

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

División de Electrónica, Computación, Información y Comunicaciones
Programa de Graduados



**Estudio de transporte de audio digital en una LANE usando
codificación MPEG-1 Capa III**

TESIS

Presentada como requisito parcial para obtener el grado académico de
Maestro en Administración de Telecomunicaciones

Yuri Islas Cortés

Mayo 2002

**Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey**

División de Electrónica, Computación, Información y Comunicaciones.
Programa de Graduados en Electrónica, Computación, Información y Comunicaciones.

Los miembros de este comité de tesis recomendamos que la presente tesis del
Ing. Yuri Islas Cortés sea aceptada como requisito parcial para obtener el grado académico de:

Maestro en Administración de Telecomunicaciones

Comité de Tesis

Ramón M. Rodríguez Dagnino, Ph.D.
Asesor

José Ramón Rodríguez Cruz, Ph.D.
Sinodal

José I. Gómez Quiñones, M.Sc.
Sinodal

Aprobada

David Garza Salazar, Ph.D.
Director del programa

Monterrey, N.L. Mayo 2002

*A Hilda y René,
mi máximo ejemplo de Amor, Amistad, Pensamiento y Humanismo.*

Agradecimientos.

Al Instituto Tecnológico y de Estudios Superiores de Monterrey y al Consejo Nacional de Ciencia y Tecnología por darme la oportunidad de realizar los estudios de maestría.

Al Dr. Ramón Martín Rodríguez Dagnino por su enseñanza y guía en esta tesis.

Al Dr. José Ramón Rodríguez Cruz por su atención y valiosos comentarios .

Al M.C. José I. Gómez Quiñones por su amistad y consejos.

Mi más grande y eterna gratitud a mis hermanos Elsa Salazar y José Alejandro López por su cariño y apoyo, que hicieron de mi estancia en Monterrey una época maravillosa e inolvidable.

A mi hermano Iván, por estar siempre cerca de mi.

A mis queridísimos amigos y compañeros del Tec: Miguel Bennetts, Isabel Villanueva, Enrique Stevens, Magda Ruiz, Oscar Rodríguez, Fortino Vázquez, Ulises Pineda, Evelia Alarcón y Vladimir Catana; por compartir como una gran familia las increíbles experiencias del posgrado y las extraordinarias carnes asadas.

Un especial agradecimiento a los colegas de la “Becky Station”: Cuitláhuac López, Fernando Solís y Germán Gutiérrez; por ser un divertido equipo de estudio, de solidaridad y excelente amistad.

A los profesores y personas con quienes conviví en el Centro de Electrónica y Telecomunicaciones (CET). A todos ellos, muchas gracias.

Resumen.

Hoy en día, la tendencia de las redes de telecomunicaciones es la convergencia. Debido al explosivo crecimiento de Internet y la popularidad de sus tecnologías, se busca la integración de servicios en una sola red. Esto es, que en solo un enlace podamos tener acceso a datos, voz y video con garantías de calidad de servicio, seguridad y confiabilidad.

En la evolución de los diferentes tipos de red hacia dicha convergencia, existen numerosas propuestas tecnológicas que pretenden lograr las ventajas de la conmutación de circuitos combinadas con el ruteo de paquetes. Por un lado tenemos a ATM que se perfilaba como la plataforma que cumpliría con los requerimientos de los nuevos servicios de comunicación y transferencia de información. El enfoque de ATM contempla sobre todo dar esas garantías de calidad de servicio y administración eficiente de tráfico por multicanalización estadística. Las redes basadas en esta tecnología tienen como rival a aquellas basadas en el protocolo de Internet (IP), cuya universalidad e implementación lo hacen parecer el candidato más fuerte para la integración de todos los servicios antes mencionados. No sin antes darle la habilidad de manejar todos aquellos requerimientos que demandan las comunicaciones en tiempo real y las aplicaciones multimedia.

Existen diversos esfuerzos por conseguir lo mejor de ambos mundos, entre ellos, una mejor y más eficiente convivencia de IP y ATM. Uno de los métodos más empleados es el enfoque del foro ATM, llamado *LAN Emulation*, que si bien no es perfecto, su desempeño es bastante aceptable en el despliegue de redes de paquetes medianas con un *backbone* de alta velocidad.

Como consecuencia y parte de la revolución de Internet, está también la del mundo digital. La manipulación, almacenamiento y transporte de video y audio no son la excepción. Gracias a los avances en codificación y compresión y procesamiento de cómputo podemos ahora llevar ese tipo de información a niveles de control extraordinarios.

Dentro del mundo del audio digital se encuentra como gran protagonista el formato MP3 que es como comúnmente se le denomina al método de codificación de MPEG-1 Capa III, cuya adopción ha sido un éxito y que es incluso motivo de polémica.

En la presente tesis, se hace un estudio de transporte de este formato a través de una LAN emulada que es la red de datos del Campus Monterrey del ITESM. El estudio nos permite conocer las capacidades de una red para este tipo de aplicaciones y utilizar esta información para futuros proyectos que involucren desempeño e implementación de aplicaciones multimedia sobre redes IP.

Contenido.

CAPÍTULO 1	10
INTRODUCCIÓN.	10
1.1 ALCANCE DE LA TESIS Y MOTIVACIÓN.	11
1.2 ORGANIZACIÓN DE LA TESIS.....	11
CAPÍTULO 2	13
AUDIO DIGITAL	13
2.1 FUNDAMENTOS DE AUDIO DIGITAL.....	13
2.2 FUNDAMENTOS DE COMPRESIÓN DE AUDIO	16
2.2.1 MPEG-1	20
2.3 CODIFICACIÓN Y COMPRESIÓN DE AUDIO MPEG 1 CAPA III	20
2.3.1 El modelo psicoacústico.....	23
CAPÍTULO 3	24
MODO DE TRANSFERENCIA ASÍNCRONO (ATM)	24
3.1 INTRODUCCIÓN.	24
3.2 TECNOLOGÍA ATM.....	25
3.2.1 Analogía con el sistema tradicional telefónico.	25
3.3 MODELO DE REFERENCIA B-ISDN	26
3.4 ENCABEZADO DE LA CELDA ATM (CELL HEADER).	28
3.5 CONMUTADORES.....	29
3.6 CIRCUITOS VIRTUALES.....	30
3.7 PARÁMETROS DE CALIDAD DE SERVICIO (QOS PARAMETERS).....	31
3.8 DESCRIPTORES DE TRÁFICO.	31
3.9 CATEGORÍAS DE SERVICIO EN ATM.....	32
3.10 CAPAS DE ADAPTACIÓN ATM (AAL)	32
3.11 CAPA DE ADAPTACION ATM 5 (AAL5).....	33
CAPÍTULO 4	35
INTERNET PROTOCOL	35
4.1 LA ARQUITECTURA DE TCP/IP.	35
4.2 FORMATO DE PAQUETES Y DIRECCIONES DE INTERNET.	39
4.3 EL PROTOCOLO IP	41
4.3.1 Ruteadores (Routers).....	41
4.3.2 Unidad Máxima de Transmisión (Maximum Transmission Unit).	41
4.3.3 Fragmentos.	41
4.4 TCP.	42
4.5 UDP.....	43
4.6 IP MULTICAST.....	43
4.7 RTP (REAL TIME PROTOCOL).	43
CAPÍTULO 5	45
IP SOBRE ATM	45
5.1 INTRODUCCIÓN.	45
5.2 EMULACIÓN LAN.	45
5.2.1 LEC (LAN Emulation Client).	46
5.2.2 LES (LAN Emulation Server).	46
5.2.3 BUS (Broadcast and Unknown Server).	46
5.2.4 LECS (LAN Emulation Configuration Server).	47

5.3 PUENTES LANE	47
5.4 PROS Y CONTRAS.	47
5.5 IMPLEMENTACIÓN MULTICAST PARA LANE.....	47
CAPÍTULO 6.....	48
PRUEBAS Y ESTADÍSTICAS DE RED.....	48
6.1 DESCRIPCIÓN DEL EQUIPO	48
6.2 DESCRIPCIÓN DEL CONMUTADOR	49
6.3 DESCRIPCIÓN DEL SOFTWARE PARA DIFUSIÓN DE AUDIO.....	50
6.4 DESCRIPCIÓN DE LAS PRUEBAS REALIZADAS.	50
6.5 RESULTADOS OBTENIDOS.....	52
6.6 CÁLCULO DE ENTREGA Y RETRASO DEL CONMUTADOR	60
CAPÍTULO 7.....	64
CONCLUSIONES.....	64
REFERENCIAS.	66
APENDICE A. REGISTRO DE ESTADÍSTICAS.....	67
APÉNDICE B. ACRÓNIMOS.	85

Lista de Figuras.

FIGURA 2.1 PROCESO DE AUDIO DIGITAL	13
FIGURA 2.2 ENMASCARADO DE FRECUENCIAS.....	16
FIGURA 2.3 UMBRAL ABSOLUTO.....	17
FIGURA 2.4 MASKING THRESHOLD.....	18
FIGURA 2.5 CODIFICACIÓN POR SUB-BANDA.....	19
FIGURA 2.6 PROCESO DE CODIFICACIÓN Y DECODIFICACIÓN MPEG	21
FIGURA 3.1 TRANSMISIÓN BÁSICA TDM.....	25
FIGURA 3.2 MODELO DE REFERENCIA B-ISDN.....	27
FIGURA 3.3 FORMATO DE LA CELDA ATM Y SU ENCABEZADO	28
FIGURA 3.4 CONMUTADOR GENÉRICO TÍPICO.....	30
FIGURA 3.5 PROCESO AAL5	34
FIGURA 4.1 PROTOCOLOS DE INTERNET	36
FIGURA 4.2 ENCAPSULADO DE PDUs EN TCP/IP E INFORMACIÓN DE DIRECCIÓN EN LOS ENCABEZADOS	37
FIGURA 4.3 INTERNET Y LAS CAPAS DE INTERFASE DE RED	38
FIGURA 4.4 ENCABEZADO DEL PAQUETE IP	39
FIGURA 4.5 FORMATOS DE DIRECCION DE PAQUETES IP	40
FIGURA 5.1 CONFIGURACIÓN DE LAN EMULADA	46
FIGURA 6.1 ESQUEMA DE LA RED DEL CAMPUS[12].....	51
FIGURA 6.2 TIEMPO DE RECORRIDO DE PAQUETES A DIRECCIÓN LANE DEL SERVIDOR	53
FIGURA 6.3 TIEMPO DE RECORRIDO DE PAQUETES AL SERVIDOR POR INTERFASE ETHERNET	54
FIGURA 6.4 RETARDO CALCULADO POR TRACERROUTE HASTA EL SERVIDOR.....	55
FIGURA 6.5 SEGMENTOS TCP RECIBIDOS/SEGUNDO DEL NODO EXTERNO	56
FIGURA 6.6 DATAGRAMAS RECIBIDOS/SEGUNDO DURANTE RECEPCIÓN DE AUDIO	57
FIGURA 6.7 BYTES RECIBIDOS/SEGUNDO EN LA TARJETA DE RED	57
FIGURA 6.8 PAQUETES RECIBIDOS/SEGUNDO	58
FIGURA 6.9. SEMENTOS TCP RECIBIDOS/SEGUNDO	58
FIGURA 6.10 ANCHO DE BANDA OBSERVADO DEL FLUJO DE AUDIO.....	59
FIGURA 6.11 DATOS GENERALES DEL FLUJO DE AUDIO	59
FIGURA 6.12 ESTADÍSTICAS DE TRANSMISIÓN. APLICACIÓN REAL PLAYER	60
FIGURA 6.13 ENTREGA DEL CONMUTADOR EN CADA TASA DE TRANSMISIÓN.....	62
FIGURA 6.14 RETRASO GENERADO EN EL CONMUTADOR EN CADA TASA DE TRANSMISIÓN	63

Lista de Tablas.

TABLA 2.1 ESTÁNDARES DE CODIFICACIÓN MONOAURAL DE VOZ COMÚNMENTE USADOS[3]	15
TABLA 2.2 AUDIO DE BANDA AMPLIA[8].....	15
TABLA 2.3 CAPAS MPEG AUDIO	22
TABLA 6.1 TIEMPOS DE RETRASO A NODOS EXTERNOS Y DIFUSORA DEL CAMPUS.....	55
TABLA 6.2 DATOS DE LA DIFUSIÓN CON NÚMERO MÁXIMO DE USUARIOS	56
TABLA 6.3 RETRASO ENTRE WIENER Y GABOR	60
TABLA 6.4 RETRASO Y ENTREGA DEL CONMUTADOR.....	62

Capítulo 1

Introducción.

El panorama actual de telecomunicaciones presenta un esquema de redes separadas para proveer los distintos tipos de servicios, voz, video y datos. La tendencia en el desarrollo de las futuras redes es la convergencia. La integración de estos servicios en un solo enlace pretende lograr la eficiencia máxima en el uso de la red, reducir los costos de operación y proveer una plataforma flexible para adaptarse a una constante evolución, a un impredecible y cada vez más competitivo escenario de las comunicaciones. Una red que manipule un solo flujo de datos es donde se enfoca el interés de la consolidación de estos tres tipos de tráfico.

Las redes de paquetes de hoy están optimizadas para el tráfico de datos, sin embargo no pueden soportar tráfico de voz o aplicaciones de tiempo real con la suficiente calidad para satisfacer los requerimientos del usuario. Para que una red de paquetes multiservicio tuviera éxito, las áreas de calidad de servicio(QoS), características de funcionalidad, y confiabilidad deben estar al mismo nivel o más arriba que las de las redes de conmutación de circuitos. La calidad de servicio(QoS) será el concepto más importante en el mundo de las telecomunicaciones en los próximos años[4].

En el proceso de convergencia han surgido diversas tecnologías. Cada una está optimizada para diferentes aplicaciones, tienen diferentes objetivos de diseño y una herencia diferente, lo que determinará la extensión del papel que jugará cada una en las redes del futuro.

ATM (Asynchronous Transfer Mode) ofrece categorías que comprenden las altamente solicitadas capacidades de calidad de servicio. Aunque ATM fue específicamente diseñado para la consolidación de tráfico de voz, video y datos, la mayoría de los usuarios a la fecha lo usan para aplicaciones de datos a alta velocidad. Solo recientemente ha habido un incremento en la actividad con voz sobre ATM en el área de proveedores de servicios. Sin embargo hay un modelo claramente emergiendo que usa ATM como una técnica de consolidación con acceso multiservicio para proveer conectividad a redes tradicionales de conmutación de voz y redes de paquetes de datos tales como Frame-Relay e Internet. ATM está diseñado para dirigir los asuntos que se presentan en cuanto a retrasos y estado latente transportando diferentes tipos de tráfico, incluyendo voz o tráfico sensible a retrasos, a través de la red. Tales retrasos dieron lugar a la selección de la longitud fija de 53 Bytes de la celda. El tamaño de la celda no es óptimo para ningún tipo de tráfico en particular, pero está optimizado para una mezcla de estos[1].

Los estrictos requerimientos del tráfico de voz y de aplicaciones multimedia han presentado tradicionalmente un enorme reto para las redes IP orientadas a no-conexión. Numerosas técnicas son empleadas para cumplir con el objetivo de proveer calidad de servicio en este tipo de redes. El protocolo de Internet no fue diseñado para tráfico de voz o aplicaciones en tiempo real, sino para datos en ráfagas. No contempla niveles ni calidad de servicio. Existen aplicaciones que manejan audio y voz pero con bajos niveles de calidad de servicio y aunque la versión 6 del protocolo(IPv6) ya contempla estos aspectos, la tecnología permanece aún inmadura y poco probada, mientras diversos estándares compiten por posicionarse en el mercado.

La especificación más importante es la familia de estándares H.323 ampliamente implementada para voz sobre IP. Sin embargo, existen puntos como algoritmos de compresión, el uso de Q.931 como mecanismos de llamada, interoperabilidad, etc., que indican que no es la solución óptima[10]. La IETF(Internet Engineering Task Force) ha estado también trabajando en desarrollar especificaciones para habilitar aplicaciones en tiempo real. La iniciativa más prominente es la de servicios diferenciados(DiffServ). La IETF ha trabajado en otro mecanismo que puede ser usado en conjunto que permitirá a los proveedores ofrecer múltiples clases de

servicio. Este mecanismo es el MPLS (Multiprotocol Label Switching). Esencialmente MPLS combina los beneficios de tecnologías orientadas a conexión con la flexibilidad y universalidad del IP orientado a no-conexión para habilitar múltiples clases de servicio y ser ofrecidas sobre la columna vertebral de un proveedor. Diversas tecnologías pueden ser usadas en la columna vertebral, incluyendo ATM, Frame Relay y ruteadores de alta velocidad[4].

Dentro de las técnicas de compresión de material multimedia para redes convergentes, la capa tres de la especificación MPEG(Motion Picture Experts Group) audio es entre otros uno de los formatos más populares para la compresión y procesamiento de audio actualmente. Las tasas de compresión y calidad que mantiene lo hacen una opción muy atractiva para transportar y proveer audio.

1.1 Alcance de la tesis y motivación.

Al inicio de esta investigación se plantearon como objetivos el identificar los protocolos y procesos más eficientes para transportar audio en una red IP/ATM, así como la comprobación y comparación del desempeño de los mecanismos que ofrecen una QoS satisfactoria probando el formato de codificación(MP3) como la mejor alternativa viable para el transporte de audio en una red de alta velocidad.

MPLS es una de las técnicas más interesantes para experimentar el transporte de diferentes servicios en una red de las características que proporciona el modelo de IP sobre ATM. Sin embargo debido a la disponibilidad de herramientas y equipo, el transporte de audio digital se hace sobre una red que presenta un modelo distinto de convivencia entre la dos tecnologías. Este es el caso de una red tipo campus que corre sobre un esquema del foro ATM que es LAN Emulation (LANE).

El propósito es describir el comportamiento de la entrega de audio digital a través de la red del campus mediante experimentación y registro de estadísticas con herramientas basadas en software de código abierto. Pretendiendo con esto mostrar el desempeño y los distintos retrasos involucrados tales como el retraso de propagación y retraso de conmutación de la red ya sea ATM, IP, o ambas. La experimentación de entrega de audio en broadcast o bajo demanda se hace sobre el estándar de audio MPEG 1172-3 Capa III conocido popularmente como MP3, ya que debido a su amplia adopción se considera como la mejor alternativa para la manipulación y transporte de audio digital comprimido.

La realización de este tesis permitirá también contar con herramientas de difusión de material de audio con estándares abiertos que serán de utilidad para futuros proyectos que involucren estudios sobre entrega de material e información con requisitos de calidad de servicio (QoS) y restricciones en tiempo real.

1.2 Organización de la tesis.

En el capítulo 2 se examinan los fundamentos de audio digital y de compresión de audio. Se hace una breve descripción de como trabaja el formato MPEG/Audio Capa III mencionando algunas otras técnicas de compresión estandarizadas.

El capítulo 3 describe los fundamentos de la tecnología ATM, las categorías de servicio, capas de adaptación, los parámetros de servicio y los descriptores de tráfico.

En lo que concierne al capítulo 4, se examinan los conceptos principales de IP y la familia de protocolos que conforma, mencionando los aspectos más importantes en la forma como se comunican las computadoras en una red de este tipo. Se revisan también las técnicas usadas en el transporte de multimedia en Internet como lo es el RTP (Real Time Protocol).

Después de dar un enfoque del protocolo de Internet, se describe en el capítulo 5 el modelo propuesto por el foro ATM para la interconexión de redes IP/ATM, LAN Emulation,

cuyas características serán de gran utilidad para la comprensión del funcionamiento de la red del campus basada en este modelo.

El capítulo 6 hace una descripción de la arquitectura del campus y de las herramientas usadas que en este caso es el software de código abierto SHOUTcast Distributed Network Audio Server, el cual se utiliza para difusión de flujos de audio MP3 a diferentes tasas de bit. El capítulo se concentra también en el registro de estadísticas y la descripción de las pruebas realizadas a través de la red del campus, así como la interpretación y análisis del experimento.

En el capítulo 7 se concluye el estudio de tráfico de audio en una ELAN (Emulated LAN) con comentarios sobre futuras consideraciones en cuanto a redes de alta velocidad y mejoras en la compresión de material de audio.

Capítulo 2

Audio Digital

2.1 Fundamentos de audio digital

La compresión de audio digital permite el eficiente almacenamiento y transmisión de información de audio. Las diferentes técnicas de compresión ofrecen diferentes niveles de complejidad, calidad y compresión. La representación digital de información de audio ofrece muchas ventajas: alta inmunidad a ruido, estabilidad y reproducción. El audio en forma digital permite también la implementación eficiente de muchas funciones de procesamiento (ej: mezcla, filtrado y ecualización) a través de una computadora.

La conversión del dominio analógico al digital empieza por muestrear el audio de entrada en intervalos discretos de tiempo y cuantificando los valores muestra en un número discreto de niveles. La información de audio digital consiste de una secuencia de valores binarios representando el número de niveles de cuantificación para cada muestra de audio. El metodo de representar cada muestra con una palabra de código independiente es llamado PCM (Pulse Code Modulation). La figura 2.1 muestra el proceso de audio digital.

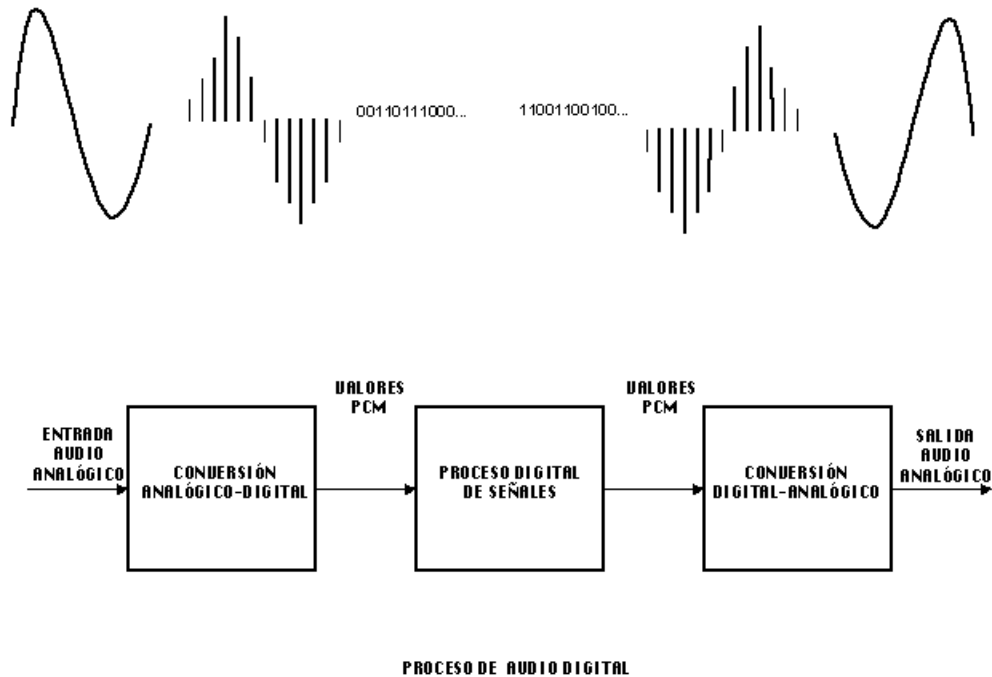


Figura 2.1 Proceso de Audio Digital

De acuerdo a la teoría de Nyquist, una señal muestreada en el tiempo puede fielmente representar señales hasta la mitad de la frecuencia de muestreo. Las tasas de muestreo típicas van de los 8 kHz a los 48 kHz. Las tasas de 8 kHz cubren un rango de frecuencia de hasta 4 kHz y por lo tanto cubre la mayoría de las frecuencias producidas por la voz. La tasa de 48 kHz cubre un rango de frecuencia de hasta 24 kHz donde cubre adecuadamente el rango de frecuencias audibles. En la práctica, el rango de frecuencia es algo menos que la mitad de la frecuencia de muestreo debido a las limitaciones de los sistemas en la práctica[7].

El número de niveles de cuantización es típicamente una potencia de 2 para hacer un uso completo de un número fijo de bits por muestra de audio para representar los niveles cuantizados. Con un espacio uniforme de pasos de cuantificación, cada bit adicional tiene el potencial de incrementar la razón señal-ruido, por 6 dB aproximadamente. El número típico de bits por muestra usado para audio digital va de 8 a 16 bits. La capacidad de rango dinámico de estas representaciones va entonces de los 48 a 96 dB, respectivamente. Para poner estos rangos en perspectiva, si 0 dB representa el nivel de presión de sonido audible más débil, entonces 25 dB es el nivel mínimo de ruido en un estudio típico de grabación, 35 dB es el nivel de ruido dentro un hogar tranquilo, y 120 dB es el nivel más fuerte antes de sentir molestia. En términos de percepción de audio, 1 dB es el cambio mínimo audible en el nivel de presión¹ de sonido bajo las mejores condiciones, y duplicando el nivel de presión de sonido se aumenta un paso perceptual en volumen[7].

Comparado a la mayoría de los tipos de información digital (excluyendo video digital), las tasas asociadas con audio digital sin compresión son sustanciales. Por ejemplo, la información de audio en un CD (2 canales muestreados a 44.1 kHz a 16 bits por muestra) requiere de una tasa de aproximadamente 1.4 megabits por segundo (Mbps). Hay una clara necesidad por alguna forma de compresión para un almacenamiento y transmisión más eficiente de esta información.

Las diferentes técnicas de compresión de audio difieren en las compensaciones entre la complejidad del codificador y decodificador, la calidad de audio comprimido y la cantidad de compresión de información. Las técnicas utilizadas van desde ley- μ , un algoritmo de mediana calidad de audio de baja complejidad y compresión que se utiliza en DPCM y ADPCM, hasta MPEG/audio, el cual es un algoritmo de alta complejidad, alta compresión y alta calidad de audio y el tema central de este capítulo. Estas técnicas aplican a las señales de audio en general y no están específicamente afinadas para señales de voz. Este documento no cubre algoritmos de compresión diseñados específicamente para señales de voz. Estos algoritmos están generalmente basados en una modelación de la zona vocal y no trabajan bien para señales de audio de banda amplia (alta fidelidad). Los estándares LPC (Linear Predictive Coding) y CELP (Coded Excited Linear Prediction) caen en esta categoría de compresión de voz.

La ITU tiene una plétora de técnicas de compresión de voz y estándares de codificación, como se muestra en la tabla 2.1 reflejando las diferentes necesidades para la eficiente transmisión de voz sobre una variedad de redes de telecomunicación. Por ejemplo, ITU-T G.711 es una simple codificación PCM de 8 bits a 64 kbps con una calidad de audio telefónica y una no linealidad previa a la conversión A/D y siguiendo la D/A. ITU-T G.722 utiliza una codificación DPCM con una calidad de audio de radio AM monoaural para comprimir la información a 64 kbps. ITU-T G.728 emplea aún más tecnología de compresión, específicamente codificación predictiva lineal y cuantización vectorial, para comprimir la voz telefónica a 16 kbps. Y ahora la G.723 es capaz de obtener una calidad razonable de voz a tasas tan bajas como 5.3 kbps.

¹ Nivel de presión es la variación dinámica de la presión estática del aire y es medida en fuerza por unidad de área, representada en una escala de amplitud logarítmica.

ITU-T Designation	Bandwidth (Hz)	Sampling Rate (KHz)	Bit Rate (Kbits/s)
G.711	200 a 3200	8	64
G.722	50 a 7000	16	64
G.721	200 a 3200	8	32
G.728	200 a 3200	8	16
G.723	200 a 3200	8	5.3 y 6.3

Tabla 2.1 Estándares de codificación monoaural de voz comúnmente usados[3]

La compresión de audio de acuerdo a la ley \square es una técnica de compresión de audio básica especificada por la recomendación de la ITU G.711. La transformación es esencialmente logarítmica natural y permite códigos de salida de 8 bits por muestra para cubrir un rango dinámico equivalente a 14 bits de valores cuantizados linealmente. Esta transformación ofrece una razón de compresión de 8 a 1. A diferencia de la cuantización lineal, el espaciamiento de pasos logarítmicos representa muestras de audio de baja amplitud con mayor exactitud que los valores de amplitud más alta. Esto es que la relación señal a ruido de la salida transformada es más uniforme sobre el rango de amplitudes de la señal de entrada.

ADPCM (Adaptive Differential Pulse Code Modulation) toma ventaja del hecho de que las muestras de audio vecinas son generalmente similares unas con otras. En lugar de representar cada muestra de audio independientemente como en PCM, un codificador ADPCM calcula la diferencia entre cada muestra de audio y su valor predicho, y entrega el valor PCM del diferencial.

ISO tiene relativamente pocos estándares de compresión de audio comparado con la ITU. ISO MPEG Audio oficialmente conocido como ISO 11172-3 es básicamente para dos canales estéreo. ISO MPEG-2 Audio oficialmente conocido como ISO 13818-3 extiende su capacidad a cinco canales de sonido envolvente con la opción de un sexto canal de baja frecuencia. MPEG-2 viene en dos opciones, uno que es compatible con MPEG-1 y otro que no.

El audio de banda amplia tiene un ancho de banda considerablemente más grande que la voz telefónica. Esto ocasiona que las tasas de información sin compresión sean también mucho más grandes como se muestra en la tabla 2.2. El audio sin compresión típicamente comprende 16 bits por muestra y dos canales estéreo para una tasa de bits total de 32 veces la tasa de muestreo. Sin embargo los dispositivos de almacenamiento óptico y magnético son susceptibles a errores y requieren de 100 a 200 % de bits extra para redundancia correctiva de errores. Por lo tanto, el bitrate total es de 2 a 3 veces el del audio.

Medio de Almacenamiento	Ancho de Banda (KHz)	Tasa de Muestreo (KHz)	Bitrate de Audio (Kbps/s)	Codificación
CD	20	44.1	1411.2	PCM
DAT	16	32.0	1024.0	PCM
ProDAT	16	44.1	1411.2	PCM
DAT	16	48	1536.0	PCM
DCC	20	44.1	384	PASC
MD	22	44.1	292	ATRAC

Tabla 2.2 Audio de Banda Amplia[8]

Los métodos explicados aquí explotan las propiedades del oído humano mucho más allá que los investigadores en codificación de video han podido hacer para utilizar las características de la visión humana.

2.2 Fundamentos de compresión de audio

La mayoría de la codificación de banda amplia y MPEG en particular, explota altamente los efectos subjetivos de enmascaramiento del oído humano para hacer cualquier ruido de codificación tan inaudible como sea posible. Más específicamente, si un tono de cierta frecuencia y amplitud se presenta, entonces otros tonos o ruido de similar frecuencia, pero con mucho menos amplitud, no puede ser escuchado por el oído humano. Entonces el tono más fuerte enmascara al más suave, y no hay necesidad de transmitir el tono más suave.

El nivel de amplitud máximo no perceptible del tono más suave es llamado *Masking Threshold*. Este depende de la frecuencia, amplitud y tiempo del tono más fuerte (*masker*), y es usualmente representado como una función de la frecuencia de ruido más suave (*maskee*) como se muestra en la figura 2.2.

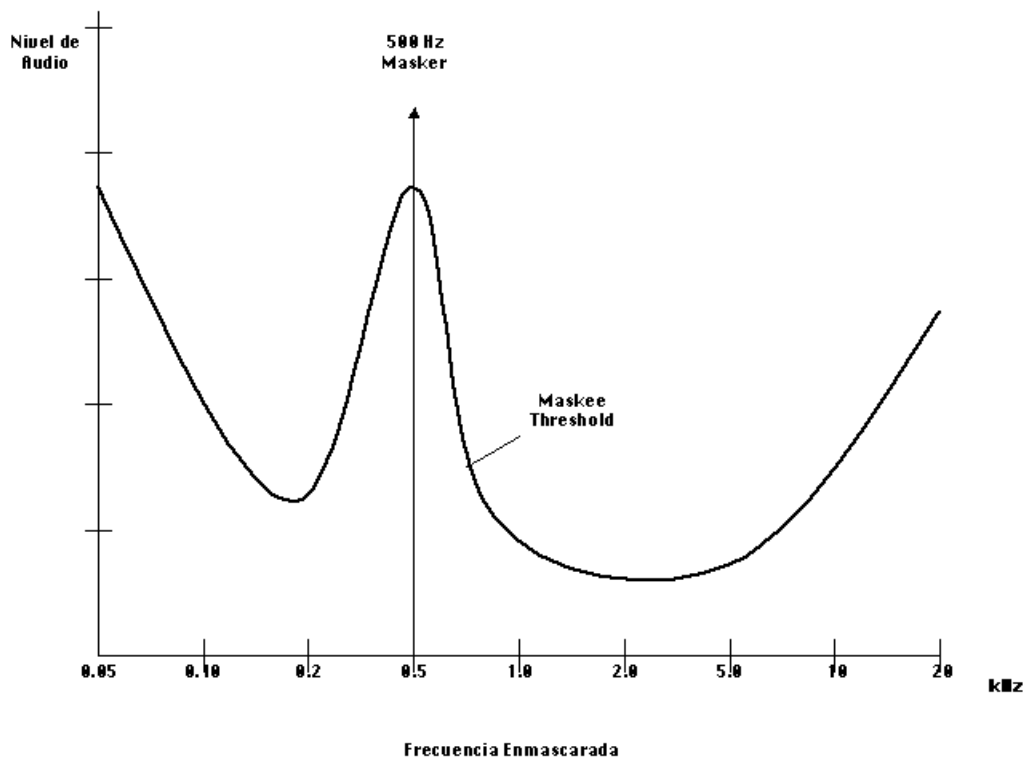


Figura 2.2 Enmascarado de Frecuencias

Además de la frecuencia de enmascarado descrita anteriormente, el oído humano es también afectado por el enmascarado temporal. Esto es, que un tono fuerte de duración finita cubrirá un tono más suave que lo sigue o que lo precede muy en cerca en el tiempo. Por lo tanto en la figura 2.2, si el tono *masker* o tono enmascarante desaparece repentinamente, el *Masking Threshold* no desaparece simultáneamente. En su lugar, desaparece en un periodo de hasta 200 ms, dependiendo de la frecuencia y amplitud del *masker*.

Si el *masker* decrece en amplitud, entonces generalmente el *Masking Threshold* también decrece. Sin embargo, hay un límite menor por debajo del cual el *Masking Threshold* no caerá sin importar que tan pequeño sea el *masker*. Este límite, mostrado en la figura 2.3 es llamado el

absolute threshold. Un tono sencillo o ruido aún sin *masker* es inaudible si está por debajo del umbral absoluto.

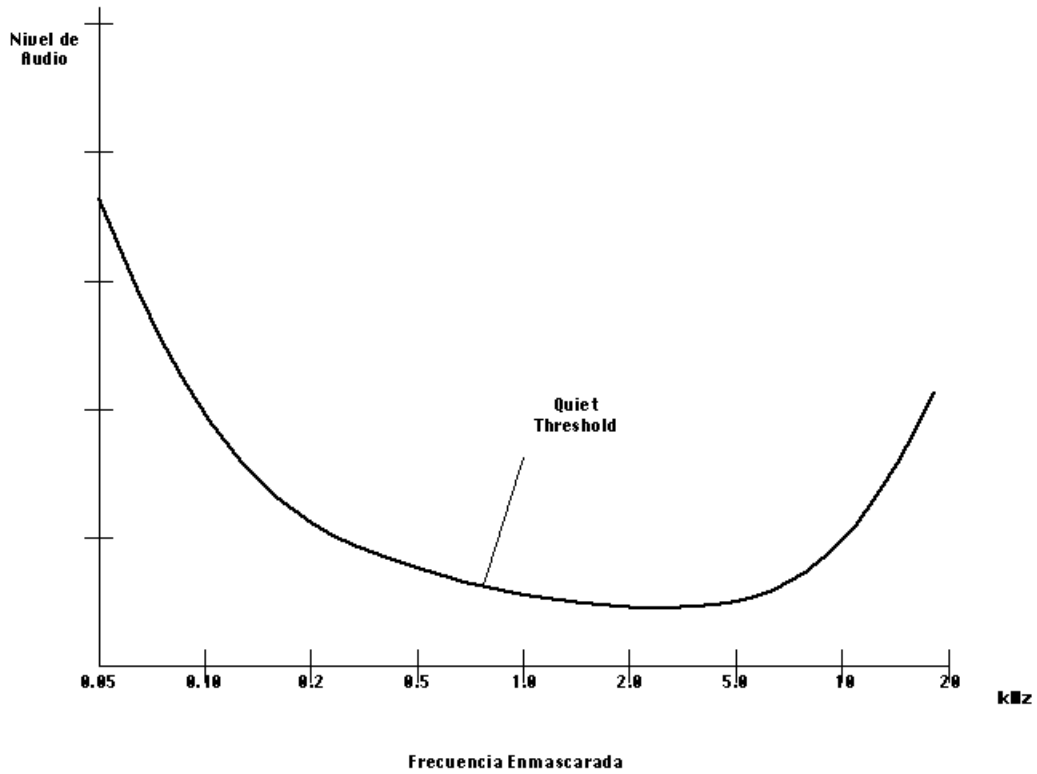


Figura 2.3 Umbral Absoluto

Con señales de audio normal, una multitud de *maskers* están usualmente presentes en una variedad de frecuencias. En este caso, tomando en cuenta la frecuencia de enmascarado, el *temporal masking* y el *absolute threshold* para todos los *maskers*, el *masking threshold* total puede ser derivado. Por lo tanto, el *masking threshold* es una función de la frecuencia variable en el tiempo que indica el máximo ruido inaudible en cada frecuencia en un momento dado, como se ilustra en la figura 2.4

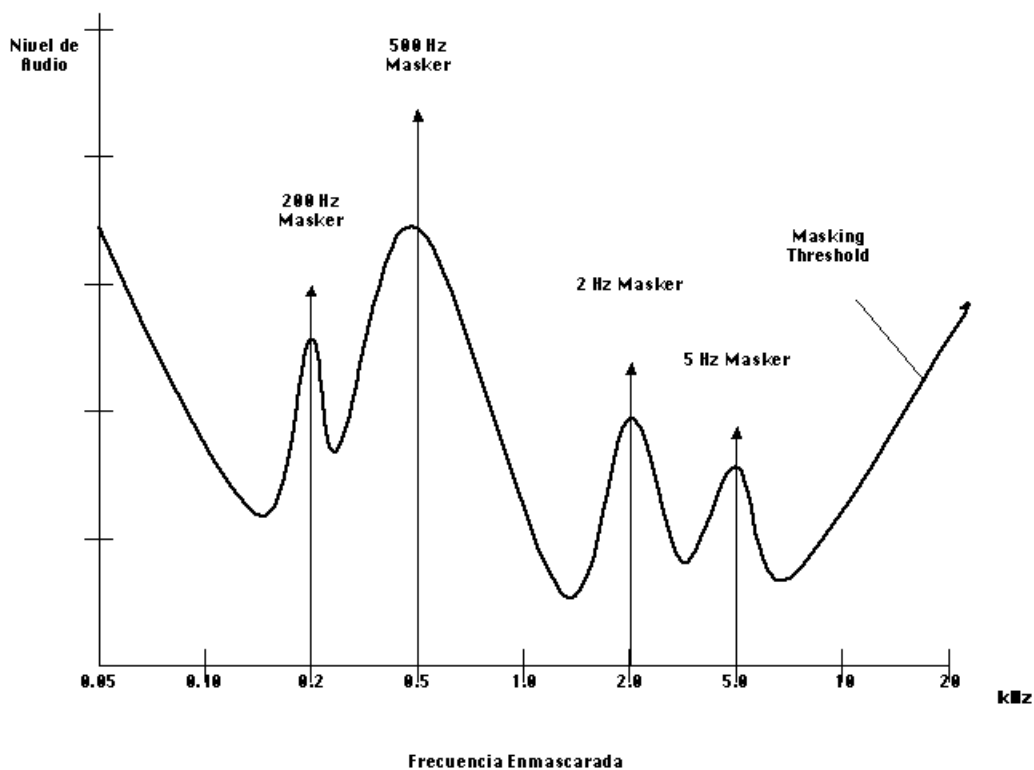


Figura 2.4 Masking Threshold

El *Masking Threshold* es usado por el codificador de audio para determinar el ruido de cuantización máximo permitido en cada frecuencia para minimizar la percepción del ruido.

Otra tecnología utilizada por los codificadores de audio de banda amplia es la llamada *Subband Coding* (codificación por sub-banda). Con este enfoque, mostrado en la figura 2.5, la señal digital PCM entrante es primero descompuesta en N señales digitales pasabanda por un banco paralelo de N filtros digitales pasabanda de análisis. Cada señal pasabanda es entonces submuestreada (o decimada) por un factor de N para que el número total de muestras en todas las sub-bandas sea igual que el PCM de entrada. Después de la decimación, cada una de las señales pasabanda se convierte, de hecho, en una señal de banda base de baja frecuencia que puede ser codificada y comprimida usando cualquier algoritmo de compresión de banda base.

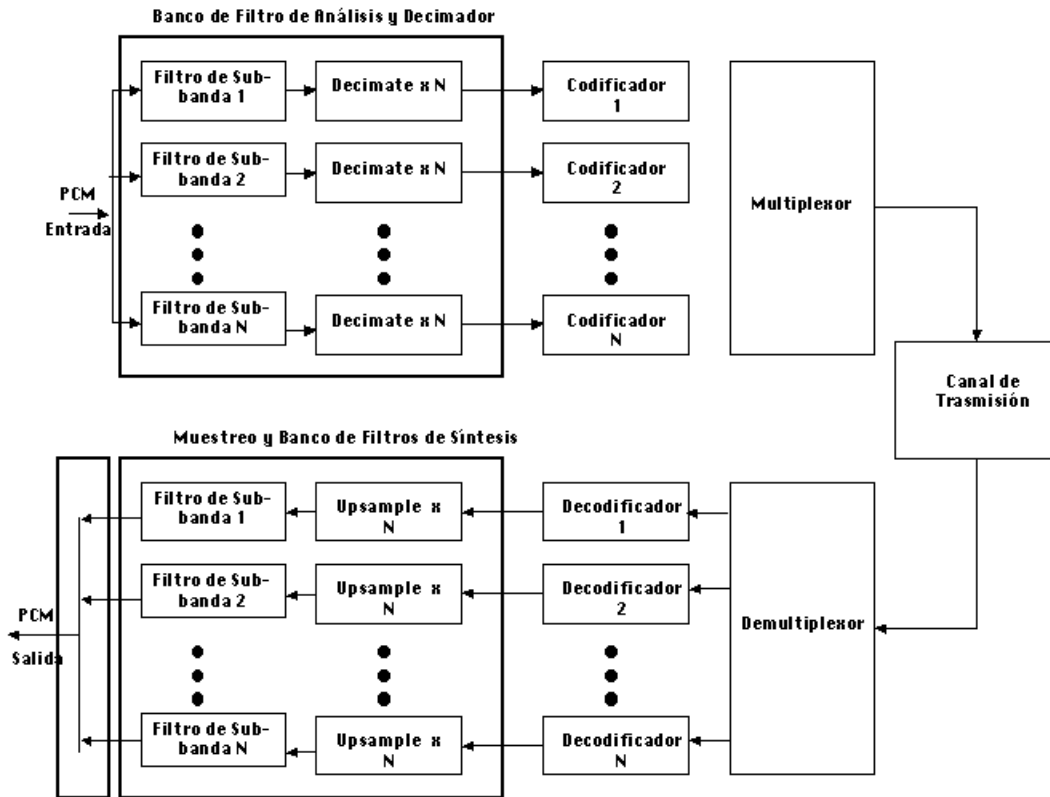


Figura 2.5 Codificación por Sub-banda

La reconstrucción de la señal original es realizada insertando primero $N-1$ muestras cero entre cada muestra de la señales sub-banda decimadas, lo cual incrementa la tasa de muestreo de cada señal pasabanda a su valor original. Las señales pasabanda muestreadas (Upsampled) son entonces alimentadas a sus respectivos filtros pasabanda de síntesis y finalmente sumadas para obtener la señal de salida PCM con el ancho de banda completo.

Dado que en la práctica, los filtros pueden solamente aproximar la característica ideal de pasabanda, cada señal submuestreada contendrá algunos componentes de traslape espectral (aliasing) que normalmente evitarían una reconstrucción libre de distorsión de la señal PCM original. Esto implica en teoría al menos, que usando filtros pasabanda especialmente diseñados, el aliasing puede ser cancelado cuando la señales son sumadas a la salida. Un conjunto de tales filtros es llamado Quadrature Mirror Filters (QMFs). Es posible diseñar un particular banco de filtros QMF eficiente a través del uso de redes polifase (Polyphase Networks). Otro conjunto de tales filtros son los llamados LOTs (Lapped Orthogonal Transforms) o ELTs (Extended Lapped Transforms).

Aún con QMFs y ELTs, ciertas aproximaciones y compromisos son usualmente necesarios, con el resultado de que el desempeño de los filtros pasabanda no es nunca completamente ideal. también, con una codificación con compresión de pérdida, las señales pasabanda son raramente recuperadas con completa exactitud en el decodificador, lo cual contribuye a una operación no ideal.

2.2.1 MPEG-1

En la primera fase de MPEG, un estándar de compresión para audio estéreo de dos canales fue desarrollado para dar una calidad cercana al CD (Compact Disk) a una tasa de 256 kbps. MPEG-1 es oficialmente conocido como ISO/IEC 11172-3. En principio, el estándar no especifica el codificador, permitiendo una evolución conforme la tecnología progresa y los costos disminuyen. MPEG-1 solo especifica el decodificador, y aún aquí la especificación ignora el post procesamiento y recuperación por errores.

El algoritmo tiene tres posibles capas de operación (layers) nombradas I, II, y III. En principio el codificador escoge la capa dependiendo de cuanta calidad se necesita o equivalentemente cuanta compresión se desea. Sin embargo, en algunas aplicaciones, los decodificadores solo implementan una capa lo que restringe el número de opciones[3].

2.3 Codificación y compresión de audio MPEG 1 Capa III

El algoritmo de codificación MPEG 1 Capa III es un sistema de compresión de audio que permite almacenar sonido de gran calidad con un índice de compresión de aproximadamente 1:10 frente al archivo original, sin pérdida de calidad apreciable. Es decir, se consigue una calidad próxima a la de un CD con una transferencia de sólo 128 Kbits por segundo de ancho de banda.

Este algoritmo de codificación perceptual, MPEG 1 Capa III, fue desarrollado por la sociedad MPEG (Moving Picture Expert Group) en conjunto con el Instituto Tecnológico Franunhofer, que se ha estandarizado como una norma ISO-MPEG Audio Capa III, lo cual supone un importante avance respecto a los algoritmos anteriores (Capa 1 y 2).

El impulso definitivo de Capa III ha venido dado gracias a la estandarización ISO, que lo ha hecho introducirse en los entornos industriales y al aumento en la potencia de los ordenadores domésticos, pues hace aproximadamente 10 años (cuando se desarrolló este algoritmo) los ordenadores personales no estaban a la altura y no podían manejar este formato con garantías, tal y como sucede actualmente. El sistema de codificación que utiliza MP3 es un algoritmo con pérdida, es decir, el sonido original y el que obtenemos tras ser comprimido no son idénticos. Sin embargo, las pérdidas corresponden a frecuencias que el oído humano no puede captar (nuestros oídos sólo trabajan con frecuencias que van desde los 20 Hz hasta los 20 KHz más o menos), por lo que la pérdida de calidad del MP3 no es percibida por las personas. Por otra parte, cuando dos señales de frecuencia similar se superponen, sólo podemos escuchar la que posee el mayor volumen, tal como se mencionó anteriormente es un efecto que se conoce como enmascaramiento.

El MP3 toma ventaja de todas estas deficiencias del oído humano, de forma que elimina toda la información que no necesitamos. Para ello utiliza el sistema de Codificación de Sub-Bandas, proceso por el cual la señal se descompone en sub-bandas a través de un banco de filtros o cualquier otro método semejante. Las sub-bandas obtenidas, se comparan a continuación con la señal original mediante un modelo psicoacústico que determina cuáles son las bandas importantes y cuáles se pueden eliminar. Dependiendo de la calidad que deseemos obtener, se eliminarán más o menos bandas. Para finalizar el proceso, se cuantifican y codifican las sub-bandas que han pasado la selección, y el resultado final se comprime utilizando un algoritmo estándar tipo Huffman o LZW (Lempel Ziv Welch). El proceso de codificación es mucho más complicado que el de decodificación, por ello se tarda mucho más en codificar un flujo que en reproducirlo. En la figura 2.6 se puede observar un diagrama general de ambos procesos.

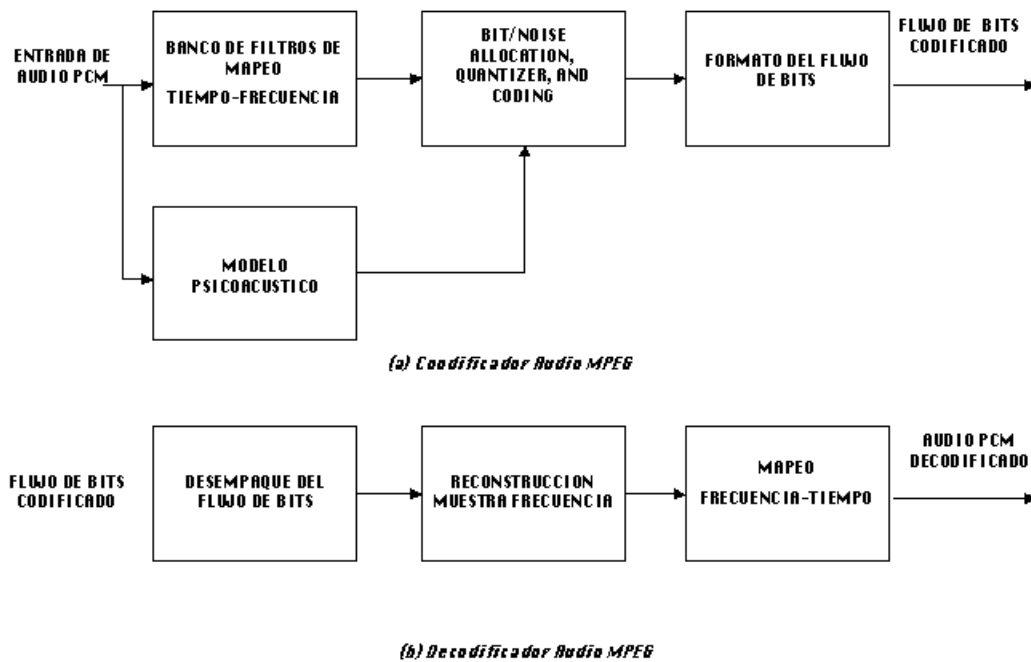


Figura 2.6 Proceso de Codificación y Decodificación MPEG

Como ya fue citado anteriormente, MP3 utiliza para la codificación un algoritmo con pérdida, que incluye en su definición tres capas (layers) que van incrementando la eficacia y la complejidad de la compresión. Cada una de ellas mejora las características de la anterior para conseguir el mejor sonido en el menor espacio, es decir, una tasa de transferencia menor. La capa III que es la que nos interesa, utiliza técnicas más complejas y consigue relaciones de compresión de hasta 12:1 manteniendo una calidad próxima a la calidad CD, para lo que necesita un velocidad de transferencia para sonido estéreo de 112 a 128 kbits/s. La tabla 2.3 muestra las tres capas con su relaciones de compresión y velocidad de transferencia[6].

La capa III es considerablemente más compleja y utiliza mucho más funciones que las capas I y II. La primer mejora sobre la capa II es incrementar la resolución de frecuencia. Esto se logra procesando más allá cada salida de los filtros de análisis de sub-banda usando lo que llamamos MDCT (Modified Discrete Cosine Transform). Llamamos a las salidas MDCT sub-subbandas. El incrementar la resolución de sub-subbanda permite una mejor codificación perceptual, especialmente a bajas frecuencias donde el Masking Threshold puede tener una variación considerable dentro de una sub-banda. Esto también permite una cancelación de aliasing causado por los filtros subbanda de análisis polifase dentro de una sub-banda.

Capa	Compresión	Transferencia
1	4:1	384 kbits/s
2	6:1 a 8:1	256 a 192 kbits/s
3	10:1 a 12:1	128 a 112 kbits/s

Tabla 2.3 Capas MPEG Audio

Todas las capas tienen el mismo esquema básico de codificación denominado “modelación perceptual del ruido” o también codificación perceptual en sub-bandas (SBC), utilizando los modelos psico-acústicos para realizar la codificación. Se han realizado multitud de estudios de percepción acústica (transformación de los estímulos sonoros en representaciones comprensibles por el cerebro, es decir la transformación de las ondas sonoras que llegan al tímpano en palabras o frases entendibles por nosotros) descubriendo que hay una serie de efectos que pueden ayudarnos a la codificación del sonido con el objetivo de reducir todo lo posible la cantidad de información inútil o redundante[5]. Algunos de ellos son:

- El límite mínimo y máximo de audición. Ningún oído humano es capaz de oír por debajo de los 20Hz y por arriba de los 20KHz aproximadamente, por tanto no será necesario codificar los sonidos que se encuentren fuera de este umbral
- Efecto de enmascarado. Cuando se escuchan sonidos de distintas intensidades los sonidos más débiles resultan enmascarados por los más intensos y como consecuencia no se oyen, por tanto puede ser obviado en la codificación.
- Redundancia de estéreo. Cuando se consideran los componentes tonales (componente de una señal de audio similar a una senoide) y no tonales (componente de una señal de audio similar al ruido) del sonido en los dos canales estéreo se observa que existen redundancias entre ellos, y, además, por debajo de una frecuencia determinada el oído humano no es capaz de percibir la direccionalidad del sonido, por lo cual por debajo de estas frecuencias es posible incluso codificar un solo canal, junto con información complementaria para restaurar la sensación espacial para el otro canal.
- En Codificación. El codificador procesa la señal de audio digital y produce la trama de datos empaquetada para su almacenamiento y/o transmisión. El algoritmo de codificación puede utilizar enmascaramiento, cuantización variable y escalamiento. Las muestras se introducen en el codificador y a continuación el mapeador crea una representación filtrada y submuestreada de la señal de entrada. El modelo psicoacústico crea una serie de datos que sirven para controlar la cuantización y codificación. El bloque de empaquetamiento de trama se encarga de agrupar como corresponde todos los datos, pudiendo añadir algunos datos adicionales
- En Decodificación. El decodificador debe procesar el *bitstream* para reconstruir la señal de audio digital. Los datos del bitstream son desempaquetados para recuperar las diversas partes de la información. El bloque de reconstrucción recompone la versión cuantizada de la serie de muestras mapeadas. El mapeador inverso transforma estas muestras de nuevo a PCM.

2.3.1 El modelo psicoacústico.

El modelo psicoacústico analiza la señal de audio y calcula la cantidad de enmascaramiento de ruido disponible como una función de la frecuencia. La habilidad de enmascaramiento de cierto componente de la señal depende de su posición de frecuencia y volumen. El codificador usa esta información para decidir como representar mejor la señal de audio de entrada con su número limitado de bits de código. El estándar de audio MPEG provee dos ejemplos de implementación del modelo psicoacústico. El modelo 1 es menos complejo que el modelo 2 y tiene más compromiso de simplificar los cálculos. Cualquiera de los modelos trabaja para cada una de las capas de compresión. Sin embargo, solo el modelo 2 incluye modificaciones específicas para acomodar la capa III.

Hay considerable libertad en la implementación de el modelo psicoacústico. La exactitud requerida de el modelo depende de la compresión y aplicación deseada. Para bajos niveles de aplicación, donde hay una provisión generosa de bits de código, una aproximación completa del modelo psicoacústico deber ser adecuado para uso del consumidor. En este caso, el proceso de colocación de bit puede iterativamente asignar bits a la sub-banda con la menor relación señal a ruido. Para el almacenamiento de música, el modelo psicoacústico puede ser mucho más riguroso[6].

Capítulo 3

Modo de Transferencia Asíncrono (ATM)

3.1 Introducción.

La tecnología llamada Asynchronous Transfer Mode (Modo de Transferencia Asíncrono) es el corazón de los servicios que ofrecerían las nuevas redes digitales de servicios integrados de Banda Ancha (B-ISDN). Para muchos ya no hay cuestionamientos; el tráfico de Internet, con su voluminoso y tumultuoso crecimiento, impone a los operadores de redes públicas y privadas una voraz demanda de anchos de banda mayores y flexibles con soluciones robustas. La versatilidad de la conmutación de paquetes de longitud fija, denominadas celdas ATM, parece ser la tecnología más calificada para soportar esta demanda.

Algunos críticos establecen una analogía de la tecnología ATM con la red digital de servicios integrados o ISDN. Al respecto se escuchan respuestas de expertos que desautorizan esta comparación aduciendo que la ISDN es una gran tecnología que llegó en una época equivocada[4].

Ahora el mercado está cambiando, las redes convergentes están encontrando una gran cantidad de aplicaciones. De toda forma la tecnología ATM se proyecta para diferentes necesidades, a pesar de su estrecha relación con ISDN, en términos de volúmenes de datos, flexibilidad de conmutación y facilidades para el operador.

Los conmutadores ATM aseguran que el tráfico de grandes volúmenes es flexiblemente conmutado al destino correcto. Estas necesidades se ajustan muy bien para los proveedores de servicios públicos de salud, con requerimientos de videoconferencias médicas, redes financieras interconectadas con los entes de intermediación y validación, o con las exigencias que pronto serán familiares como vídeo y audio en demanda para nuestros hogares con alta definición de imágenes y calidad de sonido de un CD, etc.

Para el operador, con la flexibilidad del ATM, una llamada telefónica con tráfico de voz será tarifado a una tasa diferente a la que estaría dispuesto a pagar un cirujano asistiendo en tiempo real a una operación al otro lado del mundo. Eso es una de las fortalezas de ATM, se paga solamente por la carga de celdas que es efectivamente transportada y conmutada para uno. Además la demanda por acceso a Internet ha tomado a la industria de telecomunicaciones como una tormenta. Hoy día los accesos conmutados a Internet están creando "Cuellos de Botella" en la infraestructura. Para solucionar este problema los fabricantes no solo han desarrollado sistemas de acceso sino aplicaciones para soluciones de fin a fin con conmutadores ATM, con solventes sistemas de administración de la red (Network Management).

En varios aspectos, ATM es el resultado de una pregunta: ¿Cómo se puede transportar un universo diferente de servicios de voz, vídeo por un lado y datos por otro, de manera eficiente usando una simple tecnología de conmutación y multiplexaje?.

ATM contesta esta pregunta combinando la simplicidad de la multicanalización por división en el tiempo (Time Division Multiplex TDM) encontrado en la conmutación de circuitos, con la eficiencia de las redes de conmutación de paquetes con multiplexaje estadístico. Por eso es que algunos hacen referencia de perspectivas de conmutación de circuitos mientras que otros lo hacen a redes de paquetes orientados a conexión[9].

3.2 Tecnología ATM

En las redes ATM toda la información es puesta en un formato de celdas de longitud fija con 48 bytes de datos y 5 bytes de encabezado. El tamaño de celda fija asegura que la información crítica en cuanto a tiempo, tal como la voz y el video, no sea adversamente afectada por largos paquetes o tramas. El encabezado está organizado para una eficiente conmutación en implementaciones de *hardware* de alta velocidad y transporta el tipo de información, identificadores de circuito virtual, y detección de errores de encabezado.

ATM es orientado a conexión. Organizando diferentes flujos de tráfico en transmisiones separadas, permite al usuario especificar los recursos requeridos y permite a la red asignarlos bajo estas necesidades. Multicanalizando flujos múltiples de tráfico en cada enlace físico, combinado con la habilidad de enviar estos flujos a diferentes destinos, habilita el ahorro de costos a través de la reducción en el número de interfases e instalaciones requeridas para la construcción de una red.

3.2.1 Analogía con el sistema tradicional telefónico.

Cuando se hace una llamada ordinaria, las señales analógicas de voz viajan a la central donde son convertidas a su forma digital. Esta digitalización se realiza mediante alguna forma de Modulación de código de pulso (PCM), de manera que las señales pueden ser conmutadas y ruteadas en la red telefónica. Las señales digitales son reconvertidas a analógicas antes de ser presentadas a su destino. El PCM es transportado en una ranura de tiempo particular de una trama a través de una conexión física particular por la duración de la llamada. La asignación de estos espacios en el tiempo permite que múltiples llamadas sea colocadas o multicanalizadas en una trama sencilla. A esto se le llama TDM (Time Division Multiplexing).

La red TDM tradicional esta construida por circuitos T1/E1 canalizados con 24/30 canales de 64Kbps. La figura 3.1 muestra N tramas, donde el orden de transmisión es N, 1,2,3, etc. Cada espacio es dedicado al usuario durante la llamada, aún cuando no tiene nada que transmitir. Esto crea una utilización del ancho de banda ineficiente.

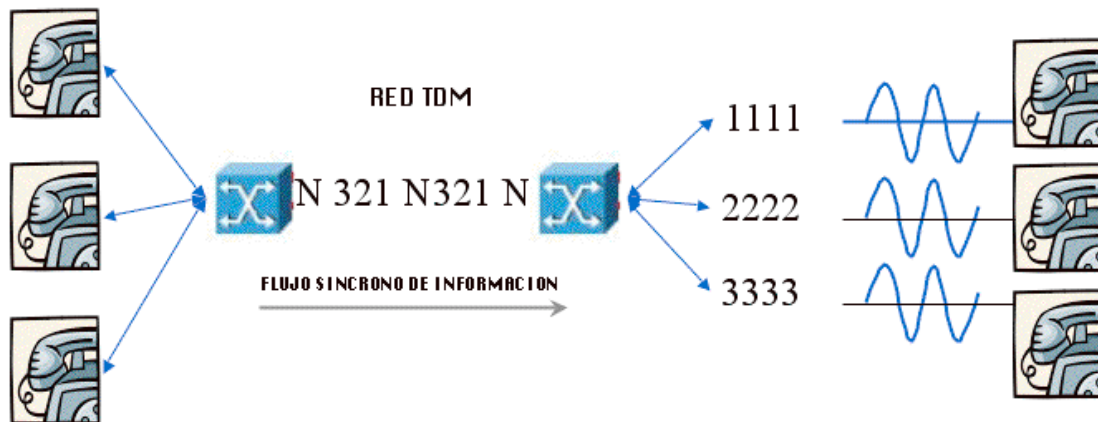


Figura 3.1 Transmisión básica TDM

TDM está basado en tramas que son enviadas a través de un circuito físico para transportar una llamada. Se ha hecho que la red telefónica cumpla con ciertos niveles de confiabilidad y disponibilidad forzando al uso de circuitos redundantes o de respaldo para satisfacer estos

requerimientos. Durante periodos en que el equipo de conmutación TDM se encuentra a su máximo en el procesamiento de llamadas, si un circuito falla, tal vez no haya otro disponible como respaldo para manejar la llamada en progreso. Por lo tanto todas estas llamadas se perderán, resultando en la inconformidad del usuario.

La red telefónica tradicional usa señalización en banda para control de llamadas, lo que significa que parte del ancho de banda usado para transportar las muestras de voz es usado para información de control. La señalización fuera de banda permite que ATM establezca rutas para dichas muestras usando circuitos especiales sólo para señalización que manejan un gran número de llamadas.

La ruta actual para cada conjunto de muestras de voz, no es establecida hasta que es necesario. La ruta, que en realidad es virtual, consiste de un flujo de paquetes que es construida cuando hay información de voz a transmitir. Esto reduce la cantidad de ancho de banda requerido para darle servicio a una llamada. Las celdas ATM no necesitan ir sobre la misma ruta física para llegar a su destino. Por lo tanto el equipo de conmutación puede manejar llamadas de manera más eficiente.

La parte operacional de ATM es el enrutamiento y manejo de los paquetes que proveen los servicios de transporte. Usando un concepto de paquetes, permite a las redes implementar rutas redundantes de alta confiabilidad para la información ofreciendo al cliente la integridad de información que necesita. En el caso de una falla, los paquetes pueden ser simplemente re-enrutados a su destino usando un camino diferente sin tener que desechar las llamadas en progreso.

ATM presenta los paquetes que manejan el control de información o muestras de voz a la red cuando hay ancho de banda disponible para manejarlos. Por esto, la transferencia es asíncrona relativamente desde el punto de vista de la fuente original. El ordenamiento de paquetes es garantizado por ATM.

Otra ventaja de ATM es su inherente formato de transmisión digital. Las celdas de longitud fija pueden ser procesadas extremadamente rápido y enviadas sobre cualquier ruta disponible usando fibra óptica como medio físico con velocidades de hasta 622Mbps.

Además de voz, ATM puede transportar audio, datos y video con igual confiabilidad. Cuando los paquetes pasan a través de la red, existe la oportunidad de realizar procesamiento adicional en el contenido de los paquetes, si se desea, así como de implementar varios tipos de procesamiento digital de señales, incluyendo compresión, supresión de silencio, encriptado y conversión de formatos.

3.3 Modelo de referencia B-ISDN

Para comprender la terminología en ATM, es necesario entender el modelo de referencia B-ISDN. Justo como el modelo ISO de siete capas definidas para el software de red. Este modelo define capas para la red ATM (Figura 3.2).

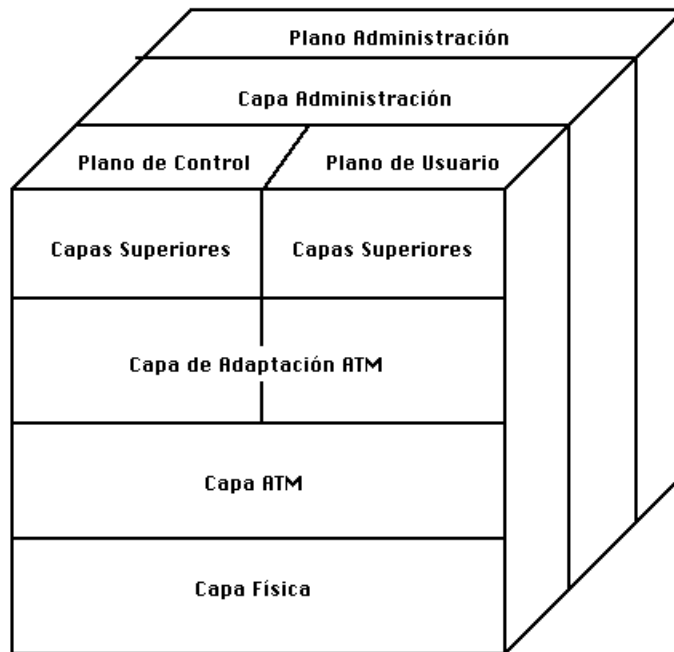


Figura 3.2 Modelo de Referencia B-ISDN

En la parte inferior está la capa física, la cual tiene que ver con cómo los bits son codificados en el cable o fibra. Esta capa presentará una interfase uniforme a la capa ATM superior, pero desempeña tareas específicas al medio físico usado. Enseguida está la capa ATM, la cual tiene que ver con el formato, envío, y conmutación de celdas. Entonces viene la capa de adaptación ATM (AAL) que mapea las unidades de información de protocolo (PDUs, o comúnmente llamados “paquetes”) en celdas de manera apropiada para el tráfico. La AAL consta de tres partes o sub-capas. El nivel más bajo es la sub-capa de Segmentación y Reensamble (SAR) que es responsable de la segmentación o recibir PDUs de las capas superiores, partiéndolas en celdas ATM y mandarlas a la capa ATM. También desempeña reensamble, esto es recibir celdas de la capa ATM y reconstruir las PDUs y mandarlas a las capas superiores.

La sub-capa media de la AAL es llamada Sub-capa de Convergencia de Parte Común (CPCS, Common Part Convergence Sublayer) y la sub-capa superior se llama Sub-capa de Convergencia Específica de Servicio (SSCS, Service Specific Convergence Sublayer). El término “sub-capa de convergencia” significa que los flujos de información de diferentes aplicaciones pueden ser concentradas aquí y enviadas hacia la red, por eso convergen en este sitio. La SSCS puede ser considerada como una “calza” para ajustar los protocolos de las capas superiores a la AAL, y la CPCS desempeña funciones comunes a todos los protocolos.

Finalmente en la parte de arriba se encuentran las capas superiores de la pila de protocolos y así sucesivamente.

Sin embargo, hay también planos, los cuales son las rebanadas verticales del pastel de la capa de protocolo. El plano de usuario es con lo que nosotros estamos acostumbrados a tratar: maneja la transferencia de datos hacia y desde los programas de aplicación. El plano de control es

nuevo, sin embargo. Maneja establecimiento y liberación de llamadas así como otras funciones de control orientadas a conexión.

Detrás del plano-U y del plano-C está el plano M. El plano de Administración, que maneja toda esta complejidad. De hecho, mucha de la complejidad aquí tiene más sustancia que el modelo de siete capas de ISO. Su principal beneficio es para discutir asuntos de diseño; la separación conceptual de control y flujos de información de usuario, por mencionar alguno[9].

3.4 Encabezado de la celda ATM (Cell Header).

Diferentes encabezados han sido definidos para su uso en la UNI (User Network Interface) y en la NNI (Network-Network Interface). La UNI es el punto interfase entre usuarios finales ATM y un *switch* ATM público o privado, o entre un *switch* privado y una red ATM pública. La NNI es la interfase entre dos nodos (*switches*) en la misma red ATM. La figura 3.3 muestra el encabezado para la UNI. A continuación se describe las funciones de los diferentes campos.

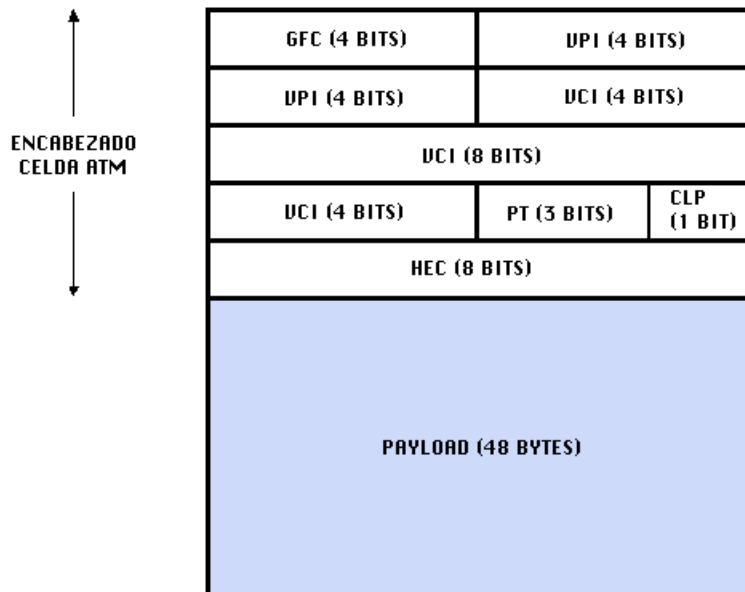


Figura 3.3 Formato de la celda ATM y su encabezado

Control de flujo genérico: el campo GFC (Generic Flow Control) es de 4 bits de largo y su intención es proveer control de flujo y acceso de medio compartido a a diferentes terminales en la UNI. Actualmente está indefinido y se pone en cero. El campo GFC solo tiene significado en la UNI y no es portado de fin a fin a través de la red. Los encabezados de UNI y NNI difieren en que el campo GFC no aparece en el encabezado de una celda NNI; en su lugar el campo VPI es aumentado a 12 bits.

Identificador de ruta virtual (Virtual Path Identifier): El campo VPI es de 8 bits, por lo tanto permite la definición de hasta $2^8=256$ rutas virtuales en un enlace UNI. Cada ruta virtual consiste de un conjunto de canales virtuales que son conmutados como una unidad sobre la secuencia de los nodos de la red que corresponden a la ruta.

Identificador de Canal virtual (Virtual Channel Identifier): El Campo VCI es de 16 bits, permite la definición de hasta $2^{16}=65,536$ canales virtuales por ruta virtual. El campo VPI/VCI es el identificador local para una conexión dada en un enlace dado, y el valor del campo cambia en cada switch.

Tipo de carga útil (Payload Type): el campo de tipo de carga útil de 3 bits permite 8 tipos de carga ATM. El bit más significativo entre celdas de información ($b_3=0$) y operaciones, administración y celdas de mantenimiento (OAM) ($b_3=1$)
Para celdas de información ($b_3=0$) el segundo bit sirve como indicador de congestión EFCI (Explicit forward congestion indication) que es puesto por los switches para indicar congestión y es usado por el mecanismo de control de congestión para la tasa de bits disponible ABR (available bit rate).
Para celdas de información, el bit menos significativo (b_1) es portado transparentemente a través de la red. $b_1=1$ es usado por la capa de adaptación 5 (AAL5) para señalar que la celda transporta el final del SDU (service data units).

Prioridad de pérdida de celdas (CLP, Cell Loss Priority): el campo CLP establece 2 niveles de prioridades para celdas ATM. Una celda que tiene $CLP=0$ debe ser tratada con una prioridad más alta que una celda con $CLP=1$ durante periodos de congestión, En particular, las celdas con $CLP=1$ deberan ser desechadas antes que las celdas con $CLP=0$. El bit CLP puede ser puesto por terminales para indicar tráfico menos importante o puede ser puesto por la red para indicar flujos con calidad de servicio de baja prioridad o celdas que han violado el contrato de tráfico.

Control de error del encabezado (header error control): Un CRC de 8 bits, usando un polinomio generador, es calculado sobre los primeros 4 bytes del encabezado. Este código puede corregir todos los errores sencillos y detectar todos los errores dobles en el encabezado. El *checksum* provee protección contra entrega errónea de celdas por errores que pudieran ocurrir durante el tránsito. Dos modos están definidos. En el modo de detección, cuando hay celdas con checksum inconsistente son desechadas. En el modo de corrección, los errores sencillos de bit son corregidos. El modo de corrección es adecuado solo en medios donde los errores sencillos predominan sobre errores múltiples. El HEC necesita ser recalculado en cada switch, dado que el VPI/VCI cambia en cada salto[1].

3.5 Conmutadores.

Una parte central del sistema ATM es el conmutador. El conmutador es conceptualmente muy simple: tiene puertos de entrada, recurso de mapeo VCI, un *fabric* de conmutación y puertos de salida (figura 3.4).

Cuando una celda llega a un puerto de entrada, el conmutador mira en los campos VPI y VCI del encabezado de la celda y consulta una tabla para determinar el puerto de salida donde debería ir. Entonces reescribe el encabezado de la celda para que tenga el nuevo VCI para el siguiente salto y finalmente arregla que la celda sea ruteada a ese puerto de salida por el *fabric* del conmutador.

Cada puerto de entrada y salida pueden tener buffers o colas que pueden mantener un número de celdas. Qué tan largos deben ser estos buffers depende del tipo de tráfico esperado y es un asunto de gran controversia. Si son regulares, cargas predecibles son esperadas, entonces pequeños buffers de entrada de unas pocas docenas de celdas será suficiente. Si se espera un tráfico en ráfagas –y el tráfico de Internet tiende extremadamente a ser de esta manera- entonces son necesarios buffers más largos para permitir una congestión momentánea y proveer tiempo para retroalimentación de control de tasa para llegar al remitente.

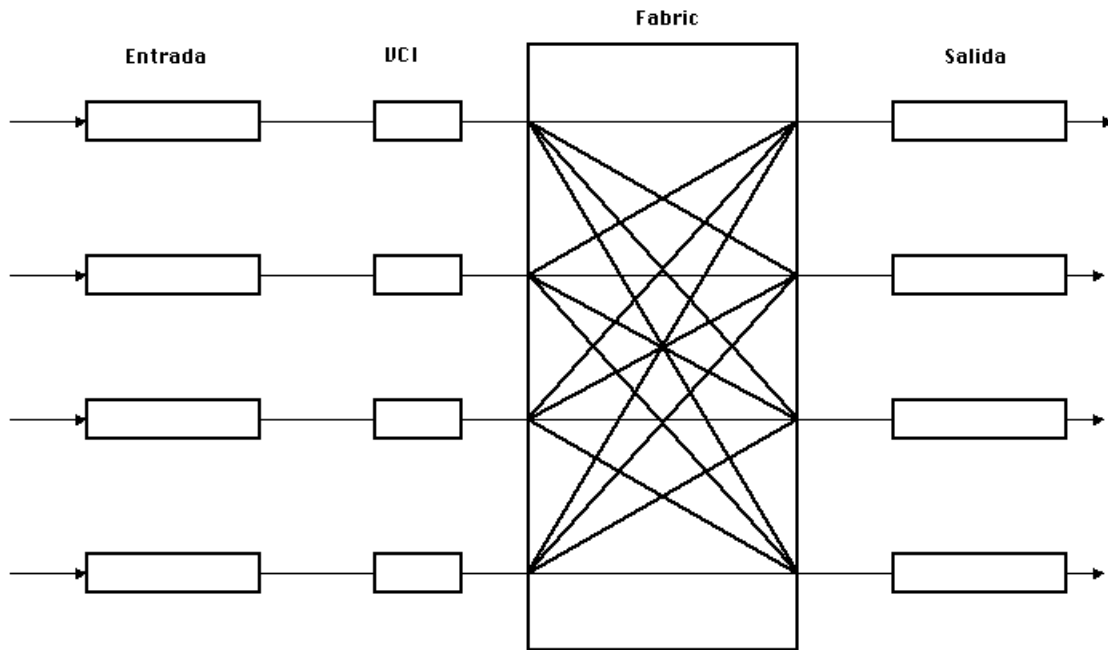


Figura 3.4 Conmutador genérico típico

Muchos esquemas han sido pensados para tratar estos (y otros) problemas. Algunos conmutadores tienen ambos buffers de entrada y salida, permitiendo un buen desempeño con un poco de resistencia al tráfico en ráfagas (bursty).

3.6 Circuitos virtuales.

El término circuito virtual es usado para designar una ruta a través de un conmutador o conmutadores hasta la máquina destino. El VCI (Virtual Circuit Identifier) es un número usado para representar el circuito. Es importante darse cuenta que este número no tiene relación con las direcciones de red de los sistemas involucrados, las direcciones de puerto en el conmutador, o alguna otra dirección.

Las rutas virtuales (Virtual Paths) son usadas para rutear grupos de circuitos virtuales entre conmutadores. Un conmutador ruteando celdas basado en una ruta virtual enviará todas al

puerto de destino designado para aquella ruta sin mirar en los VCIs, estos son decodificados más tarde en otro conmutador.

PVCs (Permanent Virtual Circuits) son ampliamente usados en redes ATM de área local. Son configurados a mano para conectar sistemas y por lo tanto son útiles pero incómodos. Es más, en redes “grandes” pueden rápidamente saturar las memorias de ruteo VCI del conmutador. Por lo tanto, los circuitos virtuales conmutados (SVCs, Switched Virtual Circuits) son preferidos viendo que cada conexión es establecida cuando es necesaria y dada de baja al terminar. Esto requiere software para administrar las conexiones[9].

3.7 Parámetros de Calidad de Servicio (QoS Parameters)

Un objetivo central de ATM es proveer garantías de calidad de servicio en la transferencia de flujos de celdas a través de una red. En ATM, la calidad de servicio proporcionada por una red es especificada en términos de los valores de varios parámetros a nivel de celda de fin a fin. Un total de seis parámetros de desempeño QoS han sido especificados.

Los siguientes tres parámetros están definidos en los estándares ATM. Estos parámetros son negociados al tiempo de iniciarse la conexión y son indicadores del desempeño intrínseco de una red.

Razón error en celdas (Cell Error Ratio): el CER de una conexión es la razón del número de celdas que son entregadas con uno o más errores de bit durante la transmisión al número total de celdas transmitidas.

Tasa de inserción errónea de celdas (Cell Misinsertion Rate): el CMR es el número promedio de celdas/segundo que son entregadas erróneamente a un destino de conexión dado (esto es, que se originó desde la fuente no correcta).

Razón de bloqueo de celdas con errores severos (Severely-Errored Cell Block Ratio): un evento de este tipo ocurre cuando más de M celdas son perdidas por error o mal entregadas en cierto bloque de N celdas recibido. Donde M y N son definidos por el proveedor de la red.

Los siguientes tres parámetros de QoS pueden ser negociados entre el usuario y la red al inicio de una conexión.

Razón de pérdida de celdas (Cell Loss Ratio): el CLR para una conexión es la razón de celdas perdidas al número total de celdas transmitidas.

Retraso en transferencia de celdas (Cell Transfer Delay): el CTD es el tiempo que transcurre desde el instante cuando una celda entra a la red en la UNI fuente, al instante que existe en la UNI destino. El CTD incluye retraso por propagación, por procesamiento, y retrasos de cola en los multiplexores y conmutadores.

Variación en retraso de celda (Cell Delay Variation): el CDV mide la variabilidad del retraso total encontrado por las celdas en una conexión.

3.8 Descriptores de tráfico.

La capacidad de una red para proveer ciertos niveles de calidad de servicio a una conexión depende de la manera en la cual la conexión produce celdas para transmisión. Esto es, a una tasa

suave constante o en una modalidad altamente explosiva. La capacidad también depende de la cantidad de recursos de la red. Por lo tanto, el contrato de conexión entre el usuario y la red debe especificar la manera en la cual la fuente producirá celdas.

Los siguientes parámetros de tráfico de la fuente han sido definidos para especificar este patrón de demanda para transmisión.

Tasa de celdas pico (Peak Cell Rate): el PCR especifica la tasa en celdas/segundo que una fuente no está permitida exceder.

Tasa sostenible de celdas (Sustainable Cell Rate): el SCR es la tasa de celdas promedio, en celdas/segundo producida por la fuente durante un largo intervalo de tiempo.

Tamaño máximo de ráfaga (Maximum Burst Size): el MBS, en un número de celdas, especifica el máximo de celdas consecutivas que podrían ser transmitidos por una fuente en la tasa de celdas pico (PCR).

Tasa mínima de celdas (Minimum Cell Rate): el MCR es la tasa de celdas promedio mínima, en celdas/segundo, que la fuente siempre esta permitida en enviar.

3.9 Categorías de servicio en ATM.

El establecimiento de conexiones ATM con propiedades de flujo de tráfico arbitrario y calidad de servicio QoS arbitraria es posible seleccionando valores para el descriptor de tráfico y los parámetros de QoS negociable. El foro ATM ha definido 5 categorías de servicio. Las dos primeras aplican a conexiones que son en tiempo real en el sentido de tener requerimientos exigentes de tiempo y retraso.

CBR. *Constant Bit rate.* Definido para tráfico con rigurosos requerimientos de tiempo, tales como voz, emulación de circuitos y ciertos tipo de video que requieren una transmisión constante de celdas para toda la duración de la conexión.

rt-VBR. *Real time variable bit rate.* Definido para tráfico variable como ciertos tipos de video con requerimientos rigurosos de tiempo.

nrt-VBR. *Non real time variable bit rate.* Definido para fuentes de ráfagas que no presentan rigurosos requerimientos de tiempo, como transferencia de datos.

ABR. *Available bit rate.* Esta categoría esta definida para fuentes que pueden adaptar dinámicamente la tasa a la cual transmiten celdas en respuesta a la retroalimentación de la red. Este servicio permite a las fuentes explotar el ancho de banda disponible en la red en cierto punto en el tiempo.

UBR. *Unspecified bit rate.* Esta categoría no provee ninguna garantía de calidad de servicio QoS. Este servicio es apropiado para aplicaciones no críticas que pueden tolerar o ajustar la pérdida de celdas.

3.10 Capas de Adaptación ATM (AAL)

Una aplicación que opera a través de una red ATM tiene la elección de las 5 categorías mencionadas anteriormente. Para la transferencia de uno o más bloques o flujo de información se

requiere de una conversión de estos bloques de información a celdas ATM. Uno de los propósitos de la capa de adaptación es proveer el mapeo de bloques a celdas. Las aplicaciones especifican sus requerimientos de calidad de servicio en términos de sus bloques y no en términos de celdas. Puede suceder que la capa de adaptación ATM no cumpla con los requerimientos de la aplicación. Por ejemplo, esta no provee por si misma un servicio de flujo confiable dado que la pérdida de celdas puede ocurrir. Diferentes aplicaciones requieren diferentes combinaciones de funciones. Por ejemplo, aplicaciones de emulación de circuito que la información sea transferida como si la conexión ATM fuera una línea de transmisión dedicada. Tal es el caso de aplicaciones en tiempo real como voz o ciertas aplicaciones de video. Otras como las de Frame Relay o IP tienen distintas características que hacen que sus requerimientos demanden otras funciones a la capa de adaptación.

Se ha tratado de categorizar las aplicaciones en un conjunto pequeño de clases y diseñar AALs para cumplir con los requerimientos de cada clase. Esto no se hace con total éxito y no existe una clara correspondencia entre clases de aplicaciones, AALs y categorías. Las capas de adaptación son básicamente las siguientes:

AAL1. Soporta servicios que requieren que la transferencia de información sea a una tasa constante y con exigencias de tiempo.

AAL2. Originalmente diseñada para aplicaciones que generan información a una tasa de bits que varía dinámicamente con el tiempo y que tiene requerimientos de tiempo de punto a punto como la voz o video con compresión.

AAL3/4. Originalmente fue definida como dos capas distintas. AAL3 fue diseñada para soportar transferencias orientadas a conexión, con datos VBR y sin requerimientos de tiempo. AAL4 fue similar pero para datos orientados a no conexión. Después estas capas fueron fusionadas pero AAL3/4 es ahora ampliamente cubierta por las capacidades de AAL5.

AAL5. Es la capa más implementada actualmente. Provee una alternativa eficiente a AAL3/4 y soporta modos de flujo y mensaje así como entrega asegurada y no asegurada.

Signaling AAL. La AAL de señalización ha sido estandarizada como la capa de adaptación en el plano de control. La SAAL provee transporte confiable para los mensajes de señalización que son intercambiados entre sistemas terminales y conmutadores para configurar circuitos virtuales.

3.11 Capa de Adaptación ATM 5 (AAL5)

AAL5 fue desarrollada como una respuesta a la complejidad de AAL3/4. Parecía que AAL3/4 había sido diseñada deliberadamente para hacer imposible implementaciones eficientes. AAL5 (conocida por un tiempo como la eficiente y simple capa de adaptación [SEAL]) fue diseñada para tener menos encabezado, ser capaz de una implementación eficiente y comportarse como las interfaces de comunicaciones de datos existentes. AAL5 provee una eficiente alternativa a AAL3/4. AAL5 incluye la capacidad de multicanalización de AAL3/4 para modos de flujo y mensaje, así como entrega asegurada y no asegurada.

La figura 3.5 muestra la operación de AAL5. Una PDU (Protocol Data Unit) de usuario es aceptada por la capa AAL y procesada por la sub-capas de convergencia de servicio específico (SSCS) si es necesario. La SSCS (Service Specific Convergence Sublayer) pasa entonces un bloque de datos a la CPCS (Common Part Convergence Sublayer) que añade 0 a 47 bytes de

padding y un trailer de 8 bytes para producir una CPCS PDU que sea múltiplo de 48 bytes. La máxima carga útil de una CPCS-PDU es de 65,535 bytes.

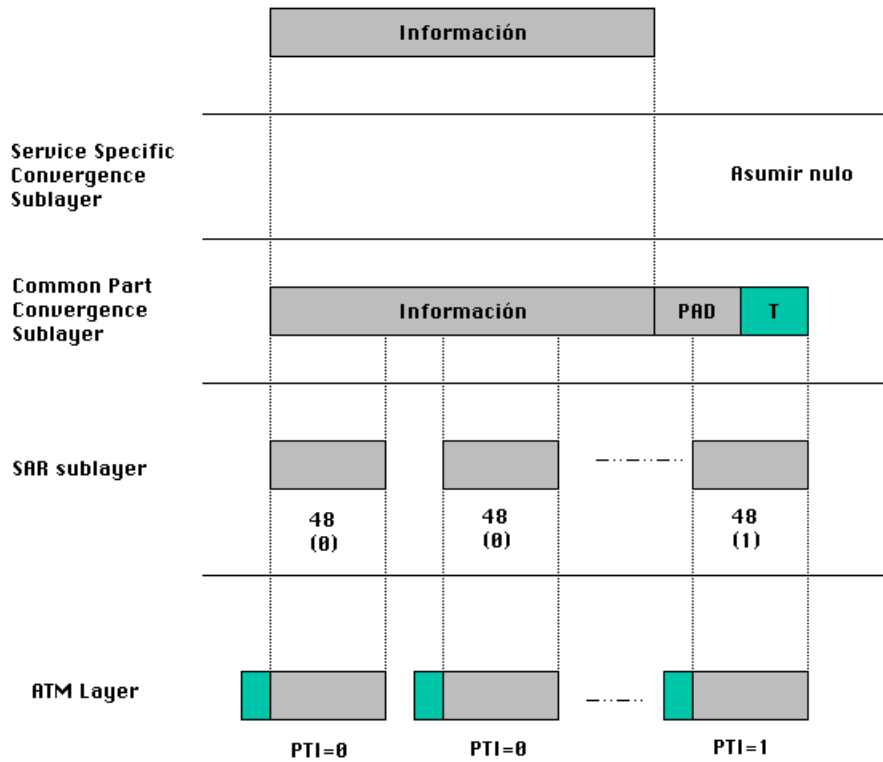


Figura 3.5 Proceso AAL5

La sub-capa SAR (Segmentation and Reassembly) parte la CPCS-PDU en cargas de 48 bytes que son pasadas a la capa ATM. La SAR da instrucciones a la capa ATM de ajustar el campo PTI (tipo de carga) en el encabezado de la última celda del PDU. Este paso señala el límite entre grupos de celdas que corresponden a diferentes mensajes a ser distinguidos en su destino. Nótese que a diferencia de AAL3/4, AAL5 puede tener solo un paquete a la vez en un circuito virtual (VC) porque las celdas de diferentes fuentes no pueden ser entremezcladas.

AAL5 es mucho más eficiente que AAL3/4 en el sentido de que no añade ningún encabezado en la sub-capa SAR. AAL5 no incluye números de secuencia para las PDUs SAR y en vez de esto confía más en el checksum del CRC para detectar pérdida, inserción errónea o celdas fuera de secuencia. AAL5 es hasta la fecha la capa de adaptación ATM más implementada[1].

Capítulo 4

Internet Protocol

4.1 La arquitectura de TCP/IP.

El corazón de la Internet de hoy es IP, el Internet Protocol. Este protocolo es el cimiento para la transmisión de información en Internet. El protocolo de Internet (IP) habilita comunicaciones a través de una colección vasta y heterogénea de redes que están basadas en distintas tecnologías. Cualquier computadora o *Host* conectado a Internet puede comunicarse con otras computadoras. Por lo tanto, Internet ofrece conectividad universal y las economías de escala que resultan de grandes despliegues.

Internet ofrece dos servicios de comunicación básicos que pueden operar por encima de IP: TCP (Transmission Control Protocol), un servicio de flujo confiable y UDP (User Datagram Protocol), un servicio de datagramas. Cualquier protocolo de la capa de aplicación que opera sobre TCP o UDP automáticamente opera a través de Internet. Por lo tanto Internet provee una plataforma universal para el despliegue de servicios basados en redes[1].

El conjunto de protocolos TCP/IP se refiere no solo a los dos más conocidos protocolos sino también a otros protocolos como UDP, ICMP (Internet Control Message Protocol) y las aplicaciones básicas tales como HTTP, Telnet y FTP. La estructura básica de TCP/IP se muestra en la figura 4.1.

Como cualquier edificio, el cimiento del conjunto del protocolo de Internet es la capa más baja, la cual se encuentra encima del transporte usado cualquiera que este sea, Ethernet, FDDI, o ATM. Este protocolo provee transporte de datagramas de un lado a otro, ruteando de red en red hasta que llegan a su destino. Es llamado apropiadamente Protocolo de Internet.

IP no provee garantías de entrega u orden. Eso se deja a los protocolos de nivel más alto. El más popular es el Protocolo de control de transmisión o TCP (Transmission Control Protocol). TCP al proveer un flujo confiable y ordenado de bytes a las aplicaciones, servicios como Telnet, http, y correo electrónico usan TCP.

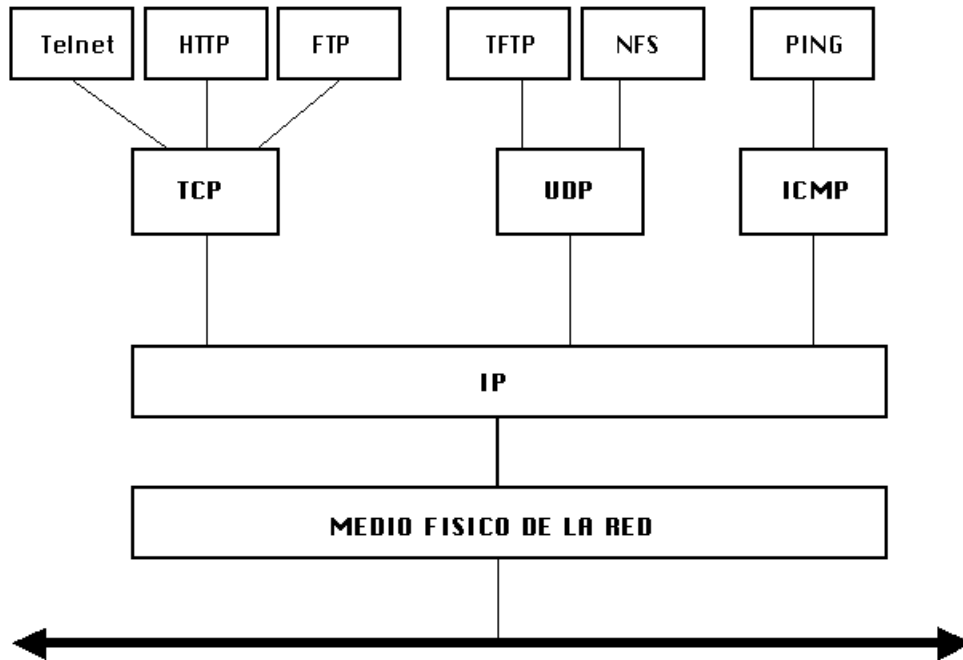


Figura 4.1 Protocolos de Internet

UDP (User Datagram Protocol), un protocolo de más alto nivel, se usa para la entrega de datagramas para procesos del usuario. Ejemplos de “usuarios” UDP son el TFTP (Trivial File Transfer Protocol) y el NFS (Network File System). El ICMP (Internet Control Message Protocol) proporciona mecanismos de control y retroalimentación. Sus atributos más visibles son el programa Ping y el Traceroute. Ping envía paquetes “echo request” a los servidores y espera de vuelta un “echo reply”. Esto es muy útil para saber si una máquina está en servicio o no.

Si se envía un mensaje con un “time to live” pequeño, un ruteador enviará un mensaje de error “time exceeded”. El programa Traceroute inteligentemente usa esto para probar la ruta de cierto destino, enlistando todos los ruteadores en el camino y el tiempo de ida y vuelta que toma a un paquete durante el viaje. Esto es extremadamente útil en problemas de diagnóstico de ruteo.

ICMP también provee retroalimentación acerca de la red. Los mensajes tales como “Host unreachable” y similares permiten a un servidor dejar de intentar transmitir información a un lugar que no la obtendrá[9].

Protocolos de la capa de aplicación tales como FTP y HTTP envían mensajes utilizando SNMP (Simple Network Management Protocol). DNS (Domain Name Service) envía sus mensajes usando UDP. Las PDUs intercambiadas por protocolos par TCP son llamadas segmentos TCP o segmentos, mientras aquellas intercambiadas por protocolos UDP son llamadas datagramas UDP o datagramas. IP multicanaliza segmentos y datagramas y desempeña fragmentación, si es necesario, entre otras tareas mencionadas más adelante. Las unidades de datos de protocolo intercambiadas por los protocolos IP son llamadas paquetes IP o paquetes (aunque algunas veces son llamadas datagramas IP, pero para evitar confusión con los datagramas UDP, se usará el término “paquetes” para referirse a las PDUs en la capa IP). Los paquetes IP son

enviados a la interfase de red para su entrega a través de la red física. En el receptor los paquetes son demultiplexados por el protocolo apropiado (IP, ARP, RARP). La entidad IP receptora necesita determinar que paquete tiene que ser enviado a TCP o UDP. Finalmente TCP (UDP) envía cada segmento (datagrama) a la aplicación apropiada basada en el número de puerto.

La PDU de cierta capa es encapsulada en una unidad de la capa inferior como se muestra en la figura 4.2. En esta figura un comando HTTP GET es pasado a la capa TCP, la cual encapsula el mensaje en un segmento TCP. El encabezado del segmento contiene un número de puerto efímero para el proceso del cliente y el conocido puerto 80 para el proceso del servidor HTTP. El segmento TCP en turno es pasado a la capa IP donde es encapsulado en un paquete IP. El encabezado del paquete IP contiene una dirección de red IP para la fuente y una para el destino. Las direcciones de red IP se dicen ser lógicas porque están definidas en términos de la topología lógica de los ruteadores y los sistemas terminales. El paquete IP es entonces pasado a través de la interfase de red y encapsulado en una unidad de la red subyacente. En la figura 4.2 el paquete IP es encapsulado en una trama de LAN (Local Area Network) Ethernet. El encabezado de la trama contiene direcciones físicas que identifican los puntos finales físicos para el transmisor y receptor. Las direcciones lógicas IP necesitan ser convertidas a direcciones físicas específicas para hacer la transferencia de bits de un dispositivo a otro. Esta conversión es hecha por el protocolo ARP (Address Resolution Protocol).

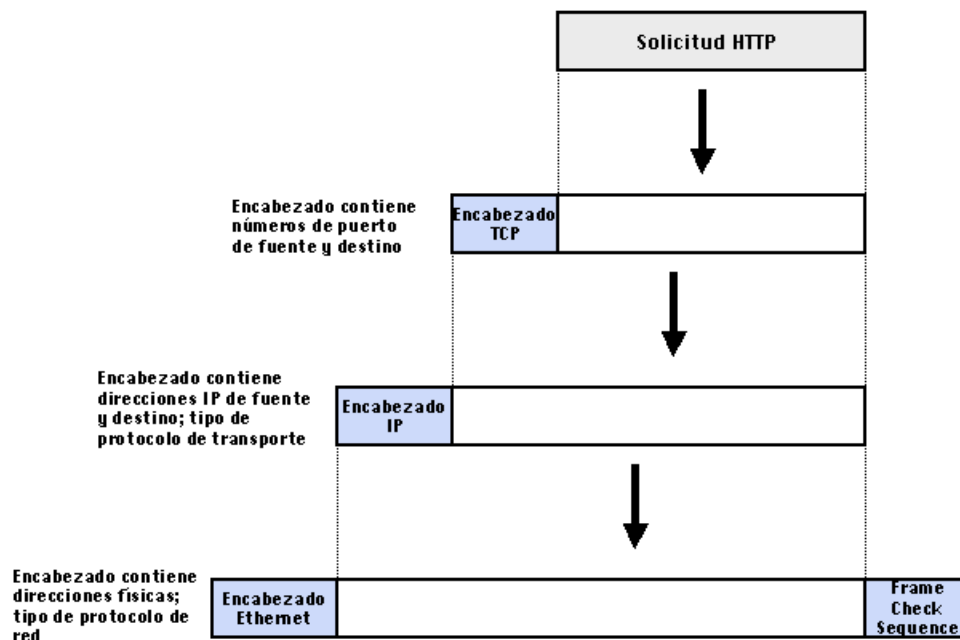


Figura 4.2 Encapsulado de PDUs en TCP/IP e información de dirección en los encabezados

La capa de Internet provee la transferencia de información en múltiples redes a través del uso de ruteadores (figura 4.3). También provee un servicio sencillo, llamado "transferencia de paquetes sin conexión de mejor esfuerzo" (best-effort connectionless packet transfer).

Los paquetes IP son intercambiados en ruteadores sin una configuración dispuesta, son ruteados independientemente, y pueden seguir diferentes rutas. Los *gateways* que interconectan las redes intermedias pueden descartar paquetes cuando hay congestión. La responsabilidad de recuperar estas pérdidas se pasa a la capa de transporte.

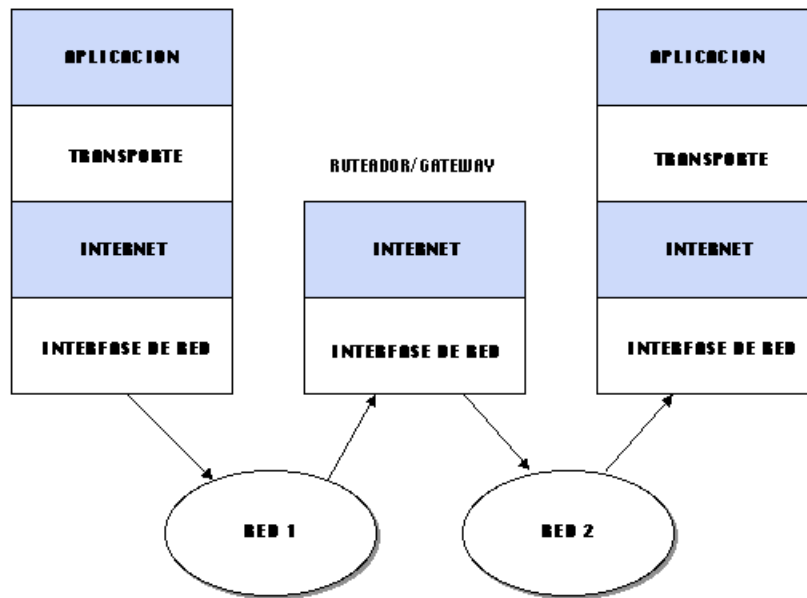


Figura 4.3 Internet y las capas de Interfase de Red

La capa de interfase de red está particularmente ocupada de los protocolos usados para tener acceso a las redes intermedias. En cada gateway el protocolo de red es usado para encapsular el paquete IP en una trama de la red o enlace subyacente. El paquete IP es recuperado en el ruteador de salida de dicha red. Este ruteador debe determinar el siguiente salto en la ruta al destino y encapsular el paquete IP o trama del tipo de la siguiente red o enlace. Este enfoque provee una clara separación entre la capa de Internet y la capa de interfase de red que depende de la tecnología. Este enfoque también permite a la capa de Internet proveer un servicio de transferencia de datos que es transparente en el sentido de no depender en los detalles de las redes subyacentes. Diferentes tecnologías de red imponen diferentes límites en el tamaño de los bloques que pueden manejar. IP debe acomodar la unidad máxima de transmisión (MTU) de una red o enlace subyacente implementando segmentación y reensamble como es necesario[1].

4.2 Formato de paquetes y direcciones de Internet.

Cada paquete contiene un encabezado de al menos 20 bytes que tiene 2 direcciones, la de la fuente y la de destino (figura 4.4).

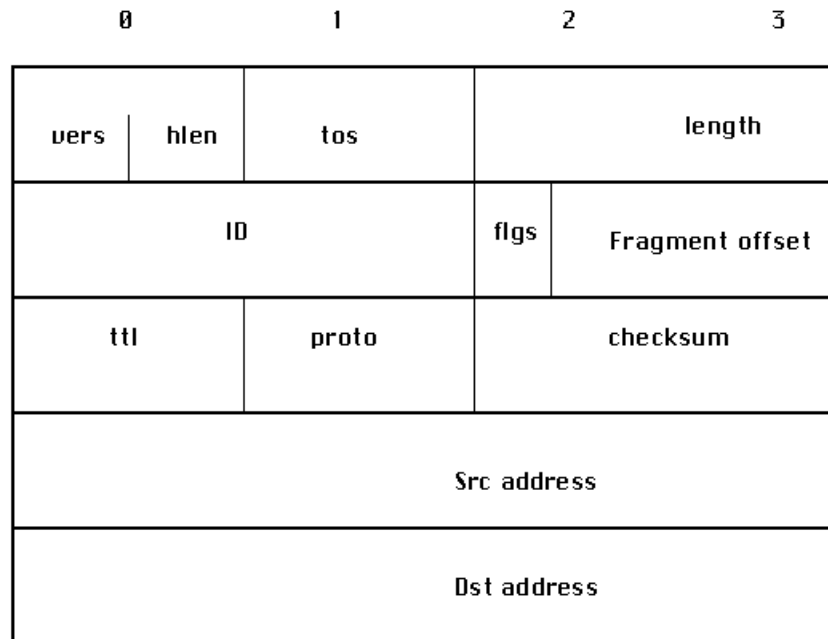


Figura 4.4 Encabezado del paquete IP

Las direcciones de IP son de 32 bits y están divididas en dos partes: el número de red y el número de servidor. Las direcciones están comúnmente escritas como números de 4 decimales, uno para cada byte, separados por puntos. Dado que 1 byte puede tener números de 0 a 255, el rango de direcciones va de 0.0.0.0 a 255.255.255.255. Los otros campos en el encabezado son:

- Vers: La versión IP usada. Siempre 4 (a menos que sea Ipv6, la cual aún no es usada ampliamente en Internet).
- Hlen: La longitud del encabezado del paquete en palabras de 32 bits. Usualmente 5, pero los encabezados de paquetes IP pueden tener partes opcionales que los hagan más grandes.
- Tos: Tipo de servicio. Especifica ya sea mínimo retraso, máxima entrega, máxima confiabilidad o minimizar costo. Este campo no es muy usado y usualmente se pone en ceros.
- Length: La longitud total del paquete IP
- ID: Este campo únicamente identifica el paquete; usado en fragmentación y reensamble.
- Flgs: Un bit es la bandera de “more fragments”, puesta en cada fragmento excepto el último.

- Fragment offset: El offset de la información de este paquete en el paquete original sin fragmentar.
- TTL: Time to live. Este campo es puesto por el remitente y decrementado cada vez que el paquete pasa a través de un ruteador. El paquete es desechado cuando llega a cero; esto previene que los paquetes caigan en trayectorias de ruteo cíclicas por siempre.
- Proto: Se refiere a que protocolo de capa superior pertenece la información en este paquete: TCP, UDP, ICMP, etc.
- Checksum: El checksum está diseñado para tener un encabezado correcto, la parte que contiene información en el paquete no es chequeada para errores en esta capa.
- Addresses: Las direcciones IP de la fuente y destino.

Las direcciones IP están divididas en partes, las partes de red y host. Típicamente una red pequeña será asignada a un número que forme parte de otra red, y las máquinas adjuntas a ella serán numeradas en el rango posible en la parte de host. Si la parte de host es todos unos, se considera una dirección de *broadcast* local y los paquetes enviados a esa dirección deberían ir a todos los hosts adjuntos a la red local.

Las direcciones pueden ser divididas en clases de acuerdo a cuantos bits de los 32 son asignados para la parte de host y cuantos para la parte de red. (figura 2.5).

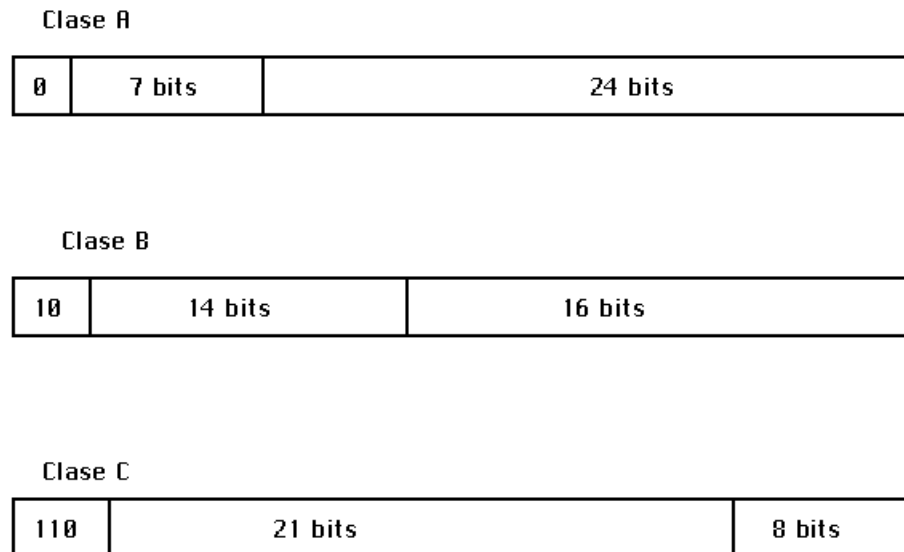


Figura 4.5 formatos de dirección de paquetes IP

Si la parte de red es de 8 bits, la red es llamada de clase A. Las partes de red clase A pueden ir de 1 a 127; el bit más significativo siempre es cero.

El esquema de direccionamiento es arreglado de tal manera que si los dos bits de más alto orden son 110, es una dirección de clase C. Recordando como trabaja la matemática binaria, esto

significa que el primer byte de una clase A puede ser de 1 a 127, uno de clase B de 128 a 191, y de clase C de 192 a 223. (después de 123, con 3 bits altos obtenemos direcciones en la clase D, las cuales son usadas en IP multicast. Cuatro bits altos indican direcciones clase E.)

Es fácil imaginar que una clase F empezaría con 5 bits altos, pero de hecho, no hay tal familia de direcciones estandarizada. Lo importante aquí es notar que mientras pueden ser 2^{24} direcciones (cerca de 16 millones) en una dirección de clase A, solamente puede haber 127 redes clase A en total. Similarmente, aunque puede haber 2^{21} redes clase C, solamente puede haber 256 hosts en cada una. Con el reciente crecimiento explosivo de Internet, hay un creciente escaseo de direcciones IP. Con la siguiente generación del protocolo IP (Ipv6) se intenta resolver estos problemas con más direcciones.

4.3 El protocolo IP

IP es un protocolo de datagramas, lo cual significa que no hay conexión involucrada en la entrega de paquetes, y que el paquete contiene suficiente información en él para ser ruteado hasta su destino durante todo el trayecto. El host remitente solo lanza al paquete en la red y se espera que éste llegue a su destino. La mayoría de las veces, cuando las cosas trabajan, lo hace, y esto se vuelve suficiente cimiento para construir un buen número de protocolos sofisticados.

4.3.1 Ruteadores (Routers)

Si los *hosts* de fuente y destino están en la misma red local, entonces el paquete solo va de uno a otro. Si están en diferentes redes, entonces, el paquete tiene que ser re-enviado a la otra red. El ruteador recibe los datagramas, checa la dirección de destino, consulta una tabla de ruteo para determinar adonde debe ser enviado el datagrama, y mandarlo. Eventualmente llega a un ruteador en el destino, el cual manda el paquete al host final.

Dado que Internet es muy grande y en constante crecimiento, la información acerca de a cual *host* están conectados por tales ruteadores y estos a su vez conectados unos con otros, está cambiando constantemente. Los ruteadores intercambian esta información usando protocolos de ruteo para mantener sus tablas internas exactas.

4.3.2 Unidad Máxima de Transmisión (Maximum Transmission Unit).

Los ruteadores se conectan unos con otros diferentes tecnologías de enlace, todo desde una línea telefónica con un modem hasta enlaces ATM Gigabit. Diferentes enlaces permiten diferentes tamaños de paquete. El paquete más grande permitido, por ejemplo, en una red Ethernet es de 1500 bytes, mientras que un enlace ATM usando AAL5 podría usar cerca de 65,535. El tamaño más grande legal de un paquete IP que puede ser transmitido en un enlace dado es llamado Maximum Transmission Unit (MTU).

4.3.3 Fragmentos.

¿Entonces que pasa cuando un paquete de 8000 bytes llega a un ruteador y tiene que ser enviado a través de un enlace donde se tiene un MTU de 1500? IP permite que los paquetes se fragmenten, cada pieza del mensaje original se envuelve en otro paquete IP de tamaño legal para el enlace de salida. La fragmentación IP usa el campo de identificación en el encabezado IP para identificar a cual paquete pertenece el fragmento y el campo de *fragment offset* para identificar en que parte del paquete va el fragmento.

El host receptor debe almacenar los fragmentos y reensamblarlos cuando todas las piezas llegan. Si un fragmento llega a un ruteador con un enlace aún más pequeño, entonces el

fragmento debe ser refragmentado. Dado que cada encabezado transporta la longitud del fragmento y su desplazamiento en el paquete entero, el reensamble es tedioso.

En cualquier caso, la fragmentación y el reensamble de paquetes consume muchos recursos. El host receptor debe no solo asignar un buffer en el cual reensamblar el paquete y conservarlo hasta que el último fragmento llegue, debe también mantener rastro de que paquetes han llegado y cuando y periódicamente checar en paquetes parciales para ver si debe desecharlos.

Es mejor evitar la fragmentación al enviar paquetes de un tamaño que puedan pasar a través de la red sin ser fragmentados. Para hacer esto, algunas pilas TCP/IP soportan “Path MTU discovery” donde se manda un paquete grande con el bit “*Don't fragment*” puesto y se recibe un mensaje de error ICMP de los ruteadores intermedios que dicen que la fragmentación es necesaria pero no pueden hacerlo, e incluyen el MTU del próximo salto en la red, aquel que no pudo tomar el paquete.

Reduciendo el tamaño del paquete e intentando de nuevo, eventualmente el MTU mínimo de la ruta completa puede ser determinado, y si todos los paquetes enviados no son más grandes que eso, la fragmentación se puede evitar para tal conexión, conduciendo a un mejor desempeño, o al menos, evitando fragmentación que evite una pérdida en desempeño.

4.4 TCP.

Con IP, tenemos una manera de entregar información de computadora a computadora a través de Internet. IP por sí mismo no hace absoluta garantía de que la información será entregada. Por lo tanto depende de un protocolo de capa superior imponer secuencia y confiabilidad en el flujo de paquetes IP. El protocolo usado para entregar flujos confiables de información a las aplicaciones es TCP.

TCP es un protocolo orientado a conexión que provee un flujo confiable de bytes a las aplicaciones. No hay fronteras de registro en flujos TCP, y aún las fronteras de los paquetes no pueden ser confiablemente determinadas por una aplicación.

Cuando se usa TCP, una aplicación que desea comunicarse con otra debe establecer primero una conexión. Una vez que esta hecha, cualquier lado puede enviar información hasta que una decide terminar la conexión.

Los paquetes TCP son transportados como información dentro de los paquetes IP o encapsulados. Los ruteadores intermedios no necesitan checar los encabezados TCP; el encabezado IP provee la suficiente información para llegar a su destino.

Cuando TCP recibe información de la aplicación, la rompe en segmentos. Cada punto final de la conexión puede anunciar el MSS (Maximum Segment Size) y no puede ser por supuesto, más grande que el MTU en la red local.

TCP usa números de secuencia para identificar el lugar en un flujo donde va la información del segmento actual. El receptor envía un segmento con el bit ACK puesto para indicar que ha recibido información hasta el byte identificado por el número en el campo Acknowledgement number en el encabezado TCP. Un encabezado de segmento TCP usualmente es de 20 bytes.

Cada conexión TCP no solo conecta dos *hosts*, también conecta dos puertos en los *hosts*. El puerto es una manera de permitir múltiples conexiones en un host sencillo. Usualmente un programa en el host es asociado con un puerto dado. La combinación de direcciones de fuente y destino y puertos de fuente y destino, identifica completamente una conexión TCP. Note que esto significa que un cierto servidor (o cliente) puede tener múltiples conexiones en su puerto dado que la dirección IP y el puerto de los extremos remotos serán diferentes.

4.5 UDP.

En contraste con TCP, UDP no es protocolo de flujo. Donde TCP esconde (o por lo menos no expone) las fronteras entre paquetes, con UDP cada operación de salida causa que un paquete UDP sea enviado. No hay garantía de entrega. Aún así, UDP puede ser muy útil. El encabezado de un paquete UDP es usualmente de 20 bytes como en TCP.

UDP es un protocolo de la capa de transporte orientado a no conexión. Es muy simple y provee solo dos servicios adicionales más allá de IP: demultiplexar y checar errores en la información. IP sabe cómo entregar los paquetes al host pero no sabe como entregarlos a una aplicación específica en el host. UDP añade un mecanismo que distingue entre múltiples aplicaciones en el host. IP solo checa la integridad del encabezado. UDP pueden opcionalmente la integridad del datagrama completo. Aplicaciones que usan UDP incluyen TFTP, DNS, SNMP, y el RTP (Real Time Protocol)[9].

4.6 IP Multicast.

Multicast es el término que se utiliza para la función que consiste en enviar un paquete a un grupo de hosts en vez de a un host sencillo. Transmisiones normales “*unicast*” involucran enviar un paquete a un solo destino y es tan común que rara vez pensamos en esto en dichos términos. “*Broadcast*” es la práctica de enviar un paquete a todos los hosts (en una área limitada).

Multicast cae entre estos dos extremos. Es útil para construir aplicaciones pseudo-tiempo real distribuidas tales como videoconferencia y audio conferencia. Sin embargo, su uso no está restringido a este tipo de aplicaciones. Cualquier aplicación que involucre el envío de copias de información a múltiples destinos puede beneficiarse. Por mencionar una aplicación, uno podría distribuir tablas de ruteo a todos los ruteadores de una empresa, sin cargar a todas las estaciones de trabajo con el procesamiento (y eliminación) de todos los mensajes[2].

Los programas *multicast* usan UDP en vez de TCP. Por una cosa, la naturaleza de la mayoría de los flujos multicast es tal que si un paquete o dos se pierden no se vuelve un desastre; tal vez algún efecto raro en el audio o una parte de un cuadro de video no se actualiza por otros 30 ms. Además TCP usa reconocimientos (ACKs) para controlar la ventana y eso crea problemas al tener varios usuarios. UDP es mejor para estos propósitos e incluso puede ser usado para una transferencia de datos “confiable” apoyandose en conexiones extra para tratar con paquetes descartados.

Dado que no todos los ruteadores tienen funcionalidad *multicast*, se puede usar tunneling. Cuando un ruteador necesita mandar un paquete *multicast* hacia otro ruteador al cual está conectado por tal túnel, encapsula el paquete dentro de un paquete IP unicast y lo envía a través del túnel. El ruteador destino desempaqueta el mensaje y descarta el paquete *multicast* en su red local.

IP *multicast* no es suficiente por si solo para cumplir con todos los requerimientos de las aplicaciones IP multicast. Los problemas más grandes al correr aplicaciones multicast multimedia sobre redes de paquetes tienen que ver con el cumplimiento de los requerimientos de ancho de banda y latencia de estas aplicaciones. Ciertamente, los protocolos de ruteo multicast ayudan, pero lo que se necesita es una red capaz de discriminar flujos de tráfico que tienen diferentes requerimientos de nivel de servicio y entonces darle prioridad al tráfico a través de la red basada en estos requerimientos, que son los asuntos de Calidad de Servicio (QoS)[1].

4.7 RTP (Real Time Protocol).

Las redes de paquetes fueron desarrolladas básicamente para el transporte de datos que no tenían exigentes requerimientos de tiempo. Los avances en algoritmos de compresión y el poder de

procesamiento de cómputo combinado con las mejoras en el ancho de banda y conmutación de paquetes están haciendo posible el soporte de comunicaciones de tiempo real en redes de paquetes. Este tipo de comunicaciones hacen posible explotar la universalidad de Internet. Las comunicaciones de paquetes pueden traer capacidades tales como multicasting que no son fácilmente provistas por redes de circuitos conmutados. Sin embargo, las comunicaciones por paquetes tienen que tratar con problemas inherentes en las redes de paquetes, las cuales incluyen retraso y jitter, pérdida de paquetes y entrega fuera de secuencia o duplicado de paquetes. El RTP provee funciones de transporte de fin a fin para aplicaciones que requieren transmisión en tiempo real, tales como audio o video sobre servicios de red de paquetes unicast o multicast. Los servicios RTP incluyen identificación del tipo de carga útil, numeración de secuencia, estampado de tiempo y monitoreo de entrega. RTP normalmente corre encima de UDP, pero también puede correr sobre otros protocolos de transporte o red. RTP no provee medios de la entrega a tiempo de la información o garantías de calidad de servicio. Para esto, RTP debe depender en los servicios de la capa de red. RTP en su lugar provee mecanismos para tratar con problemas tales como jitter y pérdida de información, así como recuperación en tiempo y sincronización intermedia.

El RTP Control Protocol (RTCP) es un protocolo de complemento para monitorear la calidad de servicio observada en el receptor. RTP intencionalmente, no es un protocolo completo e intenta ser suficientemente flexible para que pueda ser incorporado en el proceso de la aplicación en vez de ser implementado como una capa separada. El uso de RTP en una aplicación en particular requiere entonces de uno o más documentos complementarios. Un documento de especificación de perfil define atributos y/o modificaciones y extensiones a RTP para una clase de aplicaciones, por ejemplo, audio y video. Un documento de formato de información en turno define como una codificación de audio o video en particular será transportada sobre RTP, por ejemplo video MPEG-2 o audio ADPCM.

Los protocolos RTP y RTCP tienen un amplio uso en Internet soportando aplicaciones de flujo de audio y video así como Internet Telephony y otras aplicaciones de tiempo real[1].

Capítulo 5

IP sobre ATM

5.1 Introducción.

Debido a las capacidades de calidad de servicio y circuito virtual de ATM, este es ampliamente empleado por ISPs (Internet Service Providers) como protocolo de enlace de datos con SONET/SDH como capa física. Los circuitos virtuales están creados entre los ruteadores IP. Típicamente, múltiples rutas VC son creadas entre los ruteadores, y los esquemas de ingeniería de tráfico son usados para balancear la carga de tráfico y posiblemente ofrecer una ruta preferencial a tráfico de alta prioridad. El circuito virtual es un concepto de capa 2 que provee un transporte eficiente porque todos los ruteadores están a un salto de cada uno. ATM es atractivo porque su habilidad de crear rutas virtuales y canales virtuales significa que el tráfico de múltiples usuarios puede ser transportado dentro de la misma red sin preocupaciones de seguridad.

Hay entonces diferentes enfoques y técnicas desarrolladas con la intención de dar a IP capacidad y mejoras en el concepto de calidad de servicio, así como una mejor integración con una infraestructura basada en ATM. Este capítulo se concentra en uno de los modelos de overlay de IP sobre ATM más implementados, que es LANE (LAN Emulation) del foro ATM, y que es precisamente la arquitectura que presenta la red de datos del Campus Monterrey del Sistema ITESM, donde se hicieron las pruebas de tráfico.

5.2 Emulación LAN.

Es reconocido que una de las claves en la amplia adopción de ATM es su habilidad para interoperar con las redes existentes. LAN Emulation (LANE) es el intento del foro ATM para hacer trabajar ATM en conjunto con redes Ethernet y Token Ring. El concepto básico es que los protocolos LANE y ATM se vean exactamente como Ethernet o Token Ring para las computadoras existentes y sus aplicaciones, mientras se usa las capacidades de la red ATM para proporcionar un desempeño más alto.

LANE trabaja presentando la capa de red con una interfase que es idéntica a aquella de las LANs de herencia. Por lo tanto, no se necesitan hacer cambios a los protocolos o aplicaciones en las computadoras existentes, y los controladores que cumplen con LANE proporcionan interfaces idénticas (ej. NDIS) a los niveles más altos. LANE no intenta traducir los protocolos Token Ring en Ethernet o viceversa. Una LAN emulada debe ser una o otra, y si se desea una comunicación entre ellas, se debe involucrar un ruteador.

Hay diferentes partes que intervienen en una LAN emulada (ELAN) usando LAN Emulation (figura 5-1). Muchos de los documentos del foro ATM se refieren a “entidades”; dichas entidades son módulos de software residentes en los *hosts* que procesan los protocolos LANE.

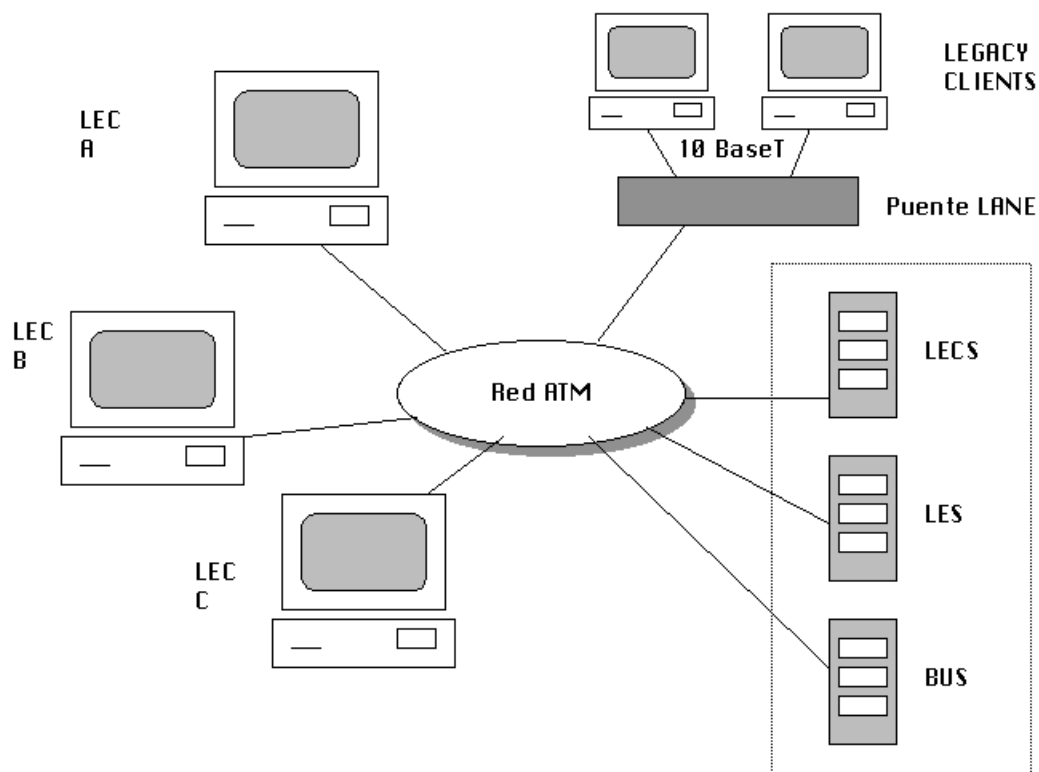


Figura 5.1 Configuración de LAN Emulada

5.2.1 LEC (LAN Emulation Client).

La primera entidad es el LEC (LAN Emulation Client). Esta entidad reside en los sistemas terminales. El LEC desempeña la resolución de direcciones ATM y presenta la interfase LAN estándar al resto de la máquina, enmascarando como una Ethernet. Cada LEC debe tener una dirección ATM única.

5.2.2 LES (LAN Emulation Server).

La siguiente entidad es el LES (LAN Emulation Server). Hay un LES por cada LAN emulada y eso significa tener una conexión con servidor LANE. Cada LES debe tener una dirección ATM única.

5.2.3 BUS (Broadcast and Unknown Server).

The broadcast and Unknown Server (BUS) manipula tráfico broadcast y multicast para una LAN dada. Puede haber uno o más por ELAN, pero típicamente es uno. Cada BUS debe tener una dirección ATM única, la cual el LES asociará con la dirección MAC (“todos unos” o por ej. FF:FF:FF:FF:FF:FF para Ethernet).

5.2.4 LECS (LAN Emulation Configuration Server).

Finalmente, está el LECS, su rol es asignar nuevos clientes iniciados a la ELAN correcta. Conoce el apropiado LES para cada uno a conectar. Habrá un LECS por “dominio administrativo” que servirá todas las ELANs ya sean Ethernet o Token Ring.

Estas entidades servidor pueden residir en cualquier máquina adjunta a una red ATM, pero típicamente se encontrarán en switches o ruteadores. Dado que la fase 1 de LANE permite solamente un BUS o LES por LAN emulada, representa un punto sencillo de falla potencial. Por supuesto, dado que un BUS o LES usualmente reside en un switch, si el switch fallara, la red se iría abajo. La fase 2 especifica protocolos que permiten múltiples servidores redundantes.

5.3 Puentes LANE.

Un cliente LANE puede actuar como puente, reenviando tramas hacia (y desde) los hosts y segmentos LAN a los que está conectado. Desempeña servicios de cliente LANE en favor de las máquinas conectadas a él y envía notificaciones de cambio en la topología al LANE Server cuando su topología cambia. El LES propagará la información de la topología a otros clientes.

5.4 Pros y Contras.

LANE es un protocolo puente de capa 2 y un poco complicado. Escalar LANE a redes verdaderamente grandes no es muy factible. Es muy bueno si se tiene un backbone sobrecargado conectando los grupos de trabajo. Se puede actualizar poniendo un switch ATM y reemplazando todos los hubs 10baseT con puentes LANE. Las estaciones de trabajo locales seguirán usando Ethernet 10baseT para comunicarse con los hubs, pero ahora el backbone es 15 veces más rápido. LANE tiene varios inconvenientes, el requerimiento de que LANE esconde los detalles de la red ATM que se encuentra debajo desde la capa de red, también implica que los atributos de calidad de servicio (QoS) no pueden estar disponibles a los protocolos de la capa de red. LANE está definido para operar conexiones UBR y ABR lo cual se acerca más a los servicios proporcionados por los protocolos LAN MAC.

En el concepto de paquetes sobre ATM, debido a que el circuito virtual opera en la capa de enlace de datos, hay un encabezado de esta capa asociado con el circuito virtual. Otra desventaja es porque cada ruteador establece un circuito virtual con los demás ruteadores, ello implica que varios circuitos virtuales sean requeridos. También como cada ruteador está lógicamente dirigido a cada ruteador, un número muy grande de actualizaciones de ruteo deben ser procesados, y una base de datos de estado de enlace muy grande debe mantenerse[4].

5.5 Implementación multicast para LANE.

Multicast en sistemas LANE es muy simple, pero consiste en esconder la complejidad de la habilidad para establecer circuitos virtuales señalados como punto a multipunto. Cuando un LEC quiere enviar un paquete multicast, lo envía sobre su previamente establecida Multicast-Send VC al BUS. El servidor lo envía de regreso en el Multicast-Forward VC y todos los LEC reciben el paquete.

Esta manera de enviar multicast permite a la LAN emulada comportarse como una LAN tradicional de medio compartido. Las desventajas es que todas las estaciones tienen que mirar el paquete y decidir si lo descartan o lo aceptan; y esto puede ser ineficiente. Además, agregar ancho de banda multicast en la LAN emulada está restringido a aquel de un VC o cuantos puede manejar el BUS[9].

Capítulo 6

Pruebas y Estadísticas de Red.

6.1 Descripción del Equipo

El equipo con el que se realizaron las pruebas consistió de 2 estaciones de trabajo, 2 computadoras portátiles y un conmutador ATM que se describe con más detalle posteriormente. Las estaciones están conectadas al conmutador por medio de una interfase a 155Mbps con fibra óptica multimodal en circuito virtual permanente. Las características de las estaciones es la siguiente:

Nombre: Wiener

- Sun UltraSparc 60
- Sistema operativo: SunOS 5.7
- 512 MB RAM
- Dirección IP: 131.178.26.43
- Dirección LANE: 131.178.107.89

Nombre: Gabor

- Sun UltraSparc 60
- Sistema Operativo: SunOS 5.6
- 512 MB RAM
- Dirección IP: 131.178.26.38
- Dirección LANE: 131.178.107.88

La tarjeta de red para interconexión con el conmutador muestra las siguientes especificaciones:

- ForeRunner 200E NIC (Network Interface Card)
- Coprocesador Intel i960
- Arquitectura Marconi SAR ASIC
- ATM UNI 3.0/3.1, LANE 1.0, RFC 1577
- Proceso de celdas por especificación ITU 1361/ ANSI T1S1 5/92-002R3
- Soporte para AAL5 (ATM Adaptation Layer 5)
- Cableado: Duplex 62.5 μ /125 μ fibra multimodo (2,000m max, 10dB de perdida)

Las computadoras portátiles utilizadas para la recepción y análisis tienen las siguientes características:

Compaq Notebook 100

- Sistema Operativo: Windows XP Pro.
- Procesador AMD K6-II 475Mhz
- 160 MB RAM
- Interfase Ethernet 10/100 Mbps
- Dirección IP de fuente de transmisión: 131.178.48.230

Apple iBook

- Sistema Operativo: Mac OS X (Darwin)
- Procesador PowerPC G3 500Mhz
- 320 MB SDRAM
- Interfase Ethernet 10/100 Mbps

6.2 Descripción del Conmutador

El conmutador empleado para este experimento es el Cisco LightStream 1010. El Cisco LightStream 1010 es un conmutador modular a 5 Gbps diseñado para uso en grupos de trabajo o un campus, dependiendo de la naturaleza de las interfaces empleadas. El conmutador 1010 tiene un chasis modular de 5 ranuras con la opción de fuentes de poder duales tolerantes a fallas y de carga compartida. La ranura central en el 1010 está dedicada a un módulo sencillo reemplazable ASP (ATM switch processor) que soporta memoria compartida de 5Gbps, de tarjeta y *fabric* sin bloqueo, y un procesador RISC de alto desempeño. El módulo ASP puede ser escalable en campo permitiendo al conmutador seguir con los cambios en las especificaciones ATM. Las ranuras restantes hasta cuatro módulos intercambiables (CAMs), cada uno con soporte para 2 módulos adaptadores de puerto (PAMs), para un máximo de 8 PAMs por conmutador para una amplia variedad de interfases ya sea para *backbone*, o escritorio ATM de área amplia.

Los mecanismos de administración de tráfico pueden soportar ráfagas, tráfico de mejor esfuerzo, y entregar calidad de servicio. El mecanismo de control de congestión ABR (Available Bit Rate) permite al switch a frenar fuentes de tráfico antes de que la congestión se vuelva excesiva. Soporta los protocolos Forum ATM para PNNI. Como aplicación de administración utiliza el ATMDirector que puede ser integrado con plataformas de administración SNMP o totalmente independiente.

Señalización y Ruteo del Cisco LS 1010

- UNI 3.0./3.1
- LANE 1.0
- ILMI
- PNNI Phase 1, IISP
- Soporte Soft PVC/PVP
- Listas de acceso ATM Y Firewalls
- Modo Plug-and-Play con imagen PNNI
- Soporte para enlace redundante con balance de carga
- Conexiones: 32,000 punto-punto, 2048 punto-multipunto
- Conmutación VC y VP, tunneling VP
- PVC y SVC
- OAM Ping en IP o dirección ATM

CiscoView es una aplicación de administración para el conmutador con interfase gráfica que provee estatus dinámico, estadísticas e información para configuración. Puede ser corrido en estaciones de trabajo UNIX como aplicación independiente y también puede ser iniciado desde el ATMDirector.

El conmutador puede ser usado para correr en LANE (LAN Emulation), lo cual representa una de las características más importantes para los propósitos de este proyecto.

6.3 Descripción del Software para difusión de audio.

La transmisión y difusión de audio codificado en MPEG Capa III estuvo a cargo del ShoutCast DNAS (Distributed Network Audio Server). Es una aplicación sin costo propiedad de Nullsoft que puede correr en los siguientes sistemas operativos: Windows 95, 98, NT, FreeBSD, Linux y Solaris 7.

Esta solución comprende 2 sencillos componentes, que son el SHOUTcast Server y el SHOUTcast source Plug-in para Winamp. El sistema puede entregar audio en vivo (Streaming) y bajo demanda. Este servidor se encuentra instalado en la estación Wiener desde la cual se hicieron todas las transmisiones.

El servidor tiene una operación sencilla y ocupa poca memoria (14KB aprox. por escucha). Al correr en plataforma UNIX necesita de la alimentación del Plug-in para Winamp, el cual sólo existe en una versión para Windows. Winamp, propiedad también de Nullsoft, es un popular reproductor de audio codificado en MP3 y algunos otros formatos como WMA, WAV, etc. La función básica del Plug-in es mandar la información de audio desde la aplicación Winamp hacia el servidor UNIX, que a su vez se encarga de transmitir el audio a los usuarios conectados. El Plug-in puede ser configurado para mandar la información en las diferentes tasas (Bit Rates) y se apoya en el codec Fraunhofer IIS MPEG-Capa III. El DNAS tiene también la capacidad de entregar audio en demanda que este almacenado en el directorio "content" bajo la solicitud desde una pagina web previamente definida con la referencia del material existente.

El servidor en activo provee información del flujo tal como nombre, artista, tasa de transmisión, bit rate, género y dirección del servidor; estadísticas de difusión como el numero de usuarios conectados, registros de conexión, tiempo de transmisión, etc.

En el sistema, el proceso de difusión y recepción se realiza solo mediante solicitudes y conexiones bajo HTTP, que es parte del protocolo TCP. Por lo tanto, todo el análisis se centra en este esquema de transferencias en IP.

6.4 Descripción de las pruebas realizadas.

El experimento basado en difusión de audio se hizo bajo diferentes escenarios, los cuales contemplaron tráfico en horas pico y no pico, distintas localidades de recepción en cuanto a subredes, diferentes tasas de transmisión y alcance fuera de la red del campus.

El servidor estuvo activo las 24 hrs. durante el periodo de pruebas. Sin embargo, el DNAS solo transmite al recibir de la fuente el audio codificado y tener usuarios conectados. El número de usuarios posible se limita al ancho de banda disponible. En este caso, la limitante es el gateway o ruteador de la subred del usuario receptor solo si la salida del servidor es por el conmutador en el dominio de ATM (155Mbps). Por lo general, dentro de la red del campus, se presenta un ancho de banda disponible de 10Mbps en teoría. La estación de trabajo esta configurada por default para tener una salida directa por la dirección LANE, pero puede realizar la comunicación en el dominio de Ethernet si así se le solicita. Tomando en cuenta esto y suponiendo que pueden iniciarse sesiones en ambas modalidades, se limito el número máximo de usuarios acorde con el ancho de banda disponible del gateway de la red del CET (Centro de Electrónica y Telecomunicaciones) y el resto de las subredes del campus, considerando que un flujo demasiado alto podría afectar el desempeño de la red a nivel local.

Al tener en teoría 10 Mbps disponibles, se decidió configurar el servidor con un número máximo de usuarios de 32. En realidad nunca se alcanzó dicho número de usuarios pero la cifra servía para poner una barrera en la cantidad de información transmitida. Si suponemos que hay 32 usuarios conectados al mismo tiempo a una tasa de 128kbps, entonces tenemos que aproximadamente el ancho de banda consumido es de 4.096 Mbps, lo cual es un nivel aún aceptable.

Si se tuviera el todo el ancho de banda disponible y se desea calcular el número máximo de usuarios, se multiplican los 10Mbps por 0.9 tomando en cuenta el encabezado, y se divide entre el *bit rate* con el cual se transmitirá. Tenemos entonces:

$$\frac{10Mbps \times 0.9}{128Kbps} = 70.31245 \text{ usuarios.}$$

En la práctica y durante el experimento, el número de usuarios no excedió a 5. Lo importante fue ver qué tanto soportaba la red las tasas empleadas manteniendo una transmisión sin interrupciones o saltos en el audio. Ya que tal información de audio codificado se mezclaba con el tráfico normal del campus.

La figura 6.1 muestra un esquema general de la red del campus. Podemos observar de manera aproximada el lugar que ocupa nuestro conmutador en la red y los enlaces que existen dentro de esta.

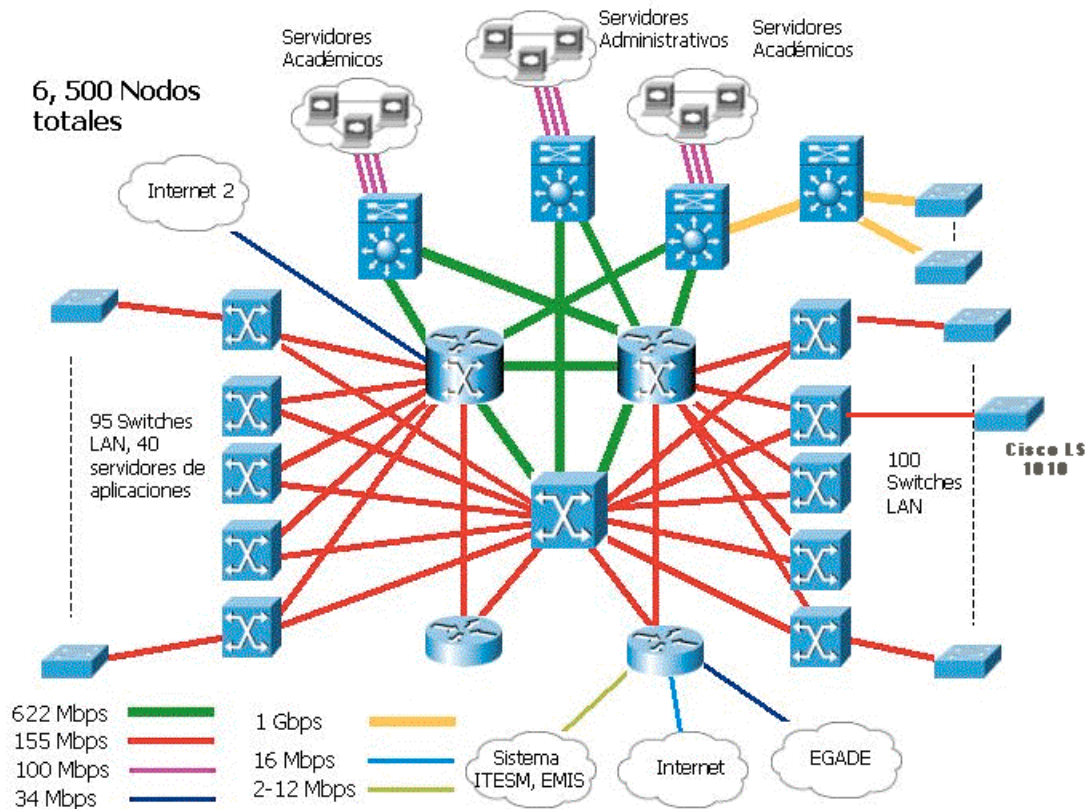


Figura 6.1 Esquema de la red del campus[12]

El flujo de audio es posible captarlo por medio cualquier reproductor de MP3 con la capacidad de conexión mediante http, tal como Winamp para Windows, XMMS, Real Player para Unix o iTunes para Mac OS. La herramienta de recepción fue la aplicación iTunes por HTTP a través del puerto 8000. El sistema opera sus transmisiones en modo *unicast*, de manera que cada usuario representa una conexión y flujo independiente. Es por esto que el ancho de banda disponible se reduce al incrementarse el número de escuchas. Por lo tanto una solución más conveniente sería utilizar multicast evitando así afectar el desempeño del servidor y hacer más eficiente el uso del

medio. Sin embargo, el uso de multicast en una ELAN tiene algunas otras implicaciones que hay que estudiar y enfrentar para un mejor desempeño de la red.

Se hicieron también una serie de pruebas de transmisión entre las dos estaciones de trabajo. El flujo de audio sale del servidor en Wiener a través del conmutador pasando por el core de la red del campus y llega a su destino final el receptor Real Player en Gabor, donde se tomaron estadísticas de red.

6.5 Resultados Obtenidos.

Las transmisiones se realizaron en ambos modos de operación del servidor, difusión en vivo y bajo demanda, durante varios días y en diferentes locaciones (subredes) del campus, escogiendo principalmente horas de alta actividad en la red.

Las diferentes tasas a las que se transmitió pueden considerarse altas, con una calidad cercana o equivalente a la de un CD de audio. Por lo tanto son flujos que un enlace telefónico con un modem de 56K no puede soportar.

Los bit rates empleados fueron los siguientes: 128 Kbps, 160 Kbps y 192 Kbps. Aunque durante el experimento se observó que una buena recepción podía llegar hasta los 256 Kbps en la red, 128 es una cifra que representa una calidad bastante aceptable en audio. Las pruebas mostraron que la red del campus tiene la suficiente capacidad para soportar una difusión de estas características sin interrupciones o degradación aparente del audio, pues todas las transmisiones estuvieron libres de problemas de recepción. Es importante mencionar que es esto se dio aún cuando la fuente de audio se encontraba en una subred distinta a la del servidor SHOUTcast, viéndose sometida al posible tráfico que pudiera encontrar en esa subred.

Se realizaron pruebas extras con servidores externos comerciales localizados generalmente en Estados Unidos. El Streaming proveniente de estos servidores sí presentó interrupciones en la recepción de audio a pesar del buffer del reproductor. El bit rate correspondía a 128 Kbps y dichos problemas se presentaban principalmente en horas pico de la red. Podemos suponer que la salida del campus a Internet presenta un cuello de botella para recibir un servicio o correr una aplicación de estas características.

Dentro de las pruebas realizadas al interior del campus con una difusión en vivo encontramos los siguientes resultados. El recorrido promedio de un paquete en la red en viaje redondo a la dirección LANE (131.178.107.89) desde las diferentes locaciones y horarios se muestra en la figura 6.2

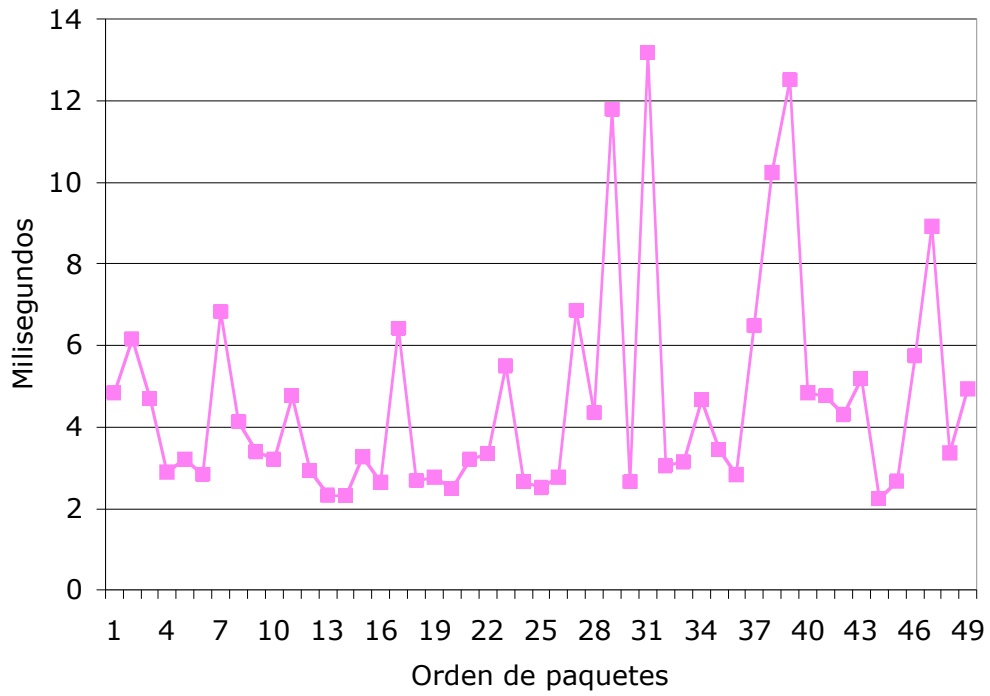


Figura 6.2 Tiempo de recorrido de paquetes a dirección LANE del servidor

Por lo tanto, el tiempo promedio se encuentra en 4.627612245 ms. El mínimo es 2.251 ms. El valor máximo alcanzado es 13.175 ms.

Las pruebas hechas para mostrar el tiempo de recorrido de los paquetes a la dirección IP (131.178.26.43) utilizando la interfase Ethernet resultaron en los valores indicados en la figura 6.3

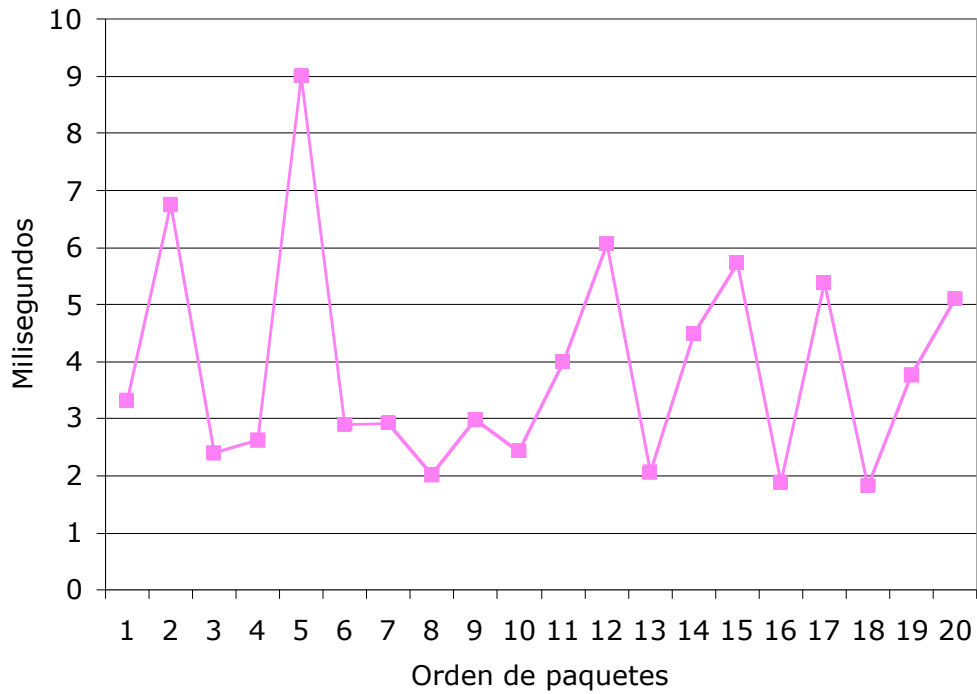


Figura 6.3 Tiempo de recorrido de paquetes al servidor por interfase Ethernet

El promedio resulta en 3.88345 ms. Valor máximo, 9.003 ms y el mínimo, 1.829 ms, lo cual indica que no hay una gran diferencia de los resultados de la gráfica 6.2. Todas las pruebas de recorrido fueron hechas enviando un Ping con un paquete de 56 Bytes de longitud. El paquete de retorno tenía una longitud de 64 Bytes

Otra manera para calcular el grado de retardo en la red (latency) es utilizar la herramienta Traceroute. Para los diferentes puntos del campus donde se realizó la recepción se hicieron pruebas con esta herramienta.

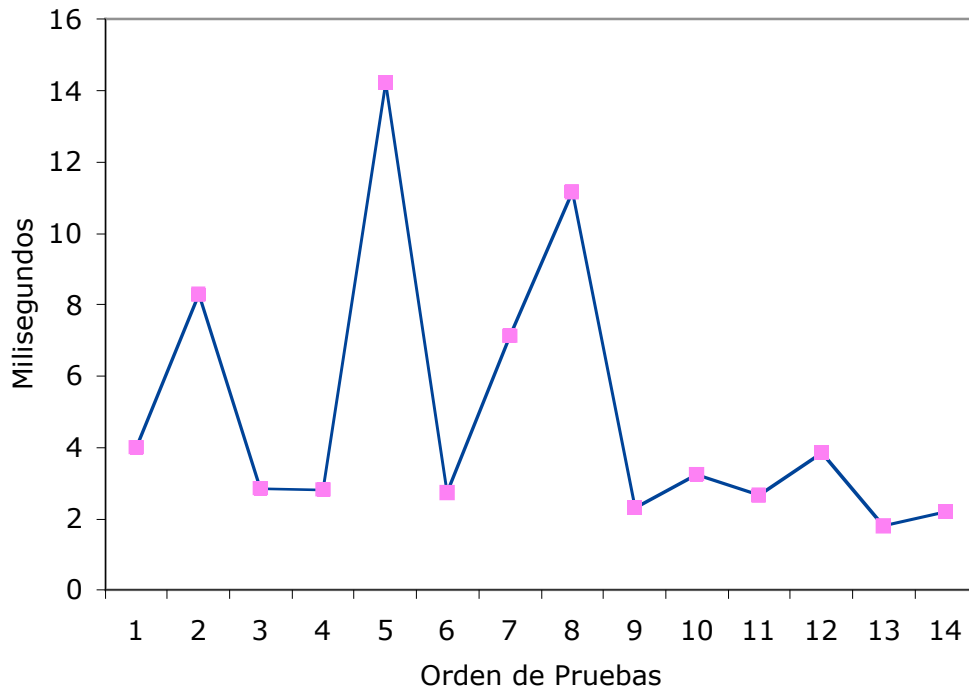


Figura 6.4 Retardo calculado por traceroute hasta el servidor

El promedio de estos valores en la figura 6.4 se encuentra en 4.9488 ms de retardo desde cualquier punto del campus hasta el servidor Wiener.

En las pruebas realizadas a puntos externos de recepción se encontraron retrasos mucho más significativos. Aunque en algunos de los casos existía una recepción satisfactoria, una prueba hecha a un nodo en el Campus Guadalajara del ITESM mostró los siguientes tiempos como se muestra en la tabla 6.1. En la misma tabla se indica los tiempos obtenidos correspondientes a la radiodifusora del Campus, cuya transmisión se hace a un Bit Rate de 32 Kbps.

C. Gda.	IP: 10.40.12.20	Min: 18.555 ms	Prom: 22.167 ms	Max: 28.43 ms
Difusora	IP: 131.178.34.74	Min: 1.863 ms	Prom: 2.763	Max: 6.348

Tabla 6.1 Tiempos de retraso a nodos externos y difusora del campus

Las gráficas obtenidas para el nodo externo indican el siguiente desempeño en segmentos TCP recibidos/segundo.

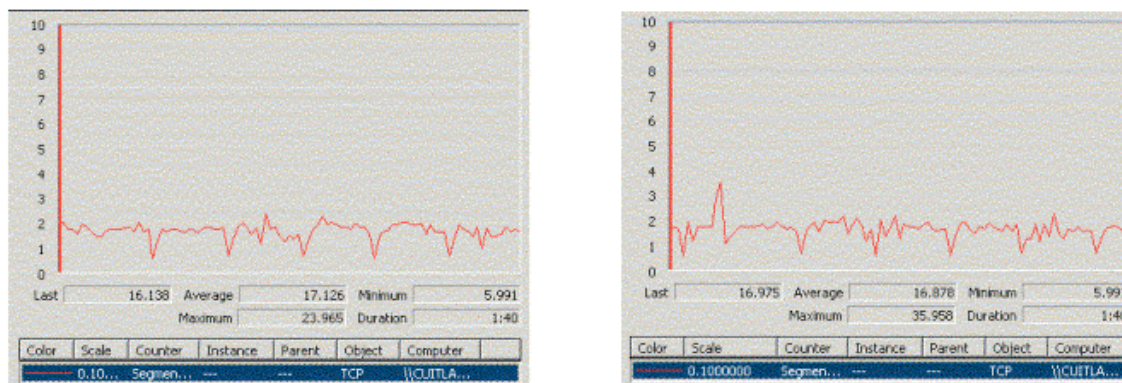


Figura 6.5 Segmentos TCP recibidos/segundo del nodo externo

Los valores mostrados en ambas gráficas de la figura 6.5 con una diferencia de tiempo de media hora en tomar la muestra de minuto y medio, no revelan una variación amplia de los valores. Lo que representa un flujo constante en general. Esto fue hecho bajo condiciones de poco tráfico en la red, es decir, una hora no pico.

Durante las transmisiones hechas en vivo, se obtuvo también información del tiempo en que el servidor tuvo a su máximo número de usuarios, con el fin de observar el ancho de banda consumido en una difusión de tipo *unicast*, así como también obtener retroalimentación de la percepción del usuario en cuanto a la calidad del audio transmitido. La siguiente tabla muestra datos de la difusión realizada, la cual se hizo bajo condiciones de alta actividad de tráfico y con un total de 5 usuarios (máximo) al momento de tomar estadísticas.

Dirección IP Usuario	Tasa de transmisión	Retraso promedio en (traceroute)	Problemas con audio	Tiempo de Recepción
131.178.154.183	160 Kbps	8.307 ms	No	60 min
131.178.48.244	160 Kbps	2.084 ms	No	73 min
131.178.134.140	160 Kbps	10.169 ms	No	16 min
131.178.113.239	160 Kbps	6.433 ms	No	28 min
148.233.91.136	160 Kbps	30.212 ms	Si	7 min

Tabla 6.2 Datos de la difusión con número máximo de usuarios

El ancho de banda total consumido por los usuarios para el enlace del servidor se encuentra aproximadamente en 872 Kbps, el cual disminuye o aumenta de acuerdo al número de usuarios que inicia o termina una sesión, afectando de manera directa el ancho de banda disponible para la salida del servidor y el desempeño de su procesador. La recepción de los usuarios no se vió afectada por el tráfico de la red, a excepción del último, cuya dirección IP corresponde a la red del Campus Estado de México. Siendo un nodo externo a la LANE examinada, muestra un retraso promedio significativamente mayor y la transmisión de audio se ve afectada por el enlace y las distintas redes que atraviesa. El usuario reportó retrasos en el audio de hasta 15 segundos, lo que hace difícil una recepción con estas características.

Para las transmisiones de material bajo demanda, se obtuvieron las siguientes gráficas con los diferentes métricas de TCP e IP. Para estas pruebas se tuvieron condiciones de hora pico en la red del campus. El material de audio empleado correspondió siempre a un mismo archivo de MP3 a 128 Kbps, con 3 minutos 38 segundos de duración y 3.3 Mbytes de tamaño.

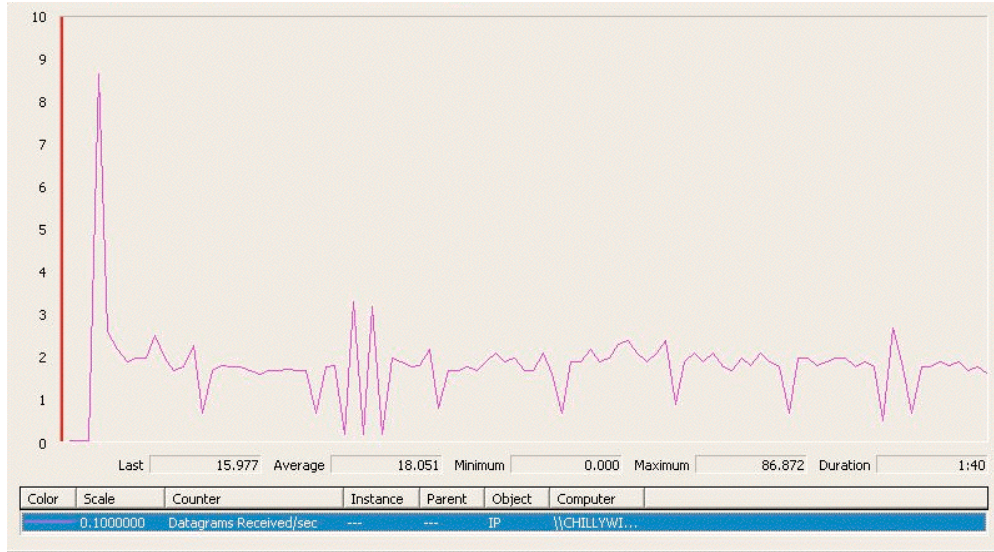


Figura 6.6 Datagramas recibidos/segundo durante recepción de audio

La figura 6.6 indica los datagramas recibidos/segundo. La toma de muestras comienza uno segundo antes del inicio de la transmisión. Debido a esto observamos un valor mínimo de cero. También se observa un pico al inicio de la transmisión, el cual se presenta en todas las pruebas hechas.

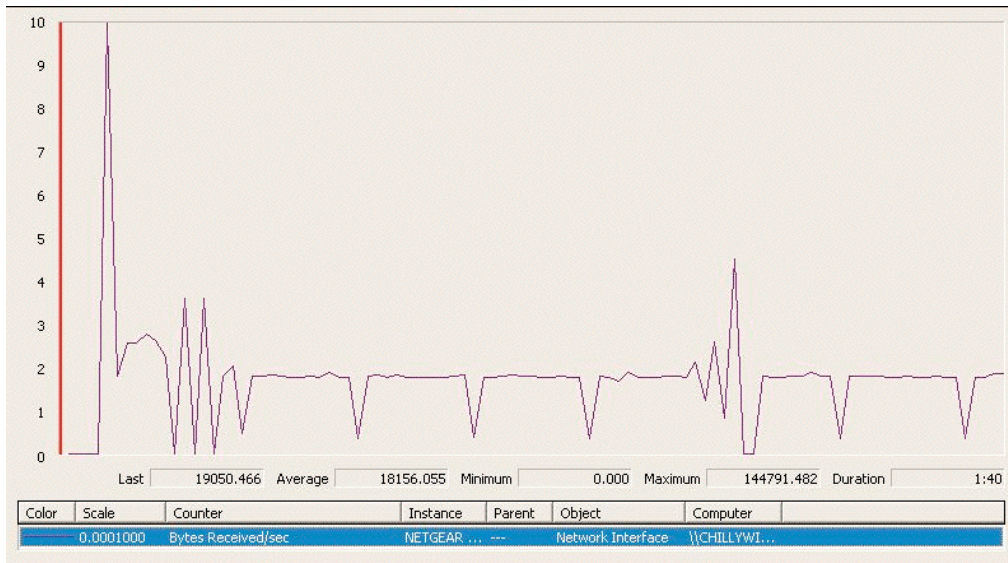


Figura 6.7 Bytes recibidos/segundo en la tarjeta de red

Para las muestras de Bytes recibidos/segundo, podemos encontrar en la figura 6.7 un promedio de 18156.055. Se observa la misma situación al inicio de la transmisión.

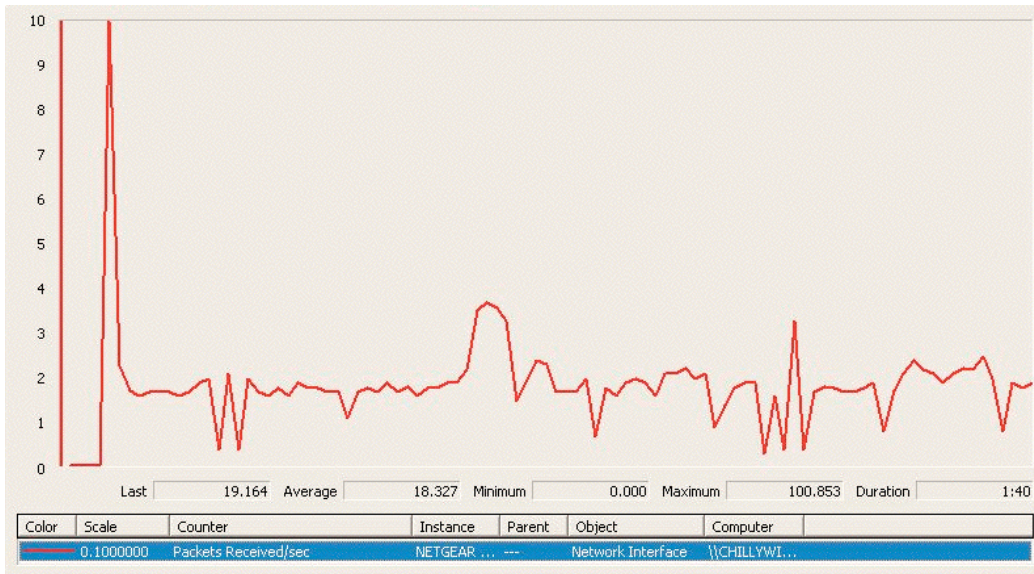


Figura 6.8 Paquetes recibidos/segundo

El promedio de paquetes recibidos por la tarjeta de red en la figura 6.8 es de 18.327 con un pico de 100.853 durante el periodo de muestra.

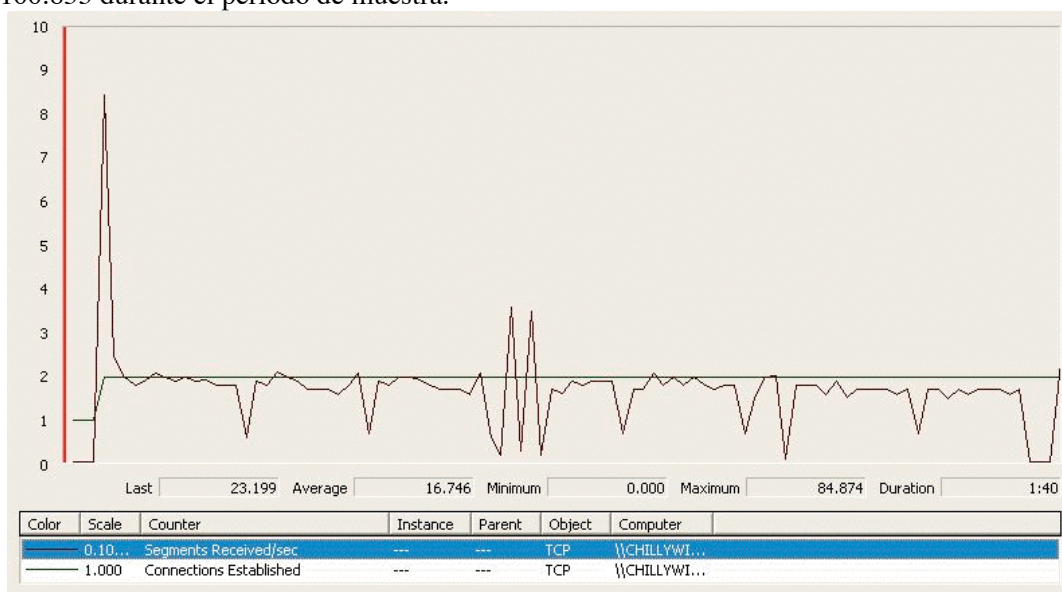


Figura 6.9. Sementos TCP recibidos/segundo

Los segmentos TCP recibidos/segundo contempla un promedio de 16.746 en la muestra señalada por la figura 6.9. Se indica también un contador de conexiones TCP establecidas durante la transmisión. Se observa que al inicio de esta se da de alta la conexión correspondiente al flujo de audio y coincide con el pico común en todas las pruebas. Este pico se debe a la cantidad de información enviada para iniciar la sesión incluyendo los primeros bits de datos de audio. Durante la prueba se encuentran variaciones que pueden considerarse no muy grandes, ya que se registran cercanas al promedio. En general, podemos observar que este valor promedio esta

dentro un rango común en todas las pruebas ya sea hora pico o no, claro, a excepción de las realizadas con puntos o servidores externos.

En cuanto a las transmisiones hechas entre las dos estaciones de trabajo Wiener y Gabor, como era de esperarse, no se registro ningún retraso en el flujo de audio. La salida del servidor se hizo a través de la interfase ATM de Wiener y la llegada por la misma correspondiente a la de la estación Gabor. El cliente utilizado para la recepción de audio en Gabor fue la aplicación Real Player propiedad de Real Networks. Dicha aplicación nos permite realizar estadísticas de la recepción. El resultado fue un flujo sin interrupciones ni pérdidas de paquetes. La solicitud del material de audio se hizo bajo demanda y con el mismo archivo empleado en las pruebas anteriores a 128 Kbps. Una muestra de las estadísticas se puede observar en la figura 6.10, 6.11 y 6.12

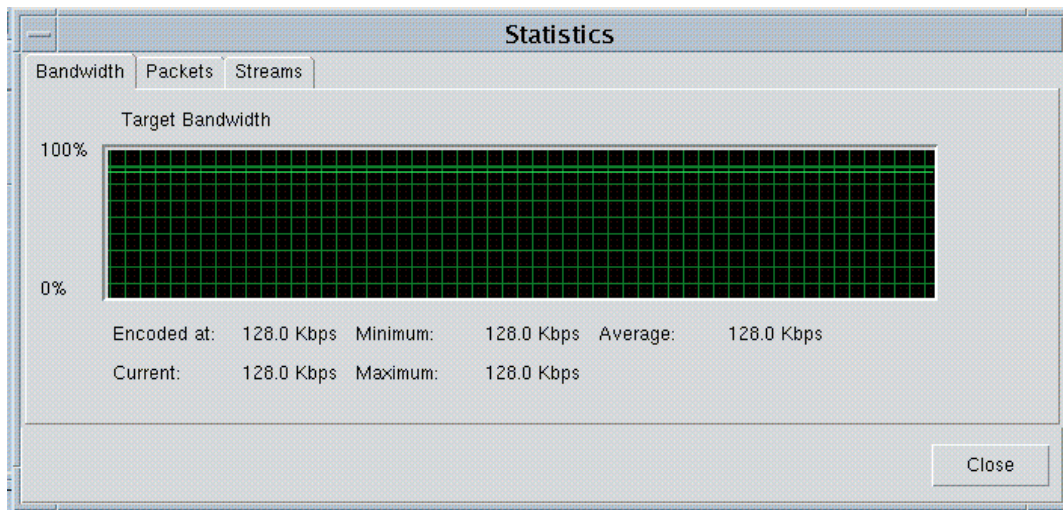


Figura 6.10 Ancho de banda observado del flujo de audio

En la figura 6.10 podemos ver el flujo de audio a 128 Kbps sin interrupciones.

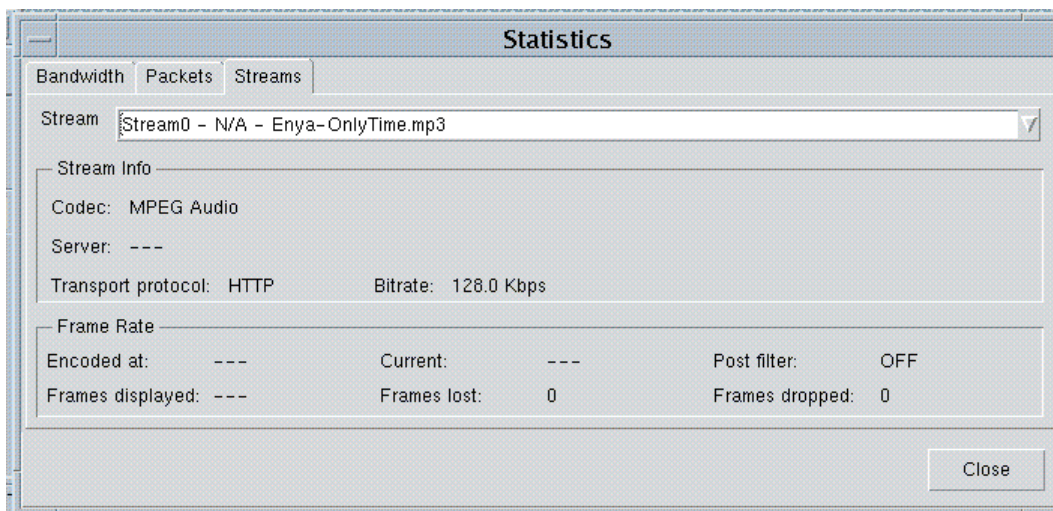


Figura 6.11 Datos generales del flujo de audio

En las figuras 6.11 y 6.12 se puede apreciar la ventana de la aplicación Real Player con información del audio recibido, estadísticas de paquetes, transmisión, etc. Se observa una transmisión sin problemas ni pérdidas.

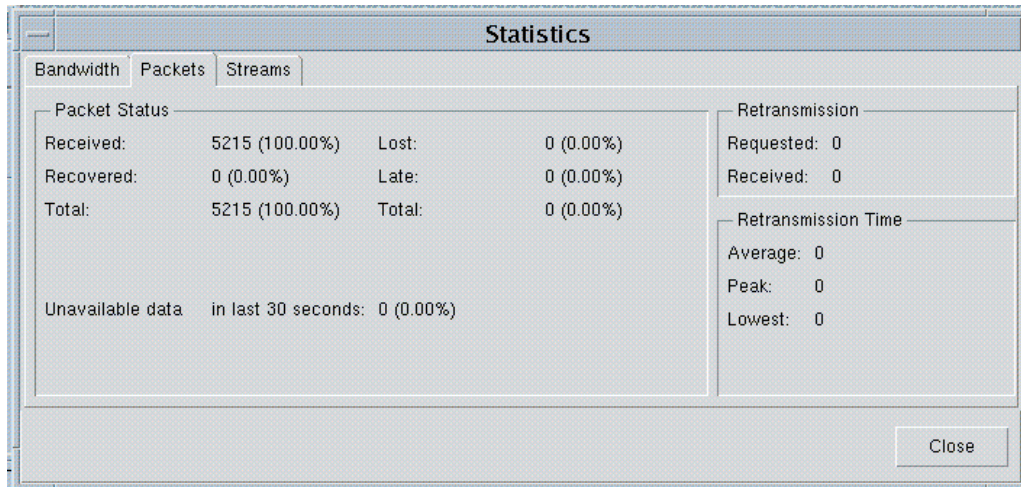


Figura 6.12 Estadísticas de transmisión. Aplicación Real Player

El retardo calculado entre las dos estaciones usando las dos interfases (ATM, Eth.) mediante las herramientas Ping y Traceroute arrojó los siguientes resultados en la tabla 6.2

Herramienta	Pruebas	Valor Máximo	Valor Mínimo	Valor Promedio
Ping	10	0 ms	0 ms	0 ms
Traceroute	4	0.532 ms	0.230 ms	0.3642 ms

Tabla 6.3 Retraso entre Wiener y Gabor

Estos resultados muestran que el retraso entre las 2 terminales es muy pequeño e incluso descartable considerando el tipo de interfase a través de la cual viaja el flujo de audio.

6.6 Cálculo de Entrega y Retraso del conmutador

El calculo de la Entrega que hace el conmutador en cierta cantidad de tiempo se realizo siguiendo el procedimiento que a continuación se muestra.

La Entrega (Throughput) puede definirse de la siguiente manera:

$$E = \frac{\sum_{n=1}^p PDU_n + PAD_n + T_n + LPDU + LPAD + T}{48} \text{ (celdas/seg.)},$$

donde PDU es el unidad de información del usuario en bytes antes de ser procesada por AAL5. Esta unidad puede definirse de la MTU (Maximum transmisión Unit) permitida dentro de la red en donde viajará. Por lo tanto si consideramos 20 bytes de encabezado TCP y 20 bytes de encabezado IP, podemos decir que la trama original antes de los encabezados se define como:

$$Frame = PDU \square 40(\text{bytes}),$$

y p lo definimos como el numero total de paquetes de longitud definida en el PDU y está dado por:

$$p = \text{int} \left\lfloor \frac{B}{Frame} \right\rfloor,$$

donde B es la tasa de transmisión utilizada o ancho de banda consumido. T es el *trailer* añadido por AAL5 a cada paquete que comprende 8 bytes. De la misma manera PAD es el *padding* agregado que va de 0 a 47 bytes y que se obtiene a partir de:

$$PAD = 48 \square \text{mod} \left\lfloor \frac{PDU + T}{48} \right\rfloor,$$

$LPDU$ y $LPAD$ que corresponden al último paquete resultado de la primera segmentación se calculan así:

$$LPDU = \text{mod} \left\lfloor \frac{B}{Frame} \right\rfloor + 40,$$

y

$$LPAD = 48 \square \text{mod} \left\lfloor \frac{LPDU + T}{48} \right\rfloor.$$

AL Obtener las celdas por segundo que pasa a través del conmutador podemos obtener el retraso generado mediante la siguiente fórmula[1]:

$$Delay = \frac{E}{S},$$

donde S es la capacidad de conmutación del switch y que es de 365,566 celdas/seg. Dadas las formulas, los valores asignados para el realizar los cálculos fueron los siguientes. El Tamaño PDU: 576, 1000, y 1500 (bytes). El Bit Rate: 128, 160, y 192 (Kbps)

Los valores para la tasa de transmisión son los usados en las pruebas. El tamaño del PDU se tomo en base a la MTU permitida en una LAN Ethernet que es el que corresponde a 1500 bytes. Los 576 bytes son la longitud por default en IP cuando no se especifica el tamaño al iniciar una conexión TCP. 1000 bytes son un valor intermedio que es cercano a los valores obtenidos durante las pruebas. Los resultados del cálculo se muestran en la tabla 6.3.

Tasa de transmisión	Tamaño Trama(bytes)	Retraso (sec)	Celdas/seg
128 Kbps	576	0.00106	388
128 Kbps	1000	0.00096	351
128 Kbps	1500	0.00096	351
160 Kbps	576	0.00132	486
160 Kbps	1000	0.00119	438
160 Kbps	1500	0.00120	439
192 Kbps	576	0.00159	582
192 Kbps	1000	0.00143	525
192 Kbps	1500	0.00144	527

Tabla 6.4 Retraso y Entrega del conmutador

De la misma manera, las gráficas 6.13 y 6.14 muestran el desempeño de la entrega y retraso del conmutador.

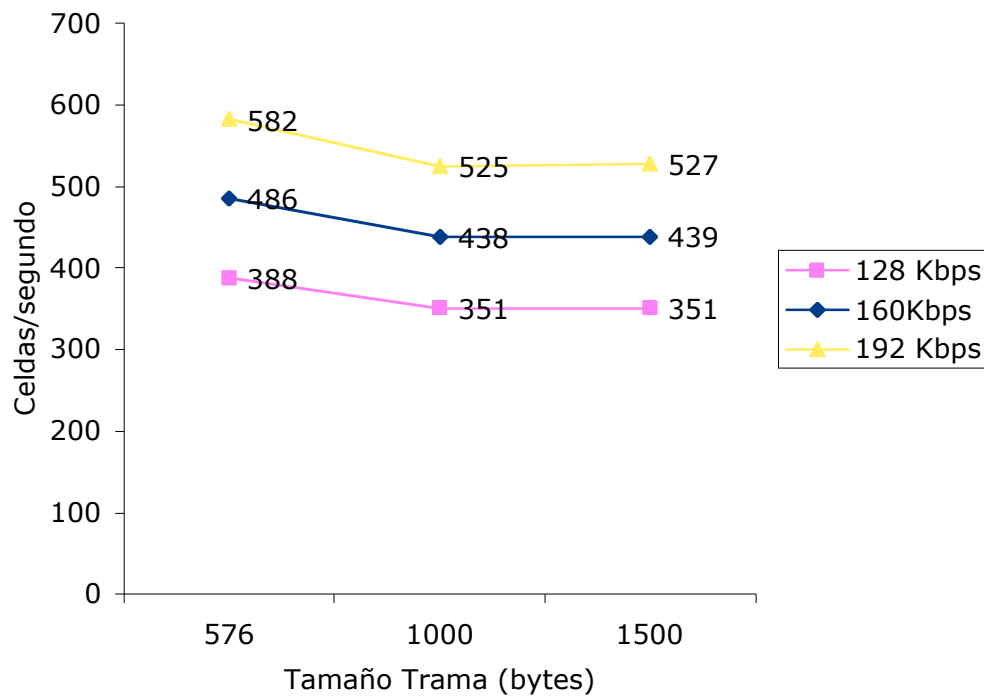


Figura 6.13 Entrega del conmutador en cada tasa de transmisión

Podemos observar que la carga disminuye al aumentar el tamaño de la trama. Esto provee un mejor desempeño dado que el costo por paquete de procesar un encabezado IP puede ser diferido con más bytes de información y ocurrir en una fragmentación menor.

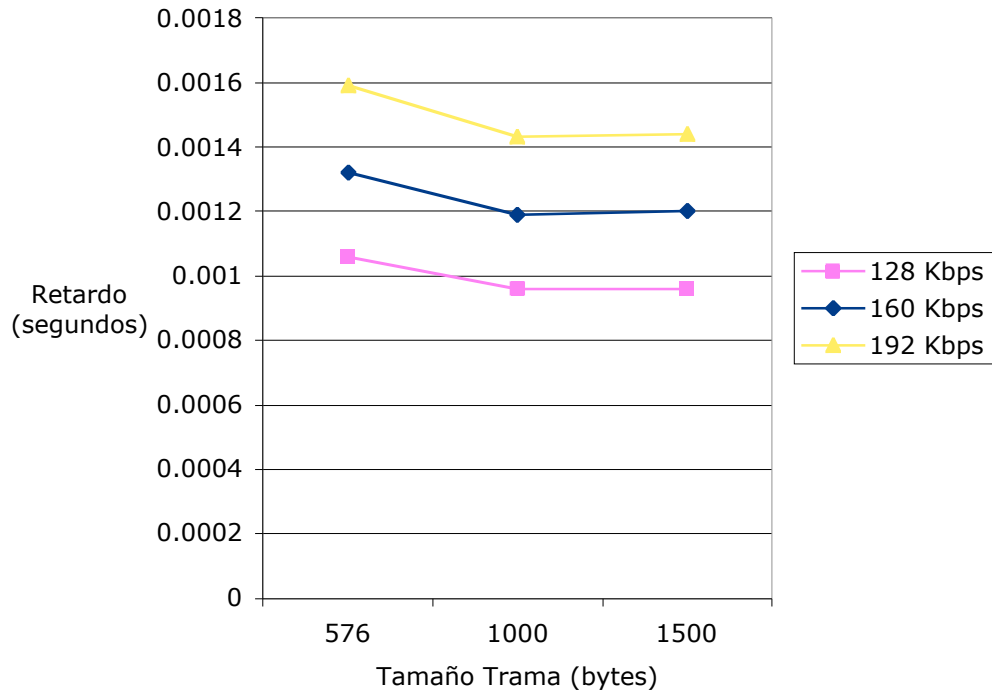


Figura 6.14 Retraso generado en el conmutador en cada tasa de transmisión

La latencia mostrada en la figura 6.13 se comporta de la misma manera por el tamaño de la trama que se refleja en un uso del ancho de banda más efectivo. Obviamente, este ultimo aumenta conforme se utiliza una resolución más alta o grado menor de compresión en la información de audio.

Capítulo 7

Conclusiones.

En la presente tesis se realizaron pruebas de tráfico transmitiendo flujos de audio a través de la red del campus, la cual está basada en un modelo *overlay* de IP sobre ATM designado por el foro ATM como *LAN Emulation*.

Las pruebas realizadas con el software de difusión de audio propiedad de Nullsoft, demostraron que la red del campus es capaz de soportar flujos (streaming) de audio de hasta 256 kbps sin problemas de interrupción o retraso perceptible durante horas de alta actividad de la red en cualquiera de las locaciones elegidas para recepción. Los problemas presentados con la transmisión fueron con algunos usuarios fuera de la red del campus localizados en otros estados. El objetivo de las pruebas fue observar el comportamiento de estas transmisiones mezcladas con el tráfico normal del campus.

Se hicieron pruebas de transmisión entre las dos estaciones de trabajo conectadas al conmutador ATM localizado en el Centro de Electrónica y Telecomunicaciones para comprobar la perfecta recepción del material de audio con la modalidad bajo demanda y registrar sus estadísticas.

Se calculó el retraso de la red y se registraron estadísticas mediante herramientas IP. El cálculo del retraso se mantuvo bastante cerca de lo obtenido en las pruebas a las diferentes tasas de transmisión utilizadas. Así como también en el cálculo de la entrega (throughput) y latencia generada del conmutador. Todo esto con la finalidad de obtener datos importantes para saber las capacidades de la red para proveer al usuario ciertas aplicaciones. La información obtenida de este experimento es representativa del desempeño de un servicio de difusión de audio en una red con las características de aquella que presenta el campus (LANE) y se considera válida dadas las circunstancias bajo las cuales se hicieron las transmisiones. Circunstancias que comprenden periodos de alta actividad dentro de la institución y significativo tráfico en ráfaga (bursty) en su red de datos.

De acuerdo a esto se observa que la red cuenta con el ancho de banda suficiente y el retraso mínimo para implementar servicios basados en distribución de audio con una calidad cercana a la que provee un disco compacto. El tipo de aplicación utilizada no consume demasiado ancho de banda y tampoco tiene requerimientos tan exigentes como lo sería una aplicación de transporte de voz en tiempo real o videoconferencia. Pero es importante analizar su comportamiento independiente ya que es parte esencial de una aplicación o servicio basado en video con el cual interactúa de manera síncrona.

La difusión de audio hecha basa su funcionamiento en el protocolo HTTP que es parte del protocolo TCP. Puede decirse que bajo este esquema, el rendimiento fue satisfactorio, aunque puede no ser eficiente ya que la transferencia de datos son de tipo de *unicast* y el ancho de banda disponible se ve afectado al incrementarse el número usuarios y por lo tanto, el número de flujos *unicast*. Una solución adecuada podría ser el uso de *multicast*, aunque esto también afecta otras áreas de la LAN emulada. En muchos casos, como el de entrega de video, el protocolo UDP resulta ser mejor que TCP aún considerando que este no da garantías de entrega. El uso de RTP es también una ayuda complementaria para el transporte de información usando aplicaciones multimedia por Internet en tiempo real.

LANE es un modelo aceptable para la integración de una red IP a un backbone basado en ATM, pero su escalabilidad se vuelve muy compleja. En el caso de una red tipo campus como esta, el ancho de banda se ve beneficiado, pero no se aprovechan al máximo las capacidades de ATM en cuanto a calidad de servicio para aplicaciones como lo sería el transporte de voz u otras aplicaciones con requerimientos de tiempo y sincronía.

Los resultados obtenidos pueden ser de utilidad para el uso de aplicaciones que involucren audio digital, incluso en aquellas que están basadas en compresión y transporte de video, especialmente al usar los estándares de ISO MPEG. Vemos también que se vislumbran más y mejores innovaciones en el área de la compresión de material de audio y video como lo es el desarrollo del estándar MPEG-4 y la liberación del formato MP3 Pro, el cual consigue tasas de compresión aún más altas, generando archivos de hasta la mitad de aquellos en formato MP3 original. Esto sin duda abrirá más posibilidades de integración multimedia de calidad a las redes de paquetes.

En cuanto a consideraciones futuras, podemos observar que en las técnicas desarrolladas para darle a IP las características deseadas para ser una plataforma de convergencia, la más prometedora parece ser la conmutación de etiquetas o llamada también por la IETF, MPLS. Es claro que IP se perfila como el ganador. Sin embargo, estos desarrollos no han hecho más que adaptar la idea original con la que fue diseñado ATM y contemplando los mismos aspectos como son la conmutación de alta velocidad, manejo de tráfico y sobre todo calidad de servicio.

Referencias.

- [1] León García, Alberto. Indra Widjaja. Communication Networks. Fundamental Concepts and Key Architectures. Mc GrawHill 2000.
- [2] Crowcroft, Jon. Handley, Mark. Wakeman, Ian. Internetworking Multimedia. Taylor & Francis 1999.
- [3] Barry G Haskell, Atul Puri, Arun N Netravali. Digital video: an introduction to MPEG-2. Chapman & Hall 1997.
- [4] Paul Izzo. Gigabit Networks. Standards and Schemes for Next Generation Networking. Wiley 2000.
- [5] Painter Ted, Spanias Andreas. Perceptual Coding of digital audio. IEEE Proceedings Vol. 88 No. 4 Abril 2000
- [6] Davis Pan. Tutorial on MPEG audio compression. IEEE Multimedia. Summer 1995. pp 60-74
- [7] Davis Yen Pan. Digital Audio Compression. Digital Technical Journal Vol. 5 No. 2, Spring 1993
- [8] José I. Gómez Quiñones. Análisis y Tutorial del codificador de audio MPEG 11172-3. ITESM Campus Monterrey, Julio 2000.
- [9] Berry Kercheval. TCP/IP over ATM: a no-nonsense internetworking guide. Prentice Hall series in computer networking and distributed systems. 1998.
- [10] Bo Li, Mounir Hamdi, Dongyi Jiang, Xi-Ren Cao, Y. Thomas Hou. QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges. IEEE Communications Magazine • April 2000.
- [11] <http://www.shoutcast.com/support/docs>
- [12] http://tyr.mty.itesm.mx/aservin/schem/red_jun_01.jpg

Apendice A. Registro de estadísticas.

Estadísticas de red
Marzo 11, 2002. 12:10 hrs.
Tipo de transmisión: Live Streaming
Fuente de audio: 131.178.48.230
DNAS: 131.178.107.89 (lane)
Lugar: Biblioteca
Dirección IP destino: **131.178.154.129**
Bit Rate: **128 Kbps**
Enlace: 10 Mbps
Paquetes recibidos en 1 min: 1062
No Multicast
Interrupciones/retraso: 0
Errores en Audio: 6

Routing tables 131.178.154.129

Internet:

Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	131.178.154.1	UGSc	2	0	en0	
localhost	localhost	UH	9	569	lo0	
131.178.154/24	link#2	UC	0	0	en0	
131.178.154.1	0:90:6d:e5:40:0	UHLW	3	0	en0	1185

ip:

148788 total packets received
0 bad header checksums
0 with size smaller than minimum
0 with data size < data length
0 with header length < data size
0 with data length < header length
0 with bad options
0 with incorrect version number
0 fragments received
0 fragments dropped (dup or out of space)
0 fragments dropped after timeout
0 packets reassembled ok
148788 packets for this host
0 packets for unknown/unsupported protocol
0 packets forwarded (0 packets fast forwarded)
0 packets not forwardable
0 packets received for unknown multicast group
0 redirects sent
137884 packets sent from this host
9 packets sent with fabricated ip header
0 output packets dropped due to no bufs, etc.
0 output packets discarded due to no route
0 output datagrams fragmented
0 fragments created
0 datagrams that can't be fragmented

icmp:

1 call to icmp_error
0 errors not generated 'cuz old message was icmp
Output histogram:
destination unreachable: 1
0 messages with bad code fields
0 messages < minimum length
0 bad checksums
0 messages with bad length

```

0 multicast echo requests ignored
0 multicast timestamp requests ignored
Input histogram:
    echo reply: 10
    destination unreachable: 2
    time exceeded: 6
0 message responses generated
ICMP address mask responses are disabled
igmp:
0 messages received
0 messages received with too few bytes
0 messages received with bad checksum
0 membership queries received
0 membership queries received with invalid field(s)
0 membership reports received
0 membership reports received with invalid field(s)
0 membership reports received for groups to which we belong
0 membership reports sent
tcp:
72207 packets sent
    749 data packets (56538 bytes)
    0 data packets (0 bytes) retransmitted
    0 resends initiated by MTU discovery
    3593 ack-only packets (2597 delayed)
    0 URG only packets
    0 window probe packets
    2208 window update packets
    65657 control packets
147640 packets received
    914 acks (for 56599 bytes)
    557 duplicate acks
    0 acks for unsent data
    15472 packets (17617253 bytes) received in-sequence
    51 completely duplicate packets (1498 bytes)
    0 old duplicate packets
    0 packets with some dup. data (0 bytes duped)
    374 out-of-order packets (405209 bytes)
    445 packets (445 bytes) of data after window
    445 window probes
    0 window update packets
    0 packets received after close
    0 discarded for bad checksums
    0 discarded for bad header offset fields
    0 discarded because packet too short
65576 connection requests
41 connection accepts
0 bad connection attempts
0 listen queue overflows
86 connections established (including accepts)
65614 connections closed (including 1 drop)
    2 connections updated cached RTT on close
    2 connections updated cached RTT variance on close
    0 connections updated cached ssthresh on close
65531 embryonic connections dropped
914 segments updated rtt (of 66446 attempts)
0 retransmit timeouts
    0 connections dropped by rexmit timeout
0 persist timeouts
    0 connections dropped by persist timeout
0 keepalive timeouts
    0 keepalive probes sent
    0 connections dropped by keepalive
31 correct ACK header predictions

```

```
14735 correct data packet header predictions
udp:
  1130 datagrams received
  0 with incomplete header
  0 with bad data length field
  0 with bad checksum
  3 dropped due to no socket
  976 broadcast/multicast datagrams dropped due to no socket
  0 dropped due to full socket buffers
  0 not for hashed pcb
  151 delivered
  141 datagrams output

PING 131.178.107.89 (131.178.107.89): 56 data bytes
64 bytes from 131.178.107.89: icmp_seq=0 ttl=253 time=4.83 ms
64 bytes from 131.178.107.89: icmp_seq=1 ttl=253 time=6.145 ms
64 bytes from 131.178.107.89: icmp_seq=2 ttl=253 time=4.687 ms
64 bytes from 131.178.107.89: icmp_seq=3 ttl=253 time=2.885 ms
64 bytes from 131.178.107.89: icmp_seq=4 ttl=253 time=3.194 ms
64 bytes from 131.178.107.89: icmp_seq=5 ttl=253 time=2.837 ms
64 bytes from 131.178.107.89: icmp_seq=6 ttl=253 time=6.835 ms
64 bytes from 131.178.107.89: icmp_seq=7 ttl=253 time=4.132 ms
64 bytes from 131.178.107.89: icmp_seq=8 ttl=253 time=3.384 ms
64 bytes from 131.178.107.89: icmp_seq=9 ttl=253 time=3.194 ms

--- 131.178.107.89 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 2.837/4.212/6.835 ms

traceroute to 131.178.107.89 (131.178.107.89), 30 hops max, 40 byte packets
 1 131.178.154.1 (131.178.154.1)  2.973 ms  1.844 ms  1.861 ms
 2 gwipv6.mty.itesm.mx (131.178.100.8)  3.891 ms  4.439 ms  6.106 ms
 3 wiener.mty.itesm.mx (131.178.26.43)  4.004 ms * 8.291 ms
```

Estadísticas de red

Marzo 11, 2002. 17:00 hrs.
Tipo de transmisión: Live Streaming
Fuente de audio: 131.178.48.230
DNAS: 131.178.107.89 (lane)
Lugar: Aulas V
Dirección IP destino: **131.178.146.105**
Bit Rate: **128 Kbps**
Enlace: 10 Mbps
Paquetes recibidos en 1 min: 1035
No Multicast
Interrupciones/retraso: 0
Errores en Audio: 0

Usuario externo: Cuitláhuac López
Hora: 17:07 hrs
Tiempo de recepción: 17 min.
Dirección IP: 10.40.12.20
Lugar: Campus Guadalajara
Interrupciones/retraso perceptible: 0
Errores en audio: 0

Routing tables 131.178.146.105

Internet:

Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	131.178.146.1	UGSc	2	0	en0	
localhost	localhost	UH	9	650	lo0	
131.178.146/24	link#2	UC	0	0	en0	
131.178.146.1	0:90:6d:e5:40:0	UHLW	2	0	en0	1171

Statistics

ip:

4126 total packets received
0 bad header checksums
0 with size smaller than minimum
0 with data size < data length
0 with header length < data size
0 with data length < header length
0 with bad options
0 with incorrect version number
0 fragments received
0 fragments dropped (dup or out of space)
0 fragments dropped after timeout
0 packets reassembled ok
4125 packets for this host
0 packets for unknown/unsupported protocol
0 packets forwarded (0 packets fast forwarded)
1 packet not forwardable
0 packets received for unknown multicast group
0 redirects sent
1895 packets sent from this host
0 packets sent with fabricated ip header
0 output packets dropped due to no bufs, etc.
0 output packets discarded due to no route
0 output datagrams fragmented
0 fragments created
0 datagrams that can't be fragmented

icmp:

0 calls to icmp_error
0 errors not generated 'cuz old message was icmp
0 messages with bad code fields
0 messages < minimum length

```

0 bad checksums
0 messages with bad length
0 multicast echo requests ignored
0 multicast timestamp requests ignored
0 message responses generated
ICMP address mask responses are disabled
igmp:
0 messages received
0 messages received with too few bytes
0 messages received with bad checksum
0 membership queries received
0 membership queries received with invalid field(s)
0 membership reports received
0 membership reports received with invalid field(s)
0 membership reports received for groups to which we belong
0 membership reports sent
tcp:
1818 packets sent
    611 data packets (43071 bytes)
    0 data packets (0 bytes) retransmitted
    0 resends initiated by MTU discovery
    740 ack-only packets (537 delayed)
    0 URG only packets
    0 window probe packets
    419 window update packets
    48 control packets
3655 packets received
    675 acks (for 43058 bytes)
    53 duplicate acks
    0 acks for unsend data
    3397 packets (3253108 bytes) received in-sequence
    1 completely duplicate packet (1 byte)
    0 old duplicate packets
    0 packets with some dup. data (0 bytes duped)
    122 out-of-order packets (131232 bytes)
    30 packets (30 bytes) of data after window
    30 window probes
    0 window update packets
    0 packets received after close
    0 discarded for bad checksums
    0 discarded for bad header offset fields
    0 discarded because packet too short
18 connection requests
17 connection accepts
0 bad connection attempts
0 listen queue overflows
35 connections established (including accepts)
30 connections closed (including 0 drops)
    0 connections updated cached RTT on close
    0 connections updated cached RTT variance on close
    0 connections updated cached ssthresh on close
0 embryonic connections dropped
675 segments updated rtt (of 676 attempts)
0 retransmit timeouts
    0 connections dropped by rexmit timeout
0 persist timeouts
    0 connections dropped by persist timeout
0 keepalive timeouts
    0 keepalive probes sent
    0 connections dropped by keepalive
21 correct ACK header predictions
2802 correct data packet header predictions
udp:

```

```
470 datagrams received
0 with incomplete header
0 with bad data length field
0 with bad checksum
0 dropped due to no socket
389 broadcast/multicast datagrams dropped due to no socket
0 dropped due to full socket buffers
0 not for hashed pcb
81 delivered
77 datagrams output
```

Ping has started ...

```
PING 131.178.107.89 (131.178.107.89): 56 data bytes
64 bytes from 131.178.107.89: icmp_seq=0 ttl=253 time=4.757 ms
64 bytes from 131.178.107.89: icmp_seq=1 ttl=253 time=2.934 ms
64 bytes from 131.178.107.89: icmp_seq=2 ttl=253 time=2.326 ms
64 bytes from 131.178.107.89: icmp_seq=3 ttl=253 time=2.305 ms
64 bytes from 131.178.107.89: icmp_seq=4 ttl=253 time=3.265 ms
64 bytes from 131.178.107.89: icmp_seq=5 ttl=253 time=2.642 ms
64 bytes from 131.178.107.89: icmp_seq=6 ttl=253 time=6.418 ms
64 bytes from 131.178.107.89: icmp_seq=7 ttl=253 time=2.687 ms
64 bytes from 131.178.107.89: icmp_seq=8 ttl=253 time=2.757 ms
64 bytes from 131.178.107.89: icmp_seq=9 ttl=253 time=2.493 ms
```

```
--- 131.178.107.89 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 2.305/3.258/6.418 ms
```

Traceroute has started ...

```
traceroute to 131.178.107.89 (131.178.107.89), 30 hops max, 40 byte packets
 1 131.178.146.1 (131.178.146.1) 9.525 ms 1.717 ms 1.87 ms
 2 gwipv6.mty.itesm.mx (131.178.100.8) 5.299 ms 5.763 ms 5.27 ms
 3 wiener.mty.itesm.mx (131.178.26.43) 2.846 ms * 2.819 ms
```

Estadísticas Usuario externo (10.40.12.20)

```
traceroute to 10.40.12.20 (10.40.12.20), 30 hops max, 40 byte packets
 1 131.178.146.1 (131.178.146.1) 2.335 ms 2.033 ms 2.022 ms
 2 gwc552.mty.itesm.mx (131.178.101.12) 2.451 ms 4.138 ms 2.548 ms
 3 gwmtypvnl.mty.itesm.mx (131.178.69.117) 2.811 ms 4.236 ms 2.446 ms
 4 10.0.8.93 (10.0.8.93) 6.74 ms 7.552 ms 4.855 ms
 5 10.0.8.45 (10.0.8.45) 16.488 ms 30.193 ms 19.463 ms
 6 10.0.8.46 (10.0.8.46) 17.691 ms 19.212 ms 17.351 ms
```

Ping has started ...

```
PING 10.40.12.20 (10.40.12.20): 56 data bytes
64 bytes from 10.40.12.20: icmp_seq=0 ttl=121 time=23.145 ms
64 bytes from 10.40.12.20: icmp_seq=1 ttl=121 time=19.95 ms
64 bytes from 10.40.12.20: icmp_seq=2 ttl=121 time=18.555 ms
64 bytes from 10.40.12.20: icmp_seq=3 ttl=121 time=18.587 ms
64 bytes from 10.40.12.20: icmp_seq=4 ttl=121 time=28.43 ms
64 bytes from 10.40.12.20: icmp_seq=5 ttl=121 time=24.638 ms
64 bytes from 10.40.12.20: icmp_seq=6 ttl=121 time=20.598 ms
64 bytes from 10.40.12.20: icmp_seq=7 ttl=121 time=19.886 ms
64 bytes from 10.40.12.20: icmp_seq=8 ttl=121 time=27.494 ms
64 bytes from 10.40.12.20: icmp_seq=9 ttl=121 time=20.393 ms
```

```
--- 10.40.12.20 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 18.555/22.167/28.43 ms
```


Estadísticas Radiodifusora Frecuencia Tec 94.1

18:00 hrs
Bit Rate: 32 Kbps
Interrupciones/errores audio: 0
Calidad radio: radio
Dirección IP: 131.178.34.74
Puerto: 8000

Ping has started ...

```
PING frecuenciatec.mty.itesm.mx (131.178.34.74): 56 data bytes
64 bytes from 131.178.34.74: icmp_seq=0 ttl=253 time=2.963 ms
64 bytes from 131.178.34.74: icmp_seq=1 ttl=253 time=2.346 ms
64 bytes from 131.178.34.74: icmp_seq=2 ttl=253 time=1.888 ms
64 bytes from 131.178.34.74: icmp_seq=3 ttl=253 time=1.863 ms
64 bytes from 131.178.34.74: icmp_seq=4 ttl=253 time=6.348 ms
64 bytes from 131.178.34.74: icmp_seq=5 ttl=253 time=1.884 ms
64 bytes from 131.178.34.74: icmp_seq=6 ttl=253 time=3.255 ms
64 bytes from 131.178.34.74: icmp_seq=7 ttl=253 time=2.338 ms
64 bytes from 131.178.34.74: icmp_seq=8 ttl=253 time=2.342 ms
64 bytes from 131.178.34.74: icmp_seq=9 ttl=253 time=2.407 ms
```

```
--- frecuenciatec.mty.itesm.mx ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 1.863/2.763/6.348 ms
```

Traceroute has started ...

```
traceroute to frecuenciatec.mty.itesm.mx (131.178.34.74), 30 hops max, 40 byte
packets
 1 131.178.146.1 (131.178.146.1) 3.097 ms 5.127 ms 3.349 ms
 2 gwcronos.mty.itesm.mx (131.178.101.1) 1.598 ms 2.413 ms 1.534 ms
 3 frecuenciatec.mty.itesm.mx (131.178.34.74) 3.052 ms 2.04 ms 1.951 ms
```

Estadísticas de red

Marzo 12, 2002. 11:15 hrs.
Tipo de transmisión: Live Streaming
Fuente de audio: 131.178.48.230
DNAS: 131.178.107.89 (lane)
Lugar: Biblioteca 3er. piso
Dirección IP destino: **131.178.154.183**
Bit Rate: **160 Kbps**
Enlace: 10 Mbps
Paquetes recibidos en 1 min: 1023
No Multicast
Interrupciones/retraso perceptible: 0
Errores en Audio: 0

Usuario externo

Hora: 12:10 hrs
Tiempo de recepción: 6 min.
Dirección IP: 148.233.91.136
Lugar: Campus Estado de Mexico
Interrupciones/retraso: constante
Errores en audio: 0

Routing tables 131.178.154.183

Internet:

Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	131.178.154.1	UGSc	2	0	en0	
localhost	localhost	UH	9	597	lo0	
131.178.154/24	link#2	UC	0	0	en0	
131.178.154.1	0:90:6d:e5:40:0	UHLW	3	0	en0	1154

ip:

5101 total packets received
0 bad header checksums
0 with size smaller than minimum
0 with data size < data length
0 with header length < data size
0 with data length < header length
0 with bad options
0 with incorrect version number
0 fragments received
0 fragments dropped (dup or out of space)
0 fragments dropped after timeout
0 packets reassembled ok
5101 packets for this host
0 packets for unknown/unsupported protocol
0 packets forwarded (0 packets fast forwarded)
0 packets not forwardable
0 packets received for unknown multicast group
0 redirects sent
2320 packets sent from this host
0 packets sent with fabricated ip header
0 output packets dropped due to no bufs, etc.
0 output packets discarded due to no route
0 output datagrams fragmented
0 fragments created
0 datagrams that can't be fragmented

icmp:

0 calls to icmp_error
0 errors not generated 'cuz old message was icmp
0 messages with bad code fields
0 messages < minimum length
0 bad checksums

```

0 messages with bad length
0 multicast echo requests ignored
0 multicast timestamp requests ignored
0 message responses generated
ICMP address mask responses are disabled
igmp:
0 messages received
0 messages received with too few bytes
0 messages received with bad checksum
0 membership queries received
0 membership queries received with invalid field(s)
0 membership reports received
0 membership reports received with invalid field(s)
0 membership reports received for groups to which we belong
0 membership reports sent
tcp:
2242 packets sent
    557 data packets (39199 bytes)
    0 data packets (0 bytes) retransmitted
    0 resends initiated by MTU discovery
    984 ack-only packets (762 delayed)
    0 URG only packets
    0 window probe packets
    653 window update packets
    48 control packets
4943 packets received
    621 acks (for 39186 bytes)
    29 duplicate acks
    0 acks for unsend data
    4667 packets (5105124 bytes) received in-sequence
    0 completely duplicate packets (0 bytes)
    0 old duplicate packets
    0 packets with some dup. data (0 bytes duped)
    164 out-of-order packets (184178 bytes)
    8 packets (8 bytes) of data after window
    8 window probes
    0 window update packets
    0 packets received after close
    0 discarded for bad checksums
    0 discarded for bad header offset fields
    0 discarded because packet too short
18 connection requests
17 connection accepts
0 bad connection attempts
0 listen queue overflows
35 connections established (including accepts)
30 connections closed (including 0 drops)
    0 connections updated cached RTT on close
    0 connections updated cached RTT variance on close
    0 connections updated cached ssthresh on close
0 embryonic connections dropped
621 segments updated rtt (of 622 attempts)
0 retransmit timeouts
    0 connections dropped by rexmit timeout
0 persist timeouts
    0 connections dropped by persist timeout
0 keepalive timeouts
    0 keepalive probes sent
    0 connections dropped by keepalive
21 correct ACK header predictions
4127 correct data packet header predictions
udp:
158 datagrams received

```

```
0 with incomplete header
0 with bad data length field
0 with bad checksum
0 dropped due to no socket
78 broadcast/multicast datagrams dropped due to no socket
0 dropped due to full socket buffers
0 not for hashed pcb
80 delivered
78 datagrams output
Ping has started ...
```

```
PING 131.178.107.89 (131.178.107.89): 56 data bytes
64 bytes from 131.178.107.89: icmp_seq=0 ttl=253 time=3.209 ms
64 bytes from 131.178.107.89: icmp_seq=1 ttl=253 time=3.336 ms
64 bytes from 131.178.107.89: icmp_seq=2 ttl=253 time=5.496 ms
64 bytes from 131.178.107.89: icmp_seq=3 ttl=253 time=2.652 ms
64 bytes from 131.178.107.89: icmp_seq=4 ttl=253 time=2.515 ms
64 bytes from 131.178.107.89: icmp_seq=5 ttl=253 time=2.763 ms
64 bytes from 131.178.107.89: icmp_seq=6 ttl=253 time=6.859 ms
64 bytes from 131.178.107.89: icmp_seq=7 ttl=253 time=4.347 ms
64 bytes from 131.178.107.89: icmp_seq=8 ttl=253 time=11.789 ms
64 bytes from 131.178.107.89: icmp_seq=9 ttl=253 time=2.648 ms
```

```
--- 131.178.107.89 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 2.515/4.561/11.789 ms
```

```
Traceroute has started ...
```

```
traceroute to 131.178.107.89 (131.178.107.89), 30 hops max, 40 byte packets
 1 131.178.154.1 (131.178.154.1) 2.689 ms 2.201 ms 1.885 ms
 2 gwipv6.mty.itesm.mx (131.178.100.8) 6.991 ms 15.733 ms 4 ms
 3 wiener.mty.itesm.mx (131.178.26.43) 14.229 ms * 2.732 ms
```

Active Internet connections

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp	0	0	pto-154-183.mty..49153	msgr-ns14.msgr.h.1863	ESTABLISHED
tcp	33304	0	pto-154-183.mty..49152	wiener-atm.mty.i.8000	ESTABLISHED
tcp	0	0	localhost.768	localhost.832	ESTABLISHED
tcp	0	0	localhost.832	localhost.768	ESTABLISHED
tcp	0	0	localhost.768	localhost.779	ESTABLISHED
tcp	0	0	localhost.779	localhost.768	ESTABLISHED
udp	0	0	localhost.49156	localhost.807	

Estadísticas de red.

Marzo 13, 2002. 18:00 hrs
Tipo de transmisión: Live Streaming
Fuente de audio: 131.178.48.230
DNAS: 131.178.107.89 (lane)
Lugar: CETEC Torre Norte 1er. piso
Dirección IP destino: **131.178.113.241**
Bit Rate: **160 Kbps**
Enlace: 10 Mbps
Paquetes recibidos en 1 min: 1013
No Multicast
Interrupciones/retraso perceptible: 0
Errores en Audio: 0

Usuarios conectados: variable (5max)
Ancho de banda consumido aproximado: 160 Kbps * 5 = 800 Kbps
Problemas de recepción: 0

Routing tables 131.178.113.241

Internet:

Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	131.178.113.1	UGSc	3	0	en0	
localhost	localhost	UH	9	666	lo0	
131.178.113/24	link#2	UC	0	0	en0	
131.178.113.1	0:50:3e:72:6c:0	UHLW	4	0	en0	718

ip:
19947 total packets received
0 bad header checksums
0 with size smaller than minimum
0 with data size < data length
0 with header length < data size
0 with data length < header length
0 with bad options
0 with incorrect version number
0 fragments received
0 fragments dropped (dup or out of space)
0 fragments dropped after timeout
0 packets reassembled ok
19947 packets for this host
0 packets for unknown/unsupported protocol
0 packets forwarded (0 packets fast forwarded)
0 packets not forwardable
0 packets received for unknown multicast group
0 redirects sent
7645 packets sent from this host
0 packets sent with fabricated ip header
0 output packets dropped due to no bufs, etc.
1 output packet discarded due to no route
0 output datagrams fragmented
0 fragments created
0 datagrams that can't be fragmented

icmp:
1 call to icmp_error
0 errors not generated 'cuz old message was icmp
Output histogram:
destination unreachable: 1
0 messages with bad code fields
0 messages < minimum length
0 bad checksums
0 messages with bad length
0 multicast echo requests ignored

```

0 multicast timestamp requests ignored
0 message responses generated
ICMP address mask responses are disabled
igmp:
0 messages received
0 messages received with too few bytes
0 messages received with bad checksum
0 membership queries received
0 membership queries received with invalid field(s)
0 membership reports received
0 membership reports received with invalid field(s)
0 membership reports received for groups to which we belong
0 membership reports sent
tcp:
7566 packets sent
    645 data packets (49067 bytes)
    0 data packets (0 bytes) retransmitted
    0 resends initiated by MTU discovery
    3840 ack-only packets (3298 delayed)
    0 URG only packets
    0 window probe packets
    3001 window update packets
    80 control packets
19609 packets received
    740 acks (for 49069 bytes)
    35 duplicate acks
    0 acks for unsent data
    18997 packets (23866281 bytes) received in-sequence
    0 completely duplicate packets (0 bytes)
    0 old duplicate packets
    0 packets with some dup. data (0 bytes duped)
    450 out-of-order packets (568292 bytes)
    10 packets (10 bytes) of data after window
    10 window probes
    0 window update packets
    0 packets received after close
    0 discarded for bad checksums
    0 discarded for bad header offset fields
    0 discarded because packet too short
34 connection requests
18 connection accepts
0 bad connection attempts
0 listen queue overflows
51 connections established (including accepts)
63 connections closed (including 2 drops)
    0 connections updated cached RTT on close
    0 connections updated cached RTT variance on close
    0 connections updated cached ssthresh on close
0 embryonic connections dropped
740 segments updated rtt (of 742 attempts)
0 retransmit timeouts
    0 connections dropped by rexmit timeout
0 persist timeouts
    0 connections dropped by persist timeout
0 keepalive timeouts
    0 keepalive probes sent
    0 connections dropped by keepalive
20 correct ACK header predictions
18369 correct data packet header predictions
udp:
338 datagrams received
0 with incomplete header
0 with bad data length field

```

```
0 with bad checksum
1 dropped due to no socket
254 broadcast/multicast datagrams dropped due to no socket
0 dropped due to full socket buffers
0 not for hashed pcb
83 delivered
78 datagrams output
Ping has started ...
```

```
PING 131.178.107.89 (131.178.107.89): 56 data bytes
64 bytes from 131.178.107.89: icmp_seq=0 ttl=253 time=13.175 ms
64 bytes from 131.178.107.89: icmp_seq=1 ttl=253 time=3.044 ms
64 bytes from 131.178.107.89: icmp_seq=2 ttl=253 time=3.144 ms
64 bytes from 131.178.107.89: icmp_seq=3 ttl=253 time=4.658 ms
64 bytes from 131.178.107.89: icmp_seq=4 ttl=253 time=3.43 ms
64 bytes from 131.178.107.89: icmp_seq=5 ttl=253 time=2.831 ms
64 bytes from 131.178.107.89: icmp_seq=6 ttl=253 time=6.486 ms
64 bytes from 131.178.107.89: icmp_seq=7 ttl=253 time=10.234 ms
64 bytes from 131.178.107.89: icmp_seq=8 ttl=253 time=12.507 ms
64 bytes from 131.178.107.89: icmp_seq=9 ttl=253 time=4.828 ms
```

```
--- 131.178.107.89 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 2.831/6.433/13.175 ms
```

```
Traceroute has started ...
```

```
traceroute to 131.178.107.89 (131.178.107.89), 30 hops max, 40 byte packets
 1 131.178.113.1 (131.178.113.1)  3.945 ms  5.711 ms  3.963 ms
 2 131.178.8.3 (131.178.8.3)  3.744 ms  5.403 ms  6.339 ms
 3 gwipv6.mty.itesm.mx (131.178.100.8)  6.833 ms  8.771 ms  4.781 ms
 4 wiener.mty.itesm.mx (131.178.26.43)  7.13 ms * 11.155 ms
```

```
PING wiener.mty.itesm.mx (131.178.26.43): 56 data bytes
64 bytes from 131.178.26.43: icmp_seq=0 ttl=253 time=3.32 ms
64 bytes from 131.178.26.43: icmp_seq=1 ttl=253 time=6.749 ms
64 bytes from 131.178.26.43: icmp_seq=2 ttl=253 time=2.404 ms
64 bytes from 131.178.26.43: icmp_seq=3 ttl=253 time=2.623 ms
64 bytes from 131.178.26.43: icmp_seq=4 ttl=253 time=9.003 ms
64 bytes from 131.178.26.43: icmp_seq=5 ttl=253 time=2.899 ms
64 bytes from 131.178.26.43: icmp_seq=6 ttl=253 time=2.922 ms
64 bytes from 131.178.26.43: icmp_seq=7 ttl=253 time=2.022 ms
64 bytes from 131.178.26.43: icmp_seq=8 ttl=253 time=2.981 ms
64 bytes from 131.178.26.43: icmp_seq=9 ttl=253 time=2.431 ms
```

```
--- wiener.mty.itesm.mx ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 2.022/3.735/9.003 ms
```

```
Traceroute has started ...
```

```
traceroute to wiener.mty.itesm.mx (131.178.26.43), 30 hops max, 40 byte packets
 1 131.178.113.1 (131.178.113.1)  6.177 ms  3.554 ms  3.154 ms
 2 gwcronos.mty.itesm.mx (131.178.101.1)  2.439 ms  5.325 ms  2.114 ms
 3 wiener.mty.itesm.mx (131.178.26.43)  2.313 ms * 3.254 ms
```

Estadísticas de red
Marzo 14, 2002. 12:00 hrs.
Tipo de transmisión: Live Streaming
Fuente de audio: 131.178.48.230
DNAS: 131.178.107.89 (lane)
Lugar: CEDES Planta Baja
Dirección IP destino: **131.178.115.65**
Bit Rate: **192 Kbps**
Enlace: 10 Mbps
Paquetes recibidos en 1 min: 1302
No Multicast
Interrupciones/retraso perceptible: 0
Errores en Audio: 0

Usuarios conectados: variable

Routing tables 131.178.115.65

Internet:

Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	131.178.115.1	UGSc	4	0	en0	
localhost	localhost	UH	9	674	lo0	
131.178.115/24	link#2	UC	0	0	en0	
131.178.115.1	0:90:6d:e5:40:0	UHLW	5	0	en0	1125

ip:

6629 total packets received
0 bad header checksums
0 with size smaller than minimum
0 with data size < data length
0 with header length < data size
0 with data length < header length
0 with bad options
0 with incorrect version number
0 fragments received
0 fragments dropped (dup or out of space)
0 fragments dropped after timeout
0 packets reassembled ok
6629 packets for this host
0 packets for unknown/unsupported protocol
0 packets forwarded (0 packets fast forwarded)
0 packets not forwardable
0 packets received for unknown multicast group
0 redirects sent
3422 packets sent from this host
0 packets sent with fabricated ip header
0 output packets dropped due to no bufs, etc.
0 output packets discarded due to no route
0 output datagrams fragmented
0 fragments created
0 datagrams that can't be fragmented

icmp:

0 calls to icmp_error
0 errors not generated 'cuz old message was icmp
0 messages with bad code fields
0 messages < minimum length
0 bad checksums
0 messages with bad length
0 multicast echo requests ignored
0 multicast timestamp requests ignored
0 message responses generated
ICMP address mask responses are disabled

igmp:


```

0 messages received
0 messages received with too few bytes
0 messages received with bad checksum
0 membership queries received
0 membership queries received with invalid field(s)
0 membership reports received
0 membership reports received with invalid field(s)
0 membership reports received for groups to which we belong
0 membership reports sent
tcp:
3344 packets sent
    652 data packets (46973 bytes)
    0 data packets (0 bytes) retransmitted
    0 resends initiated by MTU discovery
    1741 ack-only packets (1123 delayed)
    0 URG only packets
    0 window probe packets
    886 window update packets
    65 control packets
6113 packets received
    733 acks (for 46968 bytes)
    26 duplicate acks
    0 acks for unsend data
    5430 packets (6491157 bytes) received in-sequence
    1 completely duplicate packet (1 byte)
    0 old duplicate packets
    0 packets with some dup. data (0 bytes duped)
    550 out-of-order packets (732314 bytes)
    0 packets (0 bytes) of data after window
    0 window probes
    0 window update packets
    0 packets received after close
    0 discarded for bad checksums
    0 discarded for bad header offset fields
    0 discarded because packet too short
27 connection requests
17 connection accepts
0 bad connection attempts
0 listen queue overflows
44 connections established (including accepts)
46 connections closed (including 0 drops)
    0 connections updated cached RTT on close
    0 connections updated cached RTT variance on close
    0 connections updated cached ssthresh on close
0 embryonic connections dropped
733 segments updated rtt (of 734 attempts)
0 retransmit timeouts
    0 connections dropped by rexmit timeout
0 persist timeouts
    0 connections dropped by persist timeout
0 keepalive timeouts
    0 keepalive probes sent
    0 connections dropped by keepalive
21 correct ACK header predictions
4795 correct data packet header predictions
udp:
516 datagrams received
0 with incomplete header
0 with bad data length field
0 with bad checksum
2 dropped due to no socket
436 broadcast/multicast datagrams dropped due to no socket
0 dropped due to full socket buffers

```

```
0 not for hashed pcb
78 delivered
78 datagrams output
```

Ping has started ...

```
PING 131.178.107.89 (131.178.107.89): 56 data bytes
64 bytes from 131.178.107.89: icmp_seq=0 ttl=253 time=4.748 ms
64 bytes from 131.178.107.89: icmp_seq=1 ttl=253 time=4.293 ms
64 bytes from 131.178.107.89: icmp_seq=2 ttl=253 time=5.197 ms
64 bytes from 131.178.107.89: icmp_seq=3 ttl=253 time=2.251 ms
64 bytes from 131.178.107.89: icmp_seq=4 ttl=253 time=2.662 ms
64 bytes from 131.178.107.89: icmp_seq=5 ttl=253 time=5.744 ms
64 bytes from 131.178.107.89: icmp_seq=7 ttl=253 time=8.918 ms
64 bytes from 131.178.107.89: icmp_seq=8 ttl=253 time=3.355 ms
64 bytes from 131.178.107.89: icmp_seq=9 ttl=253 time=4.927 ms
```

```
--- 131.178.107.89 ping statistics ---
10 packets transmitted, 9 packets received, 10% packet loss
round-trip min/avg/max = 2.251/4.677/8.918 ms
```

Traceroute has started ...

```
traceroute to 131.178.107.89 (131.178.107.89), 30 hops max, 40 byte packets
 1 131.178.115.1 (131.178.115.1) 4.206 ms 2.243 ms 1.866 ms
 2 gwipv6.mty.itesm.mx (131.178.100.8) 3.379 ms 3.774 ms 4.417 ms
 3 wiener.mty.itesm.mx (131.178.26.43) 2.659 ms * 3.858 ms
```

Ping has started ...

```
PING wiener.mty.itesm.mx (131.178.26.43): 56 data bytes
64 bytes from 131.178.26.43: icmp_seq=0 ttl=253 time=4.002 ms
64 bytes from 131.178.26.43: icmp_seq=1 ttl=253 time=6.065 ms
64 bytes from 131.178.26.43: icmp_seq=2 ttl=253 time=2.064 ms
64 bytes from 131.178.26.43: icmp_seq=3 ttl=253 time=4.492 ms
64 bytes from 131.178.26.43: icmp_seq=4 ttl=253 time=5.728 ms
64 bytes from 131.178.26.43: icmp_seq=5 ttl=253 time=1.882 ms
64 bytes from 131.178.26.43: icmp_seq=6 ttl=253 time=5.388 ms
64 bytes from 131.178.26.43: icmp_seq=7 ttl=253 time=1.829 ms
64 bytes from 131.178.26.43: icmp_seq=8 ttl=253 time=3.769 ms
64 bytes from 131.178.26.43: icmp_seq=9 ttl=253 time=5.096 ms
```

```
--- wiener.mty.itesm.mx ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 1.829/4.031/6.065 ms
```

Traceroute has started ...

```
traceroute to wiener.mty.itesm.mx (131.178.26.43), 30 hops max, 40 byte packets
 1 131.178.115.1 (131.178.115.1) 22.142 ms 11.604 ms 6.836 ms
 2 gwcronos.mty.itesm.mx (131.178.101.1) 1.766 ms 24.48 ms 7.188 ms
 3 wiener.mty.itesm.mx (131.178.26.43) 1.796 ms * 2.198 ms
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp	31998	0	pto-115-65.mty.i.49157	wiener-atm.mty.i.8000	ESTABLISHED
tcp	0	0	pto-115-65.mty.i.49153	msgr-ns14.msgr.h.1863	ESTABLISHED
tcp	0	0	localhost.768	localhost.832	ESTABLISHED
tcp	0	0	localhost.832	localhost.768	ESTABLISHED
tcp	0	0	localhost.768	localhost.779	ESTABLISHED
tcp	0	0	localhost.779	localhost.768	ESTABLISHED
udp	0	0	localhost.49156	localhost.807	

Estadísticas de Red
Marzo 19, 2002. 11:00 hrs.
Tipo de transmisión: bajo demanda
Fuente de audio: 131.178.48.230
DNAS: 131.178.107.89 (lane)
Lugar: Bibiloteca 2do. piso
Bit Rate: **128 Kbps**
Enlace: 10 Mbps
No Multicast
Interrupciones/retraso perceptible: 0
Errores en Audio: 0

Physical Address: 00-E0-98-7B-11-91
IP Address: **131.178.153.41**
Subnet Mask: 255.255.255.0
Default Gateway: 131.178.153.1
DHCP Server: 131.178.8.1
Lease Obtained: 19/03/2002 10:58:08 a.m.
Lease Expires: 19/03/2002 11:58:08 a.m.
DNS Servers: 131.178.1.16, 131.178.1.1, 131.178.38.39
WINS Server: 131.178.1.9

Active Connections

Proto	Local Address	Foreign Address	State
TCP	ChillyWilly:1106	wiener-atm.mty.itesm.mx:8000	ESTABLISHED
TCP	ChillyWilly:5000	pto-153-225.mty.itesm.mx:3937	ESTABLISHED

Estadísticas de red
Abril 8, 2002. 01:50 hrs.
Retraso entre estaciones de trabajo Wiener y Gabor
Interfase: LANE

```
$ ping -s -t 253 -i 131.178.107.89 131.178.107.88 56 10
PING 131.178.107.88: 56 data bytes
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=0. time=0. ms
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=1. time=0. ms
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=2. time=0. ms
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=3. time=0. ms
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=4. time=0. ms
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=5. time=0. ms
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=6. time=0. ms
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=7. time=0. ms
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=8. time=0. ms
64 bytes from gabor-atm.mty.itesm.mx (131.178.107.88): icmp_seq=9. time=0. ms
```

```
----131.178.107.88 PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
```

Trazo de ruta entre estaciones

```
$ traceroute -f 20 -i lane0 131.178.107.88 40
traceroute to 131.178.107.88 (131.178.107.88), 30 hops max, 40 byte packets
 20  gabor-atm.mty.itesm.mx (131.178.107.88)  0.532 ms * 0.277 ms
```

Interfase: Ethernet

```
$ traceroute 131.178.107.88
traceroute: Warning: Multiple interfaces found; using 131.178.26.43 @ hme0
traceroute to 131.178.107.88 (131.178.107.88), 30 hops max, 40 byte packets
 1  gabor.mty.itesm.mx (131.178.26.38)  0.418 ms * 0.230 ms
```

Prueba de Transmisión bajo demanda

```
Bit Rate: 128 Kbps
Total de paquetes enviados: 8347
Perdidas: 0
```

Apéndice B. Acrónimos.

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband-Integrated Services Digital Network
BUS	Broadcast Unknown Server
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CLR	Cell Lost Ratio
CRC	Cyclic Redundancy Check
CTD	Cell Transfer Delay
DCT	Discrete Cosine Transform
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ITU	International Telecommunications Union
LAN	Local Area Network
LEC	LAN Emulation Client
LECS	LANE Configuration Server
LES	LAN Emulation Server
MPEG	Motion Picture Experts Group
MPOA	Multiple Protocols Over ATM
OSI	Organization Standards International
PCR	Peak Cell Rate
PDU	Protocol Data Unit
P-NNI	Private Network to Network Interface
PPP	Point to Point Protocol
PVC	Permanent Virtual Circuits
QoS	Quality of Service
RARP	Reverse Address Resolution Protocol
RFC	Request For Comment
RSVP	Resource Reservation Protocol
RTT	Round Trip Times
RTP	Real Time Protocol
SAR	Segmentation And Reassembly
SCR	Sustained Cell Rate
SSCS	Service Specific Convergence Sublayer
SVC	Switched Virtual Circuits
TCP	Transmission Control Protocol
TTL	Time To Live
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
UNI	User to Network Interface
VBR	Variable Bit Rate
VC	Virtual Circuit
VCC	Virtual Circuit Connection
VPC	Virtual Path Connection
WAN	Wide Area Network