

ALGORITMOS PARA ENCRYPTACIÓN DE DATOS



Por:

Vega Lebrún Carlos

Arvizu Gutiérrez Diego

García Santillán Arturo

Serie

Libros y Manuales: Informática

Unidad Multidisciplinaria: CIET

Libros de Texto/01/2008



UNIVERSIDAD CRISTOBAL COLON

Campus Calasanz y Campus Torrente

Carr. Veracruz-Medellín s/n Col. Puente Moreno, Boca del Río, Ver.,

Tel. (01 229) 9230170 al 76 Ext. 2060 y 2069

<http://dgip.ver.ucc.mx>

<http://dgip.ver.ucc.mx/centros/ciet/ciet.htm>



Diseño de Portada

Julio César Hernández Rivera

Centro de Cómputo Académico UCC, Campus Calasanz

Vega, Arvizu y García.: (2008) “Algoritmos de encriptación de Datos. Elementos técnicos de estudio” Edición electrónica. Texto completo en www.eumed.net/

ISBN: En trámite

Registro Biblioteca Nacional de España No. En trámite

Índice

Capítulo 1. Introducción

Planteamiento
Antecedentes
Situación Actual

Capítulo 2. Fundamentos Teóricos

2.1 Antecedentes
2.2 Conceptos de Redes y Telecomunicaciones
2.3 Redes de comunicación de datos
2.4 Transmisión en Tiempo Real
2.5 Calidad de Servicio. Tipos de tráfico
2.6 Algoritmos de Optimización de Tráfico en redes WAN
2.7 Tendencias en la Optimización del ancho de banda

Capítulo 3. Método y Técnica de Diseño

3.1. Método
3.2. Técnica de Diseño

Capítulo 1V. Caso Estudiado

4.1 Red corporativa de estudio
4.2 Preparación del equipo de medición para pruebas
4.3 Tecnología de Optimización

Capítulo 5. Resultados

5.1. Análisis e interpretación de los resultados
5.2 Discusión

CONSIDERACIONES FINALES

Referencias Bibliográficas

Anexos

Capítulo 1.

Introducción

A. PLANTEAMIENTO DEL PROBLEMA

ANTECEDENTES

Las aplicaciones en una red de área amplia (WAN) históricamente han sido de texto ligero “pantallas verdes” o transferencia de archivos en lote en forma asíncrona. En contraste, las aplicaciones actuales son más pesadas, con interfaces de usuario que son altamente sensibles a los retrasos de la red. Los desarrolladores de aplicaciones continúan enfocándose hacia la funcionalidad de la misma, frecuentemente a expensas del rendimiento de la red, causando que en cada nueva versión sea peor que su precursor.

SITUACIÓN ACTUAL

Hay una necesidad insatisfecha en los negocios por la demanda en la capacidad de las redes de área amplia (WAN). No solo el volumen de aplicaciones se ha incrementado, sino también su importancia. Los usuarios del negocio están también demandando más recursos de la red, más disponibilidad y más velocidad implicando ello una saturación en la red. Esta saturación tiene diferentes orígenes: aplicaciones propias del negocio (ERP's), CRM, Comercio Electrónico, correo electrónico, navegación por la WEB están incrementando su demanda en la red. La reestructuración de las áreas de TI y la centralización de proyectos son también factores, tales como centros de consolidación de datos, almacenamiento, Voz sobre IP, recuperación de desastres e Intranets. Adicionalmente las arquitecturas de las aplicaciones están cambiando.

La gran cantidad de estas nuevas aplicaciones, está saturando las redes de área amplia causando numerosos problemas de funcionamiento por el tráfico de misión crítica. El área de Tecnología de Información (TI) tradicionalmente ha respondido comprando más ancho de banda a los proveedores de servicio “carriers”. Sin embargo, con el presupuesto limitado asignado al área de TI, el aumento del

ancho de banda es insostenible para cualquier organización. Por otra parte, simplemente la adición de capacidad de la red no soluciona siempre los problemas de funcionamiento del uso.

La mayoría de las empresas en sus redes WAN tiene una fracción mínima del ancho de banda disponible a través de sus redes locales (LAN). Como resultado de ello, las aplicaciones que se ejecutan en una oficina matriz frecuentemente fallan cuando estás tratan de desplegarse en una sucursal. Por lo anterior, las empresas deben comenzar a centrarse en la optimización de sus redes de datos para asegurar de que pueden soportar las aplicaciones de misión crítica de una manera rentable. Es a partir de estos argumentos, que se pretende mostrar a lo largo de este documento, el desarrollo que pudiera llevarse a cabo al interior de las empresas, para buscar optimizar sus redes de datos, que favorezca su rentabilidad en aplicaciones críticas.

Los elementos de estudio, quedan centrados en lo siguiente:

b. Propósito: Con el experimento se pretende

Optimizar el ancho de banda (entre un 50 y 100% de su actual ocupación) para la reducción de costos y hacer más eficiente una red corporativa de área amplia (WAN) al aplicar los algoritmos de optimización en un ambiente real de operación de la red.

Además se busca

- ✓ Reducción de gastos recurrentes en infraestructura de telecomunicaciones.
- ✓ Incrementar el ancho de banda entre 70% y 80%
- ✓ Incrementar el funcionamiento de la red en 40% y 50%.
- ✓ Reducción del congestionamiento del nodo corporativo de red, actualmente presentando una saturación del 120 a 150% del ancho de banda.
- ✓ Incremento en los niveles de seguridad en la transmisión de datos sin afectar el funcionamiento de la red.
- ✓ Seleccionar la mejor solución tecnológica existente en el mercado, que mejores algoritmos de optimización emplee para mayores beneficios a ofrecer a una empresa corporativa.

JUSTIFICACIÓN Y APORTACIÓN

Las diferentes actualizaciones tecnológicas de TI y las iniciativas de negocio continúan adicionando presión a la red de área amplia.

Recientemente, las soluciones a la saturación de la red WAN se han enfocado a solo parte del problema. El desarrollo de aplicaciones basadas en el WEB con manejo de información estática permitió su uso desde una oficina remota; pero tiene un pequeño efecto en aplicaciones dinámicas WEB en el manejo de bases de datos. Las técnicas de priorización, tal como la calidad en el servicio (QoS) aceleran el uso de aplicaciones de misión crítica, pero siempre a expensas de otros. En la mayoría de las organizaciones, TI, ha recurrido simplemente a adquirir capacidad WAN adicional. Con el costo total de la red (entre el 20% al 30% del presupuesto de TI) y el constantemente incrementó de la capacidad WAN no son siempre la mejor opción.

Derivado de esta necesidad del mercado, surge una nueva solución tecnológica emergente de compresión y optimización del ancho de banda como medio eficiente de aumentar la capacidad total de la red y de proporcionar un mejor control de tráfico. La compresión y la optimización del ancho de banda proporcionan a TI una manera rápida y eficaz de reducir tráfico en la red entre un 60% y 80%, liberando el ancho de banda para nuevas aplicaciones, acelerando el funcionamiento de las aplicaciones existentes y evitando así mejoras innecesarias a la red. La cantidad permisible de reducción de cualquier enlace WAN específico dependerá de la combinación del tráfico.¹

Es por ello que las empresas deben considerar dentro de sus soluciones tecnológicas, técnicas de compresión y de optimización del ancho de banda en el manejo de memoria cache, calidad en el Servicio (QoS) y la supervisión del tráfico que la demanda del negocio requiere en el funcionamiento y uso de la red. Mejores algoritmos que midan el ancho de banda son necesarios para una variedad de aplicaciones y de usos en la red.

De acuerdo a estadísticas de Gartner, antes de 2008, la inversión en soluciones WAN para la optimización del ancho de banda será la estrategia primaria de mejora para el 75% de las empresas corporativas en el mundo.

¹ Metagroup. www.metagrup.com

Es en este trabajo, que se revisan las técnicas de optimización en el ancho de banda utilizadas y así mismo, permitirá establecer una metodología que sirva de referencia para hacer la planeación correspondiente en el crecimiento futuro, de los recursos teleinformáticos de una empresa corporativa.

La aportación empírica:

- ✓ Análisis de algoritmos utilizados en la optimización del ancho de banda, haciendo una propuesta de mejora a dichos algoritmos.
- ✓ Metodología para la optimización de la red y aplicaciones

Viabilidad y Factibilidad

Cómo en la mayoría de las empresas corporativas se tiene la necesidad de la optimización del ancho de banda, se selecciona la empresa que se toma como caso de estudio.

Los recursos básicos para llevar el experimento a cabo son.

- ✓ Acceso a las instalaciones en diferentes horarios
- ✓ Estación de trabajo (PC, con características de alto desempeño)
- ✓ Software de monitoreo de red
- ✓ Software de simulación de redes
- ✓ Instrumento de medición (a ser facilitado por proveedor)
- ✓ Acceso a los equipos de red (router principal Cisco 7000)

Cualquier empresa corporativa es factible que invierta en la mejora de su infraestructura teleinformática tomando como base este tipo de análisis realizado en este trabajo.

Alcances y limitaciones

Esta investigación centra su estudio en un nodo, perteneciente a una empresa corporativa que servirá de caso de estudio, de la red WAN a analizar ubicado en la Ciudad de Veracruz; para lo cual se realiza un diagnóstico de la situación actual y basándose en dicho diagnóstico se aplican las técnicas de optimización del ancho de banda para una red WAN que trabaja en tiempo real. Se selecciona la mejor solución tecnológica existente en el mercado y se realizan las gestiones necesarias para que el proveedor del instrumento de medición que se utiliza para probar la metodología de optimización del ancho de banda, facilite el equipo necesario para realizar las pruebas necesarias en el sitio de las instalaciones de la empresa corporativa.

Las limitantes del proyecto son:

Solo se podrá analizar un nodo de toda la red corporativa por cuestiones de facilidad de los recursos y disponibilidad de tiempo de los administradores de red. Y se tratará de analizar cada uno de los enlaces de datos que dicho nodo tiene establecidos con las diferentes áreas. Y en su defecto, solo se analizarán los enlaces que mayor saturación tengan en estos momentos.

El instrumento de medición a utilizar para realizar las pruebas, mediciones e interpretación de resultados pertenece al proveedor de la solución tecnológica, por lo que se estará sujeto a que dicho proveedor de las facilidades del equipo en tiempo y forma. En caso de que no se cuente con el dicho equipo, este trabajo se limita a solo presentar resultados obtenidos utilizando software de simulación y/o comprobación matemática.

Para el caso de los algoritmos a analizar, estos se evaluarán de forma simulada y empleando técnicas matemáticas de comprobación. No se analizarán, a detalle, los algoritmos utilizados en la tecnología seleccionada de optimización del ancho de banda por ser información clasificada como confidencial por parte del proveedor.

Diseño de la Investigación

El diseño de la investigación se basa en un experimento que permita obtener información cuantitativa:

- ✓ Antes y después de aplicar la solución tecnológica propuesta de optimización del ancho de banda.
- ✓ Comprobación y/o simulación de la funcionalidad de los algoritmos analizados en la optimización del ancho de banda.

Población y muestra

En dicha investigación, la población o universo estará representada por los diferentes servicios (enlaces de comunicación) que el nodo Veracruz tiene en estos momentos habilitados, obteniendo lecturas y registros de la cantidad de Bytes utilizados en el ancho de banda durante un lapso de un mes. Por lo que se considera todo el universo de tráfico (de llegada y de salida), que se genera en dicho nodo de Veracruz.

Técnicas e instrumentos de recolección de datos

Para obtener el diagnóstico de la situación actual del nodo Veracruz, se emplean las técnicas e instrumentos existentes en la empresa corporativa (nodo Veracruz).

Para comprobar la optimización del ancho de banda, se utilizan las técnicas e instrumentos que proporciona el proveedor acerca de la mejor solución tecnológica existente en el mercado.

Y para el caso del análisis de los algoritmos, se utilizará software de simulación y los resultados que proporcione la comprobación matemática.

Confiabilidad

La confiabilidad de los instrumentos y técnicas a aplicar son altamente confiables. La primera se basa en una solución tecnológica que esta presentando un posicionamiento en el mercado en los diferentes corporativos. Y con respecto a lo segundo, se utiliza el lenguaje de demostración matemática.

Desarrollo tecnológico

Para este estudio se emplean, los recursos que la empresa corporativa tiene y el equipo que el proveedor facilita, siendo estos:

- ✓ Equipo y software de monitoreo de red marca Domino
- ✓ Switch / Router marca Cisco modelo 7000
- ✓ PC Pentium IV a 800 MHZ; 512 MB en RAM; HD de 20 GB, NIC Ethernet 10/100 mbps
- ✓ Acceso al Firewall
- ✓ Equipo del proveedor Modelo SR-100 o SR-500

El modelo que se propone para obtener los resultados, empleando las herramientas anteriores, es colocar entre el Switch / Router (principal de la red corporativa) y el Firewall el equipo SR-100 y/o SR-500 conectados por el puerto Ethernet para monitorear un extremo del enlace (tráfico de llegada/salida). Y en el otro extremo del enlace también se colocará otro equipo con similares características para monitorear el otro extremo del enlace (tráfico de llegada/salida).

Se realizan pruebas para cada uno de los diferentes enlaces, que el nodo Veracruz tiene y/o administra.

A dicho equipo SR, previamente, se le asigna una dirección IP y la mascara de red perteneciente al nodo del corporativo.

Para la funcionalidad de los algoritmos se emplea software de simulación y el lenguaje matemático que nos permita obtener los resultados esperados.

Capítulo 2.

Fundamentos Teóricos

2.1 Antecedentes

La necesidad de transmisión de grandes cantidades de información tiene un gran impacto en los sistemas de comunicación. Esta transmisión requiere no sólo el soporte de las redes, sino también de los protocolos de comunicación de niveles superiores. En este sentido se trabaja tanto en redes locales (LAN) como en redes WAN.

Podemos hablar de transmisión de información en tiempo real cuando se puede asegurar que la información llegue a su destino con unos parámetros determinados (retraso, rendimiento, fiabilidad, etc.). En este sentido se puede asumir que la transmisión de información tiene unos requerimientos temporales que necesitan del uso de esta transmisión en tiempo real.

En general, todas las aplicaciones (sistemas de información, multimedia, etc.) requieren una calidad de servicio (QoS) por parte de los servicios de red. De las nuevas redes se exige poder especificar esta calidad de servicio y asegurar su cumplimiento. Otro problema es ¿quién proporciona esta calidad de servicio?: la red (los routers, nodos, conmutadores, etc.) o el sistema operativo.

2.2 Conceptos de Redes y Telecomunicaciones

2.2.1 Diferencias entre LAN y WAN

En adición a las usualmente conocidas en cuanto a amplitud de cobertura, son relevantes las siguientes diferencias:

- ✓ Tecnología.- Una LAN se basa en tecnologías usualmente homogéneas, en las que se utiliza un medio de transmisión compartido directa o indirectamente, mientras las WAN suelen integrar diferentes tecnologías.
- ✓ Capacidad de crecimiento.- Derivado de la diferencia de tecnologías, las WAN pueden crecer de manera prácticamente ilimitada, en tanto las LAN se ven restringidas a determinado límite práctico.
- ✓ Velocidad.- Típicamente las LAN operan a una misma velocidad, que suele ser más alta que la disponible en enlaces WAN.

- ✓ **Costo.**- La tecnología empleada en las WAN, aunada al usual empleo de enlaces rentados, las hacen más caras que las LAN, donde suelen utilizarse dispositivos de menor costo y medios de transmisión locales propios.

Medios

Constituyen la vía por la cual son transmitidos los datos entre los nodos de una red, siendo los más comunes: Cable de cobre, Fibra óptica y Radiofrecuencia.

2.2.3 Protocolos.

En este caso nos referimos a los estándares que norman la transmisión de datos sobre el medio, abarcando lo referente a las funciones definidas para el nivel físico y de enlace del modelo de referencia OSI.

2.2.4 Dispositivos de interconexión para LAN

Existen dispositivos de conexión para una red LAN, tales como:

- ✓ **Repetidor** (Repeater). Su finalidad es permitir la unión de dos segmentos LAN para hacerlos ver como uno solo
- ✓ **Concentrador** (Hub). Es un repetidor de múltiples puertos, por lo que puede unir múltiples segmentos de red. Regenera la señal recibida por cualquier puerto en cada uno de los restantes.
- ✓ **MAU o MSAU** (Multiple Station Attachment Unit). Es un dispositivo de interconexión utilizado para construir redes Token Ring sobre topología física de estrella.
- ✓ **Puente** (Bridge). Opera en el nivel MAC. Se utiliza para interconectar dos segmentos de LAN. Provee filtrado de tramas, manteniendo tablas de direcciones independientes para cada segmento y solamente retransmite las tramas cuyo destino se ubica en la LAN opuesta a la del emisor o son broadcasts a nivel MAC.
- ✓ **Conmutador** (Switch). Es un bridge de múltiples puertos, por lo que puede interconectar múltiples segmentos de red.

2.2.5 Dispositivos de interconexión para WAN

Routers. Son dispositivos que operan al nivel de la capa de red del modelo de referencia OSI. Sirven para interconectar redes para formar una WAN, y proveen la lógica necesaria para elegir las rutas para mover el tráfico entre las redes que forman una inter-red. Las redes interconectadas pueden ser de diferentes tecnologías, incluyendo las existentes para LAN y enlaces seriales de varios tipos. Para ello cuentan con diversas interfaces, de diferentes tipos si es necesario.

Access server. Es un tipo especial de router que típicamente provee conectividad a una LAN o WAN mediante enlaces telefónicos conmutados. Es usado por TELECOMMUTERS y usuarios móviles para obtener acceso a la red.

Access router. Es un router con capacidades añadidas de access server.

2.2.6 Definición de ancho de banda

En el área de comunicaciones en la capa física, el término ancho de banda se relaciona con el ancho espectral de señales electromagnéticas o con las características de propagación de los sistemas de comunicación. En el contexto de las redes de datos, el término ancho de banda cuantifica la tasa de datos que un enlace o una ruta de la red puede transferir. En este trabajo de tesis nos centramos en la valoración del ancho de banda en este último contexto de la red de datos.

El concepto de ancho de banda es central en las comunicaciones digitales, y específicamente en las redes de paquetes, pues se relaciona con la cantidad de datos que una ruta de un enlace o de la red pueda entregar por unidad de tiempo.

Por lo que adoptaremos la siguiente definición:

“Cantidad de datos que puede ser enviada en un periodo de tiempo determinado a través de un circuito de comunicación dado, por ejemplo, 33,6 Kbps (miles de bits por segundo)”.

2.3 Redes de comunicación de datos

Existe un conjunto de redes que permiten la transmisión de datos en tiempo real. Algunas de ellas son un desarrollo totalmente nuevo, otras son la adaptación de redes en uso para admitir canales para transmisión de audio y vídeo. Incluso se transmite información multimedia en redes no especialmente preparadas como por ejemplo Ethernet.

Aunque el concepto de red para LAN se refiere claramente a los niveles físicos y de acceso al medio, en el contexto de las WAN está menos claro ya que proporcionan un conjunto de servicios y funciones. Estos suelen incluir niveles de red y de enlace. Haciendo un estudio de las distintas tecnologías de red en función de los siguientes parámetros:

- ✓ Ancho de banda. Al menos 1.4 Mbps (tomando cómo referencia la transmisión Multimedia)
- ✓ Retraso de transmisión. Un máximo de 10 a 15 ms (aunque puede ser mayor si no hay ningún salto)
- ✓ Comunicación multipunto: Ver si la red incorpora funciones de multicasting. En redes de tipo broadcast esto no es ningún problema.
- ✓ Fiabilidad: Control de errores y mecanismos de recuperación en la red.

2.3.1 Ethernet

Ethernet, es la red de área local más utilizada actualmente. Su ancho de banda de 10 Mbps., permite varios canales de voz/datos/vídeo. El inconveniente, es el método no determinista de acceso a la red (CSMA-CD). Con altas cargas en la red no hay control sobre el tiempo de acceso a la red o el ancho de banda disponible. Sin embargo, muchas aplicaciones multimedia usan Ethernet como su mecanismo de transporte, generalmente en un entorno controlado y protegido.

Evaluación: Ethernet no es adecuada debido a su retraso no determinista, sin embargo se puede utilizar teniendo poco canales debido a su gran ancho de banda.

2.3.2 Iso - Ethernet

Es una red híbrida que integra una Ethernet estándar de 10 Mbps con 6.144 Mbps de ancho de banda isócrono (RDSI), ofreciendo un total de 16 Mbps. Proporciona 96 canales RDSIB de 64-Kbps utilizando el

mismo cable que la Ethernet a 10 Mbps, con lo cual tiene poco costo de integrarlo con los cableados actuales.

Evaluación: Esta red es una solución destinada a aprovechar los recursos existentes. Proporciona tráfico isócrono. La estructura de los canales parecida a RDSI permite la transmisión de audio o vídeo H.261, pero no permite la transmisión de vídeo de alta calidad.

2.3.3 Token Ring

Token Ring está mejor preparada para la transmisión multimedia. Una razón es su ancho de banda de 16 Mbps. Más importante es la disponibilidad de prioridades y su acceso determinista. Se pueden usar la alta prioridad para datos en tiempo real y la baja para datos normales.

Los retrasos dependen del número de estaciones conectadas, la carga y el tamaño del paquete. En general si se trabaja en un entorno en el que no hay muchas estaciones y el paquete se escoge de un tamaño bajo (64 a 128 bytes) se pueden asegurar retrasos de como máximo 10 ms. De hecho existen algoritmos que pueden determinar en estas redes el peor retraso.

Evaluación : Proporciona el suficiente ancho de banda para un número limitado de canales multimedia. Se puede garantizar el retraso si se utiliza un esquema de acceso y prioridades definidos. También permite multicasting.

2.3.4 100Base-T

Esta red es una simple evolución de Ethernet ampliando su velocidad a 100 Mbps.

2.3.5 IEEE 802.12 (100-Mbps Demand Priority LAN)

Este nuevo protocolo de red está pensado para satisfacer las necesidades de los nuevos entornos de red. Se busca la posibilidad de enviar vídeo y audio en tiempo real utilizando la red.

Es una LAN de 100 Mbps de ancho de banda con prioridades. El objetivo de este protocolo es poder sustituir los protocolos 802.3 y 802.5. De hecho mantiene el LLC de estos protocolos y el MAC es compatible con estos dos. Se definen dos niveles de prioridad en las tramas, el más alto con el objetivo de la transmisión multimedia. El protocolo es determinista ya quien gestiona la red es el repetidor y lo hace por medio de un mecanismo round-robin con dos prioridades.

Con este protocolo se puede calcular y garantizar los tiempos de retraso para un tamaño máximo de paquete y un número de estaciones conectadas. Por ejemplo, la transmisión de un paquete de 4-Kbytes es de 0.3 ms. Así, incluso cuando hubiese 30 estaciones conectadas y todas ellas hicieran peticiones para enviar, el retraso de acceso es menor de 10 ms.

Evaluación: Esta red es adecuada para transmisión multimedia. Si se configura la red de forma adecuada y se limita el tamaño del paquete se pueden asegurar que los retrasos de transmisión estarán asegurados. Su alto ancho de banda también lo hacen adecuado para la transmisión multimedia de calidad.

2.3.6 FDDI (Fiber Distributed Data Interface)

En principio la red FDDI es una Token-Ring rápida. Este ancho de banda alto permite una comunicación multimedia. Además FDDI soporta una clase de tráfico síncrono, que permite tráfico con un retraso limitado, configurable en tiempo de inicialización del anillo.

Desafortunadamente, al configurar un retraso demasiado bajo reduce la utilización del ancho de banda. Prácticamente, el límite de retraso está entre 5 y 50 ms.

Evaluación: FDDI soporta la transmisión multimedia aceptablemente bien, gracias a su elevado ancho de banda, su clase de tráfico síncrono y el multicasting disponible.

2.3.7 FDDI-II (Fiber Distributed Data Interface)

Para evitar los problemas del FDDI aparece FDDI-II que está basado en un protocolo slotted-ring. Permite capacidades isócronas usando slots de tiempo pre-asignados a una tasa de 8-khz.

Evaluación: Esta red proporciona canales con retraso del orden de milisegundos. El ancho de banda es suficiente y el multicasting está disponible.

2.3.8 X-25

En Europa la mayoría de servicios que utilizan conmutación de paquetes utilizan como red X-25, que permite la transferencia fiable de datos sobre enlaces de baja velocidad y fiabilidad. Su control de flujo y mecanismo de recuperación de errores provoca retrasos incontrolables.

Evaluación: X-25 no sirve para la transmisión multimedia.

2.3.9 Frame Relay

Frame Relay fue desarrollado para superar las limitaciones de X-25 para un entorno de transmisión de alta velocidad. Usa la trama LAP-D (Link Access Procedure Direct) con un identificador de 19 bit para encaminamiento. Los paquetes están limitados a un máximo de 4 Kb.

El protocolo Frame Relay no proporciona control de errores y mecanismo de control de flujo. Esto significa que por parte del protocolo no hay ningún mecanismo que afecte negativamente la transferencia. Por otro lado, los conmutadores si que introducen un retraso que depende de la carga, la velocidad de la línea y el tamaño de trama. Frame Relay se puede utilizar de dos formas: La más frecuente como un protocolo simple sobre línea dedicada. En este caso no se introduce ningún tipo de retraso. Sobre redes conmutadas, en este caso el retraso vendrá dado por los conmutadores. Además el direccionamiento está limitado a 10 bit. Por lo tanto no se espera que se convierta en la red universal multimedia.

Evaluación: Frame Relay es un protocolo de nivel 2 que no añade retrasos. El protocolo en si es adecuado para la transmisión en tiempo real, pero solo si se usa en líneas dedicadas ya que en redes conmutadas el retraso no es determinista.

2.3.10 RDSI (Red digital de servicios integrados)

La Red Digital de Servicios Integrados (RDSI) fue diseñada para soportar una gran variedad de servicios, como datos, voz, fax o vídeo. Esta construido sobre canales síncronos de 64-kbps, que se pueden usar con tráfico CBO (continuous bit-stream oriented) o CBR (constant bit-rate).

La única desventaja de esta red es su ancho de banda limitado. Para construir un ancho de banda alto hay que utilizar varios canales, y estos canales pueden seguir distintas rutas por lo que requieren una sincronización en el receptor. Utilizando tráfico CBO el retraso es constante y pequeño. Pero en el caso de transmisión de paquetes existe un retraso debido al limitado ancho de banda. Este dependerá del tamaño del paquete.

Evaluación: En general RDSI es la única WAN ampliamente disponible en Europa que soporta bien la transmisión multimedia. Falta la posibilidad de multicast.

2.3.11 ATM (Modo de Transferencia Asíncrona)

ATM es la mejor tecnología para transmisión en tiempo real. ATM es una arquitectura de red basada en células, que permite la multiplexación y conmutación de paquetes. La ITU ha declarado ATM como la tecnología base para la red pública B-ISDN. La red define tres niveles:

- ✓ Nivel físico. Se han definido distintos medios de transmisión como fibra óptica a 155 Mbps, o 100-Mbps FDDI para ATM de área local, además de otras opciones. Actualmente las interfaces a ATM han evolucionado hasta 622 Mbps o incluso 2.4 Gbps.
- ✓ Nivel ATM. El nivel ATM es una capa independiente para la conmutación y el multiplexado de los paquetes. Se define la estructura de las células como contenedores de información de 53 octetos (5 octetos para cabecera más 48 de datos). El encaminado se basa en los identificadores de circuito virtual VCI (virtual circuit identifiers) y los identificadores de camino virtual VPI (virtual path identifiers). Además ATM es una red orientada a la conexión. Debido al pequeño tamaño de las células ATM, la alta velocidad de los enlaces, y la velocidad de conmutación de los nodos, ATM proporciona una latencia mínima.
- ✓ Nivel de adaptación ATM (AAL). Este nivel está diseñado como puente entre el nivel ATM y de aplicación

Calidad de servicio: En la petición de establecimiento de conexión se especifica la calidad de servicio requerida, que contiene parámetros como ancho de banda, retraso y fiabilidad.

Evaluación: El protocolo permite la transmisión de datos de forma determinista y tiene un ancho de banda impresionante, el necesario como para la transmisión de vídeo de alta calidad. ATM también proporciona comunicación multicast.

2.3.12.- Resumen de las características de las redes

La siguiente Tabla 2.1 es un resumen de las características de las distintas redes evaluadas:

Red	Ancho Banda (Mbps)	Retraso Transmisión	Variación del Retraso	Broadcast	Disponible
Ethernet	10	Aleatorio	Φ	Sí	Sí
Iso-Ethernet (*)	10+6	Fijo < 1ms.	0	No	No
Token Ring	4/16	Configuración < 20ms.	Δ	Sí	Sí
100 Base-T	100	Aleatorio	Φ	Sí	Sí
802.12	100	Configuración < 10ms.	Δ	Sí	No
FDDI	2 x 100	Configuración	Δ	Sí	Sí
FDDI-II*	N x 6	Fijo < 1ms.	0	Sí	No
X-25	<2	Aleatorio	Φ	No	Sí
Frame Relay	< 50	Aleatorio	Φ	No	Sí
RDSI	N x 0.064	Fijo < 10ms	0	No	Sí
ATM	25 – 155	Fijo < 10ms	Δ	(Sí)	Sí
* : Se refiere a los canales isócronos.					
Fijo: El retraso es fijo para un canal.					
Configuración: El retraso depende de la configuración de la red (tamaño de los paquetes, número de estaciones, etc.). Se da un valor para una red configurada para que pueda asegurar una transmisión aceptable de información multimedia.					
Φ = Red asíncrona sin control del retraso.					
Δ = Red síncrona con una variación del retraso entre 0 y un valor máximo.					
0 = Red isócrona.					

Tabla 2.1 Resumen de las redes revisadas

De la Tabla 2.1 realizada se obtiene la conclusión de que existen algunas redes que prometen un buen soporte al tráfico (por ejemplo multimedia), pero solo unas pocas están disponibles en realidad. ATM se puede configurar como una red que trabaje en tiempo real por excelencia tanto para WAN como LAN.

La diferenciación entre LAN y WAN es cada vez menor y de hecho el usuario requiere servicios de red independientes de la organización física y lógica. De hecho las nuevas redes como ATM asumen este concepto.

Debido a que la Tecnología ATM permite trabajar en tiempo real y que este trabajo comprende a redes que trabajen precisamente en tiempo real se enfocará principalmente a esta tecnología.

2.4 Transmisión en Tiempo Real

2.4.1 Introducción

Hasta ahora, era común que toda la información (imágenes, texto, voz, etc.) llegue a nosotros por canales distintos: fotografía, radio, prensa, televisión, cine, etc. El concepto multimedia, es decir, la integración de los medios, es hasta hace poco una idea que se ha convertido en una realidad.

Actualmente, es posible que toda esta información multimedia pueda ser accesible a través de las redes de comunicaciones. Esto implica un cambio en la forma de distribución de esta información: un usuario se puede conectar a un servidor para ver la película deseada, acceder a conferencias en tiempo real, operaciones quirúrgicas a distancia, etc.

En otro aspecto, la era de los medios unidireccionales, es decir sólo receptivos, ha dado paso a la era de los medios interactivos: CD-ROM, DVD y Web-TV. Además, gracias al crecimiento de las redes de comunicación, Internet en particular, ya es posible formar parte de un mundo enteramente interactivo y multimedia.

Este incremento de la velocidad y capacidad de los distintos componentes de las redes de comunicaciones así como de los equipos de acceso (computadoras personales, decodificadores de TV, etc.) ha permitido utilizar nuevos tipos de aplicaciones como la videoconferencia, visualización científica y médica, telecontrol, vídeo y audio bajo demanda, etc., pero no siempre con los resultados deseados. Estas aplicaciones tienen unos requerimientos elevados en términos de ancho de banda, retraso de transmisión permitido y tasa de errores.

La transmisión multimedia, aunque utilizada en Internet con resultados bastantes decepcionantes, es utilizada ampliamente en otros tipos de servicios. Su uso se está extendiendo a la televisión digital por cable, tercera generación de móviles UMTS, radio digital, videoconferencia, etc. También a nuevas aplicaciones avanzadas como acceso en tiempo real a instrumentación remota, tele-inmersión, realidad virtual, etc.

El esquema de servicio ofrecido por las redes actuales es totalmente inadecuado, por lo que se requieren nuevos servicios que permitan una transmisión en tiempo real. Esto implica nuevos requisitos a la redes de transmisión, que se concretan en la especificación de una calidad de servicio (QoS: Quality of Service).

En este documento se utiliza la siguiente definición de transmisión en tiempo real, introducida por Ferrari [Ferrari90b]:

“Se denomina servicio de Transmisión en Tiempo Real aquel cuyos clientes pueden especificar los requerimientos sobre prestaciones y obtener garantía del cumplimiento de estos requerimientos.”

Por tanto, se puede hablar de transmisión de información en tiempo real cuando se puede asegurar que la información llegue a su destino con unos parámetros determinados (retraso, rendimiento, fiabilidad, etc.).

Para la transmisión de información en tiempo real se requiere un sistema de comunicaciones que proporcione a las aplicaciones los servicios y el control necesarios para la gestión de la calidad del servicio (QoS). En particular, la comunicación de audio o vídeo requiere la provisión de cierto nivel de calidad, garantizándolo durante el tiempo de la transmisión. Este nivel de calidad lo tienen que establecer las aplicaciones a la hora de comunicarse. En general el medio (físico o virtual) por el que se realizan estas comunicaciones se suele denominar canal. Estos canales son definidos con unos parámetros típicos como pueden ser el ancho de banda, retraso, variación del retraso (delay jitter) y fiabilidad.

2.4.2 Estado tecnológico actual

El estado tecnológico actual es desolador. Las LAN actualmente más usadas como pueden ser las redes Ethernet o Token Ring, no están preparadas para la transmisión de datos en tiempo real. En WAN la situación es peor. El gran problema de TCP/IP es el retraso no determinista de los paquetes. El hecho es que aunque existen una serie de productos comercializados para videoconferencia, telefonía y vídeo por Internet, etc.; sus resultados dejan mucho que desear.

Otro tema es el elevado ancho de banda requerido para la transmisión de vídeo y/o audio de alta calidad. Esto supone un problema debido a las actuales limitaciones de las redes de transmisión de datos y su alto costo económico.

Ante esta necesidad se han desarrollado un conjunto de redes y protocolos para la transmisión en tiempo real. Se pueden distinguir dos grupos:

- ✓ Las redes de transmisión de datos: En este grupo se incluyen las nuevas LAN FDDIII, Isochronous Ethernet, IEEE 802.12 y WAN (ATM, RDSI-BA). Todas estas redes contemplan en mayor o menor medida la transmisión de datos en tiempo real. Además tienen anchos de banda muy elevados para soportar la transmisión de gran cantidad de datos y de vídeo de alta calidad.
- ✓ Protocolos de transporte y red: Se están desarrollando un conjunto de nuevos protocolos para la transmisión de datos en tiempo real en WAN. Se basan en la modificación de protocolos de red o bien la utilización de algún otro protocolo de red como IPv4. Entre estos se pueden distinguir: ReSerVation Protocol (RSVP), Internet Protocol v6 (IPv6), Stream Protocol-II (ST-II), Real Time Transport Protocol (RTP) y los protocolos de la Tenet Suite.

2.4.3 Requerimientos para la transmisión de datos

2.4.3.1 Tráfico multimedia

Se suele denominar tráfico multimedia aquel compuesto por vídeo y audio, aunque formalmente se refiere a cualquier medio que se puede transmitir. En general, los requerimientos para la transmisión de audio y vídeo son tratados [Hehmann90] y sobresalen los siguientes puntos.

2.4.3.2.- Ancho de banda

El tráfico multimedia, por ejemplo, maneja grandes volúmenes de información; un CD-ROM, que son 72 minutos de sonido estéreo, requiere 648 Mbytes y una película de 90 minutos ocuparía 120 Gbytes, ambos sin comprimir.

La información de vídeo, por ejemplo, actualmente se trata casi exclusivamente de forma comprimida. El ancho de banda dependerá por tanto del tipo de compresión y de la calidad con que se desee transmitir.

En la práctica, por ejemplo la transmisión multimedia, demanda un ancho de banda de 0.4 Mbps a 1.4 Mbps y esto únicamente en comunicación unidireccional.

2.4.3.3.- Retraso de transmisión

Estos requerimientos son más estrictos que los de ancho de banda. La experiencia con los sistemas de conferencia multimedia y los estándares ITU sugieren un retraso máximo de 150 ms en las aplicaciones de vídeo interactivas y en telefonía este retraso es más exigente (<25ms) para evitar el eco. En función del retraso se pueden distinguir los siguientes tipos de tráfico:

- ✓ Asíncrono : Retraso de transmisión sin restricciones
- ✓ Síncrono : El retraso de transmisión está acotado para cada mensaje
- ✓ Isócrono : El retraso de transmisión es constante para cada mensaje

La isócronia no tiene porqué ser mantenida durante todo el camino de un mensaje, ya que puede ser recuperada en el destino mediante un almacenamiento para visualización.

Otro tema son los tiempos de compresión y descompresión de las imágenes de vídeo. Siguiendo los requerimientos del CCITT de un máximo de 150 ms de fuente a destino, se pueden identificar las siguientes componentes en el retraso [Stuttgen95]:

- ✓ Retraso en la compresión y descomposición en paquetes en la fuente
- ✓ Retraso de transmisión en la red
- ✓ Almacenamiento en el destino y retraso de sincronización
- ✓ Retraso de la composición de los paquetes y la descompresión en el destino.

La imagen debe tener de 25 a 30 tramas por segundo. Esto deja un tiempo máximo de compresión / descompresión de 30 a 40 ms (aunque puede ser menor). Restando a 150 ms deja un retraso máximo de 70 a 90 ms para la transmisión en la red. Asumiendo que una ruta de tres saltos LAN-WAN-LAN es una topología frecuente, y teniendo en cuenta que los elementos de enlace (gateways, routers, etc.) también contribuyen al retraso, nos queda un retraso máximo aceptable de 10 a 15 ms por salto. Aunque estos cálculos son aproximativos y dependerían de muchos otros factores, nos sirve como una aproximación a los problemas de la transmisión.

Otro parámetro que se suele considerar es la varianza de este retraso (delay jitter), es decir, la diferencia entre los retrasos experimentados entre dos paquetes. En determinadas aplicaciones es necesario que los paquetes lleguen con poca variación del retraso, como puede ser el audio. Tener acotada esta varianza del retraso permite calcular la cantidad de memoria necesaria para eliminar esta varianza en la reproducción.

2.4.3.4.- Fiabilidad

Las redes tradicionales proporcionan comunicación fiable entre emisor y receptor. Los protocolos de transmisión tienen sistemas de control de errores y de reenvío de paquetes que aseguran que esta fiabilidad es transparente a los niveles superiores.

Para la transmisión en tiempo real esta gestión de errores puede ser negativa, debido al retraso que produciría la retransmisión de un paquete de nuevo. Para evitar este problema se plantea que el tratamiento y gestión de los errores sea a niveles superiores.

Las prestaciones de una transmisión, por ejemplo multimedia, puede ser medida en dos dimensiones: latencia y fidelidad. La latencia puede ser vital para aplicaciones interactivas como conferencias mientras que la transmisión de una película no lo es. La fidelidad de la transmisión es variable.

Hay aplicaciones que no toleran ninguna variación en la fidelidad de la imagen como podrían ser la transmisión de imágenes médicas y otras en que esta variación solo produce una cierta distorsión tolerable como la transmisión de películas o música.

2.4.3.5.- Sincronización de canales

Cuando audio, vídeo y otros datos vienen por distintos canales, se necesitan mecanismos para la sincronización de los distintos flujos en el destino. Esto se puede conseguir usando una combinación de asignación de tiempos y almacenamiento antes de visualización. Esto en general no afecta a la red y es un problema del destino.

2.4.4 Cómo se implementa la Transmisión en Tiempo Real

2.4.4.1 Introducción

A continuación se describen tres enfoques distintos de cómo implementar redes para la transmisión en tiempo real.

El objetivo de una transmisión en tiempo real es asegurar una calidad de servicio entre emisor y receptor. Esta transmisión normalmente atraviesa varios tipos de redes como se muestra en la Figura 2.1. En esta red se quiere establecer un canal de comunicación entre una computadora conectada a una red privada a un computadora conectada a una pública. Este sería el típico caso de comunicación entre un empleado de una empresa y un cliente desde su casa. La red privada de la empresa, que es una red local, está conectada a la red pública 1 y ésta a la red pública 2 donde está conectado el receptor.

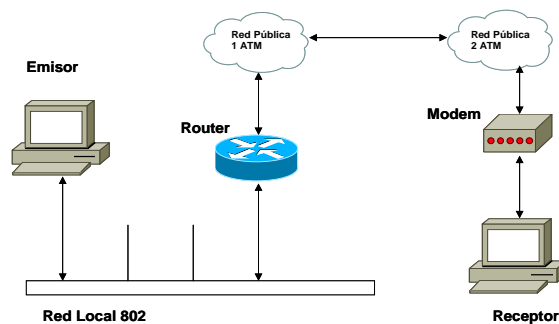


Figura 2.1. Red ejemplo

La calidad de servicio siempre se va a referir de emisor a receptor. Pero ésta va a depender de la calidad de servicio que pueda ofrecer cada subred. Por ello se necesitan mecanismos globales que gestionen esta calidad y negocien con las subredes la calidad de servicio individualmente. Cada subred tiene que proporcionar mecanismos locales de calidad de servicio como puede ser ATM o 802.1p. Así, la

calidad de servicio global será consecuencia de las calidades de servicio negociadas en cada una de las subredes. La gestión de calidad de servicio en cada subred vendrá determinada por el modelo de servicio, que podrá ser de servicios integrados o diferenciados. Este modelo determinará como se gestionan los recursos en la subred y la relación con el resto de las subredes.

Un aspecto importante para gestionar la calidad de servicio es la señalización, es decir, los mensajes que se envían los distintos componentes de la red para gestionar los recursos. Estos mensajes son necesarios para poder garantizar esta calidad de servicio pero al costo de consumir recursos. En este sentido, si se aumenta la señalización entre elementos se puede sobrecargar la red. Una alternativa es que se administre la red para que sólo determinados dispositivos críticos participen en la señalización y el control de admisión. Por ello, es muy importante al diseñar una red en tiempo real, tener en cuenta la sobrecarga que suponen los protocolos que ofrecen calidad de servicio.

2.4.4.2 Componentes comunes para la transmisión en tiempo real

Los servicios de comunicaciones para la transmisión, por ejemplo multimedia, tienen en general dos requerimientos: especificar y asegurar la calidad de servicio y ofrecer soporte para la comunicación de grupos. La comunicación de audio o vídeo requiere la provisión de cierto nivel de calidad (QoS), garantizándolo durante el tiempo de la transmisión. Este nivel de calidad lo tienen que establecer las aplicaciones a la hora de comunicarse. Por lo tanto se tienen que proveer mecanismos para

- ✓ El establecimiento y corte de los canales
- ✓ La negociación de los niveles de calidad entre emisor y receptor, sistemas intermedios y control de red
- ✓ Y el control del nivel de calidad acordado

En general, proporcionar este tipo de servicios es más complejo que implementar una pila de protocolos TCP o OSI. En cambio, cuando se ha establecido la comunicación los requerimientos son menores en conceptos como tratamiento de errores y control de flujo. La implementación de estos sistemas incluye el uso de routers que soporten este tipo de transmisión. Cada router sólo debe admitir nuevas comunicaciones si permiten obtener la calidad requerida.

Las redes basadas en paquetes como IP tienen como objetivo maximizar la utilización de la red por medio de la multiplexación de los canales. Además pueden proporcionar comunicación multipunto, y proporcionar

robustez adaptándose a la dinámica de la red. Sin embargo, este funcionamiento hace que el comportamiento sea difícilmente predecible. En cambio, las redes basadas en conexión proporcionan un servicio garantizado, pero hacen un uso no eficiente de los recursos de la red, no se adaptan a los fallos de la red y no soportan comunicaciones multipunto.

Se puede hablar de una Red de Paquetes de Servicios Integrados (RPSI) en la que se mezclan estos dos paradigmas de comunicación: combinando la comunicación multipunto y multiplexada, la robustez de la red de paquetes conmutados y la garantía de servicio del modelo de conexión.

El desarrollo de este tipo de redes requiere distintos aspectos, que incluyen:

- ✓ **Especificación del flujo:** Una especificación del flujo definiendo el tipo del tráfico, los requerimientos del receptor y la calidad de servicio que se va a proporcionar al flujo.
- ✓ **Encaminamiento:** La red debe decidir como transportar los paquetes desde la fuente al destino. Para ello se requiere un protocolo de encaminamiento que soporte calidad de servicio y rutas unicast y multicast.
- ✓ **Reserva de recursos:** Para mantener un flujo con una calidad de servicio se requiere un protocolo de reserva para crear y mantener las reservas de recursos, como son el ancho de banda o número de buffers.
- ✓ **Control de admisión:** Un algoritmo de control de admisión para mantener la carga de la red a un determinado nivel.
- ✓ **Planificador de paquetes:** Un algoritmo de servicio de paquetes para planificar la transmisión de los paquetes con el objetivo de mantener el servicio garantizado para cada flujo.

2.4.4.3 Modelo conceptual de una red en tiempo real

Se puede modelar una red en tiempo real y concretamente los canales, como una red de flujos, en el que cada canal es un flujo [Cobb97]. El emisor introduce en la red un flujo de entrada $E(t)$ (la información a transmitir) y el receptor recibe un flujo de salida $R(t)$ (la información recibida) como se muestra en la Figura 2.2.

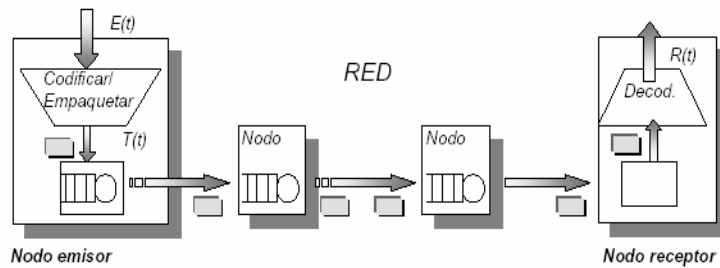


Figura 2.2 Modelo de red en tiempo real

El emisor introduce el flujo en la red a una determinada tasa. Este flujo se cuantifica en paquetes en el emisor. Cuando un paquete de un flujo llega a un nodo (a través de su enlace de entrada) lo pasa al nodo de salida aplicando una disciplina de servicio: si en ese momento no lo puede transmitir lo almacena en una cola, lo cual provocará un retraso en el flujo. Cuando este paquete llega al receptor se decodifica para formar el flujo de salida $R(t)$. Por lo tanto, el almacenamiento del flujo en la red es lo que provoca el retraso de emisor-receptor como se puede ver en la Figura 2.3a.

La función $E(t)$ representa los bits enviados en la red para el tiempo t y $R(t)$ los bits recibidos en la red para el mismo tiempo t . La representación de estas dos funciones (Figura 2.3) es útil para calcular dos valores: el retraso en función del tiempo (el valor d) y la cantidad de la información almacenada en la red (valor b o *backlog*). Una forma de calcular una cota del retraso máximo es obtener el máximo valor de d para cualquier tiempo t .

En el caso de que la red no pueda servir el tráfico introducido a la tasa pedida se producirá un almacenamiento de la red (incremento de b) hasta que se produzca pérdida de información (Figura 2.3b). Si la red puede admitir el tráfico a la misma tasa que la pedida, el almacenamiento será 0 y el retraso sólo dependerá de la latencia de la red (retraso constante) como se muestra en la Figura 2.3c.

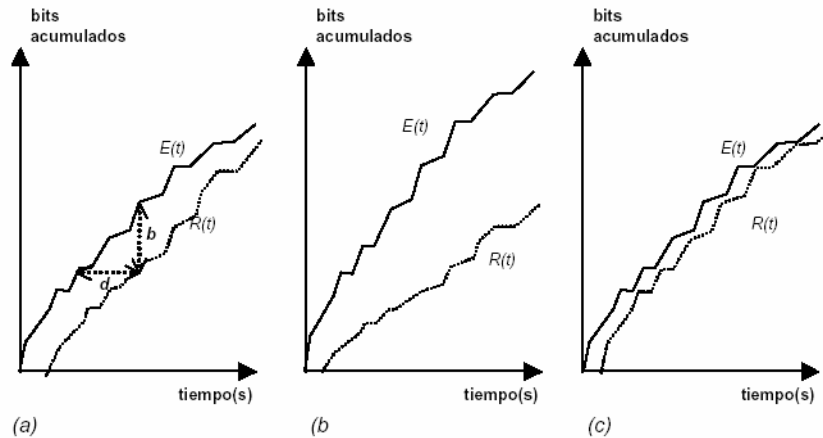


Figura 2.3. Tasa de envío vs Tasa de recepción

2.4.7 Protocolos de red en Tiempo Real

2.4.7.1 Protocolos de comunicación con calidad de servicio

Se pueden distinguir tres niveles de protocolos para ofrecer calidad de servicio [Vogel95][Stuttgen95]:

Protocolos de bajo nivel. ATM es quizá el protocolo más adecuado para la transmisión multimedia por su gestión de la calidad de servicio. Otros protocolos de bajo nivel pueden ser usados con ciertas restricciones como FDDI o Iso-Ethernet [Ross96].

Protocolos de red y transporte. El protocolo IPv4 no es apropiado para la transmisión en tiempo real, aunque IPv6 se ha diseñado para poder soportar este tipo de tráfico. Otros esquemas, como la Suite Tenet, incluyen nuevos niveles de red y transporte con el objetivo de soportar tráfico en tiempo real.

Protocolos de aplicación. En este nivel se han desarrollado distintos protocolos como RSVP (Resource Reservation Protocol) que gestiona el control de admisión y la reserva de recursos. Otros protocolos se encargan de la transmisión y sincronismo de audio y vídeo como RTSP (Real-time Streaming Protocol) o RTP (Real-time protocol)

A continuación se describen tres protocolos, uno representativo de cada nivel. No es el objetivo describir el funcionamiento de estas redes, sino presentar cuáles son las soluciones aportadas en cada protocolo para la transmisión en tiempo real. ATM es una red que está disponible y de la que se describe principalmente la especificación del tráfico y el control

de admisión. La Suite Tenet se creó con el objetivo de investigar la transmisión en tiempo real y no se ha llegado a implantar comercialmente. RSVP es principalmente un protocolo de reserva de recursos que puede ser utilizado sobre y con otros protocolos. En este momento hay varios fabricantes que han anunciado su implementación.

2.4.7.2 Tenet suite

2.4.7.2.1.- Introducción

Los protocolos Tenet están diseñados para coexistir con los protocolos Internet. Los canales en tiempo real son establecidos en la fase de conexión que precede a la transferencia de datos. Un mensaje es enviado desde el origen del canal y viaja al destino, provocando que en cada nodo se ejecuten unos test de admisión para comprobar si se puede obtener la calidad de servicio requerida. Cuando llega el mensaje al destino este envía un mensaje de aceptación de canal que llegará al origen estableciéndose el canal.

2.4.7.2.2 El diseño de los protocolos Tenet

La suite Tenet está formada por la siguiente pila de protocolos (Ver Figura 2.4 y Tabla 2.2):

TCP	UDP	RMTP	CMTP	R	R
IP		RTIP		C	T
Red de transmisión (FDDI, ATM, etc.)				A	C
				P	M
					P

Figura 2.4. Pila de protocolos de Tenet Suite

Módulo		Descripción
RMTP	Real time Message Transport Protocol	Protocolo de transporte orientado a mensaje.
CMTP	Continuos Media Transport Protocol	Protocolo de transporte orientado a flujo.
RTIP	Real time internet protocol	Protocolo de red en tiempo real.
RCAP	Real time Channel Administration Protocol	Protocolo de establecimiento y control de canales.
RTCMP	Real time Control Message Protocol	No implementado.

Tabla 2.2. Descripción de los módulos del protocolo Tenet Suite

2.4.7.2.3 Establecimiento y control de canales

El módulo RCAP (Real-time Channel Administration Protocol) proporciona los servicios de control. Su función principal son el establecimiento y liberación de los canales en tiempo real.

El establecimiento de un canal se realiza en una sola pasada. Un mensaje establish_request se envía desde el origen al destino del canal. Cada entidad RCAP mantiene una tabla de rutas para calcular el próximo salto para el establecimiento del canal. Si el canal puede ser soportado por este nodo (mediante la realización de un test de control de admisión) se reservan de forma provisional los recursos necesarios y el mensaje se envía al siguiente nodo. Si el nodo determina que no puede soportar este canal se envía un mensaje establish_denied de vuelta al origen. Este mensaje libera los recursos reservados por cada nodo por el que pasa de vuelta. Cuando el mensaje establish_request alcanza el destino, este toma la última decisión de aceptar o rechazar el canal. Si se acepta el canal se envía de vuelta al origen el mensaje establish_accept.

Cuando este mensaje llega a cada nodo intermedio el RCAP local puede reducir la asignación de recursos realizada en la ida del mensaje establish_request. Cuando la asignación definitiva de recursos se realiza se informa al agente RTIP de que se ha establecido un nuevo canal y se envía el mensaje al nodo siguiente de vuelta. Cuando este mensaje llega al origen la transmisión sobre el canal puede comenzar.

Además existen mensajes para consultar el estado de un canal en un nodo y para cerrar un canal tanto desde el emisor como el receptor. Los mensajes del RCAP son los siguientes:

<i>Path message</i>	<i>Sortido</i>	<i>Descripción</i>
Establish request	Ida	Petición de establecer un nuevo canal.
Establish accept	Vuelta	Aceptación de un canal.
Establish denied	Vuelta	Se ha rechazado el establecimiento de un canal.
Status request	Ida	Petición del estado de un canal en un nodo.
Status report	Vuelta	Retorna los datos recolectados por un mensaje status request.
Close request forward	Ida	Cierra un canal a petición del origen.
Close request reverse	Vuelta	Cierra un canal a petición del destino.

Tabla 2.3. Mensajes de Tenet

Especificación del tráfico

El cliente debe establecer los requerimientos de calidad de servicio y una descripción del tráfico que va a transmitir en la red. Los parámetros se muestran en la Tabla 2.4, donde, como se ve, el tráfico satisface el modelo Tenet (Xmin,Xave, I, Smax)

Parámetros de QoS	Descripción
D_{\max}	Límite superior de retraso del mensaje de emisor a receptor
Z_{\min}	Límite inferior de retraso del mensaje de emisor a receptor
J_{\max}	Límite superior de la variación del retraso (delay jitter)
W_{\min}	Límite inferior en la probabilidad de no pérdida debido a una sobrecarga de los buffers.
Parámetros del tráfico	
X_{\min}	Tiempo mínimo entre mensajes
X_{me}	Tiempo medio entre mensajes
T	Intervalo medio
S_{\max}	Tamaño máximo del mensaje

Tabla 2.4. Parámetros de calidad de servicio y tráfico

La implementación de este agente es a través de un *daemon*. Para la transmisión de mensajes el RCAP utiliza TCP/IP para asegurar que los mensajes llegan a su destino correctamente. Esto implica que los tiempos de conexión no son determinados.

2.4.7.2.4 Transmisión y planificación de paquetes

RTIP (Real- Time Internet Protocol) es el nivel de red de la Tenet Suite. Su principal función es la de transportar los paquetes para cumplir los requerimientos del correspondiente canal. En contraste con IP este protocolo está orientado a la conexión. RTIP no es fiable debido a que los paquetes se pueden corromper o perder debido a sobrecarga de buffers (sólo si $W_{\min} < 1$). RTIP también realiza la planificación de los paquetes basándose en los parámetros de calidad de conexión de cada canal. Ya que todos los paquetes de una conexión RTIP siguen el mismo camino, los paquetes llegan siempre en el mismo orden.

La cabecera RTIP es de tamaño fijo para permitir un rápido proceso (Figura 2.5). Para poder trabajar con IP los cuatro primeros bits de la cabecera identifican el paquete como RTIP.

0	4	8	16	24
RTIP	Sin uso		Local ID	
Longitud del paquete			Número de secuencia	
Timestamp				
Reservado			Header checksum	

Figura 2.5. Formato del paquete de Tenet

Campo	Descripción	Tamaño
Versión	Contiene identificador de paquete RTIP	4
Local ID	Identificador del canal	16
Longitud del paquete	Número de octetos del paquete	16
Número de secuencia	Número de secuencia del paquete	16
Timestamp	Tiempo en el que el paquete se recibió por el módulo RTIP del emisor.	32
Header checksum	Sólo hay <i>checksum</i> de la cabecera ya que no se asegura la integridad de los datos.	16

Tabla 2.5. Descripción de los campos de un paquete Tenet

La entidad RTIP en cada nodo tiene el objetivo de asegurar que todos los paquetes son transportados para que cumplan sus requerimientos de calidad. Cuando se establece una conexión RTIP es informado del máximo retraso “d” que puede tener un paquete en ese nodo. TIP asegura que un paquete llegado al nodo en un tiempo “T” es transmitido al siguiente nodo antes de $T + d$. RCAP informa además de la cantidad de espacio de buffer asignado a la nueva conexión y del tiempo mínimo y medio entre mensajes. (X_{min} , X_{ave} y I).

Cada entidad RTIP contiene dos módulos:

- ✓ Módulo de control de tasa: Este módulo controla el tráfico de cada conexión y lo ajusta de acuerdo con la especificación del tráfico. Así si un paquete no se esperaba antes del tiempo “ T_e ”, y llega antes, entonces el paquete no se convierte en elegible hasta el tiempo T_e . Cuando es elegible se transfiere al modulo planificador.
- ✓ Módulo planificador: Se encarga de planificar los paquetes elegibles, asegurando que se cumplan los *deadlines*. Este módulo también se encarga de llevar estadísticas.

La implementación de este agente se realiza en el kernel del sistema y coexiste con TCP, UDP y IP. Su acceso es a través de un nuevo tipo de socket IPPROTO_RMTP. Se han implementado las siguientes disciplinas de servicio: Delay-Earliest-Due Rate, Jitter-Earliest- Due-Rate y Rate-Controlled Static Priority, aunque se puede implementar cualquier otra disciplina debido a su diseño modular

2.4.7.2.5 Control de admisión

Ferrari introdujo en 1989 un control de admisión para nuevos canales basado en el protocolo EDD [Ferrari89] que es el que se detalla en este punto utilizando el modelo Tenet de tráfico. El control de admisión de nuevos canales se realiza en los nodos y el receptor. Hay dos tests en cada nodo que comprueban si pueden dar la calidad de servicio pedida sin afectar al resto de los canales, y otro test en receptor que comprueba si el retraso total está por debajo del exigido.

En general, el funcionamiento de los tests es el siguiente: en los nodos se comprueba si se puede admitir el nuevo canal y se calcula el retraso mínimo que se puede dar al canal. Con la información de todos estos retrasos mínimos el receptor comprueba si se puede garantizar el retraso máximo exigido. Si es así, distribuye el retraso para cada nodo, aceptando el canal. Este esquema funciona si las disciplinas de servicio son nonwork- conserving, en las que el retraso total de un paquete se puede obtener como la suma de los retrasos individuales en cada nodo.

Este apartado se va a centrar en los tests deterministas. Existen otros tests para un servicio estadístico que son más complejos, salen del alcance de esta tesis.

2.4.7.2.5.1 Test en los nodos

El objetivo de los tests en los nodos es asegurar que se puede dar servicio al canal sin afectar a la calidad de servicio del resto de canales. Cuando un nodo recibe una petición de establecimiento de canal se realizan dos tests en los nodos. El primero comprueba que se puede aceptar el canal y el segundo determina el retraso máximo del paquete permitido en el nodo.

El primero, denominado **Test determinista** verifica que hay suficiente poder de proceso para acomodar el nuevo canal sin perturbar al resto. Como la máxima utilización de un nodo por el canal i , cuyos paquetes tienen un tiempo de servicio máximo de t_i , es $t_i/x_{min,i}$, la condición es:

$$\sum_j t_j/x_{min,j} \leq 1 \quad (1)$$

Donde la suma se extiende para todos los canales deterministas, incluido el que se establece.

El segundo, denominado **Test del retraso límite** determina si se puede evitar la saturación del planificador en el nodo, y si es así, el mínimo retraso a asignar al canal que se va a establecer para que el objetivo se cumpla.

Para determinar si la saturación del planificador en el nodo es posible se dividen los m canales del nodo en dos conjuntos: A es el conjunto de canales cuyo retraso límite en el nodo es menor que la suma de los tiempos de servicio de todos los canales, y B es el conjunto de aquellos canales cuyo retraso límite del nodo es mayor o igual que esa suma:

$$A = \{ i \mid i = 1, \dots, a; d_i < \sum_{j=1}^m t_j \} \quad (2)$$

$$B = \{ i \mid i = a+1, \dots, m; d_i \geq \sum_{j=1}^m t_j \}, \quad (3)$$

Se numeran los a canales en A de acuerdo al orden en que se van a planificar. Asumiendo que:

$$x_{min, l} \geq \sum_{j=1}^m t_j \quad (l=1, \dots, m) \quad (4)$$

Que es lo mismo que decir que no llegarán más paquetes a un nodo que puedan interferir con la transmisión de aquellos que estamos considerando, entonces la saturación del planificador es imposible si y sólo si:

$$d \geq \sum_{j=1} t_j + \max_{i < k \leq m} t_k, \quad (i = 1, \dots, a) \quad (5)$$

Si la condición (4) no se cumple, se aplicará la siguiente fórmula al conjunto de canales que incluyen paquetes seguidos. Considerando que estos paquetes incrementan el $\sum t$, se calcula el nuevo valor de esta suma ($\sum t$), con la siguiente definición recursiva:

$$(\sum t)_o = \sum t,$$

$$(\sum t)'_{n+1} = (\sum t)'_{n+1} + Y_k t_k, \quad (h = 0, 1, \dots) \quad Y_k = 1 \quad [x_{min,k} + d_k < (\sum t)'] \quad (6)$$

$$0$$

$$\sim$$

$$\cdot$$

Con la ecuación (5) se puede calcular el mínimo valor d_i con el que se cumple la condición de no saturable. Este valor se enviará al nodo receptor.

2.4.7.2.5.2 Tests en el receptor

Cuando llega al receptor el mensaje de establecimiento de canal se realizan dos tests. El test del retraso comprueba si no se sobrepasa el retraso total:

$$D_i \geq \sum_n \min d_{i,n} + \sum_n l_{i,n} \quad (7)$$

Si se pasan todos los tests en los nodos, el receptor divide el retraso D entre los nodos atravesados por el canal, después de restar el retraso total en los enlaces de la ruta. Sea $d_{i,n}$ el mínimo retraso asignado al canal i en el nodo n . Un paquete viajando en este canal y llegando al nodo en el tiempo t se le asignará un retraso máximo de $t + d_{i,n}$. Para satisfacer el retraso total, es suficiente satisfacer el límite d en cada nodo a través de la ruta. Los valores D_i se pueden redistribuir utilizando la siguiente fórmula:

$$d_{i,k} = (\sum_n \min d_{i,n} - \sum_n l_{i,n}) / n + \min d_{i,k} \quad (8)$$

2.4.7.3 Modo de Transferencia Asíncrona (ATM)

2.4.7.3.1 Introducción

La tecnología ATM es una arquitectura de conmutación de celdas que utiliza la multiplexación por división en el tiempo asíncrona. Las celdas son las unidades de transferencia de información en ATM y se caracterizan por tener un tamaño fijo de 53 bytes. Esto permite simplificar los nodos y que la conmutación sea realizada por hardware, consiguiendo con ello alcanzar altas velocidades.

El soporte de múltiples servicios con diferentes calidades de servicio (QoS) y características del tráfico, requiere una extensa y compleja infraestructura de red.

La tecnología de red ATM estandariza un conjunto completo de funciones para la gestión del tráfico que pueden ser implementados con varios niveles de complejidad para obtener diferentes niveles de eficiencia [Giroux99].

Las redes ATM pueden interoperar con los equipos de distintos vendedores.

La red define tres niveles:

- ✓ **Nivel físico.** Se han definido distintos medios de transmisión como fibra óptica a 155-Mbps, o 100-Mbps FDDI para ATM de área local, además de otras opciones. Actualmente, las interfaces a ATM trabajan de 622 Mbps o incluso a 2.4 Gbps.
- ✓ **Nivel ATM.** El nivel ATM es una capa independiente para la conmutación y el multiplexado de los paquetes. Se define la estructura de las celdas como contenedores de información de 53 octetos (5 octetos para cabecera más 48 de datos). El encaminado se basa en los identificadores de circuito virtual (VCI: virtual circuit identifiers) y los identificadores de camino virtual (VPI: virtual path identifiers). Además, ATM es una red orientada a la conexión. Debido al pequeño tamaño de las celdas ATM, la alta velocidad de los enlaces y la velocidad de conmutación de los nodos, ATM proporciona unas latencias mínimas.
- ✓ **Nivel de adaptación ATM (AAL).** Este nivel está diseñado como puente entre el nivel ATM y de aplicación.

2.4.7.3.2 Componentes de la gestión del tráfico

Una aplicación negocia el **(1) contrato del tráfico** con la red para cada conexión virtual. Este contrato es un acuerdo entre el comportamiento del tráfico que se va a transmitir y el nivel de servicio que se requiere para la conexión. Cuando el tráfico está definido, la red aplica un **(2) control de admisión de la conexión** (CAC) para decidir si se puede admitir con la calidad de servicio exigida con los recursos disponibles en la red. Si la conexión es aceptada, la transmisión puede comenzar.

Para garantizar la calidad de servicio durante la transmisión, la red tiene que asegurar que el tráfico de la conexión cumple el contrato. Para ello, la red puede aplicar **(3) funciones de conformidad** para asegurar que un tráfico que no cumpla el contrato no afecte al resto de las conexiones. El tráfico puede ser encolado antes de ser transmitido.

Las colas son servidas de acuerdo a **(4) algoritmos de planificación** específicamente diseñados para cumplir la calidad de servicio de los distintos tipos de conexiones. Aunque se ha realizado un control de admisión puede ocurrir congestión en la red. El **(5) control de la congestión** gestiona que se eliminen las celdas de una manera no traumática.

2.4.7.3.3 El contrato del tráfico

Una aplicación que use una red ATM acuerda sus requerimientos de ancho de banda y prestaciones por medio de un contrato del tráfico. Este contrato incluye los siguientes componentes:

- la categoría del servicio,
- la calidad de servicio requerida,
- las características del tráfico de la conexión,
- la definición de cómo el tráfico se debe comportar (modelo del tráfico).

2.4.7.3.4 Categorías de servicio

Se definen 5 categorías de servicio):

2.4.7.3.4.1 Servicio CBR

El servicio CBR (Constant Bit Rate) está diseñado para soportar aplicaciones en tiempo real. Proporciona una conexión con ancho de banda dedicado, con muy baja probabilidad de pérdida de celdas, así como retraso bajo y predecible. El tiempo entre celdas es constante y está caracterizado por una tasa pico (PCR: Peak Cell Rate).

2.4.7.3.4.2 Servicio VBR

El objetivo del servicio VBR (Variable Bit Rate) es hacer más eficiente el soporte de aplicaciones de vídeo u otras aplicaciones con tráfico a ráfagas. El tráfico está caracterizado por una tasa de tráfico sostenida (SCR: Sustained Cell Rate) y la tasa pico (PCR). El valor SCR es medido sobre un periodo específico y representa la tasa de transmisión media. Hay dos subcategorías de servicio: rt-VBR y nrt-VBR. El tráfico rt- VBR tiene requerimientos estrictos de retraso. El tráfico nrt-VBR no tiene estos requisitos y se puede almacenar en la red (buffering).

2.4.7.3.4.3 Servicio ABR

El objetivo del servicio ABR (Available Bit Rate) es que los nodos terminales participen activamente en la gestión del tráfico. La red informa de cual es su estado y capacidad y los nodos terminales transmiten de acuerdo a esta información intentando evitar pérdidas de celdas.

2.4.7.3.4.4 Servicio UBR

Las conexiones UBR (Unspecified Bit Rate) comparten el ancho de banda no usado por el resto de tipos de servicio, sin ningún tipo de feedback por parte de la red. De esta forma la aplicación accede a todo el ancho de banda que la red pueda proporcionar pero sin ninguna garantía acerca de la tasa de pérdida de celdas. Es un servicio tipo “*best-effort*”.

Los tipos de servicio más interesantes para transmisión en tiempo real son CBR y rt-VBR.

2.4.7.3.5 Parámetros de calidad de servicio

La calidad de servicio en ATM está definida por un conjunto de parámetros requeridos para la conexión entre emisor y receptor. Los siguientes tres parámetros son negociados entre la aplicación y la red:

- ✓ cell loss ratio (CLR),
- ✓ maximum cell transfer delay (Max-CTD), y
- ✓ peak-to-peak cell delay variation (P2P-CDV).

y otro tres son impuestos por la red:

- ✓ cell error ratio (CER),
- ✓ severely errored cell block ratio (SECBR), y
- ✓ cell misinsertion rate (CMR).

Se van a describir únicamente los parámetros de retraso. El *cell transfer delay* se define como el tiempo transcurrido entre el envío de una celda por parte del emisor hasta su recepción en destino. Este retraso es la acumulación de los retrasos acumulados en cada nodo y enlace de la red.

Max-CTD representa el $(1 - \alpha)$ cuantil de la función de densidad del CTD, donde las celdas que exceden el máximo retraso son consideradas perdidas (véase la Figura 2.6). El ajuste de α lo realiza la red, pero se escoge para que no supere el valor CLR (*cell loss ratio*).

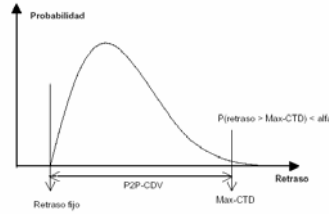


Figura 2.6 Densidad de la probabilidad del retraso

2.4.7.3.6 Modelo del tráfico

Una conexión ATM caracteriza el tráfico usando los siguientes parámetros: *peak cell rate* (PCR),

- ✓ *sustainable cell rate* (SCR) y *maximum burst size* (MBS),
- ✓ *minimum cell rate* (MCR), y
- ✓ *maximum frame size* (MFS).

Cada conexión tiene dos conjuntos de descriptores para el tráfico de entrada y salida.

PCR representa la tasa pico de emisión de la fuente. Su inversa representa el tiempo mínimo entre paquetes. La unidad de PCR es celdas/segundo. La Figura 2.7 representa un PCR contratado de 1/3 de la tasa de la línea.



Figura 2.7. Tráfico CBR

SCR es un límite superior de la tasa media de transmisión de una conexión ATM. La inversa de SCR representa un límite superior del tiempo entre celdas a largo plazo con respecto a la velocidad del enlace. El valor SCR se especifica siempre junto al valor MBS.

El valor **MBS** representa el factor de ráfagas (*burstiness*) de la conexión. Especifica el máximo número de celdas que pueden ser transmitidas por una fuente a la tasa pico (PCR) cumpliendo el valor SCR negociado. La Figura 2.8 representa una transmisión con un PCR = $\frac{1}{2}$ de la tasa de enlace y SCR = $\frac{1}{4}$ con un MBS de 5 celdas. La conexión puede enviar como máximo cinco celdas en diez unidades de *slot*. Alternativamente puede enviar una celda cada 4 pero sin aprovechar la capacidad de emitir en ráfagas.

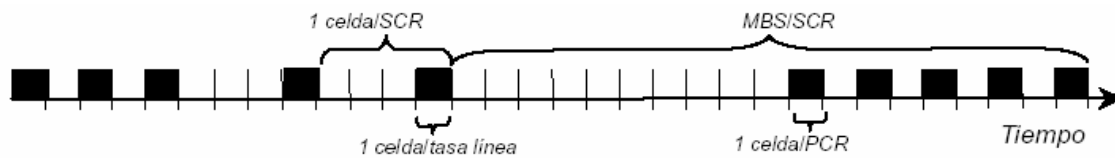


Figura 2.8. Tráfico VBR

Los parámetros MCR y MFS son usados en los tráficos GFR y ABR. La tabla 2.6 define los parámetros necesarios para cada tipo de servicio [ForumATM96].

Atributos	CBR	RT-VBR	NRT-VBR	ABR	UBR
PCR	✓	✓	✓	✓	✓
SCR,MBS		✓	✓		
MCR				✓ (Opcional)	
CLR	✓	✓	✓		
Max-CTD	✓	✓			
P2P-CDV	✓	✓			

Tabla 2.6 Parámetros para cada tipo de servicio

2.4.7.3.7 Ajuste del tráfico

La reserva de recursos en la red no es suficiente para asegurar la calidad de servicio. Si una conexión excede el tráfico contratado puede incidir en su calidad de servicio y lo que es peor en la calidad de servicio de otras conexiones. Para evitar esto, la red tiene que asegurar que el tráfico se ajuste a los parámetros contratados. Esta capacidad se obtiene por la implementación de los mecanismos de conformación y comprobación del tráfico (*traffic-shaping*, *traffic-policing*, y *soft-policing*).

Para ello se utiliza un algoritmo denominado Generic Cell Rate Algorithm (GCRA). Cuando una celda no se ajusta al tráfico

especificado puede ser degradada (baja su clase), descartada o aceptada (si no influye al resto de la conexiones). Para evitar la pérdida de celdas, la fuente de tráfico realiza un conformado del tráfico, lo que implica el retraso de ciertas celdas hasta que se ajusten a las especificaciones. El algoritmo GCRA puede ser implementado por medio de un algoritmo leaky bucket o como un algoritmo virtual-scheduling.

2.4.7.3.8 Control de admisión de conexiones

Una conexión ATM atraviesa un conjunto de nodos en la red. Para verificar la admisión de una conexión, el CAC (connection admission control) sigue el siguiente procedimiento en cada punto de la red:

1. Mapea la descripción del tráfico asociada con la conexión en un modelo de tráfico.
2. Usa este modelo de tráfico con un modelo apropiado de la cola del nodo para estimar si hay suficientes recursos para admitir la nueva conexión con la calidad de servicio pedida.
3. Reserva los recursos si son suficientes y admite la conexión. Dependiendo de los modelos de tráfico usados, los procedimientos del CAC pueden ser demasiado conservativos y reservar demasiados recursos. Esto reduce la ganancia estadística que se pueda obtener. La ganancia estadística se define como:

$$\text{Ganancia Estadística} = \frac{\text{No de conexiones admitidas con multiplexación estadística}}{\text{No de conexiones admitidas con tasa pico}}$$

Dado que las conexiones no envían continuamente datos, es posible reservar menos recursos cuando las conexiones son multiplexadas en un punto. Esto significa que se pueden admitir más conexiones que si se usa el valor PCR en cada conexión.

Un CAC eficiente debe intentar obtener la máxima ganancia estadística posible sin afectar la calidad de servicio. Las categorías de servicio CBR y rt-VBR tienen requerimientos de retraso. Esto es, el retraso en un nodo está acotado, lo que fuerza que el tamaño de los buffers sean pequeños y la congestión de celdas prevalece para estos servicios. En consecuencia es difícil obtener multiplexación estadística para estas clases.

Normalmente el CAC para el CBR podría utilizar la siguiente ecuación:

i

$$\sum_i PCR \leq \text{Capacidad del enlace}$$

Este algoritmo CAC está basado en la tasa pico, pero dado que el tráfico no conserva las características se podrían perder celdas. En cambio, para el tráfico VBR la tasa SCR/PCR define el factor de ráfagas y tiene un gran impacto en la ganancia estadística. Si este ratio es bastante menor que uno puede resultar en una utilización pobre del enlace. Por tanto, el CAC admite conexiones si:

$$\sum_i a_i \leq \text{Capacidad del enlace}$$

i

Donde $SCR_i \leq a_i \leq PCR_i$ para cada conexión i . La ganancia estadística puede ser definida como el ratio $\square PCR_i / \square a_i$. El valor a_i se llama ancho de banda efectivo o ancho de banda equivalente. Este valor depende de las propiedades estadísticas de la conexión a admitir así como de las colas de la red. En general, para una conexión es intuitivo pensar que el ancho de banda efectivo estará más cerca del valor PCR cuando el tamaño de las colas es reducido y más cerca de SCR cuando las colas son mayores.

Dos propiedades constituyen las principales ventajas de este método:

1. Propiedad de adición: El ancho de banda efectivo total necesario para N conexiones es igual a la suma de los ancho de banda efectivos de cada conexión.
 2. Propiedad de independencia: El ancho de banda efectivo para una conexión dada es sólo una función de los parámetros de la conexión. Estas dos propiedades hacen que este método sea ampliamente usado en las tecnologías ATM. Existen varios métodos para calcular el valor de a_i como el *Rate Envelope Multiplexing* (para tamaños de colas pequeños) y *Rate Sharing* [Kelly91].
- ✓ Aparte de estos métodos analíticos de control admisión, existen también métodos dinámicos que se basan en el estado real de la red para determinar la admisión de un nuevo canal.

2.4.7.4 *RSVP (Resource ReSerVation Protocol)*

2.4.7.4.1 Introducción

RSVP se ha diseñado para permitir a los emisores, receptores y *routers* de las sesiones de comunicación (tanto *multicast* como *unicast*) comunicarse con el resto para establecer una ruta que pueda soportar la calidad de servicio requerida. RSVP identifica una sesión por medio de una dirección de destino, un tipo de protocolo de transporte y un número de puerto de destino. RSVP no es un protocolo de encaminamiento; se usa meramente para reservar recursos a través de la ruta que se establezca por cualquiera de los protocolos de niveles inferiores.

Hay una extensión de RSVP para su uso en redes compartidas como LAN, que es denominada SBM (*Subnet Bandwidth Manager*). El problema en este tipo de redes compartidas es que el protocolo RSVP falla. Esto es debido a que sus mensajes deben pasar por varios dispositivos usando el nivel-2 (puentes, etc) que no conocen el protocolo RSVP de nivel 3 y superiores, con lo que no se pueden gestionar los recursos.

Aunque RSVP en principio es sólo un protocolo de reserva de recursos se suele asociar a las especificaciones de flujos definidas por el grupo *IntServ*, así como su control de admisión.

2.4.7.4.2 Objetivos de diseño

RSVP se ha diseñado con los siguientes objetivos:

1. Proporcionar la posibilidad de que **receptores heterogéneos** puedan hacer reservas de acuerdo a sus necesidades. No se debe asumir que todos los receptores tienen las mismas capacidades ni que requieran la misma calidad de servicio.
2. Debe adaptarse a las **variaciones de miembros** en grupos multidifusión. La conexión o desconexión de los miembros de un grupo debe ser dinámica.
3. Permitir a los usuarios **especificar sus necesidades** a nivel de aplicación para que los recursos reservados para un grupo multidifusión puedan reflejar con precisión los recursos necesitados por el grupo.

4. Permitir a los receptores **seleccionar entre varios canales**. El receptor debería poder seleccionar entre varias fuentes sin el riesgo de que la petición de este cambio fuera denegada, como podría ocurrir si se estableciese una nueva petición. Esto es posible si los recursos reservados por el receptor son reutilizados para la transmisión de esta fuente.
5. Debe **adaptarse a los cambios en las rutas** uni y multidifusión. RSVP no es un algoritmo de encaminamiento, utiliza el nivel de red para estos propósitos, pero si debe mantener un estado de las rutas.
6. **Controlar la sobrecarga** que produce el protocolo en la red para que no crezca linealmente o con el número de participantes.
7. Hacer el **diseño modular** para acomodar distintas tecnologías.

2.4.7.4.3 Principios de diseño: Para obtener los objetivos vistos en el punto anterior, el diseño de RSVP se basa en seis principios básicos:

1. **Reserva iniciada por el receptor.** Los receptores escogen el nivel de servicio requerido y son responsables de iniciar y mantener la reserva activa mientras quieran recibir datos. Esto es así, porque el receptor es quien conoce sus limitaciones y la calidad de servicio que recibe. Además, esto permite la gestión de peticiones heterogéneas.
2. **Filtro de paquetes.** La reserva de recursos en un *router* asigna ciertos recursos a la entidad que hace la reserva, pero no determina qué paquetes pueden usar estos recursos. Hay una función separada, llamado filtro de paquetes, que selecciona los paquetes que pueden usar estos recursos. Este filtro puede ser estático o dinámico y permite establecer varios modelos de reserva.
3. **Proporcionar varios estilos de reserva.** Por medio del filtro de paquetes se pueden definir diferentes modelos de reserva. Actualmente existen tres estilos: libre, filtro fijo y filtro dinámico.
4. **Mantener un estado (“soft-state”) de la red.** Durante una comunicación larga es posible que nuevos miembros se unan al grupo mientras que otros lo dejen, y las rutas pueden modificarse debido a cambios en la red. Por eso, RSVP debe mantener un estado de la red. Esta información se mantiene por medio de mensajes que periódicamente se envían para refrescar el estado. RSVP distingue dos clases de información en cada *router*: el estado de la ruta y el estado de la reserva.

5. **Control de sobrecarga del protocolo.** La sobrecarga de RSVP se determina por tres factores: el número de mensajes RSVP enviados, el tamaño de estos mensajes y las frecuencias de refresco de los mensajes de ruta y reserva. Para reducir la sobrecarga, RSVP funde los dos mensajes mientras atraviesan la red.

6. **Modular.** RSVP tiene interfaz con otros tres componentes en la arquitectura: (1) la especificación de flujo (flowspec) que se maneja a nivel de aplicación o sesión; (2) el protocolo de encaminamiento de red, que lleva los mensajes hasta los receptores; (3) el control de admisión en red, que realiza las decisiones basadas en el flowspec que está en los mensajes de reserva.

En la Figura 2.9. se muestra como encaja RSVP dentro de los *routers*, emisores y receptores.

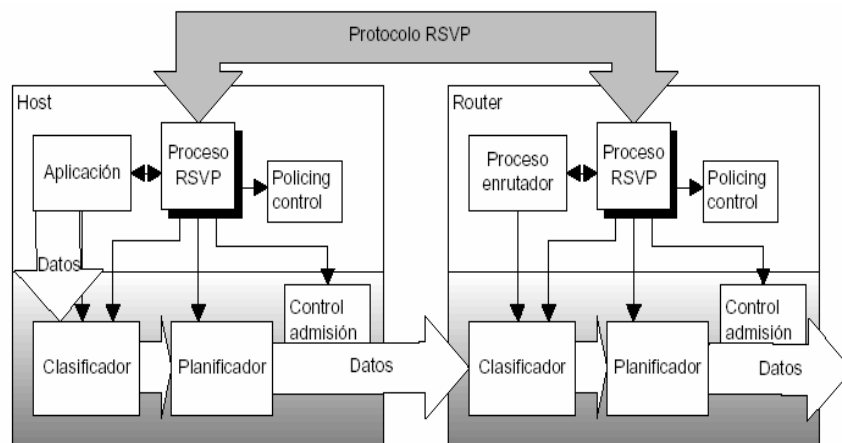


Figura 2.9 Módulo RSVP dentro del emisor y el router.

2.4.7.4.4 Clases de calidad de servicio

IETF ha considerado varias clases de calidad de servicio, aunque sólo dos han sido formalmente especificadas para RSVP: Servicio Garantizado (*Guaranteed Service*) [Schenker96] y Servicio de Carga Controlada (*Controlled-Load Service*) [Wroclawski96].

2.4.7.4.4.1 Servicio garantizado

Esta calidad de servicio está destinada para aplicaciones con requerimientos exigentes de tiempo real. Esta calidad asegura: un ancho de banda, un límite en el retraso y ninguna pérdida en las colas.

Como introducción, un flujo es descrito usando un modelo *token bucket* y dada esta descripción, cualquier elemento de la red (un *router*, una subred, etc.) calcula varios parámetros describiendo cómo va a manejar los datos del flujo. Combinando los parámetros de los distintos elementos que recorre el flujo, es posible calcular el retraso máximo que se producirá en el flujo.

Cada *router* caracteriza el servicio garantizado para un flujo determinado, asignando un ancho de banda R , y un espacio de memoria (*buffer space*) B , que representa los recursos que el flujo puede consumir. Esto representa que existe un ancho de banda R entre emisor y receptor. Así, para un servicio que siga el modelo de flujo perfecto con un cubo de capacidad b y tasa r , se puede calcular el límite del retraso como b/R siempre que R sea mayor r . Para permitir desviaciones sobre este modelo perfecto se añaden dos términos de error C y D ; con lo que el límite del retraso se convierte en $b/R + C/R + D$.

Estos errores aparecen al trabajar con paquetes. Por ejemplo, cualquier paquete puede experimentar un retraso debido a paquetes de su propia cola o debido a imprecisiones del planificador. El término C es el error dependiente de la tasa y representa el retraso que un paquete en el flujo puede experimentar debido a la tasa reservada. El término de error D es el error independiente de la tasa y representa el peor caso de variación de tiempo de tránsito a través del *router*.

Sin embargo, con el servicio garantizado se impone una cota superior a la tasa de transmisión que es la tasa pico del flujo p , que reduce los límites del retraso. Además, se tienen que tener en cuenta los efectos de la partición en paquetes en el flujo considerando el tamaño máximo del paquete, M . Esto nos permite disponer de un límite más preciso para el retraso:

$$Q_{delay} = \frac{(b-M)(p-R)}{R(p-r)} + \frac{(M+C_{tot})}{R} + D_{tot} \quad (\text{caso } p > R \geq r) \quad (9)$$

$$Q_{delay} = \frac{(M+C_{tot})}{R} + D_{tot} \quad (\text{caso } R \geq p \geq r) \quad (10)$$

Donde C_{tot} y D_{tot} representan el sumatorio de los términos de error C y D de cada *router* de la ruta. Es importante fijarse que la ecuación (9) no depende de los parámetros del flujo, debido a que como la reserva R es mayor que la tasa pico de la carga p no hay posibilidad de que los paquetes se encolen en la red.

Para que este retraso se cumpla se tiene que cumplir que el tráfico entrante esté conforme al algoritmo *token bucket*, es decir, el tráfico está limitado por la siguiente ecuación $\min[pt, b+rt]$.

Cada *router* necesita ser informado de las características del tráfico, T_{spec} , y del flujo con las características de las reservas realizadas, R_{spec} . Además, necesita los términos C_{sum} y D_{sum} que representan la suma de los términos de error C y D de cada *router* desde el origen del mensaje. Los parámetros T_{spec} y R_{spec} están enumerados en la Tabla 2.7.

Parámetro T_{spec}	Descripción	Unidad
p	Tasa pico del flujo.	Bytes/s
b	(<i>bucket depth</i>) Tamaño del cubo.	Bytes
r	(<i>token bucket rate</i>)Tasa de transmisión del cubo.	Bytes/s
m	Tamaño mínimo de un paquete.	Bytes
M	Tamaño máximo de paquete.	Bytes
Parámetros R_{spec}		
R	Ancho de banda.	Bytes/s
S	<i>Slack term</i> . Diferencia entre el retraso deseado y el obtenido usando una reserva de ancho de banda R.	μs

Tabla 2.7. Parámetros T_{spec} y r_{spec}

Para utilizar este esquema los *routers* tienen que aproximarse al modelo de flujo. Para ello, se pueden utilizar varios algoritmos de planificación como por ejemplo el visto en líneas anteriores: el WFQ (*Weighted Fair Queueing*).

Concretamente, con el planificador WFQ, los parámetros C_i y D_i pueden ser calculados de la siguiente forma: D_i es igual al *MTU* (*Maximum Transmission Unit*) del enlace dividido por el ancho de banda del enlace, con la condición de que M debe ser menor que el mínimo *MTU* del path. El valor de C_i se asume que es M para considerar la fragmentación en paquetes. Esto es:

$$D_{tot} = \sum D_i = \sum \frac{MTU_i}{AB_i} \quad C_{tot} = \sum C_i = \sum M \quad (11)$$

Respecto al tamaño del *buffer* a reservar en los nodos, el tamaño necesario en cada nodo es $b+C_{sum}+D_{sum}$, donde C_{sum} y D_{sum} son la suma de todos los parámetros C_i y D_i anteriores al nodo.

2.5 Calidad de Servicio. Tipos de tráfico

2.5.1 Introducción

En este documento se va a emplear la siguiente definición de calidad de servicio [Vogel95]:

“Calidad de servicio representa el conjunto de las características tanto cuantitativas como cualitativas de un sistema distribuido necesarias para alcanzar las funcionalidades requeridas por una aplicación.”

La funcionalidad incluye tanto la presentación de la información como la satisfacción general del usuario. Normalmente, la calidad de servicio es expresada por medio de parámetros que se negocian. Las necesidades dependerán del tipo de aplicación y pueden variar durante la transmisión.

El primer requerimiento para cualquier transmisión en tiempo real es que todos los niveles que compongan la arquitectura de la red deben garantizar unas prestaciones mínimas y deterministas. En este sentido no se puede utilizar Ethernet como protocolo de red aunque usáramos protocolos deterministas a niveles superiores [Banerjee96].

La calidad del servicio vendrá determinado por el punto de vista que se tome. Desde el punto de vista del emisor o receptor los requerimientos están relacionados casi exclusivamente con el tiempo de entrega de los paquetes de información (retraso), la tasa de pérdida de información y el ancho de banda. Otros puntos de vista pueden ser la eficiencia en el uso de la red, la tasa de errores o retransmisiones.

El tráfico se puede dividir en distintas categorías bien en función de la tolerancia a los parámetros indicados o bien por los requerimientos de los parámetros [Braden94]. En la siguiente gráfica (Figura 2.10) el tráfico es clasificado en el producto cartesiano (sensibilidad al retraso) X (sensibilidad a la pérdida). Como se observa el grado en que las prestaciones de una aplicación dependan de este retraso varía ampliamente y las podemos catalogar en aplicaciones de tiempo real y aplicaciones elásticas.

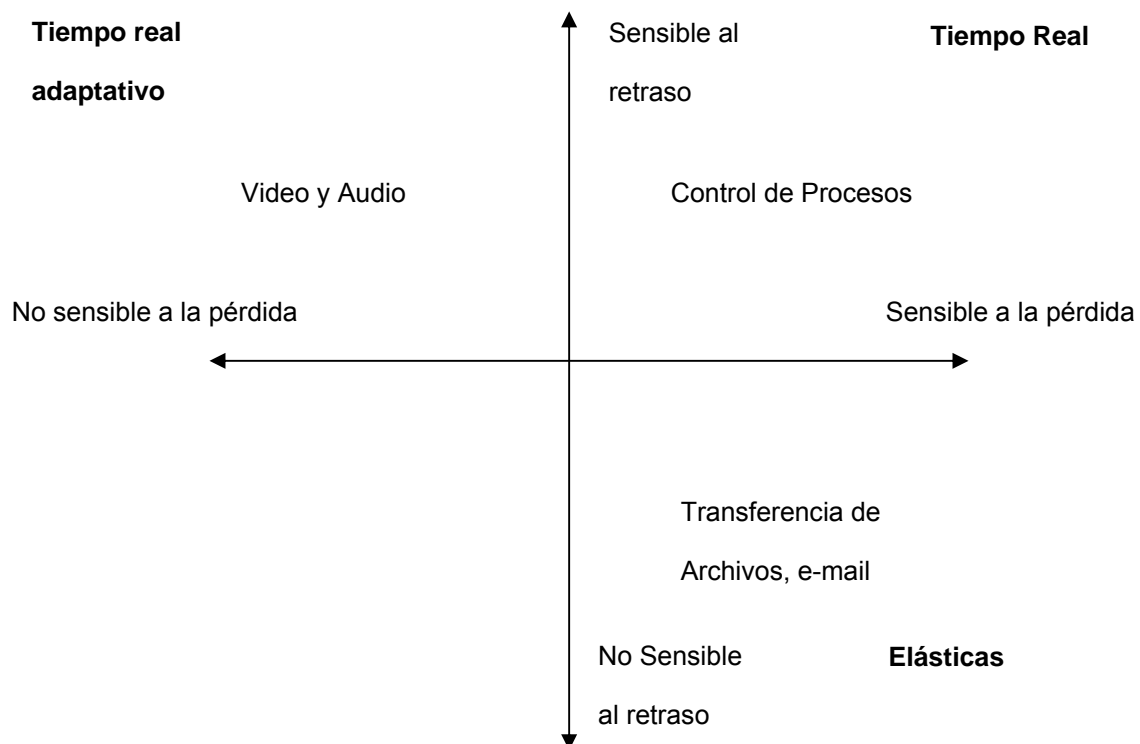


Figura 2.10. Tipos de tráfico en función de la sensibilidad al retraso o pérdida.

2.5.2 Aplicaciones de tiempo real

Una clase importante de estas aplicaciones son las de reproducción. En este tipo de aplicaciones la fuente toma una señal, la convierte en paquetes y los transmite por la red. La red introduce un retraso que debe ser tratado en el receptor. Para poder tratar correctamente los paquetes la aplicación necesita saber a priori el máximo retraso que los paquetes pueden experimentar.

El retraso puede afectar las prestaciones de las aplicaciones en dos maneras. Primero, el tiempo del retraso determina la latencia de la aplicación. Segundo, el retraso individual de los paquetes puede hacer que la fidelidad decaiga si se excede el tiempo de retraso determinado; en este caso la aplicación puede retrasar la ejecución para reproducir estos paquetes retrasados (lo que introduce distorsión) o bien simplemente descartarlos (lo que crea una señal incompleta).

En este sentido se pueden distinguir dos tipos de aplicaciones:

- ✓ **Aplicaciones intolerantes:** Estas aplicaciones no se pueden adaptar a que un paquete se retrase más que el límite predeterminado. Necesitan por lo tanto un límite superior del retraso fiable. Estas aplicaciones requieren un modelo de servicio denominado servicio garantizado.
- ✓ **Aplicaciones adaptativas:** Estas aplicaciones pueden tolerar que lleguen paquetes con un mayor retraso. Estas aplicaciones requieren un modelo de servicio denominado servicio predictivo, que proporciona un servicio bastante fiable pero no seguro. Este tipo de aplicaciones pueden aceptar una merma en la calidad presumiblemente por el menor costo de este modelo, ya que se incrementa el uso de los recursos de red.

Para proporcionar un límite en el retraso el tráfico se tiene que caracterizar y debe haber algún algoritmo de control de admisión que asegure que una petición de flujo puede ser aceptada.

2.5.3 Aplicaciones elásticas

Estas aplicaciones siempre esperan a que los datos lleguen. Este tipo de aplicaciones no requieren ninguna caracterización del servicio para funcionar. Ejemplos de estas aplicaciones son transferencias (FTP), terminales (Telnet, X, NFS), etc. Un modelo de servicio para estas aplicaciones es proporcionar un servicio “tan rápido como se pueda” (ASAP as-soon-as-possible). En contraste a los modelos en tiempo real, estas aplicaciones no están sujetas a control de admisión. Suelen estar basadas en un tipo de servicio usualmente denominado “best-effort”.

2.5.4 Ancho de banda y retraso

Otro aspecto a considerar en el tráfico es el ancho de banda y retraso necesario para la transmisión. En el siguiente gráfica (Figura 2.11), el tráfico es clasificado en el producto cartesiano (ancho de banda) X (retraso). Como se observa, la transmisión de video y audio requiere un gran ancho de banda y bajos retrasos. El caso contrario es el correo electrónico, que tiene bajo ancho de banda y tolera un alto retraso (en la mayoría de los casos).

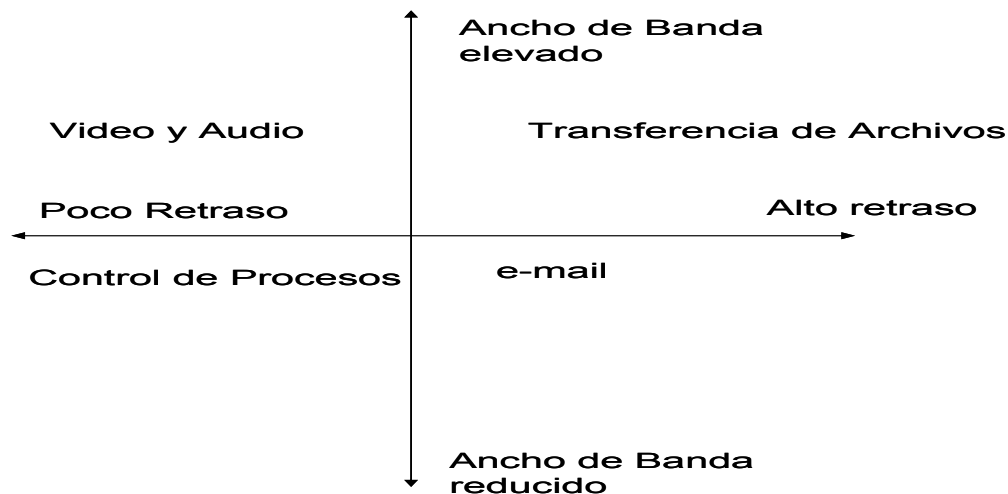


Figura 2.11 Tipos de tráfico en función del ancho de banda y el retraso

Valores típicos para distintos tipos del tráfico se describen en la siguiente Tabla 2.8:

	Retraso máximo (s)	Máxima variación el retraso	Ancho de banda (Mbps)	Tasa de pérdida de paquetes
Voz	0.25	10	0.064	10e-1
Video	0.25	10	100	10e-3
Imagen	1		2-10	10e-9

Tabla 2.8. Valores típicos de tráfico

2.5.5 Nivel de calidad de servicio

Cada tipo de aplicación requiere diferentes niveles de calidad de servicio. Bajos niveles de calidad son fáciles de implementar usando mecanismos simples de gestión, como por ejemplo controlando periódicamente el nivel de carga de una red y aumentando los recursos de la red antes de que estén congestionados. Este esquema podría servir para proporcionar un servicio predictivo.

En cambio, obtener una alta calidad de servicio, como pueda ser el servicio garantizado, es más complejo de solucionar. En general, hay que sobredimensionar los recursos de la red y tenerlos asignados para poder garantizar este nivel de servicio. En este sentido, se puede medir la calidad de servicio que ofrece la red en función de la eficiencia de recursos que usa. Este compromiso entre calidad y eficiencia es un aspecto muy importante en el diseño de las redes en tiempo real

Además, aunque los mecanismos de calidad de servicio pueden variar en distintas partes de la red, esta calidad tiene que obtenerse entre emisor y receptor.

2.5.6 Modelos de servicios

Un modelo de servicio define las propiedades que debe tener un servicio y que éste ofrece a las aplicaciones que lo usan. En general se puede hablar de dos modelos: servicios integrados (*IntServ*) y servicios diferenciados (*DiffServ*).

2.5.6.1 Modelo de servicios integrados

El modelo de servicios integrados intenta integrar todos los tipos de tráfico posibles en una misma red de uso general [Braden94]. Este modelo ofrece servicios cuantificables y medibles en el sentido que son definidos para proporcionar una determinada calidad de servicio para un tipo de tráfico cuantificado. Este modelo está típicamente asociado a mecanismos de admisión y reserva de recursos en la red.

El modelo de reserva describe cómo una aplicación negocia el nivel de calidad de servicio. El modelo más simple es que una aplicación pida una calidad de servicio particular y que la red se lo proporcione o lo deniegue.

Sin embargo, más que rechazar la petición, la red podría conceder un nivel de recursos menor que el pedido. Un esquema más complejo es el modelo de reserva de “doble pasada”. En este esquema, se propaga la especificación del tráfico inicial desde el origen a los posibles destinos. Cada router en las rutas guarda estos valores y quizá los ajusta para reflejar su capacidad disponible. Esta especificación ajustada a la red es devuelta al origen que decide si admite o no el canal.

2.5.6.2 Modelo de servicios diferenciados

Este modelo es un mecanismo de calidad de servicio de nivel 3 que ha sido utilizado durante algunos años, aunque se ha realizado poco esfuerzo para su estandarización hasta la aparición recientemente del grupo de Servicios Diferenciados de la IETF (DiffServ) [Blake98].

En este modelo, la red clasifica el tráfico en distintas clases y les aplica una disciplina de servicio diferenciada con el objetivo de proporcionar distintos niveles de calidad de servicio. En este caso no se reservan recursos por lo que no se puede garantizar a priori una calidad de servicio.

De este modo, se pueden tener varias clases de servicio para tiempo real, con varios niveles de retraso. También habrá niveles con servicio predictivo y otros sólo con garantía de entrega. El cliente escogerá el tipo de servicio en función del tráfico a transmitir y por supuesto, el precio que quiera pagar.

Otra de las ventajas de este modelo es su menor complejidad de implementación y su fácil integración con los protocolos IP, en el que cada paquete puede ser marcado con la clase de servicio que requiere. Esta marca será utilizada por los routers para diferenciar el servicio por paquete.

2.5.6.3 Tráfico agregado y por conversación

Es importante determinar cómo se va a gestionar el tráfico internamente en la red. La gestión del tráfico por conversación trata cada conversación como un flujo separado. Tradicionalmente, este tipo de gestión está asociado al modelo de servicios integrados. En este caso, la red asigna recursos independientes al resto de las conversaciones y mantiene un control de ellos.

En el núcleo de grandes redes, donde es posible soportar cientos de miles de conversaciones simultáneamente, este mecanismo no es práctico. En estos casos se utiliza el tráfico agregado. De esta forma, un conjunto del tráfico de diferentes conversaciones, se clasifica como un mismo flujo y se maneja como un tráfico agregado. Además, esta agregación permite reducir en conjunto los recursos necesarios y permite obtener una calidad de servicio casi equivalente al modelo por conversación. Los servicios diferenciados son claros ejemplos de uso de tráfico agregado.

2.5.6.4 Requerimientos para compartir recursos

Normalmente la red va a ser compartida por distintos tipos de tráfico. Mientras el aspecto más importante en la calidad de servicio es el retraso, aquí el interés primario es el ancho de banda de los enlaces. Este componente del modelo de servicio, llamado compartición de enlaces, contempla como compartir el ancho de banda de un enlace entre varios flujos de acuerdo con ciertas características [Floyd95]. Se pueden distinguir los siguientes tipos de comparticiones:

- ✓ **Compartición multi-entidad:** Un enlace puede ser usado por varias organizaciones. Se debe asegurar que los enlaces son compartidos de forma controlada, quizás de forma proporcional a lo que paguen.
- ✓ **Compartición multi-protocolo:** Se debe prevenir que una familia de protocolos (DECnet, IP, IPX, OSI, SNA, etc.) sobrecargue un enlace y

excluya al resto. Esto es importante porque las distintas familias de protocolos responden de forma diferente a la congestión.

- ✓ **Compartición multi-servicio:** Un administrador de red puede limitar la fracción de ancho de banda para cada clase de servicio.

El control de admisión será necesario de nuevo para asegurar que la compartición de recursos se va a cumplir.

2.6 Algoritmos de Optimización de Tráfico en redes WAN

2.6.1 Especificación y conformación del tráfico

Una especificación de flujo es un acuerdo entre todos los componentes de una red para especificar el tráfico que va a tener de una forma precisa y predeterminada [Tanenbaum96]. Consiste en una serie de parámetros que describen como el tráfico es introducido en la red y la calidad de servicio deseado por las aplicaciones. La idea es que antes de establecer una conexión, el origen del flujo informe sobre las características del flujo a transmitir y el servicio deseado (especificación de la calidad de servicio). Toda esta información es la que compone la especificación del flujo.

Uno de los componentes más importantes de esta especificación es la descripción de cómo se va introducir el tráfico en la red que se suele denominar modelo del tráfico. El objetivo es regular el tráfico a transmitir con el objeto de eliminar la congestión en la red debido a las características de gran variabilidad del tráfico. Este mecanismo de regulación del tráfico de acuerdo al modelo del tráfico se denomina conformación del tráfico (*traffic shaping*).

La conformación de tráfico es un mecanismo de gestión de la congestión en bucle abierto (*open loop*) que permite a la red saber cómo es el tráfico que se transmite para poder decir si lo pueda manejar. Al hecho de monitorear el tráfico para que cumpla el patrón acordado se denomina comprobación del tráfico (*traffic policing*).

Los modelos de tráfico más comunes son el leaky bucket y token bucket. Otros esquemas como el D-BIND [Knightly94], double leaky bucket o modelos multiparámetros son ampliaciones de éstos usados con planificadores complejos. También se describe el modelo *Tenet* (X_{min} , X_{ave} , I , S_{max}) [Ferrari90a] por ser un planteamiento diferente a los anteriores.

2.6.2 Algoritmo Leaky Bucket

Este algoritmo fue introducido por Turner [Turner86] y desde entonces ha sido el más ampliamente usado para describir tráficos. Este algoritmo regula el tráfico a modo de un cubo con goteo tal como se representa en la Figura 2.12. Se usan dos parámetros para describir el algoritmo: la capacidad del cubo s (bits) y la tasa de drenaje r (bits/s). El funcionamiento del algoritmo es simple: siempre que el cubo tenga contenido se envía a la red con tasa r , todo paquete entrante se introduce en el cubo y en el caso de que el cubo esté lleno, el paquete entrante se pierde. De esta forma, se limita la tasa de transmisión del tráfico al valor de r . El valor de s para un determinado flujo habrá que calcularlo de tal forma que no se pierdan paquetes.

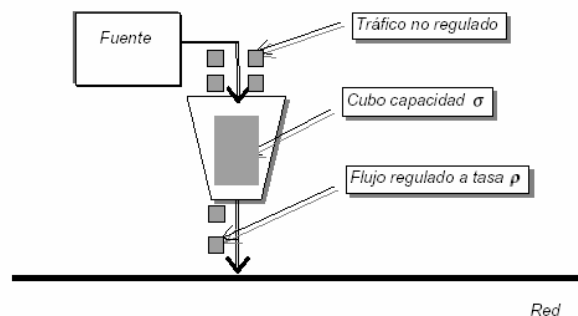


Figura 2.12. Algoritmo Leaky Bucket

Para que no se produzca ninguna pérdida de paquetes, el emisor no puede transmitir en el periodo $[0, t]$ más de $s + rt$ bits. En general, el valor r representa la tasa media de transmisión de la fuente.

2.6.3 Algoritmo Token Bucket

El objetivo de este algoritmo es permitir transmitir a mayores velocidades cuando la fuente recibe un pico. El funcionamiento del algoritmo es el siguiente: el cubo contiene *tokens* generados a una tasa r (véase la Figura 2.13). El cubo puede admitir como máximo b *tokens*, estando al inicio lleno. Para que se transmita un bit se tiene que coger un *token* del cubo y eliminarlo. Mientras existan *tokens* en el cubo, la fuente puede insertar el tráfico a la red a la tasa deseada. Cuando se acaban los *tokens* tendrá que esperar al próximo *token* que se genere, lo que implica que la tasa de transmisión disminuye a r . En esencia, lo que permite *token bucket* es poder transmitir en un determinado intervalo a tasas superiores a r .

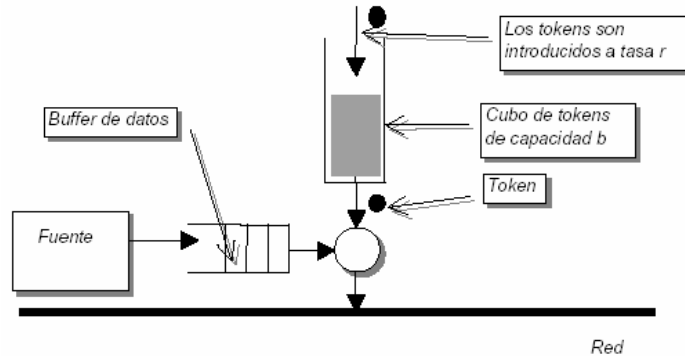


Figura 2.13. Algoritmo token bucket

El parámetro r especifica la tasa de datos sostenible continuamente, mientras que b especifica en cuánto se puede exceder esta tasa para cortos periodos de tiempo. Más específicamente, el tráfico debe obedecer la regla de que para cualquier periodo de tiempo, la cantidad de datos enviados no puede ser superior a $rt+b$, para cualquier intervalo de tiempo t . Además, se suele imponer un límite en la tasa de transmisión que es la tasa pico p . Con este límite el tráfico no puede exceder $\min[pt, rt+b]$.

2.6.4 Modelo Tenet

Este modelo de tráfico fue introducido por Ferrari [Ferrari90a] para regular el tráfico en una red en tiempo real. Un tráfico satisface el modelo $(Xmin, Xave, I, Smax)$ si el tiempo de llegada entre dos paquetes del tráfico es siempre mayor que $Xmin$, el tiempo medio de llegada entre paquetes es $Xave$ para cualquier intervalo de tiempo I , y el tamaño máximo de un paquete es menor que $Smax$.

Con este modelo, el emisor puede enviar a una tasa pico de $1/Xmin$ hasta que es forzado a parar la transmisión por el límite impuesto por $I/Xave$.

2.6.5 Planificadores

2.6.5.1 Introducción

Los planificadores de tráfico pueden ser usados en distintos entornos para satisfacer una amplia variedad de objetivos [Varma97]. Una aplicación común de los algoritmos de planificación es proporcionar una calidad de servicio a nivel de red aislando unos tráficos de otros. Los planificadores también pueden ser usados para permitir a los usuarios compartir un enlace de forma equitativa o determinista.

Un planificador puede ser contemplado como un sistema de colas que consiste en un servidor que proporciona servicio a un conjunto de clientes. Los clientes encolan paquetes para ser servidos y estos son escogidos por el planificador basándose en una disciplina de servicio definida por el algoritmo de planificación. La disciplina de servicio puede ser diseñada para cumplir los requerimientos de calidad de servicio deseados por cada cliente.

Los atributos deseables para un algoritmo de planificación son los siguientes [Varma97]:

- ✓ **Aislamiento de flujos:** Aislar un canal de los efectos indeseables de otros.
- ✓ **Retraso emisor-receptor garantizado:** El planificador debe proporcionar un retraso garantizado de emisor a receptor. Además, es deseable que este límite del retraso dependa sólo de los parámetros de la sesión y que no dependa del resto de las sesiones.
- ✓ **Utilización:** El algoritmo debe maximizar el uso de ancho de banda del enlace.
- ✓ **Equidad (*Fairness*):** El planificador debe servir a las sesiones con tasas proporcionales a su reserva en cada instante, esto es, distribuyendo el ancho de banda libre proporcionalmente entre las activas. Lo ideal sería que se comportase como un flujo perfecto repartiendo perfectamente el ancho de banda. Pero debido a la cuantificación en paquetes de los flujos esto es prácticamente imposible. Por tanto, se introduce el índice de equidad (WFI: *Worst-case fairness index*) que mide la desviación de servicio ofrecido por un planificador con respecto a un modelo perfecto.
- ✓ **Simplicidad de implementación:** El algoritmo de planificación debe ser fácil de implementar y con baja complejidad. Esto es importante si se va implementar por hardware.
- ✓ **Escalabilidad:** El algoritmo debe comportarse bien en nodos con un gran número de sesiones y con una variedad de velocidades de enlace.

2.6.5.2 Disciplinas de servicio

El objetivo de los planificadores es asignar los recursos de acuerdo a la reserva realizada con anterioridad con el objetivo de cumplir la calidad de servicio exigida. Tres tipos de recursos son asignados por los planificadores: ancho de banda (*qué* paquete es transmitido), tiempo (*cuándo* es transmitido el paquete) y memoria (*qué* paquetes son

descartados), lo que afecta a tres parámetros básicos: rendimiento, retraso y tasa de pérdida.

En general, se distinguen dos tipos de disciplinas de servicio en los nodos [Zhang98]:

- ✓ **Non work-conserving** en el que los nodos intentan mantener el modelo del tráfico, aunque esto implique que en determinados periodos no se transmita nada. En este caso, cuando entra un paquete en el nodo se le asocia un tiempo de elegibilidad. En el caso de que no haya paquetes en estado de elegibilidad, no se transmite nada. Ejemplos de estas disciplinas de servicio son las de tasa controlada (“rate-controlled”): RCSP [Zhang94], Jitter-EDD [Verma91], “Stop-and-go” [Golestani90] y “Hierarchical Round Robin” [Kalmanek90]. Cada planificador provoca un retraso acotado y calculable para cada paquete. Dado que cada nodo mantiene el modelo del tráfico el cálculo del retraso total es la suma de los retrasos en cada nodo.
- ✓ **Work-conserving** en el que si existen paquetes en el nodo por transmitir se envían. A este grupo pertenecen *Virtual Clock* [ZhangL90], *Weighted Fair Queuing* (WFQ) y GPS (*General Processor Sharing*) [Demers89] [Parekh92]. Para todos estos esquemas existen funciones para calcular el retraso máximo emisor-receptor que están basadas en el trabajo de Parekh y Gallager [Parekh93]. Normalmente, el cálculo del retraso es dependiente de la reserva de ancho de banda en los nodos.

Hay que tomar en cuenta el costo computacional de los algoritmos de planificación para su implementación en redes de alta velocidad. Por ejemplo, un planificador FCFS tiene un costo de implementación bajo, pero sólo puede soportar un límite de retraso para todas las conexiones. En el otro extremo, el algoritmo EDD es complejo ya que involucra una operación de búsqueda del paquete con el deadline más corto.

Otras disciplinas de servicio gestionan la compartición del enlace de una forma controlada, permitiendo una estructura jerárquica, como el planificador CBQ (*Class-based queueing*) [Floyd95].

A continuación se describen dos disciplinas de servicio que el autor considera los más representativos para cada tipo de servicio: el servicio RCSP del tipo *non-workconserving* y el planificador WFQ del tipo *work-conserving*.

2.6.5.3 Servicio RCSP

La disciplina de servicio RCSP (*Rate-Controlled Static Priority*) fue introducida por H. Zhang en el grupo *Tenet* [Zhang94]. Como se muestra en la Figura 2.14 un servidor RCSP está formado por dos componentes: un controlador de tasa y un planificador con prioridades estáticas. Conceptualmente, el controlador de tasa está formado por el conjunto de reguladores asociados a cada canal que atraviesa el nodo. Cada regulador es un conformador de tráfico que regula el tráfico de entrada al nodo al modelo de tráfico deseado para el planificador.

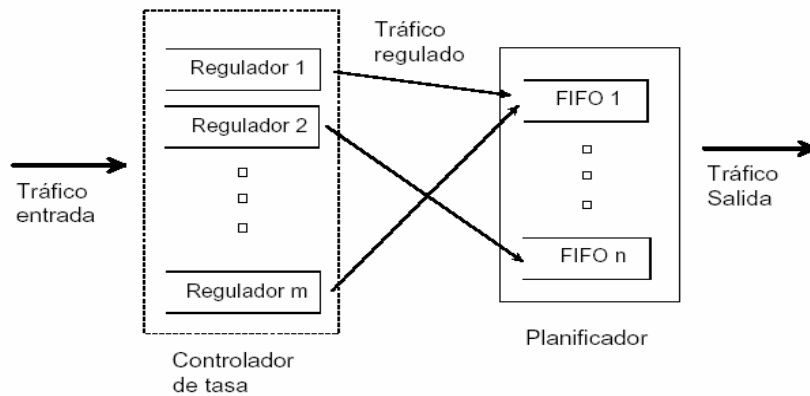


Figura 2.14. Planificador RCSP

Cuando un paquete llega, el regulador calcula un tiempo de elegibilidad y es retenido en el regulador hasta que cumpla este tiempo. A continuación, se introduce en el planificador deseado en función del nivel asignado. El planificador dispone de un conjunto de colas para cada nivel de prioridad y selecciona los paquetes de la cola más prioritaria que no esté vacía. Cada conexión tiene asignado un nivel de prioridad desde el momento del establecimiento del canal y depende principalmente del retraso exigido.

La forma de calcular el tiempo de elegibilidad depende del modelo del tráfico, el cual va a definir como se regula el tráfico. Para RCSP se utiliza el modelo *Tenet* (X_{min} , X_{ave} , I , S_{max}). Con este modelo se obtienen las ecuaciones que definen el tiempo de elegibilidad de un paquete para un canal de tal forma que el tráfico mantiene sus características a lo largo de la red.

El tiempo de elegibilidad para el paquete de orden k de la conexión j en el nodo i , $ek_{i,j}$, se define usando el tiempo de elegibilidad calculado para paquetes anteriores de la misma conexión:

$$\begin{aligned}
e_{i,j}^k &= -I, \quad k \leq 0 \\
e_{i,j}^1 &= a_{i,j}^1 \\
e_{i,j}^k &= \max(e_{i,j}^{k-1} + X_{\min}, e_{i,j}^{k-1} \left\lfloor \frac{I}{X_{ave}} \right\rfloor + 1 + I, a_{i,j}^k), \quad k > 1
\end{aligned}$$

donde $a_{i,j}^k$ es el tiempo de llegada del paquete k de la conexión j al nodo i .

2.6.6 Planificador WFQ

En este punto se describe el algoritmo de planificación más utilizado actualmente y que es la base de muchos otros. Este planificador fue introducido con el nombre de *Weighted Fair Queueing* (WFQ) en [Demers89] aunque luego se ha rebautizado como PGPS (*Packet General Processor Sharing*). Es un intento de aproximarse a un modelo de flujo perfecto (denominado GPS: *General Processor Sharing*) con un procesador compartido entre los distintos flujos de acuerdo a unos pesos predeterminados. De esta forma, el ancho de banda se reparte proporcionalmente entre los distintos flujos.

En los algoritmos WFQ, cuando llega una trama se calcula y asocia una etiqueta (*Time Stamp*, TS) que va a determinar el orden de salida y se envía a la cola de su conexión. La siguiente trama a transmitir será la que tenga el valor TS más pequeño.

TS se calcula de acuerdo a siguiente la fórmula:

$$\begin{aligned}
TS_1^0 &= 0 \\
TS_i^k &= \max(TS_i^{k-1}, v(t)) + \frac{L_i^k}{\rho_i}
\end{aligned}$$

donde:

$v(t)$ es la función de tiempo virtual calculada a la llegada del paquete.

TS_i^{k-1} corresponde al *time stamp* de la trama anterior.

L_i^k es el tamaño de la trama en bits.

ρ_i es el ancho de banda deseado para la sesión i .

La función $v(t)$ se calcula así: si $B(t)$ representa el conjunto de sesiones con algún paquete en cola (*backlogged*) en el planificador en el instante t y V el número total de sesiones, entonces:

$$v(t) = \frac{\sum_{j=1}^V \phi_j}{\sum_{i \in B(t)} \phi_i}$$

Donde ϕ es un número real que indica la porción de ancho de banda del enlace requerido por la sesión que cumple la siguiente condición:

$$\rho_i \leq \frac{\phi_i}{\sum_{j=1}^V \phi_j} r$$

donde r es la capacidad de enlace. Por ejemplo, se puede usar $\phi_i = \rho_i/r$.

Cuando una sesión tiene tráfico pendiente, se cumple que $v(t) \neq TS_{i-k-1}$ por lo que $v(t)$ no influye en TS . En caso contrario, al recibir la primera trama después de un periodo sin tráfico, se tiene en cuenta $v(t)$ para actualizar la sesión. La dificultad del planificador WFQ reside en calcular $v(t)$, por lo que se han propuesto otros algoritmos que simplifican su cálculo.

2.6.7 Gestión del tráfico: control de admisión

Dado que las redes tienen capacidad finita, si no se limita la cantidad de tráfico a transmitir en la red, el servicio ofrecido se degradará, y al final se colapsará. Si la red no puede aceptar un determinado tráfico porque no lo puede gestionar o afectase al resto de tráfico no debería admitirlo.

Las soluciones a este problema se pueden clasificar en dos grupos: reactivas (o esquema de control) y proactivas o (control de admisión) [Zhang98]. Las soluciones reactivas detectan y reaccionan dinámicamente a la congestión dentro de la red, reduciendo o eliminando parte del tráfico. Con este esquema es difícil asegurar la calidad de servicio. En las soluciones proactivas, una conexión se acepta sólo si existen suficientes recursos para satisfacer los requerimientos del nuevo canal y de los ya existentes.

Para ello, con los modelos del tráfico y disciplinas de servicio se han desarrollado unos tests que indican si se puede admitir un nuevo flujo con tales características. Estos tests pueden ser a nivel de nodo o a nivel de red. A nivel de nodo se chequea con la información disponible del nodo y del flujo si el canal se puede admitir. En el caso de que no se pueda admitir se rechaza directamente la conexión. A nivel de red, el receptor comprueba, con toda la información de los nodos por los que ha pasado el mensaje de conexión, si se cumplen los requerimientos pedidos. En el caso de ser así, se envía al emisor un mensaje de vuelta de establecimiento de canal. En caso contrario, se rechaza el canal y se envía un mensaje de rechazo.

La reserva puede ser en una pasada, en la que los recursos se van reservando en el mensaje de establecimiento de canal, o bien de doble pasada, en el que el mensaje pasa primero por todos los nodos recogiendo información sobre los recursos disponibles. Cuando el mensaje llega al receptor, éste decide si se puede admitir el canal y envía el mensaje de vuelta reservando los recursos en la red.

Los tests van a depender del modelo de los nodos y del tráfico. Cuando más exactos sean los tests se hará un uso más eficiente de los recursos de la red y se podrán admitir más canales.

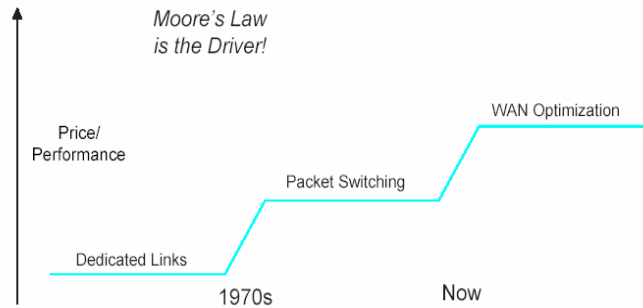
2.7 Tendencias en la Optimización del ancho de banda

La capacidad para que una aplicación adapte su comportamiento a las condiciones cambiantes de la red depende del mecanismo de estimación del ancho de banda que la aplicación o protocolo de transporte utilice. Como tal, la exactitud de los algoritmos de estimación de ancho de banda y herramientas pueden beneficiar a una gran clase de datos intensivos y aplicaciones científicas distribuidas. Sin embargo, las herramientas existentes y las metodologías para medir las métricas del ancho de banda de la red (por ejemplo la capacidad, el ancho de banda disponible, y rendimiento de procesamiento) están tratando de ser más efectivas a través de la infraestructura real de Internet.

De acuerdo a fuentes confiables de información, se puede mencionar que hasta el 90% de datos que atraviesan por una red corporativa es redundante. De acuerdo a estadísticas de Gartner, antes de 2008, la inversión en soluciones WAN para la optimización del ancho de banda será la estrategia primaria de mejora para el 75% de las empresas corporativas en el mundo.

El análisis de funcionamiento y la identificación de fallas en tiempo real en cualquier punto durante un día laborable, el Administrador de la red debe saber (o poder descubrir) qué ancho de banda se requiere para manejar el funcionamiento de la red y las aplicaciones, así como asignar el tiempo de respuesta adecuado. Sin embargo, entender las tendencias de la red no es encontrar donde ocurren los problemas inesperados en la red, el administrador de red necesita identificar el problema en ese mismo momento lo más rápido posible y poder dar una solución: ¿Servidor nuevo? ¿Nuevo servicio? ¿Interrupción de la red?.

Surgimiento y consolidación de Tecnologías emergentes. Se ha alcanzado hoy otro punto de inflexión del funcionamiento-precio. En este momento el poder de procesamiento está adquiriendo un papel muy importante. Mientras que las redes de X.25 y de Frame Relay bajaran su costo de ancho de banda para la WAN, las soluciones emergentes van un paso más lejos. Disminuyen no solo costos en el ancho de banda, también están agregando capacidad de controlar tráfico WAN y de entregar funcionamiento fiable y constante a aplicaciones de misión crítica.



La optimización WAN es no solamente una manera viable de aumentar la efectividad del ancho de banda en enlaces WAN y de reducir la latencia, se está convirtiendo en cada vez más una necesidad para las aplicaciones de soporte tales como VoIP y la comunicación de video que tienen necesidades de funcionamiento muy rigurosos. Más ancho de banda no es suficiente. El ancho de banda se debe asignar a las aplicaciones críticas e inversamente, las aplicaciones no autorizadas o recreacionales se deben restringir en el ancho de banda quitando recursos necesarios para las operaciones reales del negocio.

Capítulo 3.

Método y Técnica de Diseño

3.1 METODO

3.1.1 OBJETO ESTUDIADO

Existen actualmente varios algoritmos para la optimización del ancho de banda. Algunos de ellos emplean técnicas para la estimación del ancho de banda considerando un tráfico real (tipo cruzado) para su funcionamiento.

Para este trabajo se analizarán dos Algoritmos, el primero de ellos (envolvente empírica) considera al video como un tipo de tráfico para hacer su análisis y comprobación matemática. El segundo de ellos (reducción de secuencia molecular) considera cualquier tipo de tráfico. Ambos algoritmos cumplen con la eficiencia que demanda la complejidad computacional del análisis de algoritmos.

3.2 TÉCNICA DE DISEÑO

3.2.1 Introducción al Estudio de Algoritmos

Un algoritmo es un conjunto de reglas que da una secuencia de operaciones para resolver un tipo específico de problema, el cual tiene 5 características importantes:

1. Finito. Un algoritmo debe siempre terminar después de un número de pasos.
2. Definición. Cada paso de un algoritmo debe estar bien definido. Las acciones a ser llevadas a cabo deben especificarse sin ambigüedad y rigurosamente para cada paso.
3. Entada. Un algoritmo tiene cero entradas o más entradas, es decir, cantidades que son dadas para inicializarse antes de que el algoritmo comience. Estas entradas son tomadas de conjuntos específicos de objetos.

4. Salida. Un algoritmo tiene una o más salidas, es decir, cantidades que tienen una relación específica con las entradas.

5. Efectividad. Un algoritmo también es, generalmente pensado para ser efectivo, esto significa que todas las operaciones a ser ejecutadas en el algoritmo deben ser suficientemente básicas, tales que pueden desde un principio estar dadas exactamente y ejecutarse totalmente en un tiempo finito por el hombre usando papel y lápiz.

El estudio de algoritmos es la parte principal de la ciencia de la computación. En los últimos años se ha caracterizado por un avance significativo en dicho campo; estos avances se han enfocado principalmente hacia desarrollo de algoritmos más rápidos para ciertas aplicaciones, así como a ciertos problemas para los cuales los algoritmos son ineficientes.

3.2.2 Análisis de eficiencia de un Algoritmo

En la práctica no sólo se requieren algoritmos, sino que sean buenos en algún sentido. Un criterio para la evaluación de la eficiencia de un algoritmo es el tiempo que tarda en su ejecución misma, bajo ciertas condiciones de entrada.

El hablar de análisis de eficiencia de un algoritmo, en tiempo de ejecución, muestra una medida de complejidad; así mismo se considera que la mayoría de los algoritmos tienen un espacio limitado y que responde a una función lineal del tamaño de la entrada. O sea, que el tiempo que tarda un algoritmo en ejecutarse va estrechamente ligado con la función lineal que lo represente. Para el caso de estudio de los algoritmos a analizar la complejidad es de $O(\log m)$ donde “m” es un valor pequeño (entre 20 y 100 puntos envolventes) y $O(n)$ donde “n” es el número de símbolos de entrada. O es el nombre de la función (conocida dentro de la complejidad computacional como O – Grande).

3.2.3 Algoritmo para la optimización de recursos de red a través de una envolvente empírica

3.2.3.1 Introducción a la optimización del ancho de banda

El algoritmo de envolvente empírica fue estudiado del trabajo presentado por Enrique Hernández Orallo en su tesis doctoral, debido a que no existen fuentes bibliográficas tan completas como este estudio. Lo que se toma de este trabajo es la metodología de investigación para mostrar los avances que existen en el campo de la Optimización del Ancho de Banda.

En este apartado se aborda la optimización del ancho de banda reservado en los enlaces, utilizando el concepto de los puntos envolventes. Para realizar esta optimización se establece una condición: **es necesario conocer el tráfico que se va a transmitir**. Por ello, se plantea su utilización en la transmisión bajo demanda², siendo también aplicable a otros tipos de transmisión.

Para establecer un canal³ habrá que reservar una serie de recursos en la red para asegurar que la transmisión se realiza con una determinada calidad de servicio. Es importante, para ahorrar recursos en la red, que la reserva de los recursos sea óptima. En general, se tiene un tráfico que sufre variaciones de ancho de banda en función del tiempo, lo que se denomina tráfico VBR (Variable Bit Rate). A la hora de transmitir este tráfico, se necesita un modelo que limite esta transmisión.

El modelo de tráfico más usado es el Leaky Bucket (σ, ρ)⁴ donde el emisor no puede transmitir en el periodo $[0, t]$ más de $\sigma + \rho t$ bits. A partir de esta restricción y con los parámetros de la red se han desarrollado un conjunto de ecuaciones⁵ que nos permiten calcular el retraso del emisor al receptor a partir de una reserva de recursos de red realizada en los nodos. Para mayor detalle del desarrollo de estas ecuaciones ver Anexo 3.1 apartado 3.1.1 .

3.2.3.2 Objetivo de la Optimización

Cuando se reservan los recursos en la red para un determinado canal los parámetros de la red⁶ se conocen y son fijos. El problema reside el obtener los parámetros que caractericen el flujo y la optimización de la reserva de los recursos en la red. Como se muestra posteriormente, es iterativo y costoso, además de que se tiene que repetir para cada transmisión de un mismo video. Es costoso porque implica recorrer dicho vídeo para determinar los parámetros en cada iteración (por ejemplo, si se transmite una película se tendrá que recorrer 100 minutos de imagen que son unos 2 Gbytes).

² Es un tipo de transmisión, en donde una serie de servidores conectados a la red permiten acceder en tiempo real a cualquier película, vídeo, música, etc., que nos interese. El cliente recibirá la película en su computadora y/o decodificador.

³ En general el medio (físico o virtual) por el que se realizan las comunicaciones se suele denominar canal. Estos canales son definidos con unos parámetros típicos como pueden ser el ancho de banda, retraso, variación del retraso (delay jitter) y fiabilidad.

⁴ Es un modelo de tráfico (algoritmo) que regula el tráfico y emplea dos parámetros para describir dicho algoritmo: la capacidad del cubo σ (bits) y la tasa de drenaje ρ (bits/s). Lo anterior fue descrito en el Capítulo II de este trabajo.

⁵ Entre estas ecuaciones, las más conocidas y las utilizadas en este trabajo son las basadas en el trabajo de Parekh y Gallager [Parekh92][Parekh93].

⁶ Cómo parámetros de red tenemos: enlaces, ancho de banda de los enlaces y MTU (Unidad máxima de transmisión).

En consecuencia, se propone un método alternativo basado en un proceso denominado envoltura empírica en el que se generan off-line una serie de puntos para cada vídeo que definen la carga. Y utilizando estos puntos, se puede realizar on-line este mismo cálculo de una forma muy rápida y directa.

3.2.3.3 Cálculo del retraso emisor – receptor

El retraso de emisor a receptor se calcula normalmente a partir de una descripción de la red y un modelo del tráfico, siendo normalmente función de la reserva realizada en los nodos. Para este fin y para un servicio garantizado⁷ se utiliza el otro modelo de tráfico Token Bucket⁸ (b,r,p) en donde el número de bits que la fuente puede transmitir es menor que $b+rt$ para cualquier intervalo de t ⁹. Para la obtención del retraso se utilizan las ecuaciones (1 y 2) descritas en el Anexo 3.1 apartado 3.1.1.

De las ecuaciones de retraso descritas en el Anexo 3.1 (1, 2 y 3) se obtiene que el retraso es inversamente proporcional a la reserva de ancho de banda a realizar. Esto implica, que para que una conexión obtenga un retraso pequeño tiene que reservar un ancho de banda elevado, incluso mayor que la tasa pico. Esto puede provocar que se desperdicien recursos cuando una conexión que necesita retrasos pequeños tenga poco tráfico.

Es importante resaltar que las ecuaciones descritas en el Anexo 3.1 (1, 2 y 3) no consideran a la latencia, por lo que hay que agregar la latencia mínima de la red para calcular el retraso total. La latencia mínima es fija y dependerá principalmente de los medios de transmisión físicos, que normalmente suele ser despreciable respecto al retraso en red, aunque este valor puede ser importante en enlaces transoceánicos o vía satélite (por la distancia de estos enlaces).

En el caso que fuera necesario considerar la latencia mínima, se restará al retraso pedido esta latencia y se utilizarán las ecuaciones descritas en el Anexo 3.1 (4 y 5).

⁷ Es la calidad de servicio destinada para aplicaciones con requerimientos exigentes de tiempo real. Esta calidad asegura: un ancho de banda, un límite en el retraso y ninguna pérdida en las colas.

⁸ Es otro modelo de tráfico cuyo objetivo de este algoritmo es permitir transmitir a mayores velocidades cuando la fuente recibe un pico (p); donde “b” es la cantidad máxima de tokens permitidos en el cubo y “r” es la tasa de generación de tokens. Para mayor referencia ver Capítulo II de este trabajo.

⁹ Realmente si se establece una tasa pico p, el tráfico no puede exceder $\min[pt, b+rt]$.

En forma concreta, para calcular la reserva de recursos se necesita:

- ✓ Un modelo de la red, para obtener los parámetros de red C_{tot}^{10} , D_{tot} y M^{11} , que son determinados, y
- ✓ Calcular los parámetros del flujo $(\sigma, \rho)^{12}$. De estos parámetros, la tasa pico es de cálculo directo y determinado. El problema es cómo calcular los parámetros σ y ρ para optimizar la reserva¹³.

3.2.3.4 Optimización de la reserva del ancho de banda

Las secuencias en un vídeo constan de n tramas, donde en cada trama i se transmiten una serie de bits E_i . La transmisión de vídeo está limitada por un conformador de tráfico Leaky Bucket¹⁴ donde no se puede transmitir $\sigma + \rho t$ bits para un tiempo t . La obtención de la reserva óptima del ancho de banda varía para distintos pares de valores σ y ρ , en donde se obtienen cálculos de retrasos diferentes. Es decir a mayor capacidad de σ tenemos una menor tasa de transferencia ρ para un valor determinado de retraso implicando ello una mayor reserva. Y viceversa a menor capacidad de σ se tiene una mayor tasa de transferencia ρ para el mismo valor de retraso dando ello una menor reserva. Por esta diferencia tan grande en los cálculos de reserva, se emplea un método iterativo para encontrar los valores de σ y ρ que permitan encontrar los valores óptimos de la reserva. El desarrollo y la comprobación de este método viene descrito en el anexo 3.1 apartado 3.1.2.

Hernández Orallo demuestra que la reserva óptima (ancho de banda “R”) ocurre cuando la reserva es igual a la tasa de drenaje ρ .

¹⁰ C_{tot} y D_{tot} representan la sumatoria de los términos de error C y D de cada router de una ruta.

¹¹ M es el tamaño máximo del paquete.

¹² Si se utiliza un modelo de tráfico Leaky Bucket. En caso de utilizar el modelo Token Bucket se debe calcular los parámetros (b, r, p) .

¹³ A partir de estas líneas se utilizará la nomenclatura σ, ρ por ser la más usada para expresar tanto los parámetros leaky bucket como los parámetros b y r de token bucket.

¹⁴ Esto es también aplicable al token bucket. El emisor está limitado por la misma ecuación, la diferencia reside en que se permite una transmisión a una tasa pico p .

3.2.3.5 Descripción del método para la reserva óptima en línea de recursos.

3.2.3.5.1 Introducción

Como se ha visto en el punto anterior es necesario obtener los valores σ , ρ que optimicen la reserva de recursos, es decir, que cumplan con lo que establece la ecuación (9) descrita en el anexo 3.1. Para ello, en este punto se emplea un proceso denominado envolvente empírica con el objeto de obtener una información condensada, que permita optimizar la reserva. Esta información condensada se concreta en lo que se ha denominado puntos envolventes. A partir de los puntos envolventes se describe un método rápido para obtener la reserva óptima.

Cualquier recta $l(t)$ que contenga a la envolvente empírica estará definiendo los parámetros σ , ρ del Leaky Bucket y por definición estará limitando el tráfico como se puede ver en la Figura 3.1. Gráficamente, se puede ver, que para una misma envolvente empírica se pueden ajustar dos rectas $l_1(t)$ y $l_2(t)$, lo que proporciona dos pares de valores σ y ρ .

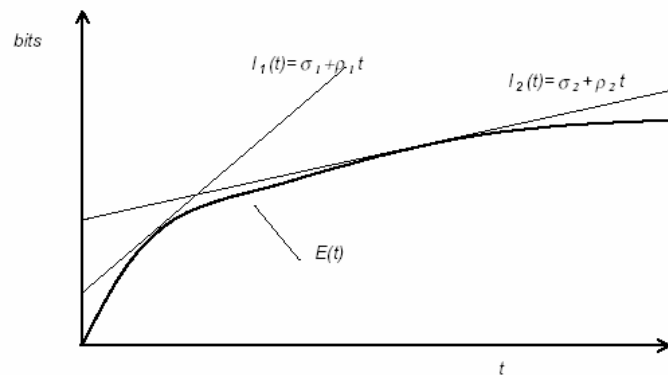


Figura 3.1 Ajuste a la envolvente empírica

Como se ha visto en el punto anterior el valor de σ se calcula en función de la carga y de la tasa ρ . Por ejemplo, si tenemos el siguiente tráfico {1, 2, 0, 4, 2, 0, 1}, los valores de σ para los siguientes valores de ρ {2, 1, 0.5, 0.25} son {2, 4, 6.5, 8.25}. Para cada valor de ρ se define por tanto una función de tráfico limitado $l(t)$. En la Figura 3.2 se tiene la envolvente empírica sobre el tráfico y las rectas anteriores. Como se puede observar, las cuatro rectas tocan la envolvente empírica pero sin cortarla en ningún punto. Por la definición de la envolvente empírica toda recta $l(t)$ que toque sin cortar será la mejor aproximación al tráfico y el valor en el origen será σ . El objetivo, por tanto, es obtener el conjunto de todas estas rectas.

Viendo la figura se puede intuir como obtenerlas: las rectas tocan a la envolvente empírica sólo en determinados puntos: el 1, 2, 5 y 7. La obtención de estos puntos servirá para calcular todas las rectas $l(t)$.

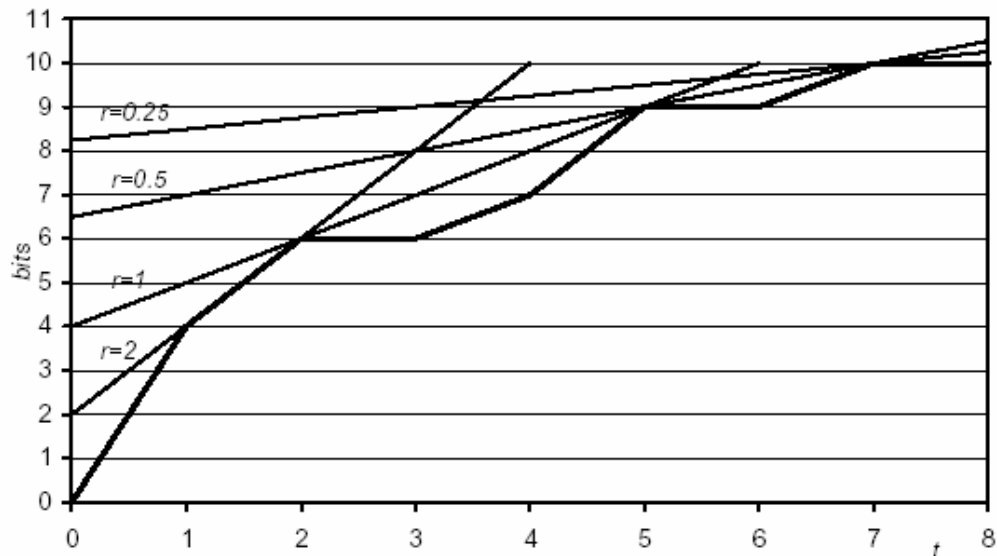


Figura 3.2 Envolvente empírica y alguna función $l(t)$

3.2.3.5.2 Cálculo de los puntos envolventes

La obtención de estos puntos se describe en el anexo 3.1. apartado 3.1.3 y permite obtener todas las posibles rectas $l(t)$ que mejor se ajustan a la envolvente empírica. Y de los resultados obtenidos permiten el desarrollo de un algoritmo para determinar los puntos envolventes. La complejidad de este algoritmo es de $O(n*m)$ donde n son los valores de t^{15} y m los puntos obtenidos. Dicho algoritmo es descrito en anexo 3.1. apartado 3.1.3

3.2.3.5.3 Obtención de la reserva óptima

Para la obtención de la reserva óptima de los recursos, se utiliza los puntos envolventes y con estos puntos se obtienen las rectas envolventes. La demostración de este apartado se detalla en el Anexo 3.1. Apartado 3.1.3

¹⁵ t es el intervalo de tiempo del tráfico.

Conociendo el intervalo en donde se encuentra la raíz ($R=\rho$) se puede obtener directamente el valor de R . La búsqueda del intervalo es fácil y acotado: se puede hacer una búsqueda binaria en los puntos envolventes hasta encontrar el intervalo. Cuando se encuentra el intervalo se aplica el teorema 3 (el cual viene desarrollado en el Anexo 3.1 apartado 3.1.3) para la obtención de R . Un algoritmo para encontrar la solución se detalla Anexo 3.1. El costo de encontrar la solución es muy rápido, con un costo computacional de $O(\log m)$, el valor de m es pequeño (entre 20 y 100 puntos para el vídeo analizado en este trabajo).

La reserva mínima de *buffer* en los nodos puede ser calculada usando las ecuaciones descritas en el Anexo 3.1 (apartado 3.1.1). El algoritmo descrito optimiza la reserva de ancho de banda dando un valor mínimo de σ para un retraso especificado. Si la restricción se impone en el tamaño del *buffer*, esto implica una limitación en el valor de σ y por tanto la reserva debe ser calculada empleando las ecuaciones descritas en Anexo 3.1 (4 y 5) directamente.

Resumiendo, el proceso completo está esquematizado en la Figura 3.3 y se compone de dos fases diferenciadas. La primera fase, que es la más costosa, se puede realizar en el momento que se planea introducir el vídeo en el servidor. La segunda fase ocurre cuando se establece un canal para la transmisión del vídeo. Detallando los procesos en cada fase sobre las Figuras 3.3 y 3.4

FASE I: El objetivo es calcular los puntos envolventes a partir del vídeo almacenado. Para ello, (a) se calcula la envolvente empírica a partir del vídeo, y (b) aplicando el algoritmo descrito en el Anexo 3.1 (Figura 3.1.6) se obtienen los puntos envolventes y se almacenan en un archivo que estará asociado al vídeo. El proceso más costoso de esta fase es el cálculo de la envolvente empírica que es de $O(n^2)$. Existen otros algoritmos más eficientes para calcular la envolvente empírica, pero dado que son procesos *off-line* no se consideraron.

FASE II: empieza cuando se establece un canal para la transmisión del vídeo. En esta fase un cliente (c) pide ver un vídeo iniciando el establecimiento de un canal con unas características determinadas, (d) en el servidor se calcula la reserva usando el algoritmo de la Figura 3.1.8 del Anexo 3.1 sobre los puntos envolventes calculados en la fase anterior y (e) se reservan los recursos calculados en la red empezando a continuación la transmisión. El costo de obtener la solución es, como se ha visto, de $O(\log m)$.

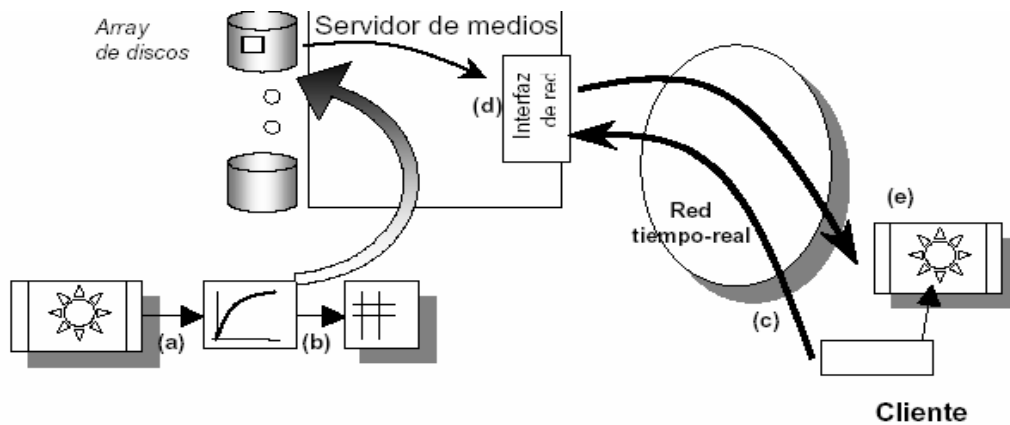


Figura 3.3 Proceso para obtener la reserva óptima

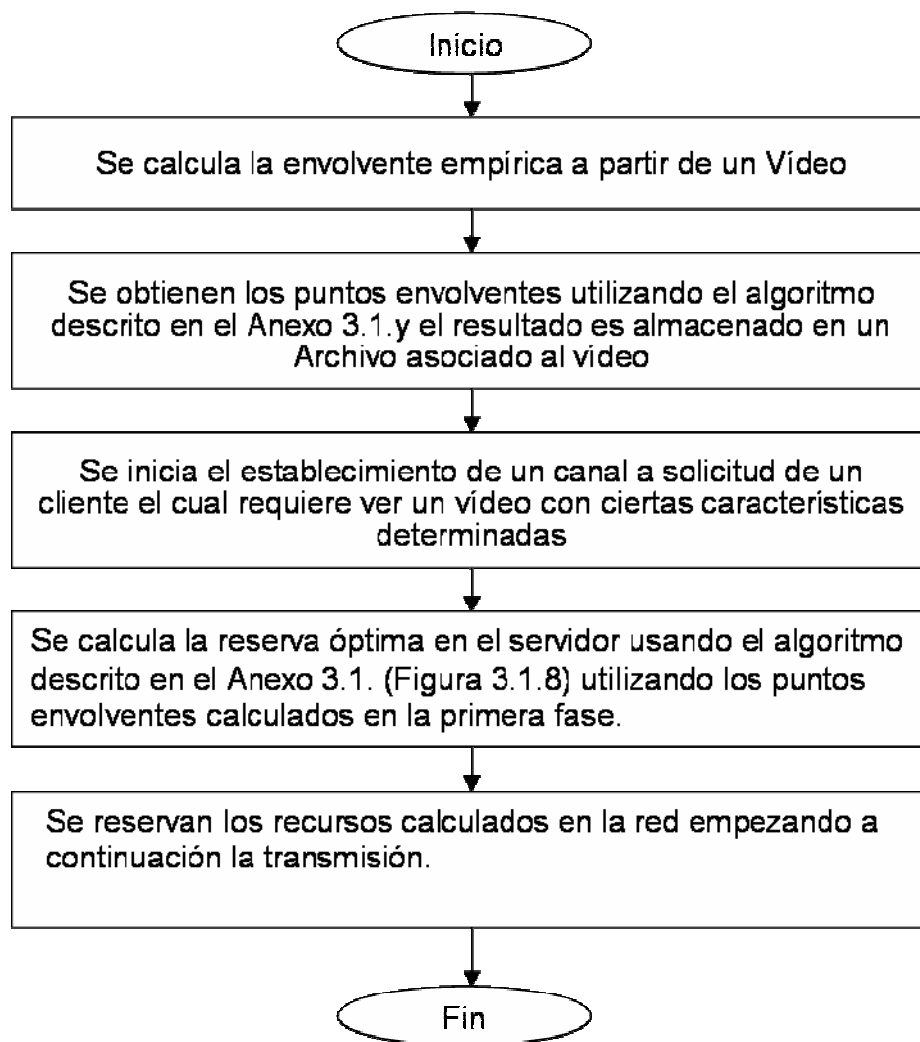


Figura 3.4 Diagrama a Bloques del proceso de Obtención de la Reserva Óptima

3.2.4 Algoritmo de optimización por Reducción de Secuencia Molecular (MSR: Reduction Sequence Molecular)

3.2.4.1 Introducción al algoritmo MSR

Este algoritmo esta patentado por un proveedor de soluciones emergentes en el mercado, y se basa en técnicas desarrolladas originalmente para analizar secuencias del genoma. El enlace clave entre ambos campos altamente dispares: el establecimiento de una red y el genoma, es que ambos campos están referidos fundamental al proceso de secuencias largas o a cadenas de datos. En el caso del genoma, los datos se basan en las moléculas biológicas del DNA que son representadas computacionalmente como una secuencia de caracteres derivados de un alfabeto de cuatro letras, A, C, G, y T. En el caso del establecimiento de una red, los datos se basan en secuencias de bytes que forman un alfabeto de 256 letras.

El genoma de un organismo consiste en un sistema sumamente complejo de las instrucciones para la construcción y el funcionamiento de ese organismo. La clave científica y el desafío médico en este campo es que estas instrucciones están escritas en un lenguaje que todavía no se entiende. El primer paso para decodificar y descifrar un lenguaje desconocido es generalmente la identificación de todos los patrones repetidos que ocurren en diversos ejemplos del lenguaje. Por ejemplo, si intentamos descifrar el lenguaje español, podemos examinar este papel y encontrar que las palabras "son", "y" y "rojo" son repetidas múltiples veces.

Una vez que se han encontrado estos patrones, las dependencias entre ellas se analizan para entender la gramática y la construcción de la lengua. Por ejemplo, la frase "la red" aparece a menudo en este trabajo, pero la frase "es" nunca aparece. De este modo, los científicos en el campo de la biología molecular computacional están aplicando varias técnicas avanzadas de comparación y aprendiendo de la computadora técnicas para continuamente incrementar su comprensión del lenguaje genoma.

Como el análisis de la secuencia del DNA, la Reducción de Secuencia Molecular se basa en descifrar y entender el origen del lenguaje del tráfico en la red. MSR entonces utiliza este conocimiento del lenguaje de la red para "traducir" los datos originales en una representación más eficiente. Un segundo dispositivo MSR en el otro extremo de la red entonces traduce de regreso el tráfico codificado para generar una copia idéntica del tráfico original. Los algoritmos avanzados del lenguaje de análisis de MSR detectan los patrones repetidos en el tráfico y construyen un vocabulario dinámico de los patrones claves transmitidos

a través de la red. En el mismo tiempo, MSR aprende las dependencias entre estos patrones para construir una base de conocimientos completa de cómo estos patrones básicos pueden ser arreglados para construir otros patrones más grandes y más complejos. Esta base de conocimientos es continuamente actualizada con más patrones y dependencias que están aprendiendo del tráfico de la red. En el mismo tiempo, patrones más viejos que son poco frecuentes y no contribuyen a la reducción de datos se envejecen fuera y se substituyen por los nuevos y por más patrones relevantes. De este modo, la base de conocimiento de MSR se mantiene sincronizada continuamente con el lenguaje dinámico y las características de tráfico de la red.

Las capacidades únicas de aprender-lenguaje de MSR, permiten que esta tecnología sea aplicada a todos los tipos de tráfico de la red y con muy baja latencia por paquete. La base de conocimientos dinámico creado por MSR permite la detección y la reducción de patrones de cualquier tamaño que son separados por un número pequeño o grande de bytes y paquetes. Por lo tanto, ambos patrones textuales pequeños que se pueden repetir dentro de distancias cortas así como patrones binarios más largos que se puedan separar por centenares de kilobytes o de megabytes de tráfico se pueden detectar y reducir por esta tecnología. La estructura única de la base de conocimientos del MSR también proporciona muy alta escalabilidad y permite que esta tecnología funcione en tiempo real a muy alta velocidad.

En contraste con MSR, las tecnologías tradicionales de compresión se basan en buscar y sustituir patrones dentro de un buffer histórico fijo. Este buffer fijo pone límites en la distancia máxima entre los patrones (es decir, sólo los patrones que son menos de 2 kilobytes y que a parte serán comprimidos) así como la escalabilidad y la velocidad de funcionamiento de estos esquemas tradicionales. Mientras que se aumenta el tamaño del buffer de búsqueda para identificar más patrones repetidos, el poder de procesamiento y latencia también aumentan dramáticamente. Por lo tanto las técnicas tradicionales de compresión han hecho frente a una compensación difícil entre la latencia y las tarifas de reducción. El resultado es que mientras la compresión está extensamente disponible en routers, esta compensación evita que la mayoría de las empresas utilicen esta característica a través de sus enlaces WAN.

La mayoría de las redes no pueden afrontar la latencia adicional (a menudo 30 milisegundos o más) para alcanzar niveles de reducción del destino. Por comparación, MSR agrega típicamente menos de 2 milisegundos de latencia mientras que alcanza mucho más tasa de reducción.

Empleando totalmente un nuevo acercamiento para aprender dinámicamente y almacenar inteligentemente la base de conocimientos de patrones repetidos, el MSR supera estas limitaciones para proveer significativamente de mayores patrones de detección y reducción con escalabilidad virtualmente ilimitada.

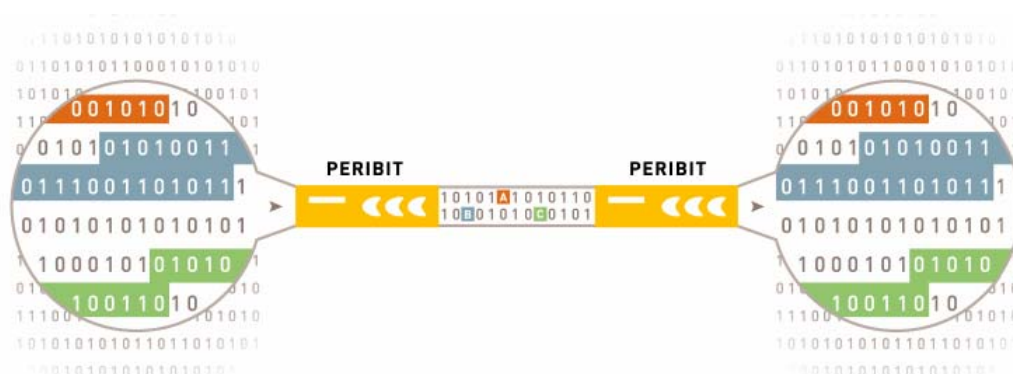


Figura 3.5 Un esquema ilustrativo de MSR de codificación y decodificación a través de un enlace WAN

3.2.4.2 El algoritmo MSR

El algoritmo de Reducción de Secuencia Molecular (MSR) funciona al encontrar los patrones de repetición de clasificación variable de cualquier secuencia de datos, a través de múltiples paquetes, aplicaciones y sesiones. MSR adaptativamente aprende los patrones que ocurren con más frecuencia posible en los datos y asigna a cada patrón una etiqueta interna única.

Continuamente aprende nuevos patrones, el MSR construye una tabla de patrones únicos y de etiquetas correspondientes análogo a un diccionario de palabras, frases, oraciones, etc., que son los más comúnmente utilizados en los datos. MSR utiliza este diccionario de patrones para codificar la secuencia de datos en un formato altamente eficiente substituyendo cada caso subsecuente del patrón por su etiqueta correspondiente. Las numerosas redundancias no-informativas que desperdician los recursos de red son así revisadas en la secuencia de datos antes de que se transmitan a través de la red en donde la mayoría de los enlaces son de capacidad restringida. Un segundo motor de MSR recibe la secuencia de datos reducida y altamente optimizada la convierte de nuevo a su forma original substituyendo cada etiqueta por el patrón original de los datos.

MSR funciona sobre el tráfico de red en tiempo real en la capa IP (capa 3). Los paquetes IP pueden ser seleccionados para el proceso de MSR basado en varios parámetros incluyendo su dirección destino. Si el paquete es destinado a través de un enlace restringido en ancho de banda, y existe un segundo dispositivo MSR a través de este enlace, entonces este es seleccionado para la reducción de datos de MSR. Una vez que un paquete se toma, todos los bytes en el paquete, el encabezado así como los datos útiles, son procesados. Cada byte, independiente de su fuente, es tratado como un símbolo de entrada independiente por el MSR. Puesto que los paquetes codificados MSR se pueden solo decodificar por un dispositivo específico de codificación MSR, todos los paquetes procesados se encapsulan en un túnel IP para la transmisión a un dispositivo decodificador. Así, al reempaquetar MSR puede reducir tanto el número de bytes transmitidos como el número de paquetes que cruzan la red.

Desde todos los paquetes IP de todas las aplicaciones, usuarios, y sesiones pueden ser procesados y reducidos, MSR proporciona ventajas significativas en la reducción de datos en el enlace de red. Además, realizando su reducción de datos en la capa IP, el MSR sigue siendo totalmente transparente al resto de la red, aplicaciones, servidores, y clientes. La Tabla 3.1 demuestra algunos de los resultados medidos reales del MSR en varias aplicaciones en vivo de la red.

Aplicación	(%) Reducción de Datos MSR
Email	75
Exchange	79
MS SQL	80
SAP	73
Telnet	67
Oracle	78
File Sharing	81
Lotus Notes	78
HTTP Servers	96
Search Servers	84
Image Servers	70

Tabla 3.1 Resultados actuales de MSR sobre una variedad de tráfico real en la red

3.2.4.2.1 Componentes principales del algoritmo MSR

Los cuatro componentes primarios del algoritmo MSR son:

- El descubrimiento de patrones
- La construcción del diccionario adaptativo
- La sincronización del diccionario
- La reducción de datos

Las secciones siguientes describen cada uno de estos componentes en más detalle.

3.2.4.2.2 El descubrimiento de patrones

El algoritmo de descubrimiento de patrones es uno de los diferenciadores clave de esta tecnología. La velocidad y la granularidad con los cuales el algoritmo de descubrimiento del patrón busca nuevos patrones es crítico a las altas tasas de reducción de datos proporcionadas por MSR y a las altas velocidades de la red en la que esta tecnología pueda funcionar. Mientras que cada nuevo símbolo de entrada es procesado por el motor de MSR, primero se comprueba para determinar si es o no parte de un patrón preexistente el cual se ha aprendido y se ha agregado ya al diccionario.

Si el símbolo no es parte de un patrón preexistente del diccionario, el algoritmo del descubrimiento del patrón de MSR busca un buffer que consiste en símbolos previamente procesados para determinar si el símbolo actual es parte de un nuevo patrón que ha ocurrido anterior en este buffer pero que aún no ha sido agregado al diccionario. Para maximizar la cantidad de datos de entrada que se puedan almacenar en este buffer histórico "mirar - atrás", el buffer así mismo se codifica en un formato altamente condensado. Por lo tanto, el tamaño físico de este buffer puede ser de 10 a 100 Mbytes, que alternadamente pueden representar 100 a 1000 Mbytes de los datos originales sin codificar.

El tamaño de este buffer histórico es determinado solamente por la cantidad de memoria física disponible en el dispositivo. La velocidad del algoritmo no es afectada por el tamaño del buffer. La capacidad del MSR de buscar este buffer grande a altas velocidades es por lo tanto una fortaleza clave del algoritmo y la fuente de las altas tasas de reducción de datos que alcanza.

Los patrones descubiertos en este buffer se comprueban para considerar si pasan un umbral significativo, en que casos son agregados al diccionario por el módulo de construcción del diccionario.

3.2.4.2.3 Construcción adaptativa del Diccionario

Cuando un nuevo patrón es descubierto en el buffer de búsqueda, se entrega al módulo de construcción del diccionario "aprende" el patrón y agregándolo al diccionario. El patrón es agregado al diccionario asignándole una etiqueta interna, que entonces será utilizada para referir únicamente a este patrón. Puesto que el número de las etiquetas del diccionario es finito, el módulo de construcción del diccionario también contiene un componente "olvidarse" que desasocia con eficiencia una etiqueta de un viejo patrón, así permitiendo que la etiqueta sea reasignada a un nuevo patrón.

Los patrones que son seleccionados a ser olvidados son los que se utilizan lo menos frecuente posible y por lo tanto de menos importancia a los datos actuales. Así, si un patrón se utiliza con frecuencia (por ejemplo la palabra "El") podría permanecer en el diccionario indefinidamente mientras que un patrón con menos frecuencia usado (por ejemplo., la palabra "Quiropráctico") seguirá permaneciendo en el diccionario por un tiempo corto y pronto será substituida solamente por un patrón que ocurre con más frecuencia.

El módulo de construcción del diccionario se adapta a los datos que están siendo procesados, siempre manteniendo solamente los patrones más relevantes y más útiles del diccionario. Los procesos de aprender y olvidar de este módulo son ambos altamente eficientes y no comprometen la linealidad y la velocidad del algoritmo total. La memoria requerida para almacenar los patrones del diccionario es reducida al mínimo codificando las etiquetas y los patrones en una forma óptima compacta. Por lo tanto, como en un diccionario verdadero de la lengua, una tabla relativamente pequeña de patrones (10 Mbytes) puede codificar un número muy grande de secuencias originales de los datos (de 100 a 1000 Mbytes). La completa originalidad adaptativa de este proceso también asegura que el MSR pueda correr continuamente e incrementalmente sobre una secuencia indefinida de símbolos de entrada.

3.2.4.2.4 La sincronización del diccionario

Cómo se mencionó en líneas arriba, MSR funciona sobre ambos extremos de un enlace de la red codificando patrones con las etiquetas en un extremo y después descifrando las etiquetas de nuevo a los patrones originales en el otro extremo. Para garantizar que este proceso es

totalmente de menor pérdida y completamente transparente, es necesario asegurar que todas las etiquetas codificadas sean descifradas correctamente exactamente a sus patrones originales. Esto es alcanzado por el módulo de la sincronización del diccionario a través de un canal de comunicaciones independiente entre cada par de dispositivos MSR.

Mientras que el diccionario de codificación es actualizado incrementalmente, el módulo de la sincronización calcula la significación de cada patrón del diccionario e incrementalmente comunica los patrones relevantes para la decodificación del dispositivo MSR. Puesto que el diccionario es actualizado incrementalmente, el overhead de la comunicación de esta sincronización es mínimo y distribuido uniformemente.

El proceso de sincronización no requiere de grandes periodos de volumen de transferencia del diccionario y en cambio programa la sincronización de paquetes tal que su overhead es funcionalmente insignificante. Puesto que el diccionario es mantenido por un módulo independiente y sincronizado a través de un canal de comunicaciones independiente, el flujo de proceso de datos se desacopla del proceso de la sincronización del diccionario. Los procesos de reducción y de reensamble de datos por lo tanto no participan en el proceso de la sincronización del diccionario, aunque utilizan los resultados de la sincronización del diccionario para garantizar la codificación y decodificación exacta.

Puesto que un sólo dispositivo MSR puede comunicarse con múltiples dispositivos remotos (es decir, configuración uno a muchos así como de muchos a uno), ahí pueden existir diccionarios de codificación múltiples en el dispositivo central del "hub". El MSR utiliza una arquitectura compartida del diccionario en este dispositivo central, para maximizar el uso de los recursos del diccionario y proporcionar mayor escalabilidad de punto final. En esta configuración compartida de punto-a-multipunto del diccionario, el módulo de sincronización transfiere solamente un patrón a un dispositivo particular MSR si el patrón está siendo actualizado utilizado en este enlace.

Por ejemplo, si un site de la matriz en San Francisco tiene enlaces a dos oficinas remotas en Nueva York y en Singapur, el dispositivo MSR en San Francisco mantendrá un diccionario compartido de los patrones para ambas oficinas remotas, mientras que los dispositivos MSR de Nueva York y Singapur mantendrán solamente esos patrones del diccionario que sean relevantes a sus sites específicos.

3.2.4.2.5 Reducción de datos

La componente MSR de Reducción de datos funciona directamente en la trayectoria de datos al codificar los paquetes individualmente basados en su diccionario de patrones. Cómo cada paquete es procesado, es primero examinado símbolo por símbolo para determinar si contiene un patrón que exista ya en el diccionario de codificación. Si no se encuentra ningún patrón preexistente en el diccionario, el símbolo entonces es comprobado por el módulo de descubrimiento del patrón para considerar si es parte de un nuevo patrón que se deba agregar al diccionario. Si corresponde un patrón encontrado en un preexistente diccionario (y el patrón del diccionario se ha sincronizado correctamente con el dispositivo correspondiente MSR decodificado), el proceso de reducción de datos substituye el patrón en la secuencia de datos por la etiqueta correspondiente para este patrón. Este proceso se realiza en cada paquete independientemente del resto de los paquetes.

Puesto que la sincronización del diccionario se realiza con un proceso separado, el MSR no impone ninguna dependencia entre los paquetes procesados. Esto permite que el proceso de la reducción de datos funcione a altas velocidades y reduzca al mínimo el proceso de latencia. Después de que el último símbolo en el paquete es procesado, el módulo de reducción de datos genera inmediatamente toda la salida codificada para el paquete y prepara el paquete codificado para la encapsulación y la transmisión.

Aunque el MSR funciona a través de una sesión cruzada encontrando patrones a través de todos los paquetes en la secuencia de datos, el reemplazo del patrón se realiza solamente dentro de los paquetes individuales. Adicionalmente para minimizar el proceso de latencia, se permite que cada paquete reducido sea tratado independientemente del resto de los paquetes. Así los paquetes que son codificados por un dispositivo MSR se pueden transmitir al dispositivo decodificador en una manera, sin conexión, sin ninguna imposición ordenada o restricción de confiabilidad ante la secuencia de datos codificada.

Puesto que los paquetes que son codificados por un dispositivo MSR se deben decodificar por un dispositivo específico decodificador MSR, los paquetes codificados se transmiten al dispositivo decodificador en una forma encapsulada. Como se encapsula el paquete entero de entrada, el encabezado así como los datos útiles del paquete se pueden codificar por el MSR. Además de reducir el número de bytes transmitidos, el MSR puede también reducir el número de paquetes transmitidos encapsulando los paquetes múltiples de entrada en un sólo paquete de salida.

MSR reduce la latencia de este proceso de la encapsulación y el proceso de reducción de paquete al fijar un límite máximo de T milisegundos para la encapsulación. El umbral T es fijado por default a 2 ms pero puede ser automáticamente ajustado basado sobre el RTT a través de la red.

3.2.4.2.6 Las fuerzas y los diferenciadores de MSR

La tecnología MSR puede ser comparada con la compresión de datos de uso general tradicional puesto que ambas tecnologías reducen datos o tamaños del objeto al quitar repeticiones y redundancias en los datos. Por lo tanto, el objetivo de ambas tecnologías es similar, es decir, encontrar y eliminar repeticiones en los datos.

La eficacia de una tecnología de compresión se mide típicamente en términos de la tasa de la reducción de datos que puede alcanzar. Pero la aplicación de la compresión de datos a la infraestructura del establecimiento de una red expone varios apremios adicionales que deben también ser considerados en la evaluación de una tecnología de reducción de datos. Estos nuevos apremios incluyen operabilidad en tiempo real, velocidad, latencia, dependencias entre paquetes codificados, transmisión de datos codificados con un medio no confiable y la operación incremental y continua.

Esta parte se examina el MSR y las técnicas de compresión de datos de uso general tradicionales con respecto a cada uno de estas ediciones y de su impacto en la reducción de datos y escalabilidad. Las técnicas de compresión de datos que se utilizan para una comparación son aquellas derivadas de los algoritmos de Lempel-Ziv [LZ77, LZ78]. Muchas de las técnicas extensamente conocidas de compresión, incluyendo gzip, compress, LZW, y LZS (Stac compresión), derivadas del trabajo original de Lempel y Ziv.

3.2.4.2.6.1 Las fuerzas primarias de MSR que distinguen estas de las tradicionales técnicas de compresión son:

- ✓ La reducción de datos lineal y la escalabilidad del proceso
- ✓ Desacople del flujo de datos y la sincronización del diccionario
- ✓ Incremental y funcionalidad continua

Las secciones siguientes discuten cada una de las características anteriores y las compara a las de las técnicas basadas LZ tradicionales de compresión.

3.2.4.2.6.2 La reducción de datos y la escalabilidad del proceso

El grado de reducción de datos que se puede alcanzar por un algoritmo de compresión es una función directa del número de patrones repetidos que puede descubrir. El número de patrones repetidos detectados depende del tamaño del buffer que guarda una copia de los símbolos previamente procesados de entrada y la eficiencia del algoritmo de búsqueda que busca repeticiones en este buffer. El tamaño del buffer histórico de entrada influye directamente en el número de patrones que pueden ser descubiertos y por lo tanto del grado de reducción de datos que el algoritmo puede alcanzar. El orden para que un patrón repetido sea detectado, el número de bytes que ocurren entre dos repeticiones subsecuentes de los patrones deben ser más pequeños que el tamaño de buffer. Es decir, ambos casos de repetición deben bajar dentro del buffer. Por ejemplo, asumiendo:

D = la distancia promedio entre cualquier dos repeticiones de un patrón

B = tamaño del buffer de búsqueda

Si $D = 100$ bytes, implica que un patrón dado está repetido, en promedio, cada 100 bytes en los datos de entrada. Para detectar eficientemente una repetición de este patrón, B debe claramente ser más grande de 100 bytes. Si B es igual a 100 bytes más dos veces el tamaño de la repetición, la probabilidad de detectar cualquier repetición dada del patrón es el 50%. Mientras que B se hace más pequeño la probabilidad de detectar la repetición disminuye, y mientras que B se hace más grande la probabilidad aumenta y se acerca a 100%. Así, mientras más grande es el tamaño del buffer mayor es el número de patrones que será detectado y por lo tanto mayor es la reducción de datos que será alcanzado.

De acuerdo con el análisis anterior, los algoritmos de compresión deben intentar claramente maximizar el tamaño de buffer B para maximizar su tasa de reducción de datos. Pero como se aumenta B, el número de los bytes o símbolos que se deben examinar para buscar los patrones repetidos también incrementa. Así, la tasa de reducción de datos y la velocidad de proceso de un algoritmo de compresión se deben examinar simultáneamente.

El factor clave que determina la velocidad de una técnica de reducción de datos es su complejidad, que se define en los términos del número de operaciones de CPU requeridas para procesar una secuencia de entrada de n símbolos. La complejidad del algoritmo de MSR es $O(n)$ o de "Orden (n)". Esto indica que el número de operaciones requeridas para procesar " n " bytes de entrada es proporcional a " n ". MSR es por lo tanto un algoritmo lineal puesto que el número de operaciones de CPU requeridas

para procesar la entrada crece linealmente con el número de símbolos de entrada. Así, si la tasa de transmisión de datos en un enlace de red fuera doblada, MSR necesitaría realizar dos veces el número de operaciones en el mismo tiempo para procesar todos los datos sobre el enlace. El algoritmo de descubrimiento del patrón de MSR es el componente clave que proporciona esta escalabilidad lineal.

El algoritmo de descubrimiento del patrón tiene la capacidad de buscar en cualquier tamaño de un buffer arbitrario en un tiempo constante, así asegurándose de que cada símbolo de entrada incurra en un número constante de operaciones. Por lo tanto, el tamaño del buffer de búsqueda, B , no aparece en la ecuación de la complejidad de MSR. Además, puesto que la tasa de reducción del algoritmo del MSR es proporcional al tamaño del buffer, podemos concluir que la velocidad del algoritmo de MSR es independiente de la tasa de reducción que puede alcanzar. Es decir, el tamaño del buffer de MSR, B , se puede hacer arbitrariamente grande para alcanzar una reducción de datos más alta sin la afectación de la velocidad o de la complejidad del algoritmo.

En contraste con MSR donde las variables del tamaño del buffer de búsqueda y la velocidad son independientes, la mayoría de las técnicas tradicionales de compresión basadas en LZ son inversamente relacionadas. Por lo tanto, las técnicas tradicionales de compresión deben compensar la velocidad para alcanzar una reducción de datos más alta o la reducción de datos de compensación para alcanzar la velocidad de proceso más alta. Esto es debido al hecho de que la complejidad de estos algoritmos de compresión es $O(n f(B))$, donde " n " es el número de símbolos de entrada y B es el tamaño del buffer de búsqueda. Esto indica que el número de operaciones requeridas para procesar una secuencia de " n " símbolos de entrada es proporcional al producto de " n " y de una función del tamaño de buffer B . Así, si el tamaño de buffer B aumenta para alcanzar una reducción de datos más alta, el número de operaciones requeridas para procesar " n " símbolos de entrada también aumenta. Esto alternadamente disminuye la velocidad y el rendimiento de procesamiento total del algoritmo de compresión.

Debido a la dependencia inversa de la velocidad al tamaño del buffer, las técnicas tradicionales basadas en LZ de compresión utilizan generalmente un buffer muy pequeño de búsqueda. Los algoritmos de compresión fuera de línea tales como "gzip", utilizan típicamente un tamaño de buffer de 8 KB a 32 KB. Los algoritmos de compresión en línea deben funcionar en tiempo real con tráfico de red en vivo utilizando incluso buffers más pequeños, típicamente 2KB o menos.

Adicionalmente, las restricciones de la red (descritas en el siguiente apartado) limitan a las técnicas tradicionales de compresión forzándolas a funcionar independientemente en cada paquete individual. Es decir, cada paquete se comprime independientemente del resto de los paquetes y el algoritmo de la compresión reajusta su estado después de que cada paquete se procesa. Puesto que el tamaño de buffer de búsqueda no puede exceder el tamaño de los datos de entrada que se están procesando, las técnicas tradicionales de compresión de la red por lo tanto se limitan a menudo a un tamaño de buffer de 1500 bytes (es decir, al MTU¹⁶ de la red).

La escalabilidad y la reducción de datos MSR y la compresión tradicional, puede ser explorada más a fondo modelando la distancia entre las repeticiones del tráfico en la red. Los enlaces de la red típicamente se actualizan a velocidades de mayor capacidad para apoyar tasas de transferencia más altas de datos, más usuarios y más aplicaciones. Por lo tanto, enlaces de mayor velocidad apoyan típicamente mayores volúmenes y variedad de tráfico. Esto además implica que a velocidades más altas, la distancia promedio entre objetos repetidos también aumenta (es decir, pues las velocidades de la red aumenta el número promedio de bytes que ocurren entre dos instancias por los incrementos mismos de patrones). Podemos por lo tanto modelar la distancia entre las repeticiones cómo una variable aleatoria normalmente distribuida, donde la media y la desviación estándar son una función de la velocidad del enlace. Para el análisis actual, la distancia media entre las repeticiones se modela para ser el 20% del número de los bytes transferidos por segundo y la desviación estándar es el 10% de la tasa de transferencia del byte (es decir, la mitad de la media).

Media: $\mu = (\text{bytes por segundo}) * 0.2$

Desviación Estándar: $\sigma = (\text{bytes por segundo}) * 0.1$

De acuerdo con este modelo, podemos calcular exactamente la probabilidad de un buffer de búsqueda de tamaño B para capturar dos casos de una repetición. Por ejemplo, si B es igual a la media, μ , entonces la mitad del promedio de las repeticiones caerá dentro del buffer y la mitad bajará fuera del buffer (asumiendo, para propósitos del modelo, que el tamaño de la repetición es insignificante comparado a la distancia

¹⁶ MTU (Maximum Transmission Unit). Unidad máxima de transmisión. Cantidad máxima de datos que pueden transmitirse por una red en un sólo paquete. Cada tecnología de red define una MTU (por ejemplo, la MTU de la Ethernet es de 1500 octetos).

entre las repeticiones). Así para $B = \mu$, el porcentaje de repeticiones que serán detectadas es el 50%. Para todos los valores de B , el porcentaje de detección de patrones se calcula como el área debajo de la curva estándar de la distribución normal menos que Z , donde Z se define como:

$$Z = (B - \mu) / \sigma$$

La Tabla 3.2 muestra el tamaño de buffer B para MSR y para la compresión tradicional a varias velocidades de la red. Puesto que MSR puede incrementar el tamaño de buffer sin afectar la complejidad del algoritmo, el tamaño de buffer para cada velocidad del enlace se fija como:

$$B_{MSR} = \mu + 2 * \sigma$$

Por lo tanto, la cuenta de Z para MSR es siempre:

$$Z_{MSR} = (B_{MSR} - \mu) / \sigma = 2$$

Una cuenta constante de Z de 2 implica que la probabilidad de una repetición que ocurre dentro de B_{MSR} bytes es de 0.97, es decir, el porcentaje de repeticiones que serán detectadas por MSR es una constante de 97%.

Debido a las limitaciones de los algoritmos tradicionales de compresión, el tamaño de buffer, B , para estas técnicas se mantiene constante en 1500 bytes. Así para cada velocidad del enlace, el porcentaje de patrones que son detectados es calculado basado en la siguiente cuenta de Z :

$$Z_{Compresion} = (1500 - \mu) / \sigma$$

La probabilidad de dos repeticiones cualesquiera que ocurran dentro de 1500 bytes entonces se calcula en cada caso como el área debajo de la curva normal estándar menos que $Z_{Compresion}$.

Velocidad	Velocidad entre Repeticiones (Bytes)		Tamaño del Buffer (Bytes)		% de Repeticiones Detectadas	
	Media (μ)	Desv. Standar (σ)	MSR	Compresión Tradicional	MSR	Compresión Tradicional
56 Kbps	1434	717	2867	1500	97.7%	53.7%
128 Kbps	3277	1638	6554	1500	97.7%	13.9%
256 Kbps	6554	3277	13107	1500	97.7%	6.1%
512 Kbps	13107	6554	26214	1500	97.7%	3.8%
1.54 Mbps	40370	20185	80740	1500	97.7%	2.7%
45 Mbps	1179648	589824	2359296	1500	97.7%	2.3%
155 Mbps	4063232	2031616	8126464	1500	97.7%	2.3%
622 Mbps	16305357	8152678	32610714	1500	97.7%	2.3%

Tabla 3.2. El porcentaje de repeticiones detectadas por MSR y la compresión tradicional a varias tasas de velocidades de datos

La Figura 3.6 muestra los resultados de este análisis en un formato gráfico. Se observa que los resultados demuestran solamente el porcentaje de repeticiones que se detectan y no la reducción de datos total alcanzada. La reducción de datos total se puede calcular como el producto de la tasa en la cual las repeticiones ocurren en los datos y el porcentaje de estas repeticiones que son detectadas. Así si la tasa en la cual las repeticiones ocurren es del 80% y el porcentaje de repeticiones detectadas es 97.7%, la reducción de datos total será aproximadamente del 78%.

Para ciertos tipos de datos (tales como texto ASCII) MSR y para las técnicas tradicionales de compresión pueden adicionalmente proporcionar reducción de datos usando esquemas de codificación estadísticos para comprimir más los paquetes de salida. Estas técnicas estadísticas de codificación son cálculos costosos y proporcionan solamente ventajas a un sistema muy limitado de tipos de datos. Así, si la reducción de datos alcanzada es alta eliminando repeticiones (como el caso de MSR) el valor incremental de esta codificación estadística es marginal y no justifica el costo computacional agregado.

Por otra parte, puesto que el porcentaje de repeticiones detectadas es bajo para las técnicas tradicionales de compresión, deben confiar ampliamente en la codificación estadística para proporcionar la compresión adicional para ciertos tipos de datos limitados.

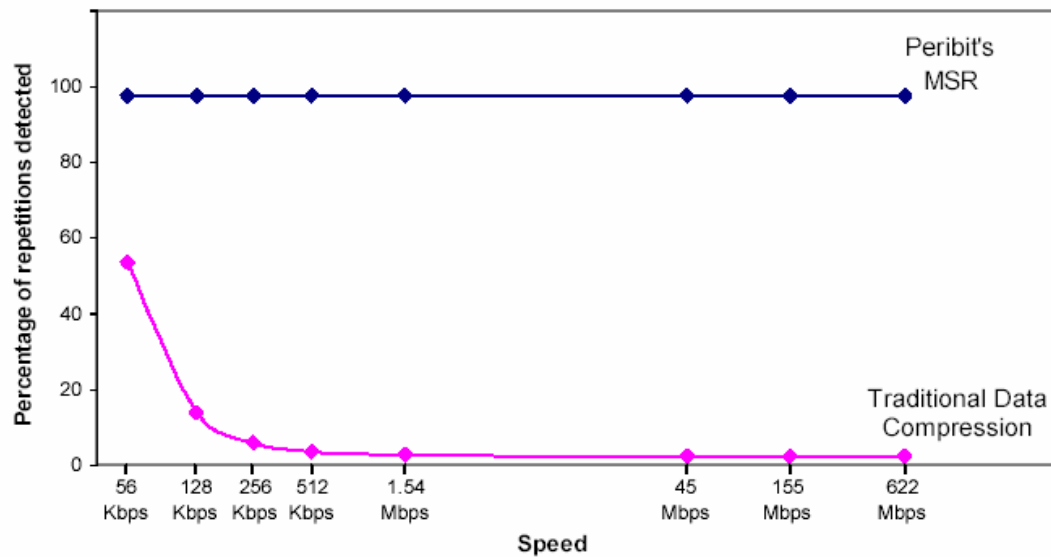


Figura 3.6 Comparación de la escalabilidad y la reducción de datos MSR vs Técnicas Tradicionales de Compresión de Datos

3.2.4.2.6.3 El flujo de datos desacoplado y la sincronización del diccionario

Un segundo punto importante de MSR es la técnica que utiliza para mantener y sincronizar sus diccionarios de patrones. El diccionario MSR consiste en una tabla explícita de patrones y de etiquetas asociadas. Este diccionario así mismo se codifica en una estructura de datos optimizada que permita maximizar el almacenaje de patrones en una cantidad pequeña de memoria física. Esta estructura de codificación optimizada también reduce al mínimo el overhead de la sincronización del diccionario permitiendo que el dispositivo de codificación MSR transmita solamente actualizaciones incrementales pequeñas del diccionario al dispositivo decodificador.

Puesto que el diccionario MSR es mantenido y actualizado explícitamente por un módulo independiente, los componentes de procesamiento de datos son desacoplados del proceso de sincronización del diccionario. Aunque la reducción de datos y los procesos re-ensamblados requieren la sincronización del diccionario para garantizar la codificación y decodificación exacta, no participan directamente en la tarea de sincronización del diccionario. Los paquetes codificados y encapsulados que se transmiten del dispositivo de codificación del MSR al dispositivo decodificador por lo tanto no tienen ninguna dependencia entre ellos.

Esto además implica que la entrega confiable y en orden de estos paquetes encapsulados no es un requisito necesario para la sincronización del diccionario y codificación exacta de los datos. Ésta es una fuerza clave de MSR, puesto que permite que cada paquete de salida sea encapsulado en una conexión independiente. Si un paquete encapsulado se pierde durante el tránsito o si los paquetes encapsulados se entregan al dispositivo decodificador fuera de servicio, el motor decodificador MSR no será afectado y continuará decodificando exactamente todos los paquetes. Puesto que la comunicación end-to-end se basa típicamente en protocolos de transporte confiables por ejemplo TCP, los paquetes encapsulados comprimidos en un segundo canal TCP pueden afectar perceptiblemente el funcionamiento de la red. Los lazos de control múltiples de la jerarquización TCP dentro de cada uno pueden causar comportamiento inestable de la red y degradar perceptiblemente el funcionamiento. Por lo tanto, la capacidad de MSR de realizar la reducción de datos a través de todos los paquetes pero de transmitir cada paquete procesado vía un canal no confiable del transporte es un diferenciador significativo de esta tecnología.

Según lo descrito en la sección anterior, las técnicas tradicionales de compresión de datos se limitan a menudo a los tamaños de buffer pequeños de cerca de 2 Kilobytes debido a las razones de funcionamiento. Por lo tanto, estas técnicas tratan cada paquete de entrada independientemente del resto de los paquetes y no buscan repeticiones a través de los paquetes múltiples. Incluso si las técnicas tradicionales de compresión pudieran superar esta limitación del funcionamiento y buscar repeticiones a través de los paquetes múltiples, entonces podrían enfocarse con la emisión de las dependencias al interior del paquete.

La mayoría de las técnicas basadas LZ de compresión de datos utilizan un esquema de codificación de repeticiones basado en apuntadores que codifican patrones repetidos señalando la posición anterior en la cadena de entradas donde ocurrió el patrón. Por ejemplo, si un patrón en el paquete que se está procesando actualmente hubiera ocurrido en el paquete anterior que fue procesado, el algoritmo de compresión codificaría el caso actual del patrón señalando su localización exacta en el paquete anterior. Así, el algoritmo de compresión requeriría que ambos paquetes lleguen al dispositivo decodificador en el mismo orden que fueron transmitidos y que ninguno de los dos paquetes esté perdido durante el transporte.

Este requisito puede ser alcanzado solamente encapsulando todos los paquetes comprimidos vía un protocolo confiable de transporte tal como TCP.

Según lo descrito anteriormente, el encapsulado de los paquetes comprimidos en un canal de TCP puede causar el comportamiento imprevisible de la red debido al anidamiento de los lazos de control múltiple de TCP. En el mismo tiempo, puesto que muchas sesiones independientes del usuario son transportadas con una sola sesión de encapsulación TCP, los mecanismos de control de congestión de TCP's podrían reducir perceptiblemente la utilización del ancho de banda y degradar a un más el funcionamiento. Debido a estos factores, las técnicas basadas LZ tradicionales de compresión no pueden codificar repeticiones a través de los paquetes y deben reajustar su estado después de cada paquete. Esto refuerza aún más las simples limitaciones del paquete de las técnicas tradicionales de compresión que fueron descritas en la sección anterior.

3.2.4.2.6.4 Incrementabilidad y funcionalidad continúa

Un tercer diferenciador significativo de MSR es su capacidad de funcionar incrementalmente y en tiempo real en una cadena continua de datos. MSR procesa cada símbolo de entrada incrementalmente y genera salida mientras que continúa procesando más datos de entrada. Los algoritmos no incrementales procesan típicamente bloques grandes de entrada a la vez y pueden generar solamente salida después de que se haya procesado un bloque completo. La naturaleza incremental de MSR es una característica importante puesto que evita la latencia adicional de tener que esperar una cantidad fija de datos de entrada antes de generar la salida reducida. Aunque el MSR busca parar y eliminar repeticiones a través de los paquetes, la salida codificada para un paquete de entrada está disponible para la transmisión tan pronto como se haya procesado el paquete de entrada. Así no hay latencia adicional introducida para cada paquete de entrada.

El MSR puede funcionar continuamente con recursos finitos en una cadena indefinida de datos de entrada. Todos los recursos, tales como espacio físico de memoria y del diccionario, son utilizados óptimamente y se reutilizan para asegurarse de que el algoritmo nunca exceda sus límites asignados. La reutilización de los recursos del diccionario se realiza a través de un patrón adaptativo "que se olvida" del proceso que substituye patrones infrecuentemente utilizados por patrones más relevantes y recientemente descubiertos. Los módulos de administración de recursos no afectan la linealidad o escalabilidad total del algoritmo de MSR de ninguna manera.

Así, el diccionario adaptativo MSR aprende y olvida procesos para poder funcionar continuamente mientras que todavía mantiene la complejidad lineal y la escalabilidad del algoritmo.

Capítulo 1V.

Caso Estudiado

Como llevar a la práctica, lo descrito en la teoría:

A continuación se describe, el Caso Estudiado

4.1 Red corporativa de estudio

El caso de estudio fue llevado a la práctica en una empresa corporativa con presencia a nivel nacional, la cual por cuestiones de seguridad y protección a la misma se omite su nombre. Por restricciones de acceso y de limitaciones de este trabajo solo se consideró el nodo Veracruz. En la Figura 4.1 se puede apreciar cómo esta integrada esta red corporativa.

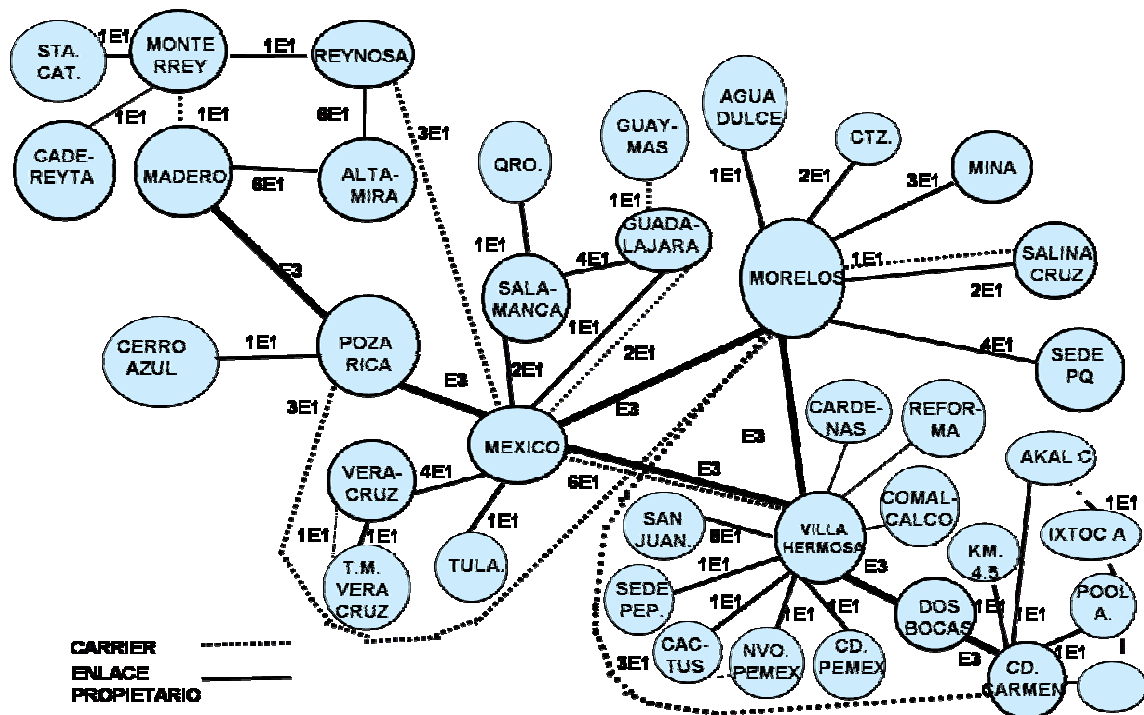


Figura 4.1 Caso de estudio: Red corporativa

De manera particular y a detalle se muestra en la Figura 4.2 cómo esta integrado el nodo Veracruz.

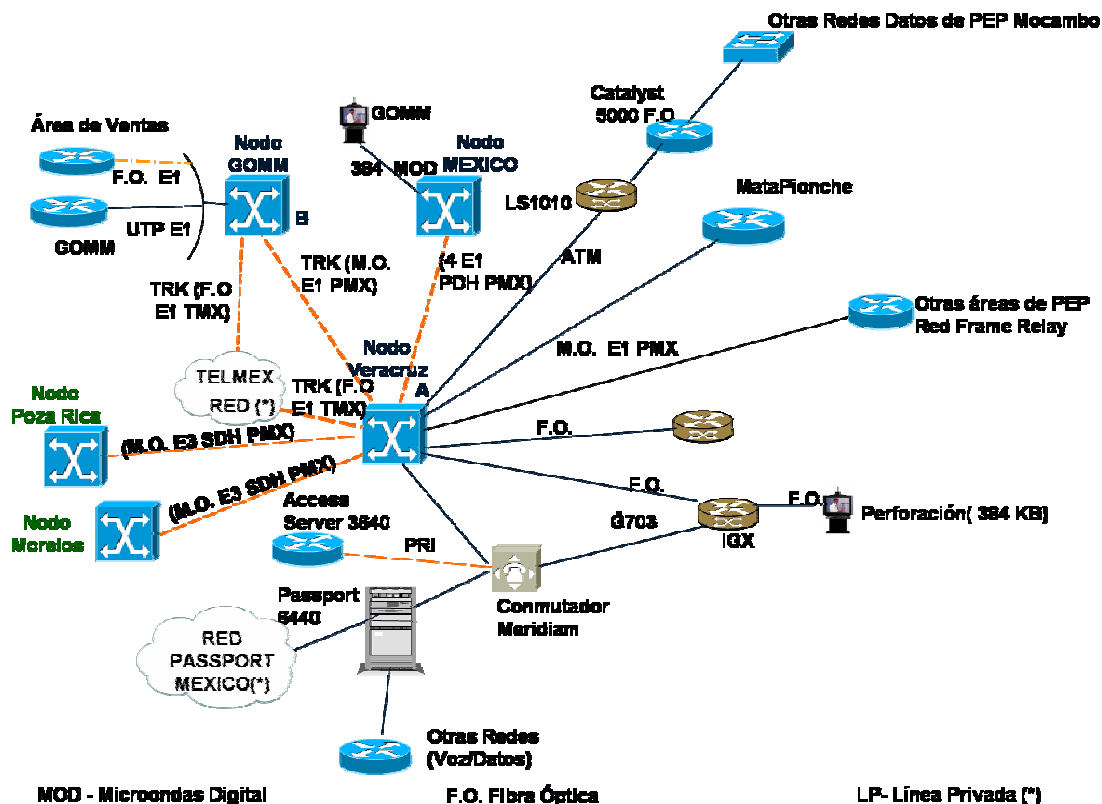


Figura 4.2 Nodo de red corporativa Zona Veracruz

4.1.1. Especificación de la situación actual de la Red Corporativa

A través del nodo Veracruz se le da Salida / Entrada a las diferentes redes locales que se encuentran integradas a través de enlaces dedicados utilizando diferentes medios de comunicación (Micro ondas digital y Líneas Privadas E1) para soportar las tecnologías de red: Frame Relay y ATM.

A través de este nodo se brinda los servicios de voz, datos y video a las diversas áreas que lo integran. Cuenta con 6 enlaces dedicados con diferente capacidad hacia diferentes puntos ubicados en un radio de 20 Km aproximadamente (en la Tabla 1 se muestra los tipos de enlace y sus destinos).

Tabla 4.1 Relación de enlaces entre las diversas áreas.

Enlace	Tipo	Destino
4 E1	Microondas	D.F
3 E1	Microondas	Poza Rica
3 E1	Microondas	Morelos
1 E1 (*)	Microondas	Torre Malecón
1 E1 (*)	F.O	Torre Malecón
1 E1	Microondas	Matapionche

Por cada uno de estos enlaces se cruza una gran cantidad de tráfico, sobresaliendo el enlace con destino hacia el D.F. Para el caso en donde se realizaron las pruebas (Veracruz GOM- Veracruz Mocambo) se presentaba una saturación del ancho de banda de la red WAN en un 150% cómo se puede apreciar en la Figura 4.3.

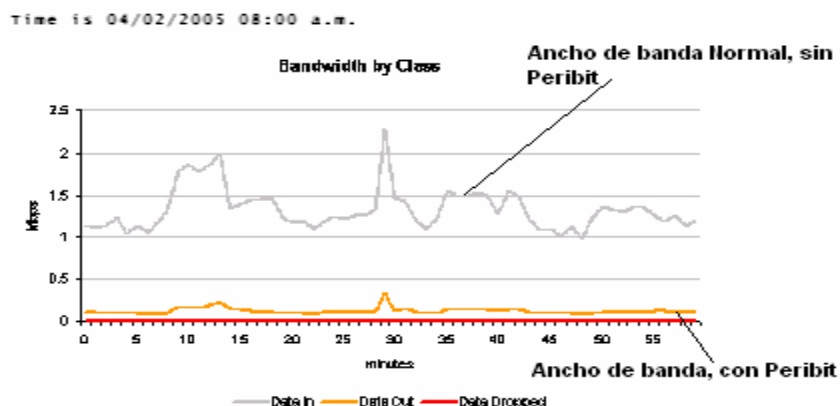


Figura 4.3 Ancho de Banda ocupado entre Veracruz GOM hacia Veracruz Mocambo

Todo este tráfico de red que se genera por las diferentes áreas a las que se les brinda el servicio se tiene integrado y preparado el nodo Veracruz con un router Cisco modelo 7000 el cual tiene las características de alta confiabilidad, disponibilidad, utilidad y funcionamiento¹⁷.

¹⁷ El router cisco 7000 soporta multiprotocolos, enrutamiento multimedia y brinda un puente sobre una variedad amplia de protocolos y cualquier combinación de Ethernet, Fast Ethernet, Token Ring, FDDI, serial, multicanal y una interfase serial de alta velocidad (HSSI). Para mayor detalle ver la dirección www.cisco.com.

El área de Tecnología de Información carece de la tecnología necesaria para hacer la optimización del uso del ancho de banda de los enlaces de red WAN disponibles, adoptando casi siempre la opción más fácil que es la contratación de más enlaces para incrementar el ancho de banda.

Aquí es donde uno de los algoritmos analizados en el capítulo 3 tiene una aplicación real y práctica al ser implementados en una tecnología existente en el mercado para la solución de esta problemática: Reducción Secuencial Molecular (MSR).

4.2 Preparación del equipo de medición para pruebas

El equipo de medición funciona en la capa (tres) IP del modelo de referencia OSI¹⁸ a través de los enlaces de área amplia. Funcionando en esta capa IP, el equipo es totalmente transparente a la red, aplicaciones, servidores, y clientes.

El esquema de configuración lógica que se empleó para llevar a cabo las pruebas se muestra en la Figura 4.4. En el Anexo 4.1 se muestran diferentes opciones de conexión y configuración que pueden ser implementadas con este equipo, todo dependerá del ambiente y equipamiento con que cuente cada empresa corporativa.

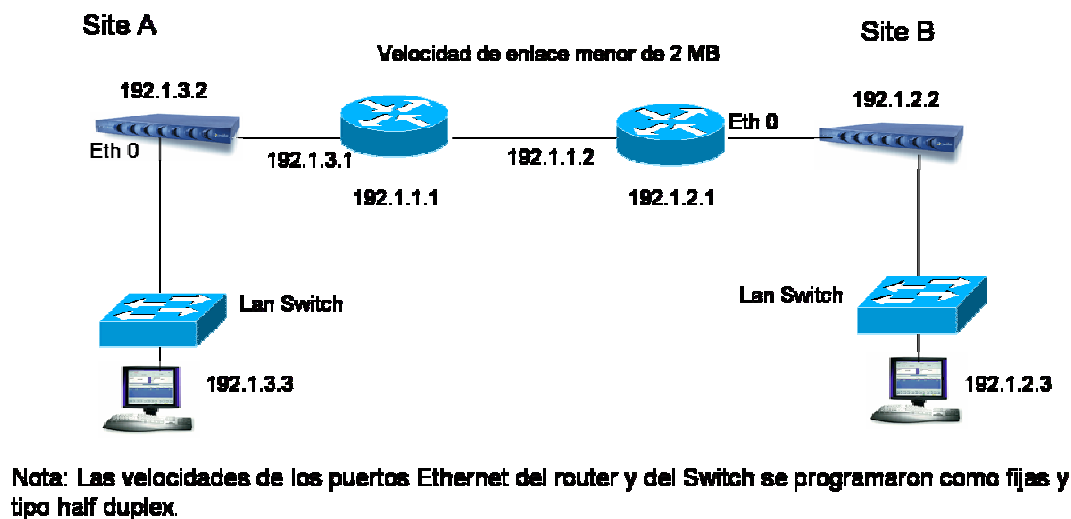


Figura 4.4 Esquema lógico de ambiente de pruebas

¹⁸ Organización Internacional de Estándares (OSI)

Para llevar a cabo la conexión mostrada en la Figura 4.4. se realizaron las siguientes actividades previas:

- Selección de dos nodos ó sitios de la red donde se instalaron los equipos empleados para la prueba (en este caso el punto “A” y el punto “B” señalados en la Figura 4.2).
- Asignación de una dirección IP y la dirección del Default Gateway correspondiente al segmento de red donde se instaló el equipo por sitio.
- Un Cable de Red uno a uno por sitio.
- Alimentación de 110 Volts.
- Espacio de 3 u de Rack por sitio.
- Conexión de una PC con acceso a la red local por sitio, para monitorear el equipo en el momento de la instalación.
- Disposición de una ventana de tiempo de 2 minutos para realizar la desconexión del cable que conecta al router de WAN con el Lan Switch para integrar el equipo a la Red. Esta ventana de tiempo es por sitio. (Nota: El tiempo que tomó desconectar el cable mencionado y conectar el equipo fue de menos de 30 segundos, los 2 minutos fueron solo por seguridad.)

El equipo de medición se interconecta a ambos extremos de las redes “A” y “B” como si fuera un LAN Switch transparente en el puerto 2, situado entre el switch y el router de la WAN. Así el equipo de medición es una tecnología basada en LAN que soporta interfaces Ethernet de BaseT de 10/100/1000 que procesa datos antes de que ellos sean transmitidos sobre los enlaces WAN. Por lo tanto el enlace WAN puede basarse en cualquier medio físico de transmisión, en este caso fue sobre un E1.

La tecnología MSR automáticamente descubre a otros equipos MSR en la red y elimina redundancias de tráfico que son destinados a las subredes que soportan a esos dispositivos remotos.

4.3 Tecnología de Optimización

4.3.1 Introducción

El área de Tecnología de Información debe comprender que la optimización del funcionamiento en el uso sobre la red WAN es un problema que tiene muchas caras que requiere más que solo aplicar

técnicas de compresión de datos. La solución metodológica y tecnológica propuesta en este trabajo, es una solución integral que proporciona una gama completa de elementos necesarios para la Optimización efectiva de la red WAN.

La tecnología propuesta de optimización integral en su arquitectura la:

- Compresión con Reducción de la Secuencia Molecular (MSR),
- Reflejo de la Secuencia de la Red (NSM)
- Reducción de la latencia con la Aceleración del Flujo del Paquete (PFA)
- Administración del ancho de banda con QoS y asignación del ancho de banda
- Optimización de la trayectoria con Política Basada en Multi trayectorias
- Visibilidad vía Web View
- Fácil empleo y permitir las operacionales de escalabilidad
- Seguridad

4.3.2 Identificación de la Fuente de datos repetidos en la empresa corporativa analizada

Las fuentes de esta redundancia son razonablemente evidentes y se desprenden sobre todo a partir de tres fuentes:

- ✓ El lenguaje común de la organización (es decir archivos, objetos, tablas, etc. similares)
- ✓ El flujo del proceso de negocio
- ✓ La redundancia en las comunicaciones de aplicaciones internas.

Cada organización y la analizada no es la excepción, desarrolla su propio "lenguaje" de frecuencia usado en archivos, objetos, descripciones, tablas, etc. que forman la base de la mayoría de las comunicaciones internas de la compañía. Por ejemplo, muchos documentos creados nuevamente, presentaciones, u hojas de balance se basan a menudo en las secciones tomadas de los archivos previamente existentes que son entonces modificados, re/arreglados e incrementados para crear un

nuevo objeto. Este incremento crece el resultado del proceso en grandes conjuntos de datos comunes de texto, tablas, cartas, etc. que aparecen en forma similar o idéntica a través de la mayoría de los archivos en una organización. Estos objetos comunes podrán consistir en cualquier cosa desde frases cortas tales como “mensaje confidencial” hasta imágenes con el logotipo de la compañía en los documentos como la carta de presentación de la compañía. Todos estos comunicados son datos dentro de la organización, este “lenguaje” común genera un alto grado de redundancia de datos a través de la infraestructura de la red de la compañía.

Adicionalmente el flujo del proceso de negocio usado también genera redundancia de datos significativo a través de las redes requiriendo compartir información con muchos compañeros de trabajo distribuidos. Para alcanzar esta comunicación compartida, los datos se envían simultáneamente a múltiples recipientes y al mismo tiempo son accedidos por múltiples usuarios en una fuente común. Por ejemplo, los documentos son distribuidos a menudo enviándolos como accesorios a los recipientes múltiples dentro de la organización. En este caso, el servidor de correo genera múltiples copias del mensaje y los accesorios son todos transmitidos como patrones de datos repetidos a través de la WAN.

Otro ejemplo son las bases de datos centralizadas donde los empleados frecuentemente accesan la misma o información similar en estas bases de datos implicando que esta información repetida sea retransmitida sobre la red de datos. De este modo éstas y otras prácticas comunes en la comunicación generan frecuentemente transmisión de datos repetidos sobre el backbone de la red.

Las aplicaciones distribuidas requieren a menudo comunicaciones frecuentes entre el cliente y el servidor de la aplicación final (back end) para asegurar la sincronización y consistencia de los datos. Estos mensajes frecuentes de actualización utilizan típicamente formatos predefinidos y protocolos que son altamente redundantes y por lo tanto generan una gran cantidad de patrones repetidos a través de la red. Estas aplicaciones de la empresa se refieren a menudo como “habladores” debido a sus muchas transacciones pequeñas en la red que en conjunto consumen recursos significativos de la red. Además de estos protocolos “habladores”, las transferencias regulares por lote a las bases de datos a través de los servidores distribuidos o grupos de clientes y de un servidor centralizado también generan un alto grado de redundancia de tráfico en la red. El resultado es que las aplicaciones distribuidas envían con frecuencia datos repetidos a través de la red de área amplia.

4.3.3 Análisis comparativo entre tecnologías existentes

4.3.3.1 Introducción

La tecnología MSR puede ser comparada con la compresión de datos de uso general tradicional puesto que ambas tecnologías reducen datos o tamaños del objeto al quitar repeticiones y redundancias en los datos. Por lo tanto, el objetivo de ambas tecnologías es similar, es decir, encontrar y eliminar repeticiones en los datos. La eficacia de una tecnología de compresión se mide típicamente en términos de la tasa de la reducción de datos que puede alcanzar. Pero la aplicación de la compresión de datos a la infraestructura del establecimiento de una red expone varios apremios adicionales que deben también ser considerados en la evaluación de una tecnología de reducción de datos.

Estos nuevos apremios incluyen operabilidad en tiempo real, velocidad, latencia, dependencias entre paquetes codificados, transmisión de datos codificados con un medio no confiable y la operación incremental y continua.

4.3.3.2 Descripción de las diferencias encontradas de la tecnología propuesta MSR con otras

- ✓ Aplica para todo aquel tráfico que pase por dicha tecnología, con la capacidad de aprender el lenguaje utilizado en dicho tráfico.
- ✓ Baja latencia por paquete (menos de 2 milisegundos).
- ✓ Creación de una base de conocimientos de manera dinámica que permite la detección y la reducción de patrones de cualquier tamaño, separados por un número pequeño o grande de bytes y paquetes. La estructura única de la base de conocimientos del MSR proporciona muy alta escalabilidad y permite que esta tecnología funcione en tiempo real a muy alta velocidad.
- ✓ La velocidad del algoritmo de MSR es independiente de la tasa de reducción que puede alcanzar y por lo tanto al tamaño del buffer de búsqueda. Las tecnologías tradicionales de compresión se basan en buscar y sustituir patrones dentro de un buffer histórico fijo. Este buffer fijo pone límites en la distancia máxima entre los patrones (es decir, sólo los patrones que son menos de 2 kilobytes y que a parte serán comprimidos) así como a la escalabilidad y velocidad en el funcionamiento de estos esquemas tradicionales. Mientras que se aumenta el tamaño del buffer de búsqueda para identificar más patrones repetidos, el poder de procesamiento y latencia también aumentan dramáticamente. Por lo tanto las técnicas tradicionales de

compresión han hecho frente a una compensación difícil entre la latencia y las tarifas de reducción.

- ✓ El resultado de lo anterior, es que mientras la compresión está extensamente disponible en routers, esta compensación evita que la mayoría de las empresas utilicen esta característica a través de sus enlaces WAN. La mayoría de las redes no pueden afrontar la latencia adicional (a menudo 30 milisegundos o más) para alcanzar niveles de reducción del destino. Por comparación, MSR agrega típicamente menos de 2 milisegundos de latencia mientras que alcanza mucho más tasa de reducción.
- ✓ MSR funciona sobre el tráfico de red en tiempo real en la capa IP (capa 3).
- ✓ El algoritmo de descubrimiento de patrones es uno de los diferenciadores clave de esta tecnología. La velocidad y la granularidad con los cuales el algoritmo de descubrimiento del patrón busca nuevos patrones es crítico a las altas tasas de reducción de datos proporcionadas por MSR y a las altas velocidades de la red en la que esta tecnología pueda funcionar.
- ✓ La construcción adaptativa del diccionario. Cuando un nuevo patrón es descubierto en el buffer de búsqueda, se entrega al módulo de construcción del diccionario "aprende" el patrón y agregándolo al diccionario. El patrón es agregado al diccionario asignándole una etiqueta interna, que entonces es utilizada para referir únicamente a este patrón. Los procesos de aprender y olvidar de este módulo son ambos altamente eficientes y no comprometen la linealidad y la velocidad del algoritmo total. La memoria requerida para almacenar los patrones del diccionario es reducida al mínimo codificando las etiquetas y los patrones en una forma óptima compacta. Por lo tanto, como en un diccionario verdadero de la lengua, una tabla relativamente pequeña de patrones (10 Mbytes) puede codificar un número muy grande de secuencias originales de los datos (de 100 a 1000 Mbytes). La completa originalidad adaptativa de este proceso también asegura que el MSR pueda correr continuamente e incrementalmente sobre una secuencia indefinida de símbolos de entrada.
- ✓ Complejidad de los algoritmos de compresión. Para el caso de MSR es de una complejidad lineal expresada por la función $O(n)$ donde "n" es el número de símbolos de entrada. Mientras que para otras técnicas de compresión son de orden $O(n \log(B))$, donde "n" es el número de símbolos de entrada y B es el tamaño del buffer de búsqueda. Para mayor detalle, vea el Capítulo 3, apartado 3.2.4.2.6.2 de este trabajo.

- ✓ La reducción de datos lineal y la escalabilidad del proceso (ver detalle en Capítulo 3, apartado 3.2.4.2.6.2).
- ✓ Desacople del flujo de datos y la sincronización del diccionario (ver detalle en Capítulo 3, apartado 3.2.4.2.6.3).
- ✓ Incremental y funcionalidad continua (ver detalle en Capítulo 3, apartado 3.2.4.2.6.4).

4.3.3.3 Tabla resumen de MSR vs otras tecnologías

Tabla 4.2 Relación de ventajas de la tecnología MSR contra otras tecnologías

Compresión	Peribit (MSR)	Expand	Packeteer	Riverbed
Los algoritmos de compresión aplican a pequeños, medianos y grandes patrones	Si	No, solo a pequeños y medianos	No, solo a pequeños y medianos	No, solo a grandes patrones
La compresión beneficia a todos los protocolos IP	Si	Si	Si	No, solo soporta TCP
Algoritmo de compresión optimiza a baja latencia	Si, 2 ms	No, 5 a 7 ms	No, > 10 ms	No, > 15 ms
Soporta los encabezados de compresión IP	Si	Si	Si	Si
El tráfico de compresión sobre múltiples VLANs	Si, 4096 VLANs simultaneas	No	No	No
Trabaja en enlaces WAN duales	Si	No	No	No
Manejo de QoS para la asignación apropiada del ancho de banda	Si	No, tiene problemas para el manejo con Framre Relay	No, requiere de TI para controlar flujos individuales	No
Permite fijar niveles de ancho de banda (mínimos / máximos)	Si	NO	No	No
Soporta la salida comprimida en los diferentes capacidades de enlace (64 K, 128 K, 256 K, 512 K, 1 Mbps, 2 Mbps)	SI	No	No	No
Soporta todo el tráfico IP	Si	No, el tráfico UDP no puede ser comprimido	No, el tráfico UDP no puede ser comprimido	No, el tráfico UDP no puede ser comprimido
Comprime todo tráfico de cualquier aplicación	Si	No, presenta detalles con algunas aplicaciones	No, presenta detalles con algunas aplicaciones	No, tiene problemas con el Exchange 2003
Escalabilidad	Si	No	No	No
Facilidad de instalar y operar	Si	No	No	No

4.4. Selección de la Tecnología de Optimización del Ancho de Banda

Del análisis anterior de las ventajas del algoritmo MSR con respecto a otras tecnologías, demuestra que es una tecnología linealmente escalable que puede proporcionar tasas constantemente altas en la detección de la repetición y de reducción de datos mientras que se aumentan las velocidades de la red. Esto es debido al hecho de que el buffer de búsqueda de MSR para descubrir patrones repetidos puede ser aumentado arbitrariamente sin la afectación de la velocidad del algoritmo. Así, como las velocidades de la red aumentan y la distancia entre las repeticiones llega a ser más grande, el tamaño de buffer de búsqueda de MSR se puede aumentar para capturar estas repeticiones distantes sin afectar la complejidad de cómputo del algoritmo.

En contraste con MSR, las técnicas tradicionales de la compresión de datos no son escalables a las altas velocidades puesto que se limitan a un tamaño de buffer fijo. Estas técnicas tienen una relación inversa entre la velocidad del algoritmo y el tamaño de buffer, y por lo tanto no pueden aumentar el buffer de búsqueda a tasas de datos más altas del juego.

Además, estos algoritmos de compresión imponen típicamente dependencias fuertes a través de la secuencia de datos comprimida, los cuales requieren los algoritmos ser reajustados a su estado después de la entrada de cada paquete. Debido a estos factores, las técnicas tradicionales de compresión funcionan sobre cada paquete de entrada independientemente del resto de los paquetes y no pueden detectar repeticiones a través de los paquetes múltiples.

Derivado de las diferentes ventajas que presenta la Tecnología propuesta MSR con respecto a otras tecnologías, se selecciona esta para llevar a cabo las diferentes pruebas de medición en las instalaciones de la empresa corporativa propuesta, para dar cumplimiento a uno de los objetivos planteados en este trabajo

Capítulo 5.

Resultados

5.1. Análisis e interpretación de los resultados

Las pruebas realizadas se llevaron a cabo en un periodo de dos semanas en diferentes horarios y de acuerdo al diagrama de pruebas propuesto en el Capítulo 4 (Figura 4.4).

Un equipo **Peribit SR 100** se conectó al segmento principal de Veracruz-GOM (192.1.3.1) con el objetivo de observar el comportamiento de la Red con el equipo Peribit Instalado en la Red, el Segundo Equipo **Peribit SR 100** fue instalado en sus oficinas de Veracruz-Mocambo sobre el segmento (192.1.2.2).

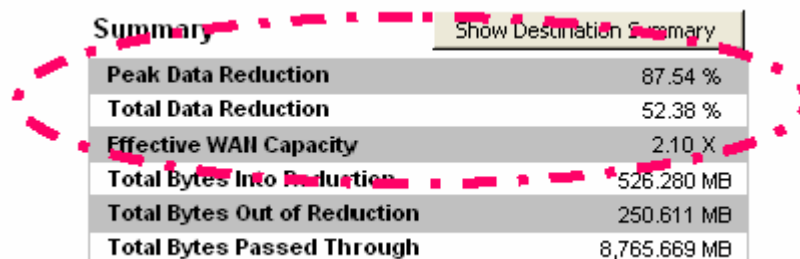
En estas pruebas los equipos instalados se configuraron para actuar sobre el 100% del Tráfico IP que se transmite del Veracruz-GOM y Veracruz-Mocambo y viceversa.

Los equipos colectan estadísticas acerca del performance de la Red y se almacenan en el mismo equipo, estas estadísticas pueden ser observadas en tiempo real desde el menú de monitoreo del equipo.

5.1.1. Detalles en la Reducción de Datos.

Comportamiento del Tráfico comprendiendo el periodo del 4 al 11 de febrero del 2005.

La siguiente Figura 5.1 muestra la cantidad total de datos que pasa por el equipo Peribit que van de **Veracruz - Mocambo hacia Veracruz - GOM** desde el lado de LAN y la cantidad total de datos que salen del equipo Peribit hacia la parte WAN y la diferencia de los datos de entrada y los de salida es la métrica del tráfico que Peribit puede reducir.



Summary		Show Destination Summary
Peak Data Reduction	87.54 %	
Total Data Reduction	52.38 %	
Effective WAN Capacity	2.10 X	
Total Bytes Into Reduction	526.280 MB	
Total Bytes Out of Reduction	250.611 MB	
Total Bytes Passed Through	8,765.669 MB	

Figura 5.1 Datos obtenidos de Veracruz-Mocambo

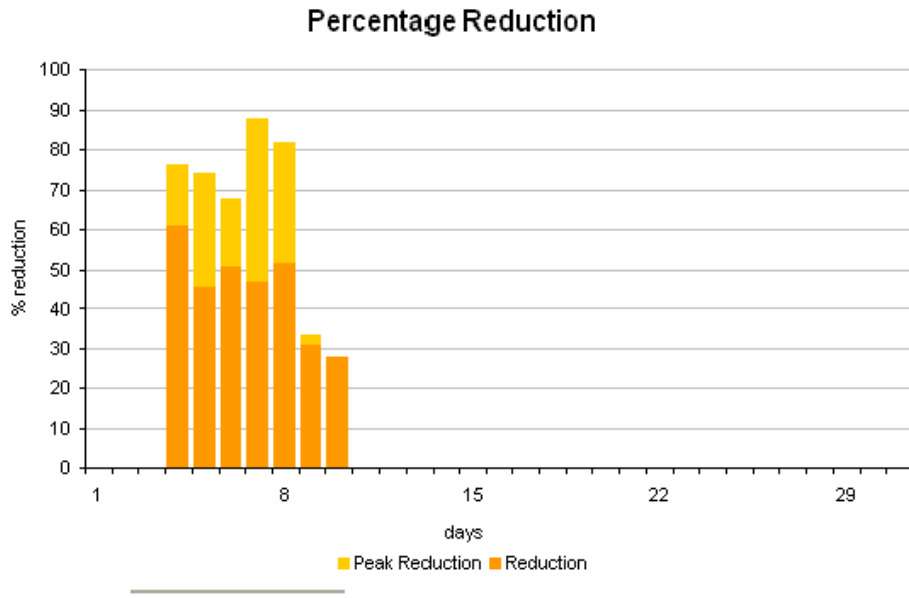


Figura 5.2 Porcentaje de reducción de Veracruz-Mocambo

En la Figura 5.2 anterior podemos observar reducción máxima del 87.54% y una reducción promedio del 52.38 % logrando **MULTIPLICAR EN 2.10 veces el throughput de la red en el sentido de Veracruz - Mocambo hacia Veracruz - GOM.**

En la siguiente Figura 5.3 muestra la cantidad total de datos que pasa por el equipo Peribit **instalado en Veracruz - GOM** desde el lado de LAN y la cantidad total de datos que salen del equipo Peribit hacia la parte WAN y la diferencia de los datos de entrada y los de salida es la métrica del trafico que Peribit puede reducir.

Summary		Show Destination Summary
Peak Data Reduction	94.09 %	
Total Data Reduction	81.11 %	
Effective WAN Capacity	5.29 X	
Total Bytes Into Reduction	127.041 MB	
Total Bytes Out of Reduction	23.997 MB	
Total Bytes Passed Through	66.573 MB	

Figura 5.3 Datos obtenidos de Veracruz - GOM

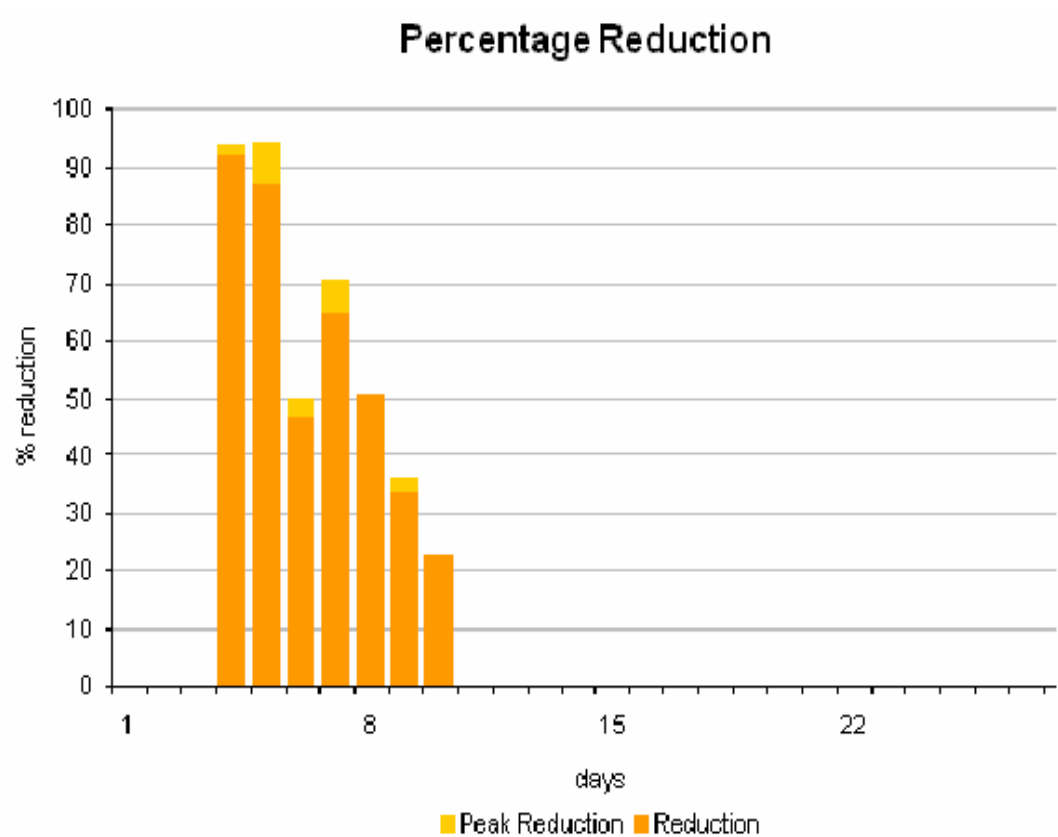


Figura 5.4 Porcentaje de reducción de Veracruz – GOM

De la grafica anterior podemos observar reducción máxima del **94.09%** y una reducción promedio del 81.11 % logrando **MULTIPLICAR EN MAS DE 5 VECES EL THROUGHPUT DE LA RED.**

5.1.2 Reporte de Tráfico por Aplicaciones

En la siguiente Figura 5.5 podemos observar el tráfico en la red por aplicaciones medido durante el periodo completo de la prueba que fue del 4 al 11 de febrero del 2005.

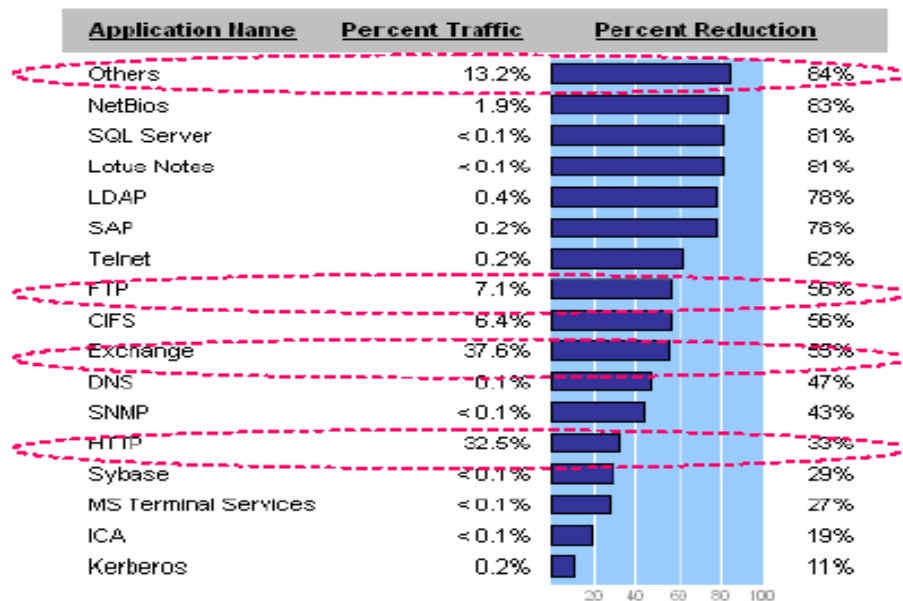


Figura 5.5 Tráfico por aplicación del lado de Veracruz - GOM

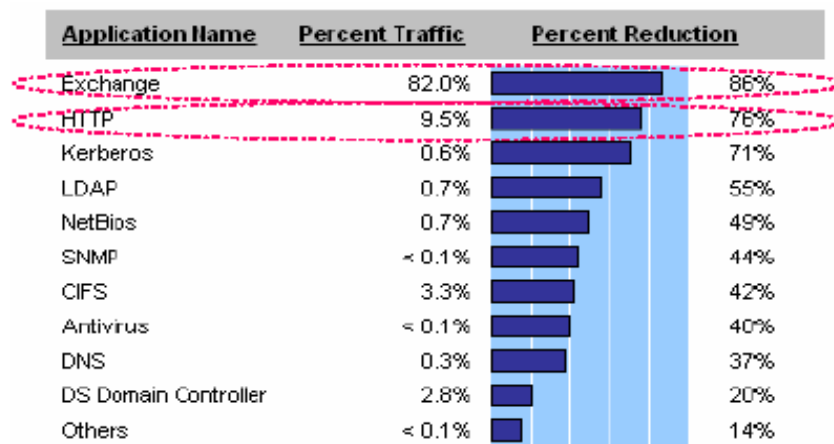


Figura 5.6 Tráfico por aplicación del lado de Veracruz - Mocambo

Se puede determinar que sobre las aplicaciones de Exchange y HTTP (Que corresponde) que ocupan el mayor porcentaje de Ancho de Banda existe un alto índice de efectividad reduciéndolo por arriba del 70% y este efecto esta en función también de la cantidad de trafico que circula por la Red.

5.2 Análisis de Resultados de la reducción por Aplicación

Las tasas reales de la reducción difieren por aplicación porque hay una amplia variación en la cantidad de datos repetidos en cualquier aplicación dada, así como una diferencia en cómo estas aplicaciones se comunican sobre la red de datos analizada. Algunas aplicaciones tienden proporcionar constantemente una reducción mejor que otras, mientras que en otras aplicaciones la reducción puede variar sobre el tiempo. Puesto que en la empresa corporativa analizada tienen aplicaciones múltiples utilizando la WAN simultáneamente, hacer un promedio de la reducción a través de estas diversas aplicaciones proporciona generalmente un nivel constante y significativo de la reducción de datos.

Por ejemplo, la aplicación de Call Center basada en WEB transmitió con frecuencia la misma forma y encabezados para cada pantalla. Estos datos son altamente redundantes y reducibles y alcanzaron tasas de reducción sobre el 85%, correspondiendo a un aumento de la capacidad de cerca de 7 veces. Por el contrario, una aplicación de respaldo que intentó solo enviar cambios en los datos tuvo menos reducción del contenido pero utilizó protocolos similares en el overhead y frecuentemente envió información de bloque del disco repetida. En estas aplicaciones se obtuvieron tasas de reducción del 60% al 75% correspondiente a incrementar la capacidad de 2.5 a 4 veces el ancho de banda de un enlace.

Las aplicaciones de base de datos se obtuvieron resultados relativamente altos de reducción. Esto es debido a varios factores importantes. Primero, muchos de los protocolos de la aplicación de base de datos definen formatos y campos de datos que son altamente repetidos y por lo tanto reducibles. En segundo lugar, los usuarios tienen acceso con frecuencia a la misma información, la cual genera múltiples patrones de datos repetidos a través de la red. En combinación con estos factores, se contribuyen a que el algoritmo MSR proporcione tasas excelentes de reducción para los tipos del tráfico de base de datos tales como Oracle, Informix y SQL.

Un diferente ejemplo es NetBIOS, un protocolo común utilizado por las varias aplicaciones de Microsoft que llevan a menudo tráfico al servidor de archivos. Este tráfico es a menudo altamente transaccional generando muchos mensajes de actualización del estado que son altamente redundantes y por lo tanto reducibles. Igual aplica al tráfico NFS dado la naturaleza similar de los datos.

En las aplicaciones ERP demuestran niveles de reducción similares a las aplicaciones de base de datos enumeradas anteriormente. El overhead de las comunicaciones internas de estas aplicaciones hacen estos datos

altamente reducibles. Además, los usuarios frecuentemente acceden a la misma o a elementos de datos similares que agregan repeticiones enviadas a través de la red. Se encontró niveles de reducción para SAP en un cluster relativamente apretado con tasas promedio de reducción alrededor del 75% y tasas de reducción de máximos alrededor del 90%.

El tráfico de la emulación terminal como Telnet y X Windows es también intrínsecamente muy reducible. Sin embargo, muchos desplegados de Telnet permiten la opción de la compresión o del cifrado de la aplicación haciendo el tráfico resultante reducible por aproximadamente 35% de tal modo que trae por debajo del promedio de la figura. En algunos desplegados donde el total del tráfico Telnet es una pequeña porción de tráfico total, MSR hará con eficacia el sitio para el tráfico de ICA reduciendo el otro tráfico. En otros casos en donde el tráfico de Telnet es la mayoría del tráfico de la red, el administrador debe de inhabilitar la compresión de la aplicación para alcanzar una tasa total mucho más alta de la reducción y por lo tanto recibir mayor rendimiento de procesamiento.

El tráfico del HTTP dió lugar a resultados altos y bajos en la reducción, dependió de si el tráfico WEB estaba basado en Intranet o en Internet. Se encontró que las tasas de reducción para un tráfico en Internet son relativamente bajas debido a la alta variabilidad en el tipo de datos accedidos por diversos empleados. En este panorama de uso, los resultados para un ambiente WEB en Internet fueron de entre 40-50%. Por otra parte para las aplicaciones WEB en ambiente de Intranet, por ejemplo un Front End en ambiente WEB para aplicaciones CRM, se determinó que el tráfico es en gran parte repetido y por lo tanto los resultados de la reducción son mucho más altos. En este caso, la reducción de datos se encontró sobre un 70%.

Finalmente, hay una variedad de otras aplicaciones pesadas de transferencia de datos como FTP que proporcionen un amplio rango de niveles de reducción. La variabilidad en este caso es debido a la naturaleza desconocida de los archivos o de los datos que se están transfiriendo. Puesto que el FTP se puede utilizar para transferir diferentes tipos de archivo con diversos grados de redundancia (e.g., documentos, presentaciones, texto plano, árboles fuente, formatos de diagramas, imágenes, video, datos cifrados, etc), los resultados de reducción cambian dependiendo del tipo de datos que se estén transfiriendo. Por lo que, las tasas de reducción de la aplicación varían dependiendo del tipo de corporativo que se este analizando.

5.3 Análisis de las Tasas de Reducción por Paquete

Además de reducir el dato agregado transmitido, MSR también disminuyó el número de los paquetes enviados sobre la red hasta un 98%. MSR alcanzó esta reducción de paquetes permitiendo que paquetes múltiples reducidos fueran encapsulados dentro de un solo paquete de salida que es entonces transmitido al dispositivo remoto SR a través de la WAN. La Secuencia de Reducción remota descapsuló este mega-paquete y descifró todas las etiquetas MSR para reconstruir los paquetes originales múltiples de entrada.

Esta reducción dramática en paquetes redujo ciclos de proceso en los routers y acortó el tiempo de tránsito a través de la red. Redujo procesamiento en el overhead en el router permitiendo que el corporativo analizado continué usando su infraestructura existente de routers, evitando así routers innecesarios y/o actualizaciones del procesador. Por lo tanto, la tecnología utilizando MSR permite a las empresas manejar más sus costos, ahorrando al maximizar la eficiencia y utilización de su infraestructura existente de ruteo.

La reducción real de la cuenta de paquetes dependió de la tasa total de la reducción en bytes y de la velocidad del enlace de la red. Cuando mayor es el nivel de reducción total, mayor es la capacidad de la Reducción de Secuencia para caber más de un paquete reducido de entrada dentro de una simple salida de un mega paquete. Al mismo tiempo, esta repaquetización se termina dentro de una ventana limitada y corta de tiempo (generalmente debajo de 2 ms) para reducir al mínimo la latencia del procesamiento total a través del dispositivo. Por lo tanto, el segundo factor en la determinación del índice en la reducción del paquete es la velocidad del enlace de la red de área amplia. Cuánto más alto es la velocidad del enlace de la red, más alta es la tasa de reducción del paquete porque la Secuencia de Reducción verá más paquetes de llegada en una cantidad de tiempo dado y por lo tanto monta más paquetes reducidos dentro de una simple salida meta paquete.

5.2 Discusión

5.2.1 Ventajas de la propuesta tecnológica

El análisis del algoritmo MSR demuestra que es una tecnología linealmente escalable que puede proporcionar tasas constantemente altas en la detección de la repetición y de reducción de datos mientras que se aumentan las velocidades de la red. Esto es debido al hecho de que el buffer de búsqueda de MSR para descubrir patrones repetidos puede ser aumentado arbitrariamente sin la afectación de la velocidad del algoritmo.

Así, como las velocidades de la red aumentan y la distancia entre las repeticiones llega a ser más grande, el tamaño de buffer de búsqueda de MSR se puede aumentar para capturar estas repeticiones distantes sin afectar la complejidad de cómputo del algoritmo. En contraste con el MSR, las técnicas tradicionales de la compresión de datos no son escalables a las altas velocidades puesto que se limitan a un tamaño de buffer fijo.

Estas técnicas tienen una relación inversa entre la velocidad del algoritmo y el tamaño de buffer, y por lo tanto no pueden aumentar el buffer de búsqueda a tasas de datos más altas del juego. Además, estos algoritmos de compresión imponen típicamente dependencias fuertes a través de la secuencia de datos comprimida, los cuales requieren los algoritmos ser reajustados a su estado después de la entrada de cada paquete. Debido a estos factores, las técnicas tradicionales de compresión funcionan sobre cada paquete de entrada independientemente del resto de los paquetes y no pueden detectar repeticiones a través de los paquetes múltiples.

De los resultados obtenidos de este análisis se demuestra que el MSR puede alcanzar constantemente un alto índice de detección de la repetición del 97% a todas las velocidades aumentando el tamaño de buffer de búsqueda para satisfacer las tasas de datos más altas. En contraste, la tasa de detección de la repetición para la compresión tradicional cae a partir del 54% en 56Kbps hasta el 2% en 622 Mbps. Esta gota dramática en la detección de la repetición es debido al hecho de que el buffer de la búsqueda para la compresión tradicional está limitado a un tamaño constante y no se puede aumentar a velocidades más altas.

El algoritmo MSR desacopla el flujo de datos del proceso de sincronización del diccionario y no impone dependencias a través de los paquetes codificados y encapsulados de salida. Aunque el algoritmo proporciona la reducción de datos significativa a través de todos los paquetes de entrada, cada paquete de salida se puede transmitir independientemente al dispositivo decodificador con un protocolo de transporte no-confiable estándar. MSR es también capaz de funcionar ejecutándose incremental y continuamente en una cadena de datos de entrada indefinida.

Los recursos del diccionario se reutilizan continuamente y eficientemente para asegurarse de que sigue estando el algoritmo siempre dentro de límites fijos de la memoria. MSR adaptativamente aprende y olvida patrones para mantener un conjunto máximo relevante de patrones en su diccionario, y sin comprometer la complejidad lineal total del algoritmo.

5.2.2. Factibilidad

5.2.2.1 Factibilidad Técnica

La tecnología de optimización integrada en la arquitectura propuesta en este trabajo incluyen la compresión con Reducción de la Secuencia Molecular (MSR), la secuencia cache con Reflejo de la Secuencia de la Red (NSM), la reducción de la latencia con la Aceleración del Flujo del Paquete (PFA), la administración del ancho de banda con QoS y asignación del ancho de banda, la optimización de la trayectoria con Política Basada Multitrayectorias y la visibilidad con Web View.

Las características de empleo fácil y operacionales del escalabilidad incluyen el protocolo de ruteo “snooping” para aprender de la topología, comunicaciones continuas entre los dispositivos de la tecnología propuesta para dinámicamente actualizar la red completa, y la flexibilidad de desplegado para implementar dichos dispositivos sin cambiar la red existente.

Actualmente TI tiene que trabajar con estas características de manera integral para tener éxito en proveer calidad en las aplicaciones de la LAN / WAN permitiendo la operación y distribución a las empresas que la utilicen

Como resultado de esta integración, la tecnología seleccionada entrega capacidades a la industria que conducen en:

- Incremento en el Funcionamiento de la aplicación
- Soporte en QoS y configuración
- Flexibilidad en el desplegado
- Facilidad de uso
- Soporte multitrayectorias
- Monitoreo
- Escalabilidad
- Seguridad

5.2.2.1 Factibilidad Económica

Los ahorros del ancho de banda, mejorar la satisfacción del usuario, el aumento de la productividad del empleado y disminuir los costos del personal, son las razones primarias para que las compañías puedan elegir la tecnología propuesta en este trabajo (MSR).

Las empresas que experimenten con esta tecnología (MSR) tienen una vuelta en su inversión a partir de tres a 10 meses, con una relación directa entre el costo del ancho de banda y la longitud del periodo de reembolso. Las compañías con un predominio de conexiones de baja velocidad, tales como Frame Relay de 64 a 128Kbps, pueden tomar hasta 10 meses para justificar una inversión con MSR. Mientras que las compañías con líneas de alta velocidad, E1 o E3, pueden alcanzar rentabilidad dentro pocos meses. Los ahorros totales del ancho de banda pueden estar en promedio en un 75%, dependiendo de las aplicaciones del cliente. Las compañías que funcionan en regiones en donde es costoso y difícil el ancho de banda pueden experimentar una vuelta en la inversión especialmente rápida.

El retorno de la inversión básica con MSR es que no se tiene que ver más al ancho de banda como un problema. Sino más bien como un costo fijo al momento de adquirir esta tecnología, contra el costo repetido de doblar la capacidad de los enlaces cada que se requiera.

5.2.2.1 Factibilidad Operativa

El valor de negocio del funcionamiento. En la mayoría de las empresas se compran nuevas aplicaciones basadas en objetivos del negocio, a menudo para mejorar procesos, mejorar la productividad, reducir los costos, o incrementar las ventas y la satisfacción de cliente. La inversión en efectivo, la gente y el tiempo es substancial. Las ventajas que resultan de las nuevas aplicaciones se miden típicamente contra los objetivos de negocio. Mucho está en juego cuando una empresa utiliza una nueva aplicación, así que la Administración hace rutinariamente diligencias para entender las ventajas del negocio así como los riesgos de la nueva aplicación.

Esta tecnología propuesta reduce grandemente la cantidad de tiempo y de ancho de banda en datos de repliegue entre los SITES. Se hizo una prueba en donde se enviaron 100 MB sobre un enlace de 384 Kbps de ancho de banda en cuatro minutos. Esto es atribuible al contenido repetido de los datos detectados y a la aceleración del flujo del paquete. Enviar 100 MB en cuatro minutos es casi diez veces el rendimiento de procesamiento posible en 384Kbps. Para alcanzar este alto rendimiento de procesamiento sin la tecnología propuesta habría requerido un enlace de 5MB.

CONSIDERACIONES

FINALES

A MANERA DE CONCLUSION SE PUEDE SEÑALAR QUE:

Del objetivo general planteado en este trabajo, podemos decir que se alcanzó exitosamente, debido a que la red corporativa analizada presentaba un problema de saturación de ancho de banda entre el enlace analizado (Veracruz - Mocambo y Veracruz – GOM) y mediante la selección/aplicación de los algoritmos de optimización por compresión propuestos en este trabajo, fue posible incrementar la capacidad del enlace y disminuir el tráfico de red.

En forma resumida podemos hacer una evaluación, tomando como referencia la tecnología seleccionada (MSR) para realizar las mediciones, con respecto a los objetivos específicos planteados:

- ✓ En promedio se obtuvo más del doble del ancho de banda disponible (la reducción del 50% es equivalente a doblar la capacidad del enlace).
- ✓ Una tasa de reducción promedio del 65% del tráfico de la red, representó un aumento de la capacidad de 2.8 veces. Durante la reducción máxima, la capacidad promedio se incrementó alcanzando 10 veces la velocidad real del enlace (reducción del 90%).
- ✓ En promedio MSR entregó una tasa de reducción del 71% durante los períodos de más alto tráfico en la red. Esto significó que los usuarios finales pueden experimentar a menudo capacidad del ancho de banda de casi 3.5 veces durante cargas máximas de la red.
- ✓ Debido a que la red de datos del corporativo analizado cuenta una mezcla única de aplicaciones y diferentes procesos de negocio, los resultados en la reducción varían. El tráfico de ciertos datos es altamente reducible y por lo tanto generó resultados muy dramáticos. Las aplicaciones de base de datos son altamente reducibles, con tiempos de más del 70%. Por otra parte, el cifrado de aplicaciones, hizo que la tasa de reducción sea relativamente baja porque el tráfico cifrado debe ser totalmente aleatorio y dicho tráfico no es muy reducible.

Resultados de la metodología propuesta

- ✓ Con la metodología propuesta nos permitió hacer una investigación a detalle sobre el análisis de algoritmos para la optimización del ancho de banda por compresión permitiéndonos establecer los criterios de selección de los mejores algoritmos existentes en la actualidad.
- ✓ La eficacia de una tecnología de compresión se mide típicamente en términos de la tasa de la reducción de datos que puede alcanzar. Pero la aplicación de la compresión de datos a la infraestructura del establecimiento de una red expone varios apremios adicionales que deben también ser considerados en la evaluación de una tecnología de reducción de datos. Estos nuevos apremios incluyen operabilidad en tiempo real, velocidad, latencia, dependencias entre paquetes codificados, transmisión de datos codificados con un medio no confiable y la operación incremental y continua.
- ✓ El factor clave que determina la velocidad de una técnica de reducción de datos es su complejidad, que se define en los términos del número de operaciones de CPU requeridas para procesar una secuencia de entrada de n símbolos. La complejidad de uno de los algoritmos analizados (MSR) es $O(n)$ o de "Orden (n)". Esto nos indicó que el número de operaciones requeridas para procesar " n " bytes de entrada es proporcional a " n ". MSR es por lo tanto un algoritmo lineal puesto que el número de operaciones de CPU requeridas para procesar la entrada crece linealmente con el número de símbolos de entrada. Así, si la tasa de transmisión de datos en un enlace de red fuera doblada, MSR necesitaría realizar dos veces el número de operaciones en el mismo tiempo para procesar todos los datos sobre el enlace. El algoritmo de descubrimiento del patrón de MSR es el componente clave que proporciona esta escalabilidad lineal.
- ✓ De la metodología propuesta, la selección del instrumento de medición y su puesta en práctica se obtuvieron y determinaron los resultados exactos de reducción, utilizando un modo de perfil del equipo MSR, que midió tasas reales de reducción usando un Reductor de Secuencia instalado en el puerto del switch observado. En esta configuración, el dispositivo procesó una copia de todo el tráfico destinado al enlace WAN sin realmente requerir dispositivos adicionales para ser desplegados a través de la WAN. Esto dio facilidad y ninguna intromisión para determinar tasas reales de reducción en el tráfico de red activo real de la empresa corporativa analizada.

- ✓ La tecnología MSR puede reducir este tráfico por hasta el 90% a través de los enlaces cada vez más restringidos de redes de área amplia identificando y eliminando las redundancias numerosas que existen en estos datos.

Aportaciones de este estudio

- ✓ Debido al continuo desarrollo tecnológico de nuevas aplicaciones, el uso de Internet / Intranet, voz sobre IP, etc. en cualquier empresa hoy en día se hace más necesario el uso y la aplicación de soluciones tecnológicas que aporten un beneficio tangible al negocio. En este trabajo el mayor aporte que se realiza es la aplicación de algoritmos de optimización por compresión para la solución de una problemática real que se tiene en cualquier empresa pequeña, mediana o grande: La saturación del ancho de banda de una red WAN.
- ✓ Adicionalmente se encontró que los motivos por los cuales se genera una gran saturación de las redes se debe al tráfico redundante entre redes y la forma de solucionarlo. Los administradores de la red corporativa se sorprendieron al aprender que hasta el 90% del tráfico de la red consiste en datos repetidos. Sobre una inspección más cercana, se encontró que las fuentes de esta redundancia razonablemente evidentes vienen sobre todo a partir de tres fuentes: el lenguaje común de la organización (es decir archivos, objetos, tablas, etc. similares), el flujo del proceso de negocio, y la redundancia en las comunicaciones de aplicaciones internas.
- ✓ Se encontró el papel que juega la latencia (el asesino silencioso del funcionamiento de aplicaciones). Tradicionalmente se han centrado los esfuerzos en el diseño y planeación de una red WAN alrededor de la utilización del ancho de banda: la determinación del ancho de banda óptimo para cada enlace, clasificación de los parámetros tales como la tasa de transferencia de la información en redes WAN y la configuración de la calidad del servicio / clase de las capacidades del servicio, en el servicio del equipo o de red para preservar el ancho de banda para las aplicaciones de misión crítica. Sin embargo, la latencia desempeña un papel más grande en la determinación de funcionamiento de la aplicación, especialmente en WANs globales. Se descubrió que aunque es crítico contar con bastante ancho de banda, los administradores de red no deben de dejar de tomar en cuenta el tiempo de la latencia para resolver los porcentajes de disponibilidad que las aplicaciones y los procesos globales del negocio exigen hoy en día.

- ✓ Se propone una metodología para la optimización del ancho de banda mediante la aplicación de algoritmos previamente seleccionados y analizados probados mediante software de simulación y tecnología existente hoy en día en el mercado.
- ✓ Solución tecnológica que nos permitió establecer una metodología cómo referencia para hacer la planeación correspondiente en el crecimiento futuro de los recursos teleinformáticos de la empresa corporativa analizada.
- ✓ Evaluación de otras soluciones tecnológicas existentes en el mercado, entre ellas la que ofrece Cisco® en sus soluciones con sus equipos routers y los motivos por los cuales no se ha incorporado cómo parte de su solución de manera integral.
- ✓ Se encontró que la aplicación de la tecnología de compresión de datos esta siendo utilizada por el programa Ubiquitous Viewer de Toshiba, "el primer software en el mundo que soporta la operación remota de una computadora personal desde un teléfono móvil, que ofrece acceso a cualquier computadora basada en el sistema operativo Windows ubicada en una empresa u hogar". Con este programa, los usuarios podrán abrir suites como MS Office, además de, por ejemplo, leer y modificar archivos. Es decir, Toshiba, desarrollaron un menú de teléfonos móviles, con el objetivo de facilitar el acceso remoto y utilizar la PC; asimismo, a fin de soportar la rápida transferencia de grandes cantidades de información entre la computadora y los celulares: están utilizando la tecnología de compresión de datos.
- ✓ De acuerdo a Gartner en donde señala que la visión en los próximos años en los corporativos será la ingeniería en sistemas de negocios (Business Systems Engineering), que surge como una nueva disciplina en Tecnologías de Información y Telecomunicaciones (TIC) consideramos que dicha visión tendrá que apoyarse en el uso de la tecnología de compresión de datos.
- ✓ Cómo experiencia adquirida durante el desarrollo del presente trabajo, se propone la adopción de los siguientes criterios para la selección del proveedor de tecnología, desde la solicitud del equipo para una prueba libre, prestando la atención particular en los siguientes aspectos:
 1. La cantidad de tiempo que toma para conseguir el equipo solicitado.
 2. La cantidad de tiempo que toma para instalar el equipo.
 3. La complejidad de la instalación.

4. Cómo se aprecia el aumento de la disponibilidad del ancho de banda (rápido, lento, a mediano plazo, a largo plazo, etc.).
5. Qué visibilidad de la red o de la aplicación se gana.
6. La facilidad para fijar políticas para apalancar la información.
7. Responsabilidad del proveedor y servicio al cliente.
8. Controlar en una hoja de balance y clasificar a los proveedores en cada una de estas áreas.
9. Obtener un rápido Retorno de Inversión.

Al final, se aporta este documento al campo de la disciplina y a la academia, en formato de *e.book*, como un manual técnico que sirva de apoyo a los usuarios (empresas) para desarrollar aplicaciones de algoritmos de optimización por compresión, como una forma de incrementar la capacidad del enlace y disminuir el tráfico de red, tomando como referencia la tecnología MSR

Referencias Bibliográficas

Bernet et al., "A Framework for Integrated Services Operation over DiffServ Networks", IETF draft, May. 2000.

Cisco Systems, Cisco Enterprise Management Solutions. México: Cisco. 2003

Douglas E. Comer. Redes de Computadoras, Internet e Interredes. México: Prentice Hall; Tom Shaughnessy (2000), Manual de CISCO. México: Mc Graw Hill. 1998

D. Ferrari, "Real-Time Communication in Packet-Switching Wide-Area Networks". *Technical Report TR-89-022. International Computer Science Institute*, Berkeley, May 1989.

E. Hernández, "Transmisión de datos en tiempo real : Síntesis de protocolos y redes para transmisión en tiempo real". Memoria trabajo 2 créditos doctorado. Jul. 1998

E.W.Knightly, H. Zhang, "Traffic Characterization and switch Utilization using a Deterministic Bounding Interval Dependent Traffic Model", *In Proceedings of IEEE INFOCOM'95*

F.L. Presti, Z.L. Zhang, J. Kurose, and D. Towsley "Source Time Scale and Optimal Buffer/Bandwidth Trade-off for Heterogeneous Regulated Traffic in a Network Node" *IEEE Transactions on Networking*, Vol 7, N°4, August 1999

J. Baker, "A distributed link restoration algorithm with robust replanning", *In Proc. IEEE GLOBECOM*, pp. 306-311, 1991.

J.Beran, R. Sherman, M.S. Taqqu, and W. Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic", *IEEE Transactions on Communications*, 43(2/3/4):1566-1579, 1995.

R. Cruz, "A calculus for network delay, part II : Network analysis". *IEEE Transactions on Information Theory*, 37(1):132-141, January 1991.

The Technology Guide Serie. Testing and Analyzing Next Generation Networks. 2000

V.S. Frost and B. Melamed, "Traffic Modeleling For Telecommunications Networks", *IEEE Communications Magazine*, March 1994, pp. 70-81.

W, Stallings. Comunicaciones y Redes de Computadores. México: Prentice Hall. 1998

Sitios WEB

<http://www.cisco.com> Sitio Oficial de Cisco Systems

<http://www.pemex.com> Sitio Oficial de Petróleos Mexicanos

ANEXOS

Anexo 3.1 Desarrollo y obtención de la reserva óptima

3.1.1 Cálculo del retraso emisor - receptor

El retraso de emisor a receptor se calcula normalmente a partir de una descripción de la red y un modelo del tráfico, siendo normalmente función de la reserva realizada en los nodos. Para este fin y para un servicio garantizado se utiliza el otro modelo de tráfico token bucket¹⁹ (b,r,p) en donde el número de bits que la fuente puede transmitir es menor que b+rt para cualquier intervalo de t^{20} . El retraso en la cola se calcula por medio de las siguientes ecuaciones:

$$d = \frac{(b-M)(p-R)}{R(p-r)} + \frac{M+C_{tot}}{R} + D_{tot} \quad (p > R \geq r) \quad (1)$$

$$d = \frac{M+C_{tot}}{R} + D_{tot} \quad (R \geq p \geq r) \quad (2)$$

Si el tráfico está limitado por un flujo leaky bucket se tiene que (σ, ρ) $l(t) = \sigma + \rho t$, entonces el retraso máximo entre emisor y receptor para la sesión i es:

$$d_i = \frac{\sigma + nL_i}{R} + \sum_{j=1}^n \frac{L_{max_j}}{C_j} \quad (R \geq \rho) = \frac{\sigma + C_{tot}}{R} + D_{tot} \quad (R \geq \rho) \quad (3)$$

donde L_i es el tamaño máximo del paquete para la sesión i , L_{max_j} es el tamaño máximo del paquete en el nodo j , C_j el ancho de banda del nodo j y n el número de nodos²¹. Esta ecuación es equivalente a (1) cuando la tasa pico p tiende a infinito (los parámetros r y b son equivalentes a los σ y ρ , $L_i = M^{22}$ y $L_{max_j} = MTU_j$). El tamaño del buffer en el nodo j será $\sigma + jL_i$

¹⁹ Es otro modelo de tráfico cuyo objetivo de este algoritmo es permitir transmitir a mayores velocidades cuando la fuente recibe un pico (p); donde "b" es la cantidad máxima de tokens permitidos en el cubo y "r" es la tasa de generación de tokens. Para mayor referencia ver Capítulo II de este trabajo.

²⁰ Realmente si se establece una tasa pico p, el tráfico no puede exceder $\min[pt, b+rt]$.

²¹ Donde R = Es el ancho de banda asignado para un flujo determinado de datos en un router; C_{tot} y D_{tot} representan la sumatoria de los términos de error C y D de cada router de la ruta. Para mayor detalle ve Capítulo II de este trabajo.

²² M es el tamaño máximo del paquete; y MTU es la Unidad de Transmisión Máxima del enlace, dividido por el ancho de banda.

Como el objetivo es calcular la reserva a partir del retraso exigido, en las ecuaciones (1) y (3) se puede despejar R (ancho de banda) quedando:

$$R = \frac{(b - M)p + (M + C_{tot})(p - r)}{(d - D_{tot})(p - r) + b - M} \quad (p > R \geq r) \quad (4)$$

$$R = \frac{\sigma + C_{tot}}{d - D_{tot}} \quad (R \geq \rho) \quad (5)$$

3.1.2 Optimización de la reserva del ancho de banda

Las secuencias en un vídeo constan de n tramas, donde en cada trama i se transmiten una serie de bits E_i . La transmisión de vídeo está limitada por un conformador leaky bucket²³ donde no se puede transmitir $\sigma + \rho t$ bits para un tiempo t . Si la tasa de transmisión del cubo es ρ y el número de tramas por segundo es f entonces el número de bits que tiene el cubo en la trama i , ignorando la capacidad finita del cubo es:

$$\sigma_i = \max\{0, \sigma_{i-1} + E_i - \frac{\rho}{f}\} \quad (6)$$

Por lo tanto, el tráfico será conforme si se escoge la capacidad del cubo σ tal que $\sigma_i \leq \sigma \quad \forall i$. Se puede expresar σ como función de ρ :

$$\sigma(\rho) = \max_{0 < i \leq n} \sigma_i \quad (7)$$

La Figura 3.1.1 representa el tráfico denominado LAMBS15009. En la figura 3.1.2 se muestran los valores de σ_i en función del tiempo para una tasa ρ de 1Mb/s.

²³ Esto es también aplicable al token bucket. El emisor está limitado por la misma ecuación, la diferencia reside en que se permite una transmisión a una tasa pico ρ .

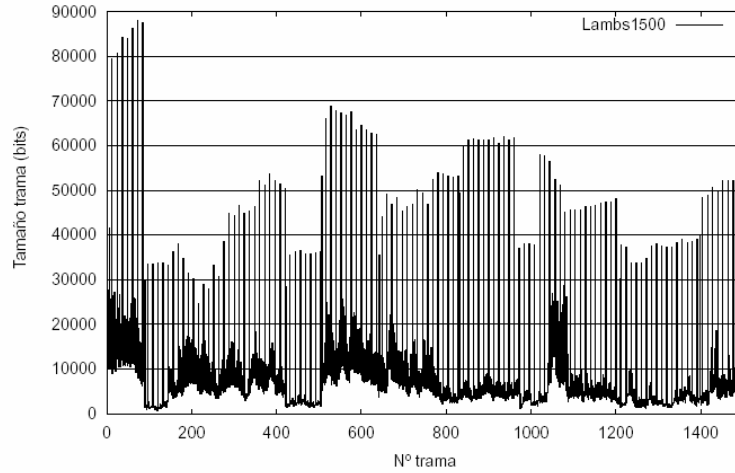


Figura 3.1.1 Ejemplo de tráfico denominado LAMBS15009

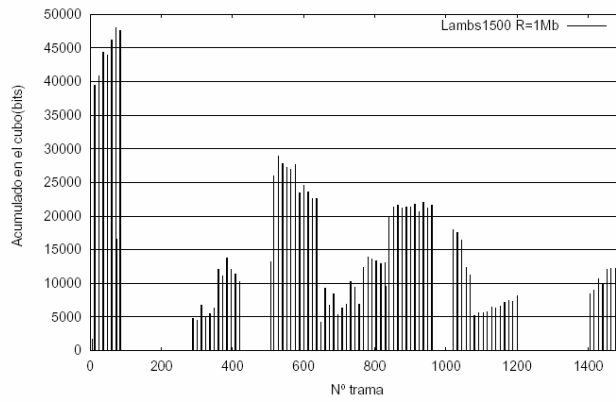


Figura 3.1.2 Valor de σ en función del No. de tramas para el tráfico LAMBS15009

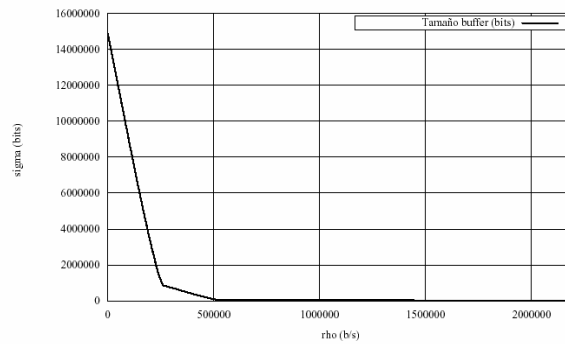


Figura 3.1.3 Función de σ (ρ): profundidad de cubo necesario en función de la tasa de drenaje (ρ)

Se comprueba que para distintos pares de valores σ , ρ se obtiene un cálculo del retraso diferente. Usando, por ejemplo, la red de la Figura 3.1.4, se tiene que los parámetros asociados a la red son $D_{tot} = 0.0045s$ y $C_{tot} = 40000b$. Usando la ecuación (1) con el tráfico LAMBS1500 para un primer par de parámetros $\rho_1 =$

240,000 b/s, $\sigma_1 = 1,464,008$ b la reserva para un retraso de 0.1s es de 2,008,639 b/s. Para el par $\rho_2 = 590,000$ b/s, $\sigma_2 = 64,392$ b la reserva para el mismo retraso es de 961,627 b/s. Se obtiene que las reservas difieren mucho.

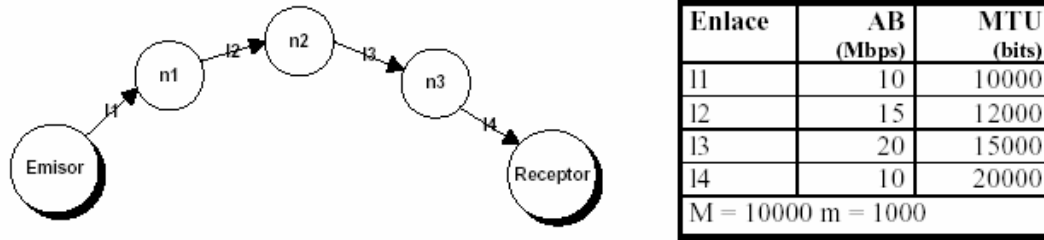


Figura 3.1.4 Red de ejemplo con tres nodos

Cómo el objetivo es optimizar la reserva R en la red, empleando las ecuaciones (4) ó (5) y utilizando un método iterativo para encontrar los valores de ρ y σ que optimicen la reserva, tenemos que el valor de ρ puede variar entre 0 y ρ . El cálculo de σ depende del valor de ρ y es una función decreciente. Aunque $\sigma(\rho)$ en principio no es continua, debido a que el cálculo se realiza usando números enteros, para poder iterar se usa la misma ecuación (7) con todos sus parámetros reales devolviendo un valor σ real.

Sustituyendo en (4) y (5) los valores de r y b por ρ y $\sigma(\rho)$ quedan las siguientes funciones que calculan la reserva R en función de ρ :

$$R(\rho) = \frac{(\sigma(\rho) - M)p + (M + C_{tot})(p - \rho)}{(d - D_{tot})(p - \rho) + \sigma(\rho) - M} \quad p > R(\rho) \geq \rho \quad (8)$$

$$R'(\rho) = \frac{\sigma(\rho) + C_{tot}}{d - D_{tot}} \quad R'(\rho) \geq \rho \quad (9)$$

En Hernández Orallo se demuestra que la reserva óptima ocurre cuando $R(\rho) = \rho$, es decir, cuando la reserva es igual a la tasa de drenaje ρ . Además, se demuestra que este valor ρ minimiza $R'(\rho)$ y que por tanto $R(\rho) = R'(\rho) = \rho$. Esto simplifica, por tanto, la búsqueda de la reserva óptima al poder utilizar la ecuación (9), que es más simple, para ambos modelos de tráfico.

3.1.3 Cálculo de los puntos envolventes

Como se intuye en el punto anterior la obtención de estos puntos permitirá calcular todas las posibles rectas $l(t)$ más ajustadas a la envolvente empírica. Para ello, en este punto se introduce el concepto de puntos envolventes. Antes de desarrollar los puntos envolventes es necesario definir el concepto de “rectas envolventes” que es una formalización del conjunto de rectas más ajustadas o próximas a la envolvente empírica.

Si una función $l(t)=\sigma + \sigma t$ describe el tráfico, cualquier función $l(t)$ (se denomina al conjunto de estas rectas $l^*(t)$) tendrá que contener $E(t)$. Esto es:

$$l^*(t) \geq E(t) \quad \forall t \geq 0 \quad (10)$$

Definición: Se llaman rectas envolventes al conjunto de rectas $l(t)$ que contiene $E(t)$ tal que no existe otra recta $l'(t)$ con la misma pendiente ρ y $l'(t) < l(t)$.

Las rectas envolventes están definiendo en origen el valor σ para la pendiente ρ . Es decir, a partir de las rectas envolventes se puede obtener el valor σ en función de ρ , como se verá en este punto y el siguiente. Para obtener estas rectas envolventes se utilizarán los puntos envolventes.

Si $E(t)$ fuese una función continua se podrían calcular las rectas tangentes para cualquier valor de t y tendríamos el conjunto de rectas $l(t)$ más próximas. Dado que $E(t)$ es creciente, las pendientes en cualquier punto de la curva serían positivas²⁴:

$$E'(t) > 0 \quad \forall t \geq 0$$

pero esto no significa que las pendientes sean siempre decrecientes. Esto implica que alguna de las rectas anteriores cortarían $E(t)$ y no cumplen la condición (10). Para que las rectas tangentes a una función creciente no corten a ésta en un intervalo definido y estén siempre por encima, está función tendría que ser cóncava por debajo. Si $E(t)$ cumpliera esta condición se obtendrían todas las rectas en función de t :

$$\rho(t) = E'(t); \quad \sigma(t) = E(t) - \rho t \Rightarrow \sigma(t) = E(t) - E'(t)t \quad (11)$$

Pero los tráficos reales son discretos, es decir, se tiene un conjunto de valores en función de unos incrementos de t , con lo que no se puede aplicar la derivada para hallar la tangente a la curva. La condición (10) se sigue cumpliendo. Por un punto t_i pueden pasar varias rectas con distintas pendientes ρ , pero ninguna de estas rectas podrá cortar a $E(t)$. Sea $l(t)$ la recta con menor pendiente que corta con un punto anterior de $E(t)$ y $l^+(t)$ la recta con mayor pendiente que corta con un punto posterior de $E(t)$. La condición para que t_i sea un punto envolvente sería que no exista otro punto anterior t_{i-1} donde la pendiente $l_{(ti-1)}$ sea menor a $l_{(ti)}$ y ningún punto posterior t_{i+1} donde la pendiente $l^+_{(ti+1)}$ sea mayor a $l^+_{(ti)}$. En la Figura 3.7a se pueden ver las dos rectas. Esto se puede formular por medio de la siguiente expresión:

²⁴ En este caso la notación $E'(t)$ representa la función derivada de E en t .

$$l(t_i) = E(t_i) \Rightarrow \begin{cases} \rho_i^- \geq \rho^- & \forall t < t_i \quad l_i(t) = E(t) \\ \rho_i^+ \leq \rho^+ & \forall t > t_i \quad l_i(t) = E(t) \end{cases} \quad (12)$$

El objetivo consiste en obtener una serie de puntos de $E(t)$ donde la pendiente de la recta que une la anterior sea siempre mayor que la pendiente de los puntos posteriores y que la pendiente de recta que une al punto posterior sea menor a las anteriores. Formalmente:

$$\forall t_i \quad \rho_{i-1}^- > \rho_i^- \quad \rho_i^+ < \rho_{i+1}^+$$

Como $\rho_{i-1}^+ = \rho_i^-$, $\rho_i^+ = \rho_{i+1}^-$, la condición anterior se puede reducir a:

$$\forall t_i \quad \rho_{i-1}^- > \rho_i^- > \rho_{i+1}^-$$

como se ve en la Figura 3.1.5b. Es decir, se obtienen las rectas que unen los puntos de forma que las pendientes son siempre decrecientes. En resumen, lo que esto implica es la eliminación de los “valles” de la envolvente empírica, quedando los “picos” (Figura 3.2 del Capítulo 3.2).

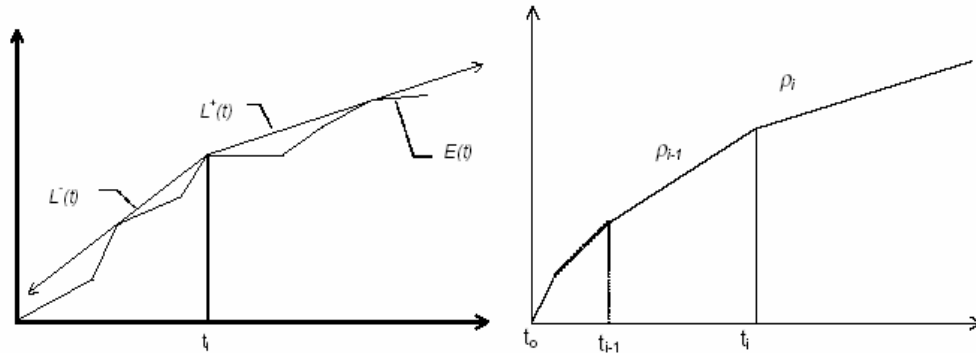


Figura 3.1.5 (a) Rectas L^- y L^+ , (b) Pendiente decreciente para t_i

Con esta condición se puede desarrollar un algoritmo para el cálculo de estos puntos el cual está detallado en la Figura 3.1.6. La complejidad de este algoritmo es de $O(n*m)$ donde n son los valores de t y m los puntos obtenidos.

OBJETIVO
Calcula los puntos envolventes.
ENTRADA :
$t_0..t_n$: Intervalo de tiempo del tráfico $E(t)$: Envolvente empírica.
SALIDA:
t_i, p_i, E_i : puntos, pendientes y valor de E en t_i m : Número de puntos generados
ALGORITMO:
$t_0 = 0, \rho_0 = \frac{E(t_1) - E(t_0)}{t_1 - t_0}$ // Primer punto $t_I = 1;$ $i = 1;$ para t entre $t_I..t_n$ $r = \frac{E(t) - E(t_{I-1})}{t - t_{I-1}}$ mientras $r \geq p_{i-1}$ // Se elimina el punto anterior $i = i - 1;$ $r = \frac{E(t) - E(t_{I-1})}{t - t_{I-1}}$ // Se vuelve a calcular la pendiente fmientras si $r < p_{i-1}$ entonces // Genera un nuevo punto $t_i = t; \rho_{i-1} = r;$ $i = i + 1;$ $E_i = E(t);$ fsi fpara $m = i;$

Figura 3.1.6 Algoritmo para calcular los puntos envolventes

El objetivo de esta sección es demostrar que con los puntos envolventes se puede obtener una función $\sigma'(r)$ equivalente a la función (7) $\sigma(r)$. Para ello, se demostrará que a partir de estos puntos se obtienen las rectas envolventes. A partir de esta demostración, se obtiene una simplificación a la hora de obtener la reserva óptima que permite una búsqueda de la solución rápida y acotada.

Para un punto i el valor de ρ está en el intervalo $[\rho_{i-1}, \rho_i]$. Por este punto pasarán el siguiente conjunto de rectas:

$$E(t_i) = \sigma + \rho t_i \quad \rho_{i-1} \geq \rho \geq \rho_i \quad (13)$$

Despejando por tanto σ y variando ρ en este intervalo se pueden calcular todas las rectas que pasan por el punto i :

$$\sigma_i(\rho) = E(t_i) - t_i \rho \quad \rho_{i-1} \geq \rho \geq \rho_i \quad (14)$$

entonces $\sigma'(r)$ es la composición de funciones lineales $\rho_i(r)$ para cada intervalo definido por los m puntos envolventes.

$$\sigma'(\rho) = \sigma_i(\rho) \quad \rho_{i-1} \geq \rho \geq \rho_i \quad i = 1..m \quad (15)$$

es decir, que para calcular $\sigma'(\rho)$ hay que buscar el intervalo donde esté comprendido ρ y aplicar entonces la ecuación (15).

Lema 3: $\sigma'(r)$ es monótona decreciente y continua en el intervalo $[0, p]$.

Demostración: ²⁵ La función (14) $\sigma_i(r)$ es monótona decreciente al depender sólo de ρ en el intervalo $[\rho_{i-1}, \rho_i]$. El valor de $E(t_0)$ es igual a cero con lo que $\sigma(\rho)=0$. Cuando ρ es cero el valor de $\sigma(0)=E(tm)$. Por definición, la recta que pasa por el punto t_i con pendiente r_{i-1} también pasa por el punto t_{i-1} y por lo tanto tiene el mismo valor en el origen σ . Por ello $\sigma_i(\rho_{i-1}) = \sigma_{i-1}(\sigma_{i-1})$. Para ρ_i es lo mismo y por tanto $\sigma_i(\rho_i) = \sigma_{i+1}(\rho_i)$.

Además, por la propia definición, $\sigma_i(\rho)$ es una función decreciente por intervalos lineales como se puede ver en la Figura 3.1.7 Se ha llegado por tanto, a un mecanismo por el cual se obtienen los segmentos comentados sobre la Figura 3.3.

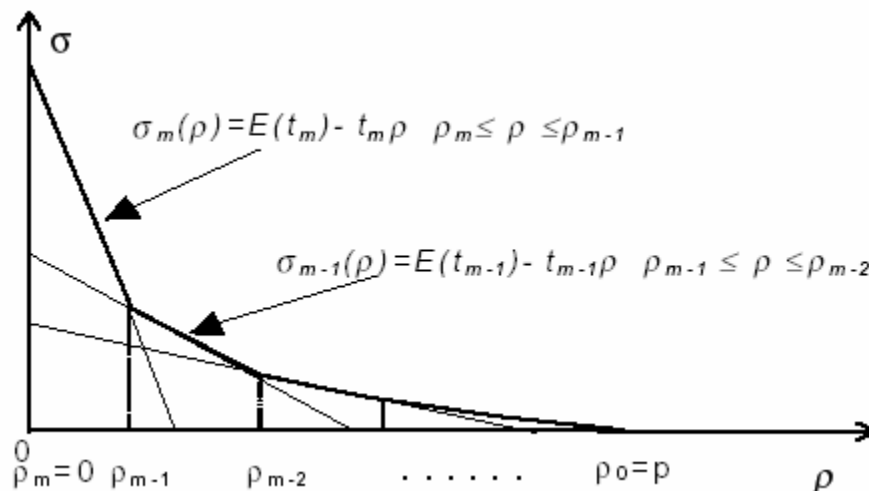


Figura 3.1.7 Función $\sigma'(\rho)$

Como se ve, el rango de $\sigma'(r)$ para el dominio $[0, \rho]$ es $[0, T]$ donde T es el total de bits transmitidos (que es igual a $E(t_n)$). Se define $L_p(t)$ como el conjunto de rectas $l(t)$ generadas para cualquier valor de ρ . Esto es:

²⁵ Hay que aclarar que los puntos que se generan tienen pendiente decreciente, es decir $\rho_{i-1} > \rho_i$. Esto significa que ρ_i es igual a la tasa pico p y $\rho_m=0$.

$$L_\rho(t) = \{ l(t) = \sigma'(\rho) + \rho t \quad 0 \leq \rho \leq p \} \quad (16)$$

Lema 4: $L_\rho(t)$ es el conjunto de rectas envolventes a la envolvente empírica $E(t)$.

Demostración: Si el rango de pendientes de $L_\rho(t)$ va de 0 a ρ , lo que hay que probar es que no existe otra recta $l'(t)$ con la misma pendiente tal que $l'(t) < l(t)$.

Por la condición (12) la recta $l(t)$ tocará la envolvente empírica en un punto t_i . Si existiera una recta $l'(t) < l(t)$ con la misma pendiente ρ , esto implicará que $\sigma' < \sigma$. Esto supone que el valor de $l'(t)$ en t_i será menor que $l(t_i)$. Dado que $l(t_i) = E(t_i)$, esto significa que $l'(t)$ corta a la envolvente empírica con lo que se incumple la condición (10).

Dado que $\sigma'(r)$ es continua y monótona decreciente (lema 3) se obtienen todas las rectas $l(t)$ para cualquier valor de ρ . Por lo que $L_\rho(t)$ es el conjunto de todas ellas.

Teorema 2: La función $\sigma'(\rho)$ es equivalente a $\sigma(\rho)$.

Demostración: Por el lema 4, $L_\rho(t)$ es el conjunto de rectas envolventes de la envolvente empírica. Esto implica, que el valor en el origen σ de estas rectas es el menor para una tasa de drenaje ρ , que es igual al valor $\sigma(\rho)$. Dado que $\sigma'(\rho)$ es continua y decreciente entonces $\sigma'(\rho) = \sigma(\rho)$ para todo ρ entre 0 y p .

Por lo tanto, dado que $\sigma'(\rho)$ devuelve los mismos valores que $\sigma(\rho)$ se puede utilizar para iterar en la ecuación (9) y encontrar raíces de la misma forma. Esto permite obtener la solución a partir de los puntos envolventes, pero se sigue utilizando un método iterativo y no acotado para encontrar la solución $R(\rho) = \rho$. Pero dado que $\sigma'(\rho)$ está formada por segmentos lineales, esto permite desarrollar un método directo para encontrar la solución que se describe a continuación:

Teorema 3: Si la raíz $R = \rho$ está en el intervalo $[\rho_{i-1}, \rho_i]$, entonces:

$$R = \frac{E(t_i) + C_{tot}}{d - D_{tot} + t_i} \quad (17)$$

Demostración: Si la raíz $R = \rho$ está en $[\rho_{i-1}, \rho_i]$, se tiene que:

$$\sigma_i(\rho) = E(t_i) - R t_i$$

y sustituyendo en la ecuación (9) y despejando R :

$$R = \frac{E(t_i) - R t_i + C_{tot}}{d - D_{tot}} \quad \Rightarrow \quad R = \frac{E(t_i) + C_{tot}}{d - D_{tot} + t_i}$$

Esto implica que si se sabe en qué intervalo está la raíz $R=\rho$ se puede obtener directamente el valor de R . La búsqueda del intervalo es fácil y acotado: se puede hacer una búsqueda binaria en los puntos envolventes hasta encontrar el intervalo. Cuando se encuentra el intervalo se aplicará el teorema 3 para obtener R . Un algoritmo para encontrar la solución se detalla en la Figura 3.1.8 El costo de encontrar la solución es muy rápido, con un costo computacional de $O(\log m)$. Como se verá en el siguiente punto, el valor de m es pequeño (entre 20 y 100 puntos para los vídeos analizados).

La reserva mínima de *buffer* en los nodos puede ser calculada usando las ecuaciones vistas en el punto 3.1.1 El algoritmo descrito optimiza la reserva de ancho de banda dando un valor mínimo de σ para un retraso especificado. Si la restricción se impone en el tamaño del *buffer*, esto implica una limitación en el valor de σ y por tanto la reserva debe ser calculada usando las ecuaciones (4) y (5) directamente.

OBJETIVO
Calcula la reserva optimizada a partir de los puntos envolventes por intervalos
ENTRADA :
Parámetros de la red C_{tot} y D_{tot} . Puntos envolventes $t_i, p_i, E_i, i = 0..n$
SALIDA:
R, ρ, σ : Parámetros de reserva y flujo.
ALGORITMO:
$i_0 = 0; i_1 = n;$ <i>// 1º Encuentra el segmento</i> hacer $i = \frac{i_0 + i_1}{2};$ $\sigma = \sigma'(\rho_i);$ $R = \frac{\sigma + C_{tot}}{D - D_{tot}}$ si $R < \rho_i$ entonces $i_0 = i;$ si $R > \rho_i$ entonces $i_1 = i;$ mientras $(i_1 - i_0) > 1;$ <i>// 2º Calcula el valor R y σ</i> $R = \frac{E_i + C_{tot}}{D - D_{tot} + t_i};$ $\sigma = E_i - R * t_i;$ $\rho = R;$

Figura 3.1.8 Algoritmo para obtener la reserva óptima

3.1.4 Algunos resultados aplicando la optimización

En este punto se muestran los resultados sobre algunas trazas considerando video tipo MPEG. Por el corolario 1 las reservas son equivalentes para las

ecuaciones del IETF/RSVP y Parekh. Por ejemplo, los puntos generados para el tráfico LAMBS1500 están en la Figura 3.1.9 Como se puede ver, una de las propiedades más interesantes de estos puntos es que es una descripción sintetizada del tráfico. Así, el primer valor de ρ es siempre la tasa pico p .

Con estos datos y usando el ejemplo puesto en el punto 3.1.2 para un retraso máximo de 0.1s, se tiene que el par de valores obtenidos σ y ρ son 50208 bits y 944590 b/s, lo que conduce a una reserva de 944590 b/s. Por el corolario 2 se puede calcular el retraso mínimo que se puede exigir que es:

$$D = \frac{C_{tot}}{p} + D_{tot} = 0.023s.$$

En la Figura 3.1.10 se muestra el cálculo de la reserva y el valor σ en función del retraso exigido desde 0s a 5s en la red de ejemplo de la Figura 3.1.4 Cuanto mayor es el retraso mayor tiene que ser el valor de σ debido a que se tiene que almacenar las tramas en la red. El incremento de este valor en función del tiempo es bastante lineal. En cambio, el valor ρ decrece exponencialmente hasta 0.25s para después decrecer linealmente.

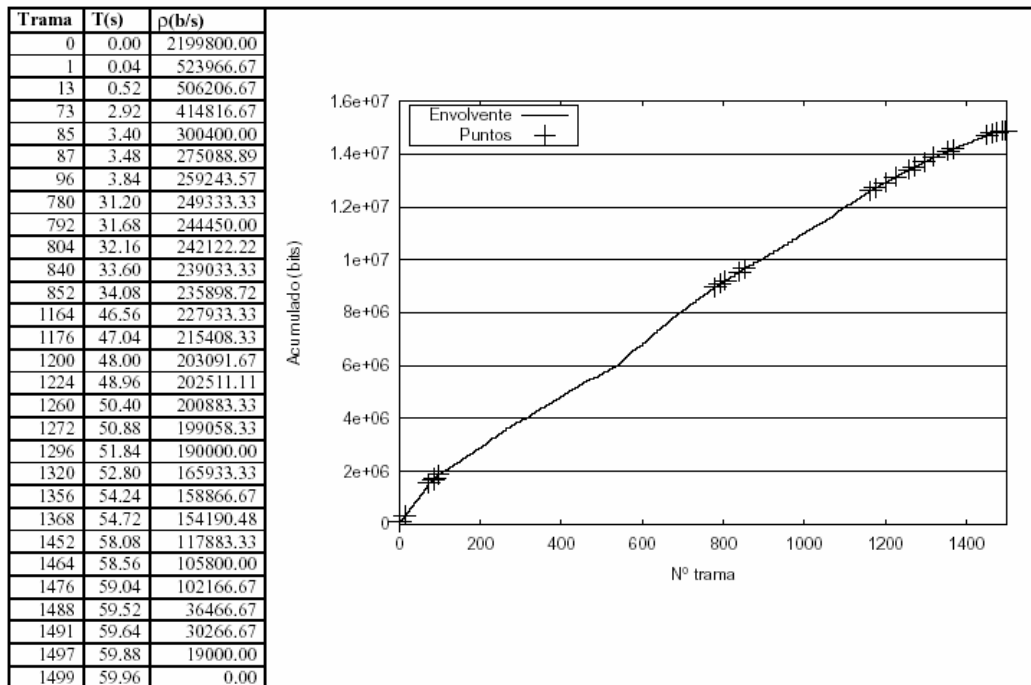


Figura 3.1.9 Puntos envolventes para el tráfico LAMBS1500

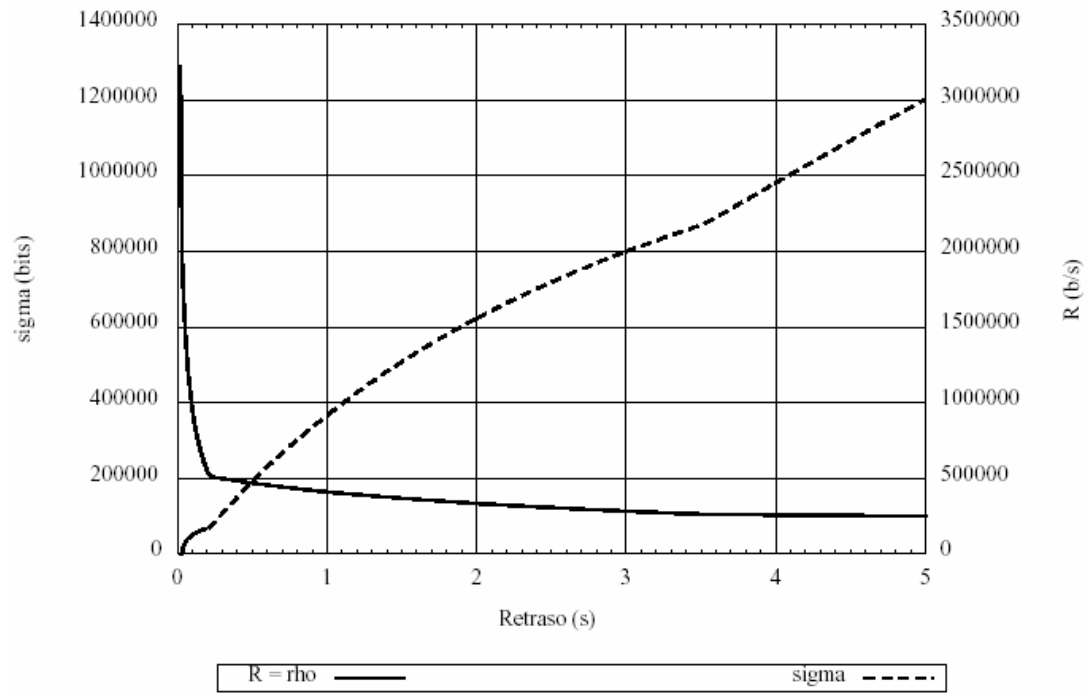


Figura 3.1.10 Retraso óptimo para el tráfico LAMBS1500

Anexo 5.1 Otros estudios realizados

Este estudio fue extraído del sitio de Internet de www.cisco.com y sirve como referencia del uso que el fabricante Cisco® esta haciendo en sus diferentes plataformas de equipos routers en lo referente al uso de la compresión de datos. Y solo sirve como referencia de los resultados obtenidos anteriormente [5].

Este estudio compara el funcionamiento del software (Cisco IOS®) basado en la compresión de datos; así como también por hardware a través de un Adaptador de Servicio por Compresión (CSA) ambas en plataformas de routers de la serie Cisco 7200 y 7500. El algoritmo de compresión usado en este estudio es el algoritmo Lossless de compresión de datos de LZS-STAC.

El Software de Cisco IOS soporta tres diversos métodos para implementar la compresión:

- ✓ Software-basado en compresión usando el procesador del router principal (Route Switch Processor) [RSP] o el Network Processing Engine [NPE]).
- ✓ Software-basado en compresión usando procesadores versátiles del interfaz (VIPs) (basado en procesador distribuido).
- ✓ Hardware-basado en compresión usando el adaptador del servicio de compresión (CSA).

5.1.1 El estudio del funcionamiento.

El estudio se hace en tres partes. Primero, el rendimiento de procesamiento de varios enlaces se mide sin ninguna compresión permitida. En segundo lugar, la compresión por software basado en Cisco IOS. Esta prueba, además de dar el funcionamiento del rendimiento de procesamiento, también proporciona el porcentaje de carga del CPU y la tasa de compresión de los datos de prueba. La última prueba realizada es la compresión basada en hardware (CSA), que también proporciona el funcionamiento del rendimiento de procesamiento, el porcentaje de carga del CPU, y la tasa de compresión.

Las pruebas se realizan para tamaños de paquetes de 103 y 551 bytes. El paquete de 103 bytes representa un paquete de tamaño pequeño, mientras que un paquete de 551 bytes representa un paquete por default del Protocolo de Transferencia de Archivos (FTP). El algoritmo de compresión usado en todos los casos es STAC-LZS Lossless que es el algoritmo de compresión de datos que se utiliza en todos los casos.

Ambos funcionamientos por software y hardware de compresión se miden para diversas velocidades del enlace: 64 Kbps y el E1, incluyendo 1.54, 2.015, 4.030, y 8.061 Mbps. Los resultados se obtuvieron en plataformas Cisco 7200 y 7500. En la plataforma del Cisco 7200, el funcionamiento de la compresión se mide contra ambos modelos de NPEs (NPE-150 y NPE-200). En la plataforma del Cisco 7500, el funcionamiento de la compresión se mide contra RSP4 y ambos VIPs (VIP2-40 y VIP2-50).

Las gráficas siguientes muestran el funcionamiento del número máximo de paquetes de datos (medidos en miles de paquetes por segundo) para ambos tamaños de paquetes de 551 y 103 bytes, para una velocidad dada del enlace (medida típicamente en Megabits por Segundo).

La tasa sin comprimir del paquete refleja el número de paquetes por segundo requerido para alcanzar una tasa lineal para varias velocidades de enlace. La tecnología de compresión reduce el tamaño del paquete, de tal modo que aumenta el número de paquetes transmitidos para la misma tasa del enlace. El efecto es virtual, al aumentar el ancho de banda total del enlace.

Por ejemplo, en la prueba 1, para una velocidad del enlace de 4Mbps, el número máximo de paquetes sin comprimir (551 bytes) que pueden ser transportados es de 0.914 kpps. Este número se traduce en 4 Mbps, que es también la tasa máxima del enlace. Usando la compresión por software, el número máximo de paquetes (551 bytes) es transportado en 1.156 Kpps. Este número se traduce en 5.095 Mbps de datos transportados en el mismo enlace físico que funciona a la misma velocidad del enlace, que es 4 Mbps. Esto representa un aumento aproximadamente de 1 Mbps del rendimiento de procesamiento de datos. Ahora, usar CSA basado en la compresión por hardware, el número máximo de paquetes (551 bytes) transportados es 2.062 kpps. Este número se traduce en 9.089 Mbps de datos transportados en el mismo enlace de 4Mbps, más de dos veces el transporte sin comprimir los datos.

5.1.1.1 CASO 1

Plataforma: VIP2-40 en Cisco 7500

Puerto Adaptador: PA-4T+ y CSA-Comp/1 (768 KB)

Cisco IOS Ver.: 11.1(14.5) CA

Tamaño del Paquete: 551 y 103 bytes

En la Figura 5.1.1 se demuestra que la compresión ayuda definitivamente a aumentar el rendimiento de procesamiento de datos, con la compresión por hardware se envían más datos sobre el enlace que por software. La compresión de paquetes de datos mejora la utilización del ancho de banda del enlace WAN disponible traducidos directamente en ahorros de costos de acceso a la WAN. En el caso de compresión por software, el funcionamiento del CPU es el cuello de botella para la compresión en enlaces de alta velocidad (véase el caso de los 8M). El hecho de que la utilización del CPU estuviese cerca de 99%, es porque el rendimiento de procesamiento por compresión por software está por debajo del rendimiento de procesamiento sin comprimir. La tasa de compresión es 2.35.

La utilización del enlace por compresión por hardware es más de dos veces que la velocidad del enlace de 8M, que es equivalente a cuatro enlaces E1. La utilización del CPU del procesador de VIP soportado por el puerto adaptador CSA es menos de 50. La tasa de compresión es 2.28.

El funcionamiento de la compresión por software depende fuertemente de la capacidad del procesador. El tamaño más pequeño del paquete pone más carga en el procesador, como es evidente en el funcionamiento reducido de la compresión para los enlaces de alta velocidad, por ejemplo para los 4M.

La compresión por hardware (CSA), se recomienda definitivamente para la compresión en enlaces de alta velocidad. Esta compresión también libera al procesador para otras funciones importantes del router tales como generar y actualizar tablas de ruteo, etc.

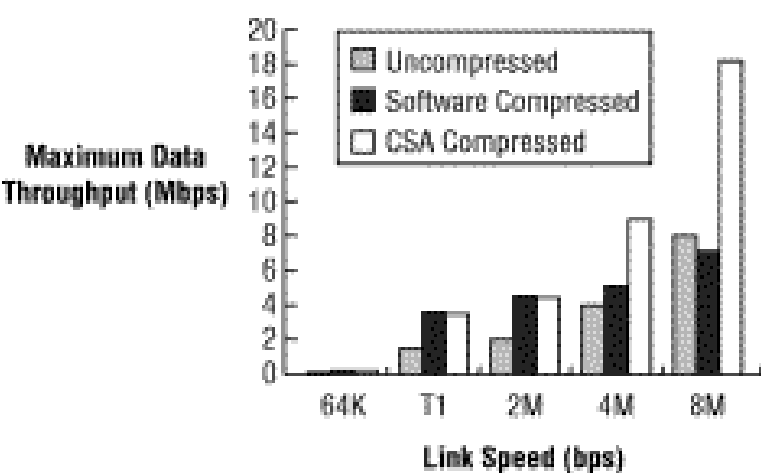


Figura 5.1.1. El Funcionamiento de la Compresión en VIP2-40 (551 bytes por paquete).

Velocidad del Enlace	64K	T1	2M	4M	8M
Sin comprimir	0.064	1.607	2.011	4.021	8.052
Compresión por Software	0.128	3.713	4.643	5.097	7.215
Compresión por Hardware (CSA)	0.128	3.603	4.507	9.093	18.070
Máximos datos procesados en Mbps para varias velocidades de enlace					

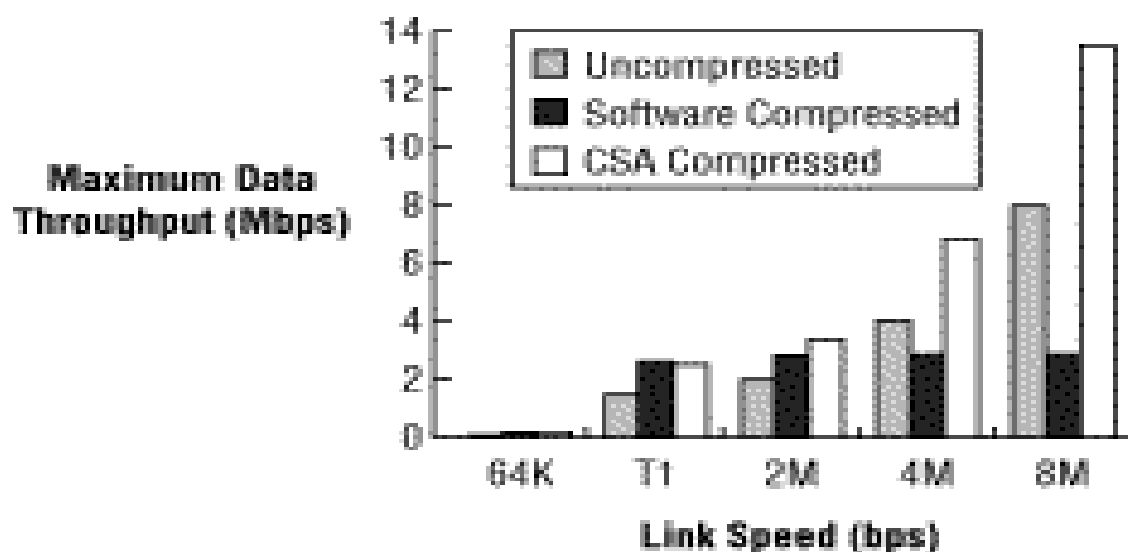


Figura 5.1.2 El Funcionamiento de la Compresión en VIP2.

Velocidad del Enlace	64k	T1	2M	4M	8M
Sin comprimir	0.063	1.604	2.002	4.011	8.025
Compresión por Software	0.106	2.801	2.903	2.914	2.923
Compresión por Hardware (CSA)	0.106	2.759	3.450	6.900	13.699
Máximos datos procesados en Mbps para varias velocidades de enlace					

5.1.1.2 CASO 2

Platforms: VIP2-50 in Cisco 7500 platform

Port Adapter: PA-4T+ and CSA-Comp/1 (768 KB)

Cisco IOS Ver.: 11.1(14.5) CA

Data Packet Size: 551 bytes and 103 bytes

El funcionamiento de la compresión por software es mejor con el VIP2-50 que con el VIP2-40 porque el VIP2-50 tiene un procesador más rápido. Esto es evidente para los enlaces de alto velocidad tales como 4 Mbps (dos enlaces E1) visto en la Figura 5.1.3. La carga del procesador se encontraba cerca de 99% para las velocidades del enlace de 4 Mbps y demás capacidad. Esto confirma de nuevo que la compresión por software es ideal para los enlaces de poca velocidad. La tasa de compresión en este caso es 2.35.

La utilización del enlace con la compresión por hardware es más de dos veces tanto como se vio en el Figura 5.1.1. El uso eficiente de enlaces WAN con compresión por hardware resulta de una orden de magnitud de utilización más alta del enlace, de tal modo que reduce los costos de acceso del enlace WAN. La utilización del CPU del VIP que soporta el adaptador en el puerto CSA fue menos del 50 por ciento. La tasa de compresión es 2.28.

El funcionamiento de la compresión por software depende fuertemente de la capacidad del procesador. El tamaño más pequeño del paquete pone más carga en el procesador, como se muestra en el funcionamiento reducido de la compresión para enlaces más altos en velocidad, como 4 Mbps y 8 Mbps por ejemplo. La utilización del procesador se encontraba cerca del 99% en ambos casos. La tasa de compresión es de 1.81.

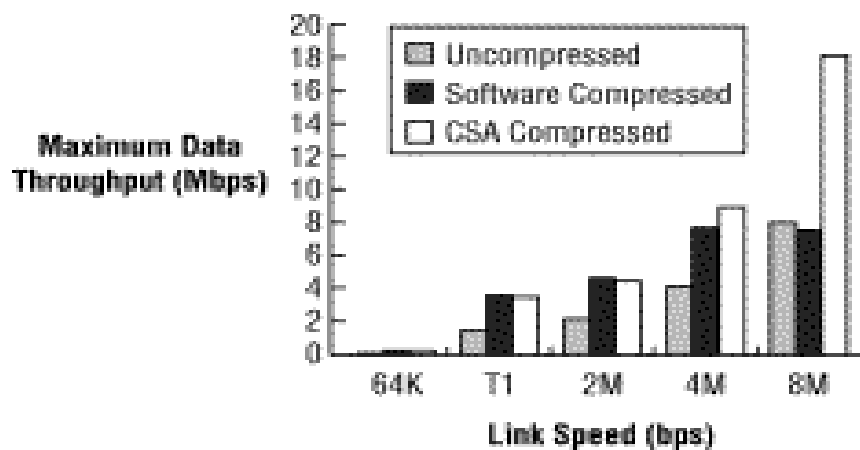


Figura 5.1.3 El funcionamiento de la Compresión en VIP2-50 (551 bytes por paquete)

Velocidad del Enlace	64k	T1	2M	4M	8M
Sin comprimir	0.064	1.607	2.011	4.026	8.061
Compresión por Software	0.128	3.713	4.643	7.88	7.713
Compresión por Hardware (CSA)	0.128	3.603	4.507	9.093	18.07
Máximos datos procesados en Mbps para varias velocidades de enlace					

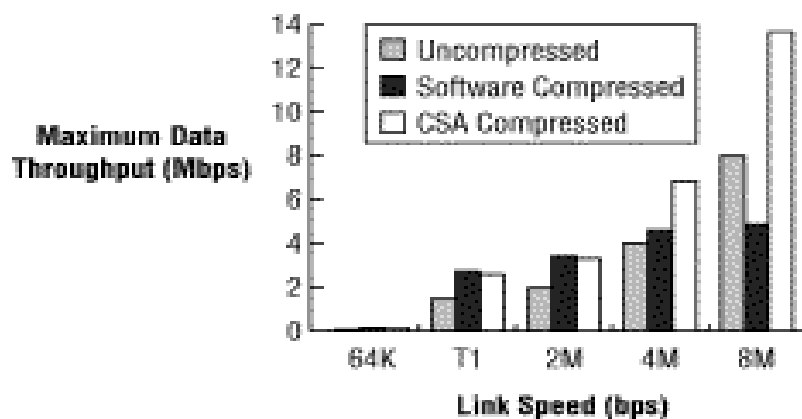


Figura 5.1.4. El funcionamiento de la Compresión en VIP2-50 (103 bytes por paquete).

Velocidad del Enlace	64k	T1	2M	4M	8M
Sin comprimir	0.063	1.604	2.014	4.029	8.025
Compresión por Software	0.106	2.801	3.511	4.696	4.861
Compresión por Hardware (CSA)	0.106	2.759	3.45	6.9	13.802

5.1.2 Análisis del algoritmo utilizado por la tecnología de compresión de Cisco®

El algoritmo de compresión usado por la tecnología Cisco®, cómo se menciona en este estudio, es el algoritmo Lossless de compresión de datos de LZS-STAC. Haciendo un análisis comparativo de lo que ofrece la tecnología de Cisco vs la tecnología utilizada en las mediciones de este trabajo, encontramos una serie de diferencias en lo referente a características técnicas, no fue posible comparar los resultados en el campo experimental debido a que la empresa Corporativa que se selecciona no tiene en sus equipos de routers habilitada la opción de compresión de datos.

Cisco® (LZS-STAC)	Peribit® (MSR)
Manejo de un buffer o tamaño de ventana fijo, se limita a menudo a la Unidad Máxima de Transmisión (MTU) de la red, es decir un tamaño de buffer de 1500 bytes.	No maneja un buffer fijo; sino más bien dinámico.
Detecta solo aquellos patrones que son igual o menor al tamaño de ventana establecido	Detecta todos aquellos patrones repetidos, independiente de su tamaño
La compresión varía, dependiendo de la velocidad del enlace, el tipo y el grado de agregación de los datos a través del enlace.	La compresión se comporta de manera lineal para cualquier velocidad de enlace.
Son eficaces en enlaces de poca velocidad, simples usuarios, o enlaces de aplicaciones simples, pero con un % de reducción de datos muy baja.	La compresión se comporta de igual manera para cualquier enlace.
La velocidad del algoritmo de búsqueda depende fuertemente de tamaño de la ventana o buffer.	Es independiente el algoritmo de búsqueda al tamaño de ventana o buffer.
Latencia aproximadamente de 30 ms (*)	Latencia de 2 ms
Añaden carga a la red, dependiendo del esquema de configuración de la compresión: Por software, SI. Por Hardware, NO	No se añade carga a la red, es todo por Hardware
Complejidad Computacional: No lineal de orden $O(n \cdot f(B))$, donde "n" es el número de símbolos de entrada y "B" es el tamaño del buffer de búsqueda.	Complejidad Computacional: Lineal de orden $O(n)$. Esto indica que el número de operaciones requeridas para procesar "n" bytes de entrada es proporcional a "n" bytes.

Nota: (*) Valor teórico, no fue posible obtenerlo de manera experimental.