomprimir por consola mediante el programa *untar*. Otro

preparativo previo a la instalación fue la generación de un paquete de instalación donde el código contenido de la librería *plugins-v1_21_0.tar.gz* se colocó junto con el contenido de la librería *h323plus-v1_21_0.tar.gz* llamado *h323plus.tar.gz*, pues de esa manera el directorio de plugins quedaba dentro del directorio de OpenH.323. De esta manera, se consiguió una manera más práctica de instalación con los pasos dados a continuación:

```
cd /home/NombrePC/Desktop
sudo su
cp ptlib-2.4.5.tar.gz /usr/src
cp h323plus.tar.gz /usr/src
cp openmcu-v2_2_5.tar.gz /usr/src
cd /usr/src
ls
```

Después de haber verificado la existencia de los paquetes de instalación en el directorio */usr/src*, se procedió a la instalación de las librerías *ptlib-2.4.5.tar.gz*, *h323plus-v1_21_0.tar.gz*, *openmcu-v2_2_5.tar.gz*. Fue necesario mantener ese orden a fin de seguir la secuencia de dependencias. De no seguir este patrón el OpenMCU se instalaría incorrectamente. Para instalar la librería PTLib se siguieron las siguientes instrucciones.

```
cd /usr/src
tar zxvf ptlib-2.4.5.tar.gz
cd ptlib-2.4.5
./configure
make bothdepend
make bothnoshared
make opt
```

```
PTLIBDIR=/usr/src/ptlib-2.4.5
```

```
export PTLIBDIR
```

Anteriormente se describió una receta para construir (build) ejecutables "noshared" (sin uso de bibliotecas compartidas) en la plataforma Linux/Unix. Se prefirió construir ejecutables "noshared" dado que son un poco más fáciles de utilizar. Por ejemplo, se puede transferir los ejecutables a otro equipo sin tener que instalar las librerías PWLib y OpenH323. Al construir la aplicación utilizando `make bothdepend`; `make bothnoshared` se generarán las versiones debug y la optimizada (release).

Inmediatamente después de haber instalado la librería PTLib, se siguieron los siguientes pasos necesarios para la instalación de la librería H323 Plus:

```
cd /usr/src
```

```
tar zxvf h323plus.tar.gz
```

```
cd h323plus
```

```
./configure
```

```
make bothdepend
```

```
make bothnoshared
```

```
cd /usr/src/h323plus/plugins
```

```
make bothdepend
```

```
make bothnoshared
```

```
PTLIBPLUGINDIR=/usr/src/h323plus/plugins/video/H.261-  
vic:/usr/src/h323plus/plugins/audio
```

```
export PTLIBPLUGINDIR
```

```
OPENH323DIR=/usr/src/h323plus
```

```
export OPENH323DIR
```

```
LD_LIBRARY_PATH=/usr/src/ptlib-2.4.5/lib:/usr/src/h323plus/lib
```

```
export LD_LIBRARY_PATH
```

Lo arriba descrito fue para la instalación del OpenH323 plus. Para instalar el OpenMCU tenemos como instrucciones lo siguiente:

```
cd /usr/src  
tar zxvf openmcu-v2_2_5.tar.gz  
cd openmcu  
make bothdepend  
make bothnoshared  
make install  
cd /usr/src/h323plus  
ls  
cd lib  
cp libh323_linux_x86_.so.1.21-beta0 /usr/local/lib  
cd /usr/src/ptlib-2.4.5  
make install  
cd /usr/local/bin
```

Luego se ejecuta dentro de directorio anterior */sbin/ldconfig*

Como se pudo observar en este proceso de instalación si se tiene PTLib y OpenH323 ubicadas en algún directorio diferente a *~/pplib* y *~/openh323*, se debe configurar dos variables en el shell (*PWLIBDIR* y *OPENH323DIR*) antes de compilarlas. Los comandos del shell son:

```
export PWLIBDIR=/directorio/diferente/ptlib-2.4.5  
export OPENH323DIR=/directorio/diferente/openh323plus
```

Los archivos objeto (*.o) y los ejecutables de oh323plus (las versiones optimizada y debug, respectivamente) fueron creadas en los subdirectorios *obj_linux_x86_r* y *obj_linux_x86_d*. Los ejecutables producidos utilizando la opción "no shared" ya contienen todo el código para PWLib y OpwnH323. Estos, sin embargo, dependen de

todas las librerías dinámicas previamente instaladas, sin las cuales esta compilación no tendría resultados satisfactorios.

Esta instalación sólo maneja el códec de video H.261 aun cuando en la instalación se genera una tabla de capacidades que muestra que existe la capacidad de manejar H.263, no se ha generado la librería correspondiente (.so). Si se quiere generar tal librería existen dos opciones. La primera es descargar esta librería previamente construida, siendo una opción de descarga el sitio web: <http://www.voxgratia.org/bin/ffmpeg-0.4.7.tar.gz>. Una vez descargado, se descomprime el contenido del fichero, escogiendo la librería el archivo *libavcodec.so* y se renombra a *avcodec* (sin extensión) hecha previa copia del fichero original. Hecho esto, se vuelve a comprimir en un nuevo paquete que incluye el nuevo archivo. Entonces se procede a instalar mediante los procedimientos descritos más adelante. La otra opción de instalación, es la de compilar manualmente la librería ffmpeg previamente a las librerías *ptlib-2.4.5.tar.gz*, *plugins-v1_21_0.tar.gz*, *h323plus-v1_21_0.tar.gz* y *openmecu-v2_2_5.tar.gz*. Esta librería puede ser descargada desde el sitio web <http://ffmpeg.org/download.html> (FFmpeg 0.8.5 "Love"). Sin embargo, el tiempo de compilación e instalación es bastante extenso por lo que la acción más recomendable es la primera opción. He aquí los pasos necesarios, para la instalación de la librería H.263 compilando desde cero:

```
cd /home/NombrePC/Desktop //si los archivos se descargaron en el //escritorio
cp ffmpeg-0.5.1.tar.gz /usr/src
cd /usr/src
tar zxvf ffmpeg-0.5.1.tar.gz
cd ffmpeg-0.5.1
./configure
make
make install
```

Luego de esto, se procede con los siguientes pasos, los cuales son a su vez de la primera opción, es decir, la opción en la que se instala una versión previamente construida.

```
cd /home/NombrePC/Desktop
cp h263.tar.gz /usr/src
cd /usr/src
tar zxvf h263.tar.gz
cd /usr/src/h323plus/plugins/video/H.263-ffmpeg
cp h263-ffmpeg_video_pwplugin.so /usr/src/h263
cd /usr/src
PTLIBPLUGINDIR=/usr/src/h263
export PTLIBPLUGINDIR
```

Al igual que la primera instalación se comprobó el funcionamiento del OpenMCU escribiendo en el terminal *openmcu*. Después, en lugar de ejecutar *openmcu-xc*, se ejecutó *openmcu-d* el cual hace que el MCU inicie como DAEMON (de sus siglas en inglés Disk And Execution MONitor), es decir, que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario y así, este continuará en ejecución o sin intervención de terceros y sin dependencia de la consola. Este modo se debe hacer en modo super usuario sino será bloqueado el comando.

Desde marzo de 2011, existen dos versiones posteriores de OpenH.323, la versión 1.22.0 (Marzo 2011) y la 1.23.0 (Agosto 2011), con mejoras considerables pero con una estabilidad baja en pruebas realizadas. Sin embargo, para el desarrollo de estudios posteriores se colocó en la sección de anexos una serie de instrucciones para su compilación e instalación, siendo esta serie la más practica pero no la mas óptima, se sugiere que para instalaciones de estos paquetes revisar bien toda la documentación sobre el compilador gcc 4.4 y sus versiones posteriores.

Una vez finalizada la instalación del OpenMCU 2.2.5, se procedió a la instalación del software elegido como cliente, el Ekiga. Su instalación se realizó de 3 maneras diferentes a fin de verificar la mayor compatibilidad con el OpenMCU instalado. Para ello se instalaron en las máquinas del laboratorio antes mencionadas, una con doble sistema operativo (Windows XP, Ubuntu 10.04) y la otra con solo el Ubuntu 10.4. La razón de instalar Ekiga en un programa privativo de código cerrado (Windows XP) es la gran cantidad de usuarios de este sistema a los cuales se les puede hacer extensivo el uso del OpenMCU sin necesidad de que cambien de sistema operativo. Aunque no es parte de este trabajo, también se probó el uso del NetMeeting como cliente solo para presentar sugerencias de uso y realizar pruebas de compatibilidad.

La primera instalación del cliente fue la del cliente Ekiga v3.1.0, por medio del gestor de paquetes Synaptic. Este paquete instalado no tuvo un desempeño aceptable, pues su compatibilidad con webcam solo se reduce a la marca de cámaras Logitech® y sus drivers controladores se deben compilar junto con los paquetes del Ekiga, algo que no es posible con el gestor de paquetes. También se observó que el programa Ekiga descargado directamente por Synaptic muchas veces presentaba fallas al comunicarse con el OpenMCU. Entonces se procedió a la instalación del cliente Ekiga por medio de la compilación manual.

Antes de realizar la compilación deben descargarse las librerías necesarias para la construcción del mismo. Estas librerías son adicionales a las descargadas para el Openmcu y son las siguientes:

```
apt-get install gnome-common libsasl2-dev gettext libgnome2-dev libldap2-dev  
libgconf2-dev autoconf libgnomeui-dev libxv-dev intltool scrollkeeper libxml-parser-  
perl evolution-data-server-dev libavahi-common-dev libavahi-client-dev libavahi-  
glib-dev gnome-doc-utils libsigc++-2.0-dev libdbus-glib-1-dev libebook1.2-dev. flex  
bison
```

```
aptitude install libspeex-dev libx264-dev libtheora-dev libavcodec-dev libavformat-dev // Librerías de códecs necesarias para Ekiga
```

Antes de instalar los paquetes de Ekiga, se deben instalar los paquetes de proyectos de código abierto de video por webcam para que el sistema operativo reconozca las cámaras y micrófonos del PC, que como se expresó en párrafos anteriores, son pocos los productos compatibles en el mercado. Esto se logró mediante la siguiente pila de comandos:

```
apt-get install libdv4-dev libv4l-dev libasound2-dev libpt-plugins-v4l
```

Luego de instalarse estas librerías se procedió con los comandos:

```
Lsmod |grep gspca  
Sudo modprobe gspca  
apt-get install qc-usb
```

Quedando de este modo el sistema operativo en capacidad de reconocer las cámaras y micrófonos.

Después se descargaron las siguientes carpetas comprimidas: *opal-3.6.8.tar.gz*, *ptlib-2.6.7.tar.gz* de el sitio web <http://www.opalvoip.org> y *ekiga-3.1.0.tar.gz*, de la página: <http://www.ekiga.org/> y fueron guardadas en el escritorio para una posición de fácil acceso.

A continuación se describen los pasos que se realizaron para la instalación de este software:

```
cp opal-3.6.8.tar.gz /usr/src  
cp ptlib-2.6.7.tar.gz /usr/src
```

```
cp ekiga.tar.gz /usr/src
cd /usr/src
ls
```

Este set de instrucciones es para copiar y verificar que las librerías se ubican en el directorio /usr/src. Se continuó la instalación con la librería PTLib:

```
tar zxvf ptlib-2.6.7.tar.gz
cd ptlib-2.6.7
./configure --enable-sunaudio=no --enable-oss=yes --prefix=/usr --disable-sdl --
enable-v4l
make
make install
```

Finalizada la instalación de PTLib se continuó con la instalación de Opal.

```
cd /usr/src
tar zxvf opal-3.6.8.tar.gz
cd opal-3.6.8
./configure
make
make install
```

Por último, para la instalación de Ekiga, se realizó lo siguiente:

```
cd /usr/src
tar zxvf ekiga.tar.gz
cd ekiga
./autogen.sh
make
```

```
make install
cd /usr/src/opal-3.6.8/lib_linux_x86
cp libopal.so.3.6.8 /usr/local/lib
cd /usr/local/bin
/sbin/ldconfig
```

Una vez completada la instalación se procedió a reiniciar la computadora y se ubicó el programa en el menú de *Aplicaciones* → *Internet* → *Softphone Ekiga*. Luego se procedió a configurar los códecs de audio y video a ser usados para la primera prueba. Esto se hizo entrando en *editar* → *preferencias* al realizar esto se abrió una ventana donde se encontraba el menú de audio y video. Luego se abre la ventana de códecs en el apartado de audio, para efectuar la selección.

La elección hecha referente al audio, fueron los códecs PCMA el cual no es sino el códec G.711 ley A, el PCMU, que es el G.711 ley μ , y el códec GSM. La razón de escogencia de estos códecs fueron; en primer lugar, los códecs G.711 ley A y ley μ , son obligatorios en el estándar H.323. Por otra parte, estos códecs son de menor compresión, por tanto, aunque exigen mayor velocidad de transmisión requieren menor índice de procesamiento, el cual es clave pues la universidad dispone de suficiente ancho de banda.

La elección del códec GSM fue hecha en base a la preparación de este sistema para ser conectado con otro módulo, el cual trata de una tesis sobre un Gateway de señalización H.323 de código abierto, en el cual el uso de este códec es requerido. Además, como se puede ver en la tabla 6, el GSM posee menor ancho de banda nominal, el cual debe ser multiplicado por dos si la comunicación es full dúplex.

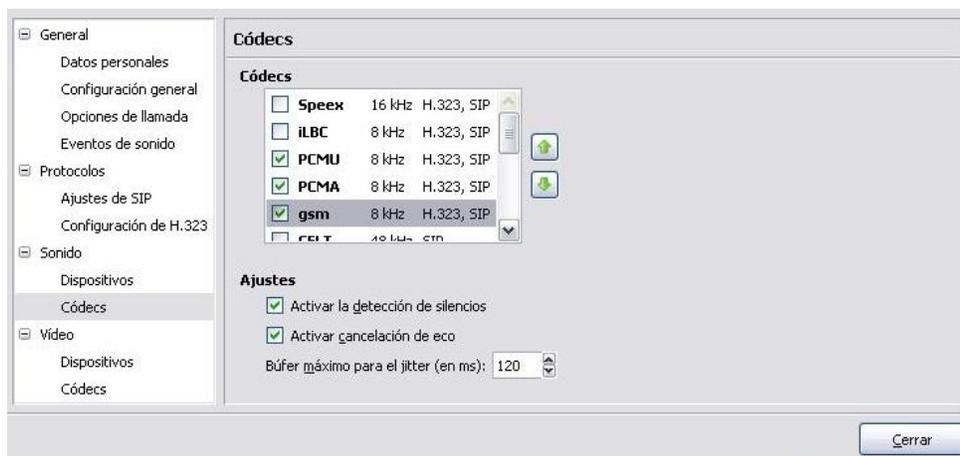


Figura 18. Menú de selección de códecs de audio.

Fuente: Elaboración Propia

Tabla 4. Características de los códecs de audio seleccionados

Fuente: Elaboración propia

Codec	Ancho Banda (kbps)	Período de Muestra (ms)	Carga útil (bytes)	PPS	Ancho Banda Nominal (kbps)
G.711	64	20	160	50	80
GSM	13	20	32.5	50	29

Otra consideración pero ya en cuanto a videoconferencia es que el incremento en ancho de banda por número de llamadas en GSM es mucho menor, como se puede observar en las tablas 5 y 6.

Para la parte de video, se escogieron los códecs H.261 y H.263. Esta selección se debe a que estos códecs de video están en la mayoría de proyectos H.323 de código abierto. Otra razón sigue siendo la capacidad de procesamiento. El H.261 tiene una compresión menor y necesita poco procesamiento. El H.263 tiene menor calidad pero exige menos velocidad de transmisión y poca velocidad de procesamiento.

Tabla 5. Ancho de Banda por llamada G.711

Fuente: Elaboración propia

Número de Llamadas	Ancho de Banda Total (kbps)	Incremento de Ancho de banda (Kbps)
1	111,48	0,00
2	217,62	106,14
3	327,01	109,39
4	412,50	85,49
Promedio		100,34
Variación		12,96

Tabla 6. Ancho de Banda por llamada GSM

Fuente: Elaboración propia

Número de Llamadas	Ancho de Banda Total (kbps)	Incremento de Ancho de banda (Kbps)
1	25,16	0,00
2	55,50	30,34
3	80,39	24,89
4	103,54	23,15
Promedio		26,13
Variación		3,75

Sus resoluciones son aceptables en cuanto a videoconferencia. El H.263 fue creado específicamente para videoconferencia. Otros códecs pueden ser soportados por el Ekiga; sin embargo nuestro OpenMCU solo maneja esos dos con la versión de trabajo instalada.

Tabla 7. Características de los códecs de video seleccionados

Fuente: Elaboración propia

Estándar/Formato	Ancho de banda típico	Compresión
H.261	64 Kb/s – 2000 Kb/s	24:1
H.263	28,8-768 Kb/s	50:1

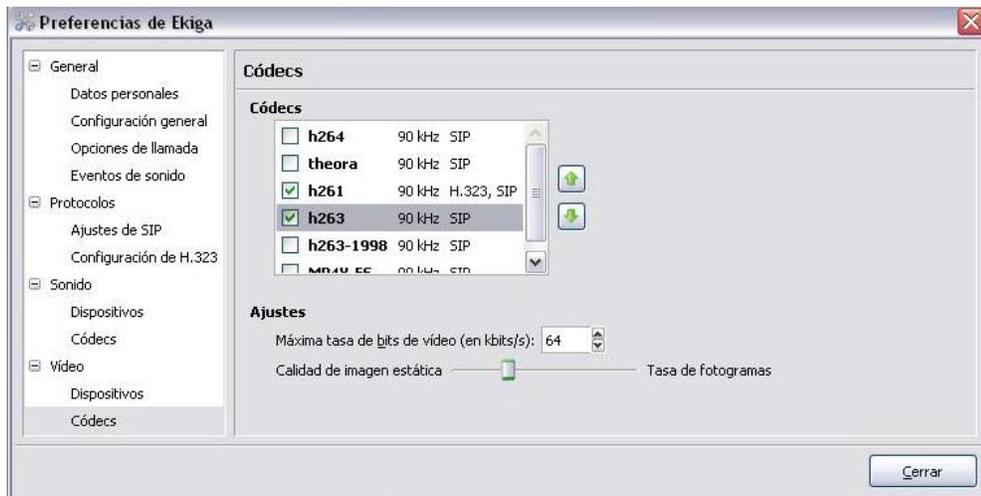


Figura 19. Menú de selección de códecs de video.

Fuente: Elaboración Propia

El sistema debería ser de al menos de 8412,5 Kbps sí se tratase de una comunicación bidireccional y cuatro personas efectuando una videoconferencia. Aunque el OpenMCU permite hasta 8 participantes por sala de videoconferencia, el sistema solo permite solo a cuatro de ellos estar en el flujo de video siendo el sistema priorizado para las personas que estén hablando en el momento. A pesar de que el ancho de banda disponible en la red de datos de la UCV parece suficiente, dispositivos como routers, bridges y hubs pueden no permitir aprovechar este ancho

de banda debido a su capacidad, por tanto se requiere que dichos dispositivos sean de gran desempeño y de alta velocidad.

Una vez configurados los programas, se procedió a realizar la llamada en cada una de las computadoras, este proceso se realizó escribiendo la dirección IP de la máquina con el MCU previamente instalado, en la barra de llamada del programa Ekiga precedido por la especificación del protocolo a usar en este caso H323. Inmediatamente luego de marcar la dirección se estableció la comunicación con el OpenMCU. Sin embargo, solo se transmitió audio y no video, y en ciertas ocasiones solo dejaba un participante. En principio, se pensó que el problema era por ancho de banda pues la interconexión de equipos se estaba realizando a través de un hub de 10 Mbps. Por ello, fue sustituido este hub por un Switch Netgear ProSafe Ethernet Gigabit 10/100 Mbps de 16 puertos. Además, se verificó que ningún Firewall propio de la red de datos de la UCV estuviese bloqueando la conexión, o estuviese ocurriendo algún otro proceso de seguridad el cual no permitía el establecimiento de comunicación entre las máquinas. Sin embargo, luego de hacer una revisión exhaustiva, se concluyó que el problema era que la computadora donde se había realizado la instalación, una PC Intel Pentium IV con 512Mb de Memoria RAM y 80Gb de Disco Duro, pues el compilador gcc 4.4 necesita preferiblemente 2 GB de RAM (además de 1GB para swap) o al menos 1531,25 MB de RAM (con aún un mayor swap) para construir la versión de depuración (debug) y la versión optimizada de OpenH323 en un tiempo razonable y sin errores. Por lo tanto, se procedió a cambiar el equipo por uno otorgado en calidad de préstamo que contaba con las siguientes especificaciones: Intel Core 2 Dúo 2.1GHz 2GB de Memoria RAM y 160GB de Disco Duro. Es importante hacer notar que las memorias deben ser del tipo DDR2 en adelante, pues son de doble tasa de transferencia, y son las que soportan la velocidad de datos transferidos por segundo al procesador, cuando se están instalando los paquetes del OpenMCU. Es decir, que si se tiene una memoria de 2 GB pero de formato DDR no será suficiente para el procesamiento de datos teniendo como consecuencia una compilación incorrecta.

Ya determinado y contrarrestado el problema se realizó una vez más la instalación del sistema operativo, la configuración de red, descarga de librerías y finalmente la instalación del MCU.

Con el equipo en funcionamiento se realizaron las pruebas para establecer la videoconferencia con las dos computadoras que contaban con el cliente Ekiga, resultando esta vez una conexión bastante satisfactoria, logrando así el establecimiento de la sesión de videoconferencia, donde se comprobó el requerimiento de ciertas especificaciones puntuales con respecto al hardware del equipo donde se instaló el OpenMCU.

De igual manera que con la versión anterior se accedió a su interfaz gráfica y se configuraron sus parámetros para así hacer la división de pantalla.

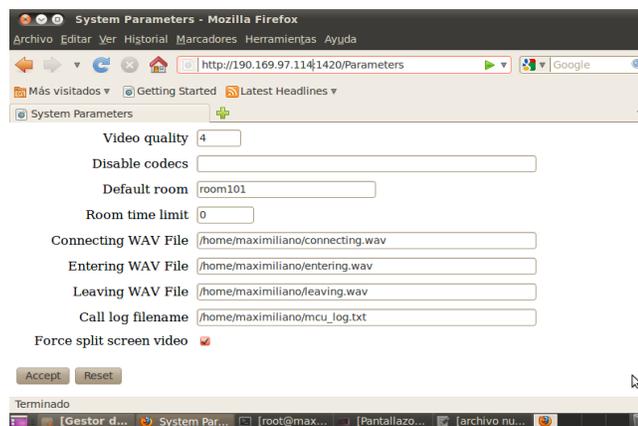


Figura 20. Parámetros de Openmcu.

Fuente: Elaboración Propia

Una vez hecho esto se procedió a hacer las pruebas necesarias. La primera topología de prueba fue de dos clientes con el OpenMcu. Una vez que se obtuvieron resultados positivos se pasó a la realización de una videoconferencia entre tres personas y luego cuatro personas. A partir de esta configuración también se procedió

a realizar el análisis de calidad de servicio para lo que se utilizó Iperf el cual es una programa que permite medir la calidad de videoconferencias. Tanto en la máquina que fungía como MCU como en un cliente, la instalación se hizo de manera idéntica, siendo a través del terminal con el comando *aptitude install iperf*.

Para lograr que el programa identifique quien es el servidor y quien es cliente, se ejecutaron los siguientes comandos: *iperf -s* el cual se ejecutó en el MCU e *iperf -c dirección IP del Servidor* en la máquina que tenía instalado el Ekiga. Los resultados del análisis de datos transmitidos, ancho de banda y pérdida de paquetes están plasmados en el siguiente capítulo.

A su vez, se hicieron llamadas de prueba a equipos Polycom pertenecientes a la universidad para la comprobación de compatibilidad entre estos equipos y el OpenMCU. También se hicieron Pruebas de interconexión con otro proyecto de tesis que consistía en un Gateway obteniendo resultados bastante satisfactorios.

3.4 Fase 4. Elaboración del Informe Final

Finalmente, en esta fase se realizó el presente documento donde se explica paso a paso y detalladamente todos los procesos mencionados anteriormente como la instalación del OpenMCU, la construcción e instalación del software cliente Ekiga en Ubuntu (la instalación en Windows esta en los anexos), el manejo del programa cliente para realizar una sesión de videoconferencia, y la solución a los problemas que presentaron durante el proceso. También se formulan una serie de conclusiones, recomendaciones y propuestas para trabajos de investigación posteriores.

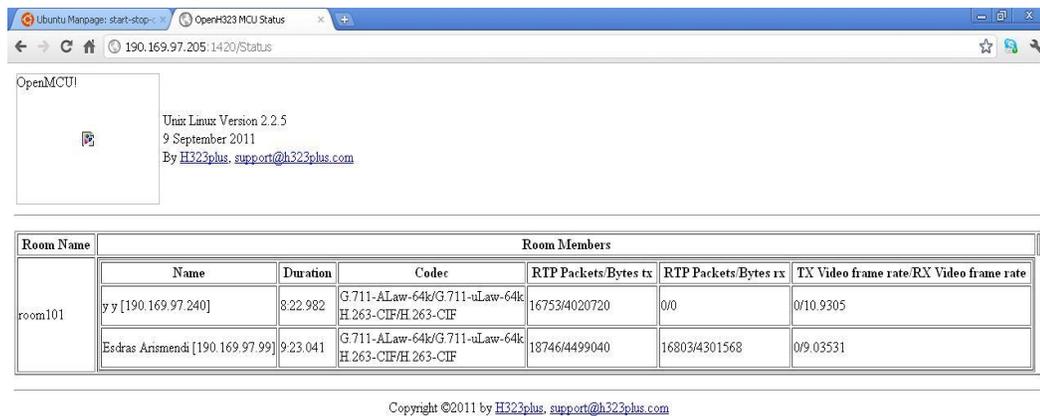
CAPÍTULO IV

4. RESULTADOS

En este capítulo se presentará la sucesión de pruebas hechas siguiendo la metodología antes presentada en capítulos anteriores. La secuencia sigue el orden lógico de desarrollo, con errores de implementación, su estudio y posterior solución.

4.1 Pruebas Iniciales

En las primeras fases de implementación se instaló el OpenMCU desde el gestor de paquetes synaptic. En estas incipientes pruebas se utilizó un cliente conocido como NetMeeting presente en todas las distribuciones de Windows XP. He aquí los resultados.



OpenMCU!

Unix Linux Version 2.2.5
9 September 2011
By H323plus, support@h323plus.com

Room Name	Room Members					
	Name	Duration	Codec	RTP Packets/Bytes tx	RTP Packets/Bytes rx	TX Video frame rate/RX Video frame rate
room101	y y [190.169.97.240]	8:22.982	G 711-uLaw-64k/G 711-uLaw-64k H 263-CIF/H 263-CIF	16753/4020720	0/0	0/10.9305
	Esdras Anismendi [190.169.97.99]	9:23.041	G 711-uLaw-64k/G 711-uLaw-64k H 263-CIF/H 263-CIF	18746/4499040	16803/4301568	0/9.03531

Copyright ©2011 by H323plus, support@h323plus.com

Figura 21. Primera conexión entre OpenMCU y NetMeeting.

Fuente: Elaboración Propia

Como puede observarse en las figuras anteriores el OpenMCU reporta una conexión exitosa con los códecs de audio G.711 en la parte de envío y recepción, y el códec H.263 en la parte de video. Sin embargo, en el apartado de paquetes de transmisión enviados no está recibiendo ni enviando paquetes de video, lo que se traduce en que el terminal no muestra video como se puede observar en la figura siguiente.

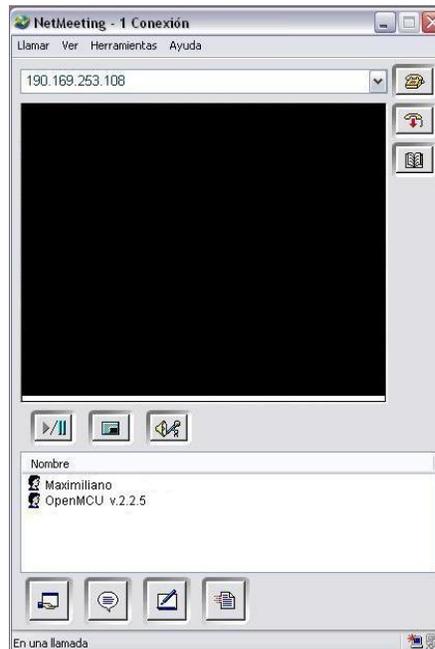


Figura 22. NetMeeting conectado a OpenMCU.

Fuente: Elaboración Propia

Con esta Prueba se confirma que el OpenMCU instalado por Synaptic tiene problemas, que los desarrolladores reconocieron, que en su proyecto, la parte de video era muy inestable.

Luego de instalar Ubuntu 10.04, descargar las librerías necesarias, instalar el OpenMCU, se prosiguió con las siguientes pruebas, pero esta vez ya instalando el Ekiga. Como se explicó en el capítulo anterior se estableció la comunicación con el OpenMCU. Sin embargo, solo se transmitió audio y no video, y en ciertas ocasiones solo dejaba un participante. Luego de estudios cabales, se llegó a la conclusión que

era problema de la memoria RAM, que primero no era de doble tasa de transferencia y era de 1GB. Lo cual generaba bloqueos en ciertas librerías de compilación. Se cambió de equipo y se realizó la instalación del OpenMCU una vez más. Solventado este impase, ya se establece comunicación de audio y video lo que confirma que el problema es con las librerías compiladoras en especial la *gcc-4.4*. A continuación las gráficas que reflejan lo antes descrito.



Figura 23. OpenMCU con errores de compilado.

Fuente: Elaboración Propia

Aunque en esta gráfica se ve funcionando códecs de audio y video solo permitía la conexión de un participante. En el software cliente no había ninguna traza de video, sin embargo, no se colocó a modo de no parecer redundante. En las siguientes dos figuras ya se puede observar la videoconferencia H.323, realizadas entre el Laboratorio de Comunicaciones de la escuela de Ingeniería Eléctrica de la UCV y San Antonio de los Altos. Con esta prueba se verifica la capacidad de conexión entre la red interna de la universidad y redes ADSL.

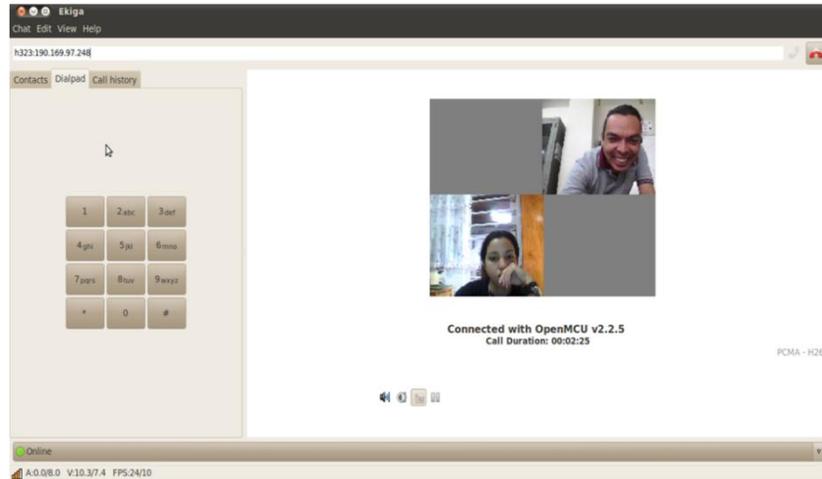


Figura 24. Prueba de videoconferencia OpenMCU con dos participantes.

Fuente: Elaboración Propia

Room Name	Room Members					
	Name	Duration	Codec	RTP Packets:Bytes tx	RTP Packets:Bytes rx	TX Video frame rate:RX Video frame rate
room101	Erdas [190.169.97.228]	3:58.960	SpeedITF Wide-20 6k/SpeedITF Wide-20 6k/H.261-CIF/H.261-CIF	5974/310648	11501/598052	10.001/5.47488
	Maximiliano (maximiliano) [190.169.97.69]	35:01.206	SpeedITF Wide-20 6k/SpeedITF Wide-20 6k/H.261-CIF/H.261-CIF	52530/2731560	105074/5463848	10.0005/12.2088

Copyright ©2011 by H323plus.support@h323plus.com

Figura 25. Estatus de OpenMCU en videoconferencia de prueba con dos participantes.

Fuente: Elaboración Propia

En la misma fecha se hizo sesión de prueba de la videoconferencia con 3 participantes:



Figura 26. Prueba de videoconferencia OpenMCU con tres participantes.

Fuente: Elaboración Propia

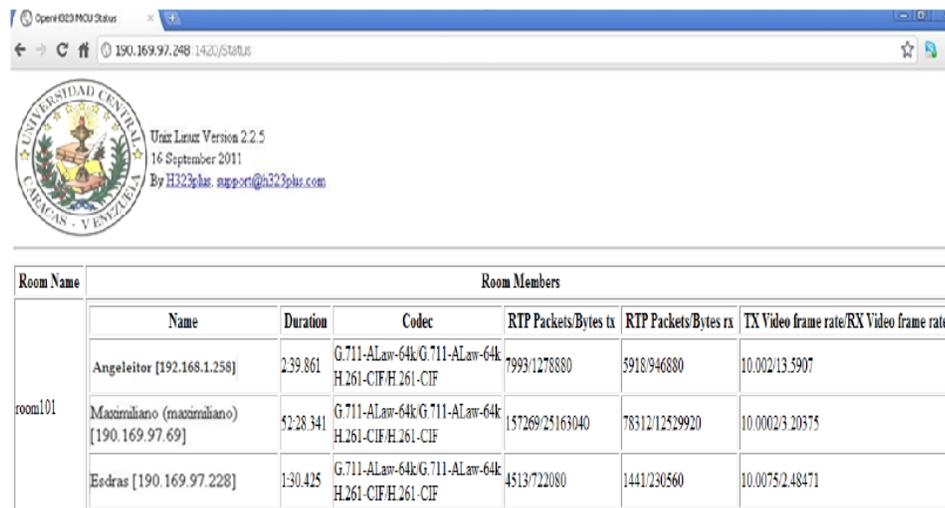


Figura 27. Estatus de videoconferencia OpenMCU con tres participantes.

Fuente: Elaboración Propia

En esta prueba se puede observar el uso de los códecs G.711 ley A y H.261 con una resolución de 352×288 pixeles (CIF) lo cual es bastante definido para usos de videoconferencia.

Luego, se logró satisfactoriamente una videoconferencia de cuatro participantes:



Figura 28. Prueba de videoconferencia OpenMCU con cuatro participantes.

Fuente: Elaboración Propia

Room Name	Room Members					
	Name	Duration	Codec	RTP Packets/Bytes tx	RTP Packets/Bytes rx	TX Video frame rate/RX Video frame rate
room101	Maximiliano (maximiliano) [192.168.1.100]	40.062	G.711- ALaw-64k/G.711- ALaw-64k H.261-CIF/H.261-CIF	1155/1784800	2/320	10.0078/4.89914
	Esdras [190.169.97.228]	3:05.009	G.711- ALaw-64k/G.711- ALaw-64k H.261-CIF/H.261-CIF	9247/1479520	1/160	10.0046/5.07631
	Alejandro [192.168.15.158]	5:06.609	G.711- ALaw-64k/G.711- ALaw-64k H.261-CIF/H.261-CIF	15320/2451200	9154/1464640	10.0027/3.93443
	Angeleitor [192.168.1.258]	5:04.853	G.711- ALaw-64k/G.711- ALaw-64k H.261-CIF/H.261-CIF	15194/2431040	10177/1628320	10.0014/5.67845

Figura 29. Estatus de videoconferencia OpenMCU con cuatro participantes.

Fuente: Elaboración Propia

Esta última prueba involucraba a la red interna de la universidad y dos localidades exteriores, demostrando así las capacidades de esta Unidad controladora Multipunto.

4.2 Interconexión con otros Equipos

Para verificar capacidades de interconexión, se hizo pruebas con OpenMCU y un signaling gateway que permite la transcodificación del estándar H.323 a SIP. Además, se hizo la conexión a un equipo Polycom desde el OpenMCU. En las primeras pruebas con el Gateway, estos fueron los resultados:



The screenshot shows a web browser window titled 'OpenH323 MCU Status' with the URL '190.169.97.248:1420/Status'. The page content includes system information for 'OpenMCU!' (Unix Linux Version 2.2.5, 16 September 2011) and a table of room members and call logs.

Room Name	Room Members					
	Name	Duration	Codec	RTP Packets/Bytes tx	RTP Packets/Bytes rx	TX Video frame rate/RX Video frame rate
701	901 [190.169.97.212]	1:54.407	G.711-A-Law-64k/G.711-A-Law-64k none/none	5720/915200	5718/914880	0/0
room101	Maximiliano (maximiliano) [190.169.97.69]	3:43.102	G.711-u-Law-64k/G.711-u-Law-64k none/none	11155/1784800	11154/1784640	0/0
	Esdras [190.169.97.228]	4:20.953	G.711-u-Law-64k/G.711-u-Law-64k none/none	13048/2087680	2/320	0/0

Copyright ©2011 by H323plus, support@h323plus.com

Figura 30. Estatus del OpenMCU en la comunicación entre equipos SIP y terminales H.323 (Audio) interconectando el Gateway con la OpenMCU de la UCV.

Fuente: Elaboración Propia

En esta figura se puede apreciar, que solo existe audio pues se deshabilitó video para la primera fase de pruebas de transcodificación. Además, se nota que el Gateway crea su propia sala de chat. Esto ocurre debido a que por defecto, el troncal del

Gateway está direccionado al MCU con el nombre de 701, lo cual posteriormente se corrigió, cambiando el nombre por la sala de conferencia correspondiente. Luego se hicieron pruebas de transcodificación completas, en las que junto con el audio se incluye el video. Las siguientes figuras muestran los resultados obtenidos:

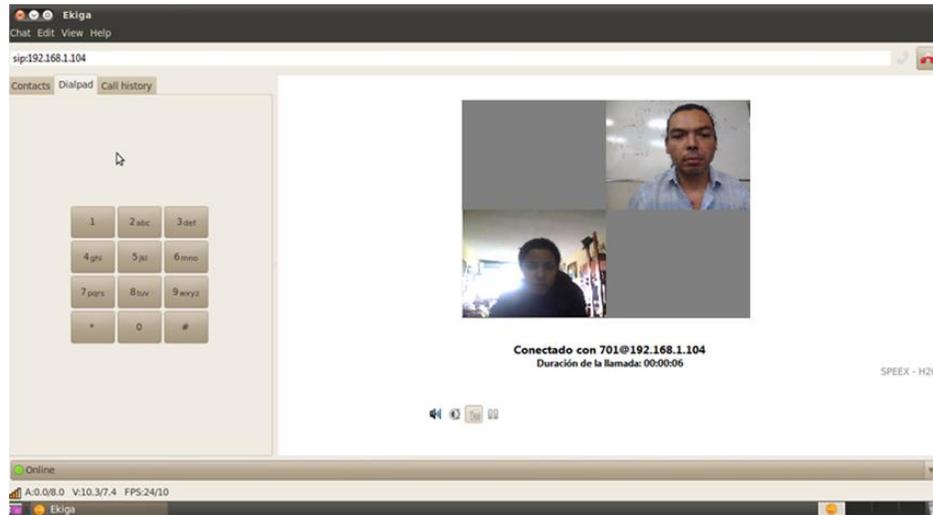


Figura 31. Prueba de llamada entre equipos SIP y terminales H.323 (Audio y Video) interconectando el Gateway con la MCU de la UCV con 2 participantes.

Fuente: Elaboración Propia

Room Name	Room Members					
	Name	Duration	Codec	RTP Packets/Bytes tx	RTP Packets/Bytes rx	TX Video frame rate/RX Video frame rate
701	901 [190.169.97.212]	3:58.960	SpeexiETFWide-20.6k/SpeexiETFWide-20.6k H.261-CIF/H.261-CIF	5974/310648	11501/598052	10.001/5.47488
	Maximiliano (maximiliano) [190.169.97.69]	35:01.206	SpeexiETFWide-20.6k/SpeexiETFWide-20.6k H.261-CIF/H.261-CIF	52530/2731560	105074/5463848	10.0005/12.2088

Copyright ©2011 by [H323plus, support@h323plus.com](mailto:support@h323plus.com)

Figura 32. Estatus de llamada entre equipos SIP y terminales H.323 (Audio y Video) interconectando el Gateway con la MCU de la UCV con 2 participantes.

Fuente: Elaboración Propia

En estos resultados se evidencia la capacidad del OpenMCU para conferencias transcodificadas y por la zona H.323 puede ser multiusuario.

Luego, este equipo se puso a prueba llamando a un equipo Polycom en modo listener, el cual fue la Sala de Triage Virtual. Los resultados de esta experiencia son satisfactorios y se muestran a continuación:

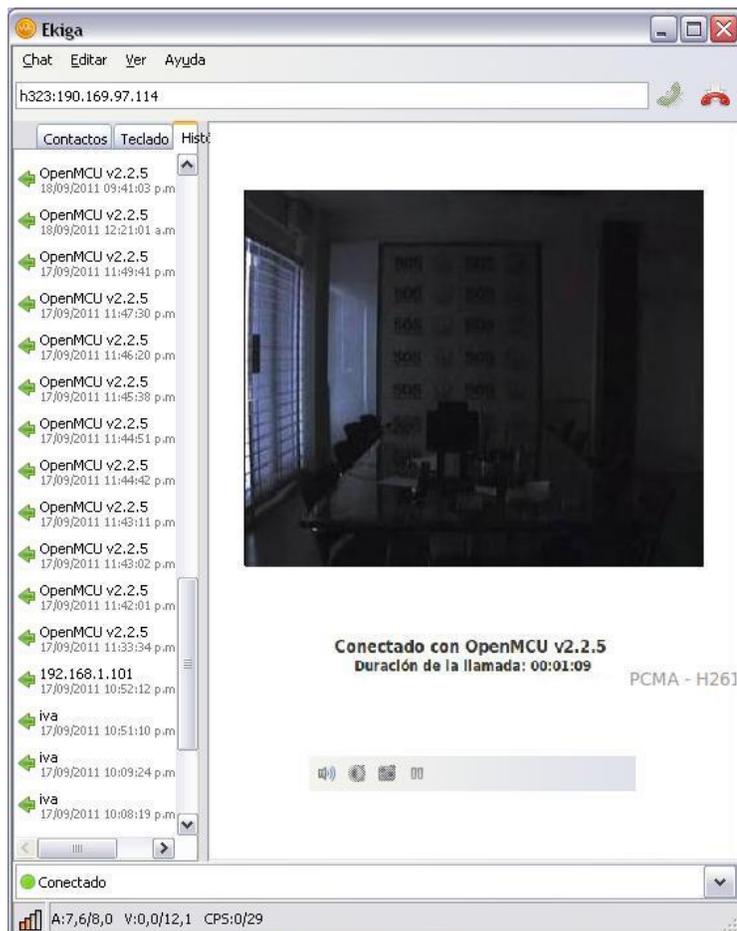


Figura 33. Prueba de llamada entre OpenMCU y Equipo Polycom

Fuente: Elaboración Propia

4.3 Pruebas de Rendimiento con Iperf

Finalmente se muestran los resultados utilizando el programa *Iperf* en la sesión de videoconferencia entre 3 clientes y el OpenMCU ubicado en el Laboratorio de Comunicaciones de la Escuela de Ingeniería Eléctrica con su respectivo análisis. Las pruebas hechas fueron sobre la cantidad de paquetes enviados y recibidos bien sean paquetes de audio o video. Lo importante a verificar son las características siguientes:

Tabla 8. Medición de latencia, jitter y pérdidas entre OpenMCU y Cliente Ekiga

Fuente: Elaboración Propia

[ID] Interval Datagramas	Latencia [ms]	Bandwidth [Mbits/seg]	Jitter [ms]	Paquetes Perdidos	Paquetes Totales	% Paq Perdidos
[3] 0.0- 1.0 sec	1.17	9.98	0.148	17	850	2.00
[3] 1.0- 2.0 sec	1.18	10	0.079	0	851	0.00
[3] 2.0- 3.0 sec	1.18	10	0.061	0	850	0.00
[3] 3.0- 4.0 sec	1.18	10	0.142	0	850	0.00
[3] 4.0- 5.0 sec	1.18	10	0.080	6	850	0.71
[3] 5.0- 6.0 sec	1.18	10	0.177	0	850	0.00
[3] 6.0- 7.0 sec	1.18	10	0.228	0	850	0.00
[3] 7.0- 8.0 sec	1.18	10	0.191	0	850	0.00
[3] 8.0- 9.0 sec	1.18	10	0.780	0	851	0.00
[3] 9.0- 10.0 sec	1.18	10	0.107	0	850	0.00
[3] 10.0- 11.0 sec	1.18	10	0.125	15	850	1.76
[3] 11.0- 12.0 sec	1.18	9.84	0.191	0	850	0.00
[3] 12.0- 13.0 sec	1.19	9.82	0.212	0	850	0.00
[3] 13.0- 14.0 sec	1.18	9.91	1.167	6	844	0.71
[3] 14.0- 15.0 sec	1.16	9.74	1.242	13	845	1.54
PROMEDIO	1.18	9.95	0.329	3.4	849.4	0.40

Retardo (Latencia)

Es la cantidad de tiempo requerido para transmitir y recibir una señal de audio y video. Esto es un aspecto muy importante en los sistemas de videoconferencia ya que si los paquetes tardan en llegar, las personas que participan en una sesión de videoconferencia notaran que se pierde el sincronismo entre lo transmitido y lo recibido afectando la videoconferencia. Una latencia de 50 ms, su efecto es casi

imperceptible, pero arriba de 150 ms ya los usuarios lo detectan y puede notarse la falta de sincronía entre el movimiento de los labios y la voz.

Analizando los resultados obtenidos de las mediciones con el OpenMCU y dos clientes Ekiga se puede ver que se obtuvo un promedio 1,18 ms de latencia con lo cual es aceptable ya que existe un buen sincronismo en ambos extremos de la videoconferencia.

Jitter

A la variación de la latencia se le conoce como jitter. Este normalmente es originado por alto tráfico que temporalmente reduce las capacidades de la red o por cambios en el camino de los paquetes. Las redes IP no garantizan la entrega de los paquetes en orden, ni de forma inmediata, mas sin embargo, los paquetes de voz y video pueden ser etiquetados para recibir un trato preferente a la hora de atravesar la red. No obstante lo anterior, tales medidas no terminan con el jitter, sólo reducen sus efectos.

Analizando los resultados obtenidos de las mediciones con el OpenMCU y dos clientes Ekiga se puede ver que se obtuvo un promedio 9,95 ms de jitter, lo cual es un valor bastante bueno ya que lo aceptable en las transmisiones de VoIp es mantenerse en un valor menor a los 50ms.

Pérdida de paquetes

La pérdida de paquetes significa que los elementos de la videoconferencia, es decir los paquetes de datos, no llegan a su destino. Sus efectos son video entrecortado durante la sesión de videoconferencia, fallas de audio, video estático e incluso hasta la misma pérdida de la comunicación. La pérdida de paquetes se produce por descartes de paquetes que no llegan a tiempo al receptor.

Para una videoconferencia aceptable se deben tomar en cuenta:

- Una pérdida de paquetes de 1% puede producir congelamiento en el video y/o pérdida del audio.
- Una pérdida de paquetes de 2% puede hacer que el video sea malo, aunque el audio puede sonar algo aceptable.
- Una pérdida de paquetes por arriba del 2% es inaceptable.

Analizando los resultados obtenidos de las mediciones con el OpenMCU y dos clientes Ekiga se puede ver que se obtuvo un promedio 0,40% de pérdida de paquetes con lo cual es aceptable y se cumple con los parámetros anteriormente señalados.

CAPÍTULO V

5. CONCLUSIONES

En el proceso de desarrollo del presente Trabajo Especial de Grado, la implementación de la Unidad de Control Multipunto, el estudio de los elementos involucrados en la misma, y los resultados obtenidos, se generan las siguientes conclusiones:

- La conclusión principal es que la implementación de la Unidad de Control Multipunto, fue exitosa. Se logró un desempeño en las funcionalidades de esta entidad de videoconferencia. Además con los estudios involucrados se logró la completa compatibilidad de todos los equipos que trabajan con el protocolo H.323. Asimismo, se escogió el software cliente de mejor adaptación y compatibilidad con el MCU.

- Otra conclusión importante, es el logro de interconexión con un Gateway de señalización el cual permite ampliar los límites y el alcance de este software para videoconferencia. Demás está decir, que queda para la comunidad universitaria este trabajo, para su uso y desarrollo posterior.

- Las videoconferencias son una herramienta indispensable hoy en los ámbitos académicos y científicos. Con la implementación de OpenMCU los límites de ambos se extenderán rápido y fácilmente.

- Esta Unidad Multipunto, puede ser replicada en cualquier instancia de la universidad y ser usada bien sea por profesores, alumnos y/o empleados todos miembros de la comunidad, que requieran los servicios de videoconferencia.

- Este trabajo permite la reestructuración de las videoconferencias en la universidad pues como ya se probó existe una completa compatibilidad con los equipos Polycom, los cuales son los equipos de videoconferencia actuales en nuestra casa de estudios.

- Otra conclusión importante es la independencia del hardware de videoconferencia pues cualquier equipo de computación que cumpla los requerimientos necesarios, se puede convertir en central de videoconferencia.

- El desempeño del OpenMCU se estableció con la medición de parámetros que fueron evaluados con la ayuda de la herramienta Iperf para la medición de la calidad de servicio. Demostrándose que el OpenMCU funcionando en la red de datos de la UCV genera bajos niveles de latencia y una pérdida de paquetes lo cual resulta bastante aceptable para una red con alto tráfico.

- Al manejar una política de código abierto, el sistema OpenMCU se hace viable para su ampliación, estudio, reconfiguración, replanteamiento, haciéndolo de fácil integración a otros equipos y redes pertenecientes a la comunidad.

- Sin embargo, esta política de código abierto al no tener fines de lucro, reduce el interés de fabricantes de hardware en abastecer de equipos a los usuarios de software libre, generando bastantes restricciones a la hora de implementación de software novedosos, mas aún en nuestro país, en el cual la filosofía de compartición de conocimiento ha tenido poco auge.

- Otro aspecto en contra, es que si bien el costo del equipo se reduce a cero, el costo operativo sube grandemente, debido a la complejidad de las redes basadas en el estándar H.323. Este tipo de estándar exige a los equipos mayor tasa de transmisión, a su vez demandan mayores tasas de procesamiento.

- El funcionamiento y la calidad de las videoconferencias proporcionadas por el OpenMCU están estrechamente relacionadas a la capacidad de procesamiento del PC en el cual se ejecutan así como también con la capacidad del ancho de banda de la red donde trabaja, aunque esta última en menor consideración.

- No se puede perder de vista el hecho de que aun para la más sencilla de las aplicaciones, la instalación se convierte en un proceso engorroso y complicado, hecho que limita la motivación a la experimentación con el software libre.

- Otra conclusión, que es importante mencionar, es que el OpenMCU no representa rival alguno para los equipos y software de fabricantes los cuales pueden manejar un gran flujo de datos.

- Es difícil el control de vista en pantalla, pues se activan con la voz. Si en la sala existen más de cuatro personas hablando al mismo tiempo sin un moderador, se presentará intermitencias de video debido a la indecisión del OpenMCU de dar video al que está hablando.
- El OpenMCU no tiene cancelación de eco, por lo tanto, si no se configuran correctamente los micrófonos corre el riesgo de un feedback intempestivo o el caso de intermitencia de video mencionado anteriormente.
- Fue necesario gran conocimiento sobre el software libre, sus limitaciones y alcances, el sistema de programación orientada a objetos, protocolos de comunicaciones, pues fueron un factor clave en el desarrollo exitoso del presente trabajo.

CAPÍTULO VI

6. RECOMENDACIONES

De las experiencias obtenidas durante la implementación de la Unidad de Control Multipunto, los diferentes inconvenientes resueltos y los resultados obtenidos, se conciben las siguientes recomendaciones:

- La primera de las recomendaciones, debe ser la medición del número máximo de usuarios por sala, y a su vez el número máximo de salas que tiene capacidad el OpenMCU, así también su desempeño y requerimientos en estas condiciones máximas.

- Además otra recomendación sería hacer el estudio de usuarios de videoconferencia, por tipología, para lograr adecuar este OpenMCU a usos específicos.

- Una recomendación necesaria es el estudio para el mejoramiento de la interfaz gráfico, pues presenta muchas deficiencias. En la configuración de parámetros suele desligarse del OpenMCU en sí, siendo necesario reiniciar el sistema. A esto se une el control nulo que se tiene sobre los participantes en cada una de las salas de videoconferencia, al no haber control de derecho de palabra digital, la posibilidad de ejecutar comandos para la admisión o no de participantes y cerrar salas.

- Hecha esta recomendación cabe la sugerencia de un estudio de la integración del OpenLDAP con el OpenMCU. OpenLDAP es una implementación libre y de código abierto del protocolo Lightweight Directory Access Protocol (LDAP) desarrollada por el proyecto OpenLDAP, el cual es un protocolo a nivel de aplicación el cual permite el acceso a un servicio de directorio ordenado y distribuido para buscar información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse

consultas. Es por eso que el OpenLDAP es una opción inmejorable a la hora de control de participantes de videoconferencia, pues habitualmente, en los servidores LDAP se almacena la información de autenticación (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc.). Además, OpenLDAP posee una interfaz grafica superior en cuanto a funcionalidades y estética y si este llega a combinarse con el OpenMCU, le daría capacidades para substituir algunos de los MCU de fabricantes presentes en la universidad.

- Otro estudio pertinente es en cuanto a las librerías y proyectos de compilación de software libre, en especial en cuanto a la optimización de recursos y tiempo de compilación. Uno de los más emblemáticos es el gcc-4.4 el cual fue clave para la instalación de los plugins de video del OpenMCU.

- También es deseable la realización de un estudio de seguridad que proteja la información que pasa a través de un OpenMCU bien sea de forma encriptado o alguna otra forma de protección de datos. Todo esto con miras a la expansión del OpenMCU.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Red Interinstitucional para Videoconferencia Universidad Autónoma del Estado de Hidalgo. *Videoconferencia* Recuperado el 15 de Noviembre de 2010 de la World Wide Web: <http://virtual.uaeh.edu.mx/riv/videoconferencia.php>
- [2] Newman Markus, *Códec G-711*. Recuperado el 12 de Agosto de 2011 de la World Wide Web: <http://www.voipforo.com/codec/codec-g711--ley.php>.
- [3] Newman Markus, *Códec G.728*. Recuperado el 12 de Agosto de 2011 de la World Wide Web: www.voipforo.com/codec/codec-g728.php.
- [4] Unión Internacional de las Telecomunicaciones (UIT), *Códec G.729*. Recuperado el 12 de Agosto de 2011 de la World Wide Web: <http://www.itu.int/rec/T-REC-G.729-200701-I/es>
- [5] Lajos Hanzo, *Códec de Voz*. Recuperado el 20 de Febrero de 2011 de http://www.mobile.ecs.soton.ac.uk/speech_codecs/standards/gsm.html
- [6] ComuSOFT, *Códec GSM*. Recuperado el 12 de Agosto de 2011 de la World Wide Web: <http://www.comusoft.com/que-es-el-codec-gsm-caracteristicas-y-tipos>
- [7] Newman Markus, *Códecs*. Recuperado el 30 de Abril de 2011 de la World Wide Web: <http://www.voipforo.com/codec/codecs.php>
- [8] Digimad, *Códec Video*. Recuperado el 3 de Mayo de 2011 de la World Wide Web: <http://www.digimad.es/h261-h263-h264-codecs-video.html>
- [9] Saruth Antoine, *Códecs*. Recuperado el 3 de Mayo de 2011 de la World Wide Web: <http://es.kioskea.net/contents/internet/tcp.php3#goprev>
- [10] Más Adelante, *Protocolo UDP*. Recuperado el 9 de Febrero de 2011 de la World Wide Web: <http://www.masadelante.com/faqs/udp>

- [11] Alegsa, Protocolos y Señalización. Recuperado el 6 de Marzo de 2011 de la World Wide Web: <http://www.alegsa.com.ar/Dic/ip.php>
- [12] Alba Enrique, Protocolo IP. Recuperado el 11 de Enero de 2011 de la World Wide Web: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/red/ip.html>
- [13] Packetizer, Protocolo H.323. Recuperado el 3 de Febrero de 2011 de la World Wide Web: <http://www.packetizer.com/ipmc/h323/papers/14>
- [14] Newman Markus, Estándar H.323. Recuperado el 2 de Marzo de 2011 de la World Wide Web: <http://www.voipforo.com/H323/H323ejemplo.php>
- [15] Ferrell Bob, Librería OpenH.323. Recuperado el 2 de Marzo de 2011 de la World Wide Web: <http://openh323.sourceforge.net/>.
- [16] H.323 Plus Project, Librería OpenH.323. Recuperado el 2 de Marzo de 2011 de la World Wide Web: <http://www.voxgratia.org/docs/openh323.html>

BIBLIOGRAFÍA

- Ferreira, P. (Septiembre de 2005). Recuperado en Diciembre de 2010, de <http://www.fccn.pt/files/documents/D2.05.2.PDF?947cda2253a1dc58fe23dc95ac31cbed=e3fbccc61614de6343e90171d1d8d1c8>
- Fernández, L. (2008). H.323. *Difusión Multimedia. Año 3*. (N° 4), 108.
- <http://www.voip-info.org/>. (s.f.). Recuperado Enero 2011, de <http://www.voip-info.org/>.
- <http://openh323.sourceforge.net/>. (s.f.). Recuperado en Diciembre de 2010, de <http://openh323.sourceforge.net/>: <http://openh323.sourceforge.net/>
- <http://www.gnu.org/home.es.html>. (s.f.). Recuperado en Enero de 2011, de <http://www.gnu.org/home.es.html>: <http://www.gnu.org/home.es.html>.
- <http://openh323.sourceforge.net/> Recuperado el 2 de Marzo de 2011 de la World Wide Web: <http://openh323.sourceforge.net/>.
- <http://www.voxgratia.org/>. Recuperado el 2 de Marzo de 2011 de la World Wide Web: <http://www.voxgratia.org/>
- ITU-T Rec. H.264 / ISO/IEC 11496-10, “*Advanced Video Coding*”, Final Committee Draft, Document JVT- E022, Septiembre 2002
- Xiph.org Foundation. Theora Specification. April 16, 2008. Ruiz, Rosalba. El H.264, Un Nuevo Estándar para Compresión de Video y su Aplicación a la Videoconferencia. (Paper). 2007