UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

ESCUELA DE COMPUTACIÓN

TRABAJO DE GRADUACIÓN.

PARA OPTAR AL GRADO DE

INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

DISEÑO Y DESARROLLO DE UN

ADMINISTRADOR DE ANCHO DE BANDA.

PRESENTADO POR:

RENÉ MAURICIO HERNÁNDEZ LÓPEZ

JOSÉ ATILIO MARTÍNEZ TURCIOS

DANILO ANTONIO RUANO GARCÍA

ASESOR:

ING. CARLOS GUILLERMO BRAN

CIUDADELA DON BOSCO, DICIEMBRE, 2006

EL SALVADOR, CENTROAMÉRICA.

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

ESCUELA DE COMPUTACIÓN

ANTEPROYECTO DEL TRABAJO DE GRADUACIÓN.
PARA OPTAR AL GRADO DE
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

DISEÑO Y DESARROLLO DE UN
ADMINISTRADOR DE ANCHO DE BANDA.

ING RAFAEL COBOS	ING. CARLOS VÁSQUEZ	ING. HERBERT ASCENCIO
JURADO.	JURADO.	JURADO.

ING. CARLOS BRAN	ING. MELVIN CARÍAS
ASESOR.	TUTOR.

CIUDADELA DON BOSCO, MARZO, 2007
EL SALVADOR, CENTROAMÉRICA.

UNIVERSIDAD DON BOSCO

AUTORIDADES

ING. FEDERICO MIGUEL HUGUET RIVERA
RECTOR

PRESBITERO LIC. VICTOR BERMUDEZ YANEZ
VICERRECTOR ACADÉMICO

LIC. MARIO RAFAEL OLMOS ARGUETA
SECRETARIO GENERAL

ING. ERNESTO GODOFREDO GIRON
DECANO DE LA FACULTAD DE INGENIERÍA

LIC. JORGE MAURICIO COTO
DIRECTOR DE ESCUELA DE COMPUTACIÓN

AGRADECIMIENTOS

José Atilio Martínez Turcios.

En primer lugar quiero agradecer y dedicar el presente trabajo de graduación a DIOS ALTÍSIMO TODOPODEROSO, por darnos las fuerzas suficientes para sacar adelante este proyecto, por haber derramado sus bendiciones, orientarnos y permitirnos la conclusión de esta obra.

A mi madre BLANCA ROSA TURCIOS, mi padre JOSÉ ATILIO MARTÍNEZ CÁCERES, mi hermana CARMEN CECILIA MARTÍNEZ TURCIOS y DEMÁS FAMILIA, por apoyarme en todo momento, tanto económicamente como moralmente. Gracias por darme ánimos cuando las cosas parecían ir fuera de lugar, por confiar en mí, por darme fuerzas para seguir adelante. No tengo palabras para agradecerles todo su apoyo, su dedicación y su esfuerzo para hacer posible mi formación académica. Este título es de Ustedes.

A mi novia MARÍA TERESA CAMPOS CAMPOS por brindarme su apoyo en todo momento, por darme palabras de aliento, por escucharme y animarme cuando me encontraba decaído y sin ánimos de seguir. Gracias por soportar mis cambios de estado de ánimo producto del estrés al que estaba sometido. Mil gracias por todo el apoyo que me bridaste y me sigues brindando.

A mis compañeros de tesis y amigos RENE MAURICIO HERNÁNDEZ LÓPEZ Y DANILO ANTONIO RUANO. Gracias por permitirme compartir con Uds. Esta experiencia tan gratificante y desde ya mis más sinceras felicitaciones por este logro que hemos conseguido.

A Brenda Jennifer VENTURA LÓPEZ y MARIA TERESA CAMPOS por su apoyo en la logística en cada una de las defensas realizadas. Gracias por su valioso apoyo en los momentos que más lo necesitábamos.

A todos MIS AMIGOS y AMIGAS que siempre estuvieron pendientes del trabajo de graduación que estábamos realizando, gracias por su apoyo y comprensión por mis cambios de ánimo.

A nuestro asesor Ing. Carlos Bran por la orientación y apoyo brindado para finalizar este proyecto.

Al Jurado evaluador, Ing. Rafael Cobos, Ing. Carlos Vásquez e Ing. Herbert Ascencio, por sus observaciones que ayudaron a enriquecer este proyecto.

A la Fundación Empresarial para el Desarrollo Educativo FEPADE, por brindarme una beca a lo largo de todos mis estudios universitarios. Sin el apoyo de ellos este título no habría sido posible.

AGRADECIMIENTOS

Danilo Antonio Ruano García

Agradezco en primer lugar a Dios Todopoderoso por la oportunidad que nos concedió de culminar de forma satisfactoria nuestra carrera, apoyándonos y dándonos fuerzas en todo momento, te doy gracias Señor por este triunfo que nos has permitido gozar y por todos los retos que nos pusiste para enseñarnos el camino del aprendizaje.

A mis padres, Oscar Benjamín Ruano García y Carmen Rosario García de Ruano, y mi hermana Diana Vanesa Ruano García, les estoy eternamente agradecido por el apoyo incondicional que me han dado en todos los momentos de mi vida, no solo académicamente, sino personal y sentimentalmente, y en esta última fase por todos sus consejos y sugerencias.

Agradezco también a mi novia, Brenda Jennifer Ventura López, por apoyarme en todas las fases de mi carrera, y la inmensa comprensión y sacrificio que ha tenido que hacer por todos esos momentos que he dejado de dedicarle el tiempo justo que se merece por tan maravillosa persona; además de todo el apoyo logístico que brindó a todo el grupo en los momentos claves.

A la familia Martínez Turcios, por permitirnos invadir la privacidad de su hogar para lograr hacer las incontables pruebas que se hicieron del proyecto hasta llegar a su final.

A mis amigos y compañeros de tesis, Atilio y René, por el placer que ha sido el haber trabajado y compartido con ellos tantos momentos de éxito como de fracaso, pero que nos han moldeado hasta conseguir este logro, creo que el grupo de tesis se formó desde el primer año de la universidad.

Agradezco a Brenda y a María Teresa Campos por el apoyo logístico que nos brindaron en los momentos que no bastábamos nosotros para preparar las presentaciones, y ayudarnos con todo lo que las rodeaba.

Al Ing. Melvin Carías, por la orientación y sugerencias que nos dio en todo el proceso de tesis.

Al Ing. Carlos Bran, por el apoyo brindado como asesor, ayudarnos en la elaboración técnica del proyecto y por facilitarnos las instalaciones de los laboratorios para poder realizar las pruebas necesarias para completar el sistema.

Al jurado evaluador, compuesto por el Ing. Rafael Cobos, Ing. Herbert Ascencio e Ing. Giovanni Vásquez, por las sugerencias y recomendaciones que enriquecieron el proyecto.

Y a todas las demás personas que de una u otra forma colaboraron en la realización y culminación de este gran logro. Gracias.

AGRADECIMIENTOS

René Mauricio Hernández López

Agradezco primeramente a Dios Todopoderoso que vive y reina por los siglos de los siglos, por este proyecto que solo gracias a su misericordia y gracia pudo haber sido completado y a quien pertenece toda la gloria por la finalización de esta tesis.

A mis padres, René Hernández y Lilian López, por el apoyo moral que me brindaron, no sólo en la realización de este proyecto sino a lo largo de toda mi carrera, y por quienes también fue posible alcanzar este logro.

A mis compañeros y amigos, José Atilio y Danilo Antonio por tolerarme como compañero a lo largo de nuestra vida en la universidad y formar un grupo en el que la amistad fue el fundamento para que las cosas siempre progresaran y tener así una muy buena química de trabajo.

A Brenda Ventura y a María Teresa por su apoyo logístico en las defensas del proyecto y que gracias a ellas todo estuvo siempre listo al momento de las presentaciones.

A la familia Ruano García, por su hospitalidad y las molestias que toleraron durante las largas jornadas nocturnas de trabajo dedicadas a este proyecto, a pesar de las cuales nos recibieron en su hogar siempre con una sonrisa y con la misma hospitalidad.

A la familia Martínez Turcios, quienes pusieron a disposición su hogar para realizar las pruebas de la aplicación y nos trataron siempre con una gran calidez humana, una atención admirable.

Al ing. Carlos Bran, por su apoyo como asesor y por facilitarnos el laboratorio para realizar las pruebas del sistema cuando lo necesitamos. Al ing. Melvin Carías por orientarnos en el proceso de tramitación y de logística en las presentaciones del proyecto, así como también al jurado evaluador compuesto por el ing. Rafael Cobos, ing. Carlos Vásquez e ing. Herbert Ascencio por sus recomendaciones y sus ideas para mejorar el proyecto.

Y por último, pero muy especialmente, a todas aquellas personas que estuvieron a lo largo de todo el proceso, dando palabras de aliento e inspirando confianza en nosotros mismos y quienes siempre estuvieron orando por el éxito de este proyecto, siempre tendrán un lugar muy especial en mi corazón: Ladislao, Carolina, Marianella, Coralia, Miguel, Cristina, Karina, Ulises, Walter, Luís, Gloria a todos ellos, muchas gracias.

Índice General

Capítulo V: Mecanismos de regulación de tráfico

Capítulo VI: Diseño de Odisea Bandwidth Manager

Índice de Imágenes

Índice de Tablas

Introducción

La navegación en Internet hoy en día presenta muchas dificultades, de las cuales la mayoría se derivan del creciente uso que se hace de Internet y otras aplicaciones de red diariamente.

La velocidad del flujo de tráfico de paquetes a través de una red está condicionado por diversos factores que hacen que éste sea lento o rápido, por lo que continuamente se buscan soluciones, ya sea a nivel de software o de hardware, dentro o fuera de la red local, para que el flujo del tráfico sea siempre lo más rápido posible o, por lo menos, que se mantenga a una velocidad regular estable o que exista un mínimo de velocidad aceptable. Pero, si no se puede regular todo el flujo que existe en la red, se intenta, por lo menos, asegurar que el tráfico más importante que fluye a través de la red sí tenga un mínimo de velocidad y continuidad aún en horas de intenso uso de la red procurando tener un mínimo aceptable de ancho de banda disponible para éste, máxime si la mayor parte de la productividad de una empresa depende de la velocidad y garantía que dicho tráfico debe tener para llegar a su destino.

El presente proyecto pretende exponer una solución a nivel de software del control y administración del ancho de banda existente en una red, además de comprender de qué forma es que se procura asegurar que exista un ancho de banda mínimo, ya sea para un host, grupo de hosts, tipo de tráfico y otros parámetros que pueden tomarse en cuenta al momento de tomar una decisión en cuanto a la importancia del tráfico o de los usuarios al momento de transmitir o recibir datos.

Se hace una breve reseña sobre la tecnología QoS, que es la encargada de asegurar un servicio cualificado al tráfico de una red cubriendo distintas tecnologías, y se exponen los distintos protocolos que conforman QoS, además de

profundizar en el protocolo en el que la aplicación se ha basado para este aseguramiento de la calidad en el servicio.

Además, se explican distintos tipos de algoritmos que rodean, apoyan y dan funcionalidad a la administración del ancho de banda y priorización del tráfico según haya sido definido por el encargado de red que opere la aplicación.

Se espera que el proyecto cuente con todos los requerimientos deseados.

CAPÍTULO I: Definición del tema

1.1 Antecedentes

Antes que Internet pasara al uso comercial, la mayoría de los usuarios eran del sector académico y sus procesos comprendían la transferencia de archivos, correo electrónico y acceso remoto. Al generalizarse el uso del Internet en la sociedad y en las empresas, el número de usuarios creció exponencialmente, pasando de 16 millones de usuarios a nivel mundial en sus inicios en 1995, hasta cerca de 650 millones de usuarios a fines del año 2004[1]. En El Salvador, la cantidad de usuarios de Internet ha variado desde las 5000 personas en el año de 1996 a 550000 al año 2003[2]

Año	Usuarios de Internet en El Salvador
1996	5000
1997	15000
1998	25000
1999	50000
2000	70000
2001	150000
2002	300000
2003	550000

Tabla 1: Usuarios de Internet en El Salvador entre los años 1996 y 2003

Fuente: Naciones Unidas

Asimismo, el tipo de aplicaciones que transitan hoy por la Internet se ha diversificado considerablemente. Cada vez hay más aplicaciones para la transmisión de video, audio y voz, video interactivo, imágenes de alta resolución,

[1] Fuente: Naciones Unidas, 2005, Millennium Indicator Database [En línea] http://unstats.un.org/unsd/mispa/mi_series_results.aspx?rowID=608

[2] Fuente: Naciones Unidas, 2005, Millennium Indicator Database [En línea] http://unstats.un.org/unsd/mispa/mi_series_results.aspx?rowID=608

etc. Esto genera una gran cantidad de tráfico de red compitiendo entre sí para lograr el acceso a un ancho de banda limitado.

Una de las razones por las cuales se ha optado por desarrollar una aplicación para administrar el ancho de banda de forma dinámica, es el hecho que se ha observado, en innumerables ocasiones, los problemas causados por las congestiones de tráfico de red en lugares tales como centros de cómputo, ciber-cafés, oficinas, e inclusive en las viviendas.

El problema observado en los ciber-cafés es que, en las horas en las cuales coinciden varias personas navegando en Internet, existe un tráfico de red intenso que produce congestión en la red. Esto afecta principalmente la velocidad de navegación, presentando una gran lentitud al momento de visitar una página Web.

Otro problema, observado mayormente en algunas oficinas, es el uso de sintonizadores de radio por medio de Internet. Aplicaciones de uso popular, tales como Windows Media Player©, Winamp, Real Player™, y algunas aplicaciones de mensajería instantánea proporcionan este tipo de servicios, haciendo que el rendimiento de la red decaiga notablemente. Estos sintonizadores de radio disminuyen al ancho de banda disponible afectando las actividades normales de las empresas, tales como videoconferencias, transferencia de archivos, conexiones a dispositivos remotos, acceso a bases de datos, etc.

Un ejemplo más de los problemas causados por el uso de aplicaciones que consumen ancho de banda, son los programas que se utilizan para compartir archivos a través de Internet. Estas aplicaciones (conocidas como P2P peer-to-peer) consumen un elevado ancho de banda, haciendo que otras aplicaciones que hacen uso del mismo, se vean mermados, afectando su normal funcionamiento.

Debido a estos problemas se ha propuesto como solución el diseño de una herramienta que ayude a los administradores de red a distribuir al ancho de banda de manera adecuada, según las necesidades que se tengan.

Entre algunos de los software que existen actualmente en el mercado, se encuentra el SoftPerfect Bandwidth Manager©, el cual ofrece una interfaz gráfica que facilita la configuración de las opciones con las que trabajará la red, teniendo la posibilidad de administrar el ancho de banda en base a distintos parámetros, tales como: dirección IP o rango de direcciones IP, protocolo de capa de transporte y número de puerto. El costo de este programa es de $99.00[3], pudiendo obtener un demo gratuito que tiene una validez para 30 días posteriores a su instalación.

Otros productos, entre ellos Solaris Bandwidth Manager (desarrollado por la empresa Sun©), ofrecen una mayor gama de servicios para controlar el tráfico de la red, sin embargo los costos de la licencia de aplicaciones como la mencionada pueden ir desde $500.00 hasta $13,000.00[4] lo cual se convierte en un gasto demasiado excesivo en ocasiones para algunos administradores de redes, ya que en muchos casos no se cuenta con presupuestos que permitan utilizar software de este tipo.

[3] Información obtenida del sitio http://www.softperfect.com
[4] Información obtenida del sitio http://www.sun.com/software/bandwidth/index.xml

1.2 Importancia de la Investigación

1.2.1 Planteamiento del Problema

El creciente uso de Internet hoy en día hace que las redes estén saturadas por el enorme tráfico que circula a través de las ellas. Por ejemplo, si un usuario está realizando la descarga de algún archivo, esta descarga acapara el ancho de banda que el programa utilizado necesite para tal función, dejando a los demás usuarios de la red sin ancho de banda suficiente, presentando lentitud en la navegación y poco o casi nulo ancho de banda para las funciones que lo requieren, afectando el rendimiento de la red.

Otras aplicaciones, como las utilizadas para video-conferencias o sintonizadores de radio o televisión en Internet, consumen gran cantidad de ancho de banda o son muy sensibles a la caída de éste, por lo que, si se tuviera un ancho de banda “dedicado” para alguna de éstas aplicaciones, de alguna manera se sentiría menos si otro dispositivo o host comienza a consumir de forma súbita el ancho de banda utilizado en la red.

En vista de lo anterior, se ha decidido desarrollar una aplicación que permita administrar el ancho de banda en una red. Dicha aplicación estará ubicada en el host que una la red LAN con el enlace WAN, de manera que la computadora que sirva para este propósito deberá ser la conectividad entre la LAN y la WAN.

La aplicación será capaz de monitorear los paquetes que provengan de la LAN y colocarlos en una cola, luego estos paquetes serán ordenados en base a prioridades establecidas mediante políticas definidas bajo los criterios de: rango de direcciones IP, usuarios, puertos y protocolos.

En síntesis, los problemas que se presentan son los siguientes:

- Generación de cuellos de botella en horas de alto tráfico de red.
- Disminución del rendimiento de la red debido a cuellos de botella.
- Necesidad de un ancho de banda dedicado para aplicaciones específicas.

1.2.2 Justificación

La razón principal por la que se desarrollará el software que administre y controle el ancho de banda en una red LAN es para asegurarse que un usuario o grupo de usuarios puedan contar con el ancho de banda mínimo necesario para satisfacer sus necesidades de conexión en todo momento y sin la preocupación de quedarse sin este recurso.

Con esto se pretende que cualquier usuario que ejecute alguna aplicación que consuma mucho ancho de banda (tales como video-conferencia o sintonizadores de radio o televisión por Internet) no acapare todo el ancho de banda disponible para la red y permita que los demás usuarios conectados a dicha red continúen realizando sus actividades normales.

El software que se diseñará será capaz de controlar el ancho de banda de manera que los cuellos de botella que se generan en las horas de alto tráfico de red no afecten de una manera significativa al rendimiento de la red.

La herramienta que se desarrollará permitirá brindar un ancho de banda dedicado a las aplicaciones que lo requieran, es decir, que por su función dentro de una empresa ó por su misma naturaleza, requieran de un ancho de banda mínimo para funcionar de manera adecuada.

En el mercado existen diversas aplicaciones que permiten administrar el ancho de banda de las redes pero la mayoría de ellos solo están disponibles por un determinado costo y no son de código abierto, el software que se pretende crear será de código abierto, para resolver dicho inconveniente.

En resumen, los beneficios que brindará la aplicación que se desarrollará son los siguientes:

- Mayor control sobre el tráfico de red y los cuellos de botella que se generan.
- Brindar ancho de banda dedicado a las aplicaciones que lo requieran.

1.3 *Objetivos*

1.3.1 General

Diseñar y desarrollar una aplicación que permita la administración del ancho de banda utilizada en una red de área local.

1.3.2 Específicos

- Investigar las técnicas y procedimientos de Calidad de Servicios (QoS) utilizados en aplicaciones diseñadas para este fin, de manera que sirvan como base para el desarrollo de la aplicación a implementar.

- Explicar las estrategias de administración de ancho de banda existentes en los equipos más comunes de Internetworking.

- Aplicar los conceptos de Calidad de Servicio en el diseño y desarrollo de una herramienta para distribuir el ancho de banda.

- Desarrollar un motor de políticas sencillo y de fácil configuración.

- Desarrollar un modulo de interfaz gráfica para el control de los siguientes criterios: direcciones de capa 3, aplicaciones, usuarios, protocolos y dirección MAC, de manera que se mejore el rendimiento de una red.

1.4 *Alcances y Limitaciones*

1.4.1 Alcances.

- Se empleará el lenguaje de programación Visual C++ para desarrollar la aplicación debido a que ofrece una mayor facilidad para el manejo de paquetes. También se utilizará la herramienta Visual Basic .Net para el desarrollo de una interfaz gráfica.

- Se podrá administrar el ancho de banda en base a las siguientes opciones: direcciones IP, rango de direcciones IP, grupos de usuarios, puertos, protocolos de red y direcciones MAC.

- La aplicación a diseñar será capaz de distribuir el ancho de banda de manera estática (el administrador de la red podrá asignar el porcentaje de ancho de banda que estime conveniente para las aplicaciones, grupos, protocolos o rangos de direcciones).

- La aplicación contará con un módulo para poder visualizar las estadísticas de los parámetros que se estarán manejando (paquetes por segundo, tráfico por protocolo, entre otros) en tiempo real.

- Se tendrá un módulo para crear un historial (log) de las estadísticas de los parámetros antes mencionados, de manera que se puedan crear reportes con los datos obtenidos.

- Se desarrollará un instalador para la aplicación, de manera que se vuelva fácil su despliegue en la infraestructura de Internetworking.

1.4.2 Limitaciones.

- Se trabajará sobre una plataforma que permita optimizar su operación para un número de usuarios concurrentes. Se ha escogido la plataforma Windows como sistema operativo en que se desarrollará la aplicación, esto debido a que las herramientas utilizadas ofrecen una mayor facilidad de programación en Windows y no afecta el consumo de recursos.

- El software estará enfocado a ser utilizado por administradores de red o personas que estén encargadas del desempeño de una red y no por usuarios comunes que no posean las nociones necesarias para manejar una red.

- En la aplicación no se implementarán las características que posee QoS, sino características similares como la de Servicios Diferenciados, pero utilizando algoritmos diferentes a los planteados por las tecnologías de otras arquitecturas.

CAPÍTULO II: Marco Conceptual.

2.1 Protocolo Ethernet

Ethernet es el nombre de una tecnología de redes de computadoras de área local basada en tramas de datos. Ethernet define las características de cableado y señalización de nivel físico y los formatos de trama del nivel de enlace de datos del modelo OSI.[5]

Para permitir el envío local de las tramas en Ethernet, se debe contar con un sistema de direccionamiento, una forma de identificar los computadores y las interfaces de manera exclusiva. Ethernet utiliza direcciones MAC que tienen 48 bits de largo y se expresan como doce dígitos hexadecimales. Los primeros seis dígitos hexadecimales, identifican al fabricante, y se conoce como Identificador Exclusivo Organizacional (OUI). Los seis dígitos hexadecimales restantes representan el número de serie de la interfaz, administrado por el proveedor del equipo. La tarjeta de interfaz de red (NIC por sus siglas en inglés) utiliza la dirección MAC para evaluar si el mensaje se debe pasar o no a las capas superiores del modelo OSI. La NIC realiza esta evaluación sin utilizar tiempo de procesamiento de la CPU permitiendo mejores tiempos de comunicación en una red Ethernet.

En una red Ethernet, cuando un dispositivo envía datos, el dispositivo origen adjunta un encabezado con la dirección MAC del destino y origen y envía los datos a la red. Cada dispositivo de la red verifica si su dirección MAC coincide con la dirección destino física que transporta la trama de datos. Si no hay concordancia, la NIC descarta la trama de datos. Cuando los datos llegan al nodo destino, la NIC hace una copia y pasa la trama hacia las capas superiores del modelo OSI.

[5] En línea http://es.wikipedia.org/wiki/Ethernet. Consultado el 16 de julio de 2006, Actualizado el 12 de julio de 2006.

2.1.1 Trama Ethernet

Una trama Ethernet tiene el siguiente formato:

Preámbulo	SOF	Destino	Origen	Tipo	Datos	FCS
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	1500 bytes	4 bytes

Figura 1. Formato de la trama Ethernet.

- **Preámbulo:** Es una secuencia de bits que se utiliza para sincronizar y estabilizar al medio físico antes de comenzar la transmisión de datos.
- **SOF (Start-of-frame delimiter):** Delimitador del inicio de la trama. Consta de un byte y es un patrón de unos y ceros alternados que finaliza en dos unos consecutivos (10101011), indicando que el siguiente bit será el más significativo del campo de dirección de destino.
- **Dirección de destino:** El campo de dirección destino es un campo de 48 bits (6 bytes) que especifica la dirección MAC hacia la que se envía la trama.
- **Dirección de origen:** El campo de la dirección de origen es un campo de 48 bits (6 bytes) que especifica la dirección MAC desde donde se envía la trama.
- **Tipo:** El campo de tipo es un campo de 16 bits (2 bytes) que identifica el protocolo de red de alto nivel asociado con el paquete o en su defecto la longitud del campo de datos.
- **Datos:** El campo de datos contiene de 46 a 1500 Bytes. Cada Byte contiene una secuencia arbitraria de valores. El campo de datos es la información recibida del nivel de red.
- **FCS (Frame Check Sequence):** El campo Secuencia de verificación de la trama contiene un valor de verificación CRC (código de redundancia cíclica) de 32 bits o 4 bytes, calculado por el dispositivo emisor en base al contenido de la trama y recalculado por el dispositivo receptor para verificar la integridad de la trama.

2.2 El protocolo IP

El protocolo de Internet, conocido como IP por sus siglas en inglés (Internet Protocol), es un protocolo no orientado a la conexión. Es utilizado para la comunicación de datos a través de una red de paquetes conmutados. Los datos en una red basada en IP son enviados en forma de bloques o paquetes conocidos también como datagramas.

IP provee un servicio no fiable, conocido como servicio de mejor esfuerzo (best effort). Esto significa que IP no provee ningún mecanismo para saber si el destino recibió el paquete. La única seguridad que proporciona se logra mediante checksums o sumas de comprobación para determinar si el paquete recibido es el mismo que fue enviado. Para proveer confiabilidad se recurre a protocolos de capa de transporte, por ejemplo TCP.

La versión más popular del Protocolo de Internet actualmente es IPv4. Éste utiliza direcciones de 32 bits divididas en 4 octetos. Sin embargo, debido al agotamiento de las direcciones IPv4 se surgió una versión superior de este protocolo, llamado IPv6, el cual utiliza direcciones de fuente y destino de 128 bits, muchas mas direcciones que las que provee IPv4.

2.2.1 Dirección IP

Una dirección IP es un número que identifica de manera lógica y jerárquicamente a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP[6].

2.2.2 Direcciones IPv4

Una dirección IPv4 se representa mediante un número binario de 32 bits. Las direcciones también pueden representarse como números de notación decimal. Para esto se dividen los 32 bits de la dirección en cuatro octetos.

[6] En línea http://es.wikipedia.org/wiki/Direcci%C3%B3n_IP#Direcciones_IPv6. Consultado el 15 de julio de 2006, Actualizado el 8 de julio de 2006.

Cada octeto puede tomar un valor entre 0 y 255, y se separan mediante el carácter ".".

2.2.2.1 Clasificación de las direcciones IPv4.

Hay tres clases de direcciones IP: clase A, clase B y clase C. En la actualidad, las direcciones de clase A se reservan para los gobiernos de todo el mundo, las direcciones de clase B para las medianas empresas, y las de clase C para todos los demás solicitantes.

- En una red de clase A, el primer octeto identifica a la red y los tres últimos octetos identifican a los hosts, de modo que la cantidad máxima de hosts es 2^{24} menos dos: las direcciones reservadas de broadcast y de red, es decir, 16,777,214 hosts.
- En una red de clase B, los dos primeros octetos identifican a la red, y los dos octetos finales identifican a los hosts, de modo que la cantidad máxima de hosts es 2^{16} menos dos, es decir, 65,534 hosts.
- En una red de clase C, los tres primeros octetos identifican a la red y el octeto final identifica a los hosts, de modo que la cantidad máxima de hosts es 2^{8} menos dos, es decir, 254 hosts.

Adicionalmente, existen otros dos tipos de clases, una de ellas es para direcciones multicast (tipo de direcciones permite direccionar a un grupo concreto de hosts dentro de una subred) y la última esta reservada para un uso posterior. La estructura de cada clase se detalla en el cuadro siguiente:

Clase	Dirección IP (R=Red, H=host)	Rango
A	0RRRRRRR.HHHHHHHH.HHHHHHHH.HHHHHHHH	1.0.0.0 a 127.255.255.255
B	10RRRRRR.RRRRRRRR.HHHHHHHH.HHHHHHHH	128.0.0.0 a 191.255.255.255
C	110RRRRR.RRRRRRRR.RRRRRRRR.HHHHHHHH	192.0.0.0 a 223.255.255.255
D	1110 (Dirección de multicast)	224.0.0.0 a 239.255.255.255
E	1111 (Reservado para uso futuro)	240.0.0.0 a 255.255.255.255

Figura 2. Clases de direcciones IP y sus rangos correspondientes.

2.2.2.2 La cabecera IPv4.

La siguiente figura muestra la estructura de la cabecera IPv4:

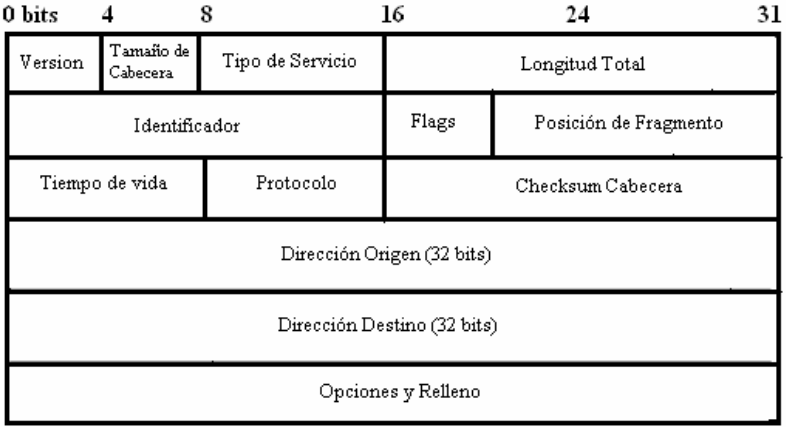

Figura 3. Formato del datagrama IPv4.

- **Versión:** Consta de 4 bits. Describe el formato de la cabecera utilizada.
- **Tamaño Cabecera (IHL):** Formada por 4 bits. Longitud de la cabecera, en palabras de 32 bits. Su valor mínimo es de 5 para una cabecera correcta, y el máximo de 15.
- **Tipo de Servicio:** 8 bits. Indica una serie de parámetros sobre la calidad de servicio deseada durante el tránsito por una red.
- **Longitud Total:** 16 bits. Es el tamaño total, en octetos, del datagrama, incluyendo el tamaño de la cabecera y el de los datos. El tamaño máximo de los datagramas usados normalmente es de 576 octetos (64 de cabeceras y 512 de datos). En caso de fragmentación este campo contendrá el tamaño del fragmento, no el del datagrama original.
- **Identificador:** 16 bits. Identificador único del datagrama. Se utilizará, en caso de que el datagrama deba ser fragmentado, para poder distinguir los fragmentos de un datagrama de los de otro.
- **Indicadores:** 3 bits. Actualmente utilizado sólo para especificar valores relativos a la fragmentación de paquetes:
  bit 0: Reservado; debe ser 0
  bit 1: 0 = Divisible, 1 = No Divisible

bit 2: 0 = Último Fragmento, 1 = Fragmento Intermedio (le siguen más fragmentos)

- **Posición de Fragmento:** 13 bits. En paquetes fragmentados indica la posición, en unidades de 64 bits, que ocupa el paquete actual dentro del datagrama original. El primer paquete de una serie de fragmentos contendrá en este campo el valor 0.
- **Tiempo de Vida (TTL):** 8 bits. Indica el máximo número de direccionadores que un paquete puede atravesar. Cada vez que algún nodo procesa este paquete disminuye su valor en, como mínimo, un direccionador. Cuando llegue a ser 0, el paquete no será reenviado.
- **Protocolo:** 8 bits. Indica el protocolo de siguiente nivel utilizado en la parte de datos del datagrama.
- **Checksum Cabecera:** 16 bits. Checksum de la cabecera. Se recalcula cada vez que algún nodo cambia alguno de sus campos (por ejemplo, el Tiempo de Vida). El método de cálculo (intencionadamente simple) consiste en sumar el complemento a 1 de cada palabra de 16 bits de la cabecera y hacer el complemento a 1 del valor resultante.
- **Dirección IP de Origen:** 32 bits. Contiene la dirección IP del origen del paquete.
- **Dirección IP de Destino:** 32 bits Contiene la dirección IP del destino del paquete.
- **Opciones:** Tamaño variable. Contiene un número indeterminado de opciones.
- Relleno: Tamaño variable. Utilizado para asegurar que el tamaño, en bits, de la cabecera es un múltiplo de 32. El valor usado es el 0.

2.2.3 Direcciones IPv6.

A diferencia de IPv4, IPv6 cuenta con direcciones de 128 bits de longitud. Estos 128 bits corresponden a 32 dígitos hexadecimales, divididos en ocho cuartetos. Cada cuarteto se separa por medio del carácter ":". El número de direcciones IPv6 posibles es de 2128 ≈ 3.4 x 1038. Esta cantidad de

direcciones posibles mediante IPv6 hace posible que desaparezcan los problemas del direccionamiento del IPv4 actual y evita el uso de técnicas como NAT para proveer conectividad a todas los host de una red.

2.2.3.1 *Clasificación de las direcciones IPv6.*

Los tipos de direcciones IPv6 pueden identificarse tomando en cuenta los primeros bits de cada dirección.

- ::/128 – la dirección con todo ceros se utiliza para indicar la ausencia de dirección, y no se asigna ningún nodo.
- ::1/128 – la dirección de loopback es una dirección que puede usar un nodo para enviarse paquetes a sí mismo (corresponde con 127.0.0.1 de IPv4). No puede asignarse a ninguna interfaz física.
- ::/96 – La dirección IPv4 compatible se usa como un mecanismo de transición en las redes duales IPv4/IPv6.
- ::ffff:0:0/96 – La dirección IPv4 mapeada es usada como un mecanismo de transición en terminales duales.
- fe80::/10 – El prefijo de enlace local específica que la dirección sólo es válida en el enlace físico local.
- fec0::/10 – El prefijo de emplazamiento local específica que la dirección sólo es válida dentro de una organización local. La RFC 3879 lo declaró obsoleto, estableciendo que los sistemas futuros no deben implementar ningún soporte para este tipo de dirección especial.
- ff00::/8 – El prefijo de multicast es usado para las direcciones multicast.

2.2.3.2 **Cabeceras IPv6.**

La longitud total de la cabecera de un paquete IPv6 tiene 40 bytes, y posee el siguiente formato:

Figura 4. Formato del datagrama IPv6.

- **Versión:** Consta de 4 bits. Describe el formato de la cabecera utilizada.
- **Clase de tráfico:** Es equivalente a ToS (Type of Service) en IPv4. Tiene una longitud de 8 bits.
- **Etiqueta de flujo:** Permite tráfico con requisitos de tiempo real. Tiene una longitud de 20 bits.
- **Longitud de campo de datos:** Es la longitud de los propios datos. Tiene una longitud de 16 bits.
- **Cabecera siguiente:** Indica cual cabecera de extensión (si la hay) le sigue a ésta o en su defecto, el protocolo de capa superior que le sigue. Tiene una longitud de 8 bits.
- **Límite de saltos:** Indica el número de saltos que un paquete pueda dar antes de ser descartado. Tiene una longitud de 8 bits.
- **Dirección Origen:** Contiene la dirección del host origen. El tamaño de este campo es de 128 bits.
- **Dirección Destino:** Contiene la dirección del host destino. El tamaño de este campo es de 128 bits.

2.3 Protocolo TCP (Transmission Control Protocol).

El Protocolo de Control de Transmisión (TCP por sus siglas en inglés), es un protocolo de comunicación orientado a conexión y fiable del nivel de transporte y es uno de los protocolos más extendidos en Internet. TCP garantiza que los datos serán entregados sin errores y en el mismo orden en que fueron transmitidos.

TCP ofrece fiabilidad en la comunicación, debido a que IP aporta un servicio de mejor esfuerzo y sin confirmación de recibo. TCP provee las funciones necesarias para proveer un servicio que brinde una comunicación libre de errores, en orden, sin pérdidas y sin duplicaciones.

2.3.1 La cabecera TCP.

La unidad de datos de este protocolo recibe el nombre de segmento TCP. Como la cabecera debe implementar todos los mecanismos de fiabilidad del protocolo su tamaño es bastante grande, como mínimo 20 bytes. El formato de la cabecera TCP se detalla a continuación:

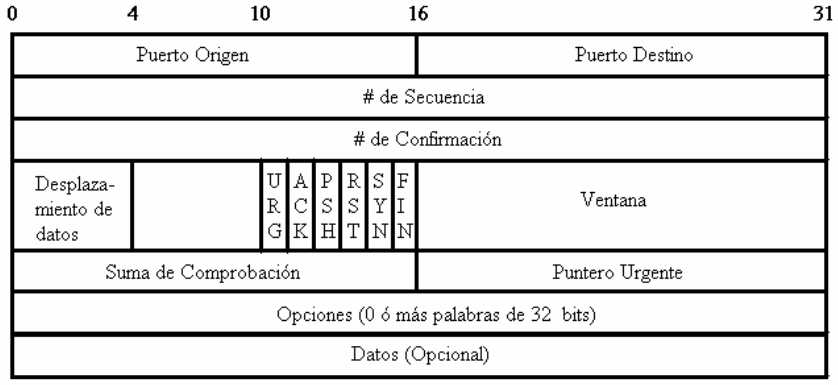

Figura 5. Formato de la cabecera TCP

- **Puerto origen:** Es el punto de acceso de la aplicación en el origen. Esta formado por 16 bits
- **Puerto destino**: Es el punto de acceso de la aplicación en el destino. Esta formado por 16 bits
- **Número de secuencia:** Identifica el primer byte del campo de datos. indica el primer byte de datos que hay en el segmento. Al principio de la conexión se asigna un número de secuencia inicial (ISN, Initial Sequence Number) y a continuación los bytes son numerados consecutivamente. Consta de 32 bits.
- **Número de confirmación (ACK):** Cuando el bit ACK está activo, este campo contiene el número de secuencia del primer byte que espera recibir.

Dicho de otra manera, el número ACK - 1 indica el último bit reconocido. Consta de 32 bits.

- **Longitud de la cabecera:** Indica el número de palabras de 32 bits que hay en la cabecera. Normalmente el tamaño de la cabecera es de 20 bytes por lo que en este campo se almacenará el número 5. Si el TCP utiliza todos los campos de opciones la cabecera puede tener una longitud máxima de 60 bytes. Este campo tiene una longitud de 4 bits.
- **Reservado:** Se ha reservado para su uso futuro y se inicializa con ceros. Posee 6 bits.
- **Indicadores o campo de control** (6 bits): Cada uno de los bits recibe el nombre de indicador y cuando está a 1 indica una función específica del protocolo.
    - **URG:** Hay datos urgentes y en el campo "puntero urgente" se indica el número de datos urgentes que hay en el segmento.
    - **ACK:** Indica que tiene significado el número que hay almacenado en el campo "número de confirmación".
    - **PSH:** Sirve para invocar la función de carga (push). Con esta función se indica al receptor que debe pasar a la aplicación todos los datos que tenga en la memoria intermedia sin esperar a que sean completados. De esta manera se consigue que los datos no esperen en la memoria receptora hasta completar un segmento de dimensión máxima.
    - **RST:** Sirve para hacer un reset de la conexión.
    - **SYN:** Sirve para sincronizar los números de secuencia.
    - **FIN:** Sirve para indicar que el emisor no tiene mas datos para enviar.
- **Ventana:** Indica cuantos bytes tiene la ventana de transmisión del protocolo de control de flujo utilizando el mecanismo de ventanas deslizantes. Contiene el número de bytes de datos comenzando con el que se indica en el campo de confirmación y que el que envía está dispuesto a aceptar. Consta de 16 bits

- **Suma de comprobación:** Este campo se utiliza para detectar errores mediante el complemento a uno de la suma en módulo 2^{16} - 1 de todas las palabras de 16 bits que hay en el segmento mas una pseudo-cabecera. La pseudo-cabecera incluye los siguientes campos de la cabecera IP: dirección IP origen, dirección IP destino, el protocolo y un campo longitud del segmento. Consta de 16 bits.
- **Puntero urgente:** Cuando el indicador URG está activo, este campo indica cual es el último byte de datos que es urgente. De esta manera el receptor puede saber cuantos datos urgentes llegan. Este campo es utilizado por algunas aplicaciones como telnet, rlogin y ftp. Consta de 16 bits.
- **Opciones:** Si está presente permite añadir una única opción de entre las siguientes:
    - Timestamp para marcar en que momento se transmitió el segmento y de esta manera monitorizar los retardos que experimentan los segmentos desde el origen hasta el destino.
    - Aumentar el tamaño de la ventana.
    - Indicar el tamaño máximo del segmento que el origen puede enviar.

  El tamaño de este campo es variable.

2.3.2 Funcionamiento de TCP.

Las conexiones TCP se componen de tres etapas: establecimiento de conexión, transferencia de datos y fin de la conexión.

Antes de transmitir datos, los dos clientes que desean comunicarse deben llevar a cabo un proceso de sincronización para establecer una conexión virtual para cada sesión entre ellos. Este proceso de sincronización asegura que ambas partes están listas para la transmisión y permite que los dispositivos determinen los números de la secuencia inicial de dicha sesión. Este proceso se llama saludo de tres vías, es un proceso de tres pasos para establecer una conexión virtual entre dos dispositivos.

2.3.3 Uso de Ventanas

A menudo, la cantidad de datos que se necesita transmitir es demasiado grande como para ser enviada en un solo segmento de datos. En este caso, los datos deben dividirse en porciones de menor tamaño para permitir su correcta transmisión. TCP tiene la responsabilidad de dividir los datos en segmentos. Además, es posible que las máquinas receptoras no sean capaces de recibir datos con la rapidez que el origen los envía, tal vez, porque el dispositivo receptor está ocupado con otras tareas o porque el transmisor simplemente es un dispositivo más robusto.

Una vez segmentados los datos, deben transmitirse hacia el dispositivo destino. Uno de los servicios que provee TCP es el control de flujo que regula la cantidad de datos enviada durante un período de transmisión dado. Este proceso de control de flujo se conoce como uso de ventanas.

El tamaño del a ventana determina la cantidad de datos que se pueden transmitir simultáneamente antes que el destino responda con un Acuse de recibo (ACK). Después que un host transmita el tamaño de ventana en bytes, el host debe recibir un ACK indicando que la información se recibió antes de poder enviar más información.

TCP usa las ventanas para determinar de forma dinámica el tamaño de la transmisión. Los dispositivos negocian el tamaño de la ventana a un número específico de bytes para transmitir antes del ACK.

2.3.4 Números de secuencia

Como en una transmisión de información no hay garantía de que los datos llegarán en el orden en que se transmitieron, TCP aplica los números de secuencia a los segmentos de datos que transmite, de modo que el receptor pueda reensamblar adecuadamente los bytes en su orden original. Los

números de secuencia le indican al dispositivo destino cómo ordenar correctamente los bytes a medida que éstos llegan al destino.

Cada segmento TCP se numera antes de su transmisión. En el host destino, TCP usa los números de secuencia para reensamblar los segmentos hasta formar un mensaje completo. Si falta algún número de secuencia en la serie, ese segmento se vuelve a transmitir.

2.4 Protocolo UDP (user Datagram Protocol).

El Protocolo de Datagramas de Usuario (UDP por sus siglas en inglés User Datagram Protocol) es un protocolo de capa de transporte del modelo OSI y basa su funcionamiento en el intercambio de datagramas, los cuales no requieren que se establezca una conexión previa. La razón por la que no es necesario establecer una conexión previa es porque el diagrama contiene suficiente información de direccionamiento en su encabezado.

En general UDP no tiene la mayoría de características de los protocolos orientados a la conexión entre ellos no ofrece confirmación, no hay control de flujo por paquetes pueden llegar en desorden, no hay comprobación de errores, de hecho ni siquiera se garantiza que el paquete haya llegado.

Sin embargo a pesar de que UDP carece de esas funciones, es más rápido y eficiente cuando la prioridad es tráfico ligero o sensible al tiempo, a la vez permite ser utilizado para consultas rápidas en la red por ejemplo un broadcast utiliza UDP, así no es necesario establecer una conexión previa con cada nodo de la red.

2.4.1 Estructura de datagrama UDP

El datagrama UDP está formado por cuatro campos, de los cuales dos son prácticamente opcionales. La siguiente figura describe el encabezado:

	Bits 0 - 15	16 - 31
0	Puerto Origen	Puerto Destino
32	Longitud	Checksum
64	Datos	

Figura 6: Detalle del datagrama UDP. En color azul se encuentran resaltados los campos opcionales.

En el encabezado del datagrama el puerto origen puede ser opcional debido a que UDP no solicita respuestas. Cuando este campo no es utilizado debe ser puesto a cero.

El segundo campo del datagrama UDP es el puerto de destino el cual identifica el puerto hacia el cual va dirigido el datagrama. Este campo, a diferencia del puerto origen, es requerido en el datagrama.

El campo de longitud especifica la longitud del datagrama UDP, este campo está representado por 16 bits y describe la longitud incluyendo el encabezado y los datos del datagrama. El valor mínimo que puede adoptar este campo es de 8 bytes, ya que esa es la longitud del encabezado mismo.

El campo checksum, al igual que el puerto origen, es opcional debido a que no se realiza una comprobación, sin embargo en la práctica este campo si es utilizado en algunas ocasiones, generalmente cuando se trabaja con aplicaciones de voz en las que se prioriza a la velocidad en lugar de la fiabilidad de la entrega. En este caso, el checksum tiene exactamente el mismo funcionamiento que el checksum del encabezado IP.

2.4.2 Aplicaciones que utilizan UDP

Debido a la fiabilidad que UDP ofrece, las aplicaciones que lo utilicen deben estar dispuestas a aceptar pérdidas, errores o incluso paquetes repetidos (duplicación). Aplicaciones como servidores DNS, Voz sobre IP y

servidores TFTP utilizan mayormente el protocolo UDP, debido a que deben responder rápidamente o bien simplemente no es necesario establecer una conexión con otros puntos de la red para realizar sus funciones.

2.5 Puertos TCP y UDP

TCP y UDP usan el concepto de número de puerto para identificar a las aplicaciones emisoras y receptoras. Cada lado de la conexión tiene asociado un número de puerto asignado por la aplicación emisora o receptora. EL número de puerto esta formado por 16 bits, por lo tanto existen 65536 puertos posibles.

Los puertos son clasificados en tres categorías: bien conocidos, registrados y dinámicos/privados. Los puertos bien conocidos son asignados por la Internet Assigned Numbers Authority (IANA), van del 0 al 1023 y son usados normalmente por el sistema o por procesos con privilegios. Las aplicaciones que usan este tipo de puertos son ejecutadas como servidores y se quedan a la escucha de conexiones. Algunos ejemplos son: FTP (21), SSH (22), Telnet (23), SMTP (25) y HTTP (80).

Los puertos registrados son normalmente empleados por las aplicaciones de usuario de forma temporal cuando conectan con los servidores, pero también pueden representar servicios que hayan sido registrados por un tercero.

Los puertos dinámicos/privados también pueden ser usados por las aplicaciones de usuario, pero este caso es menos común. Los puertos dinámicos/privados no tienen significado fuera de la conexión TCP en la que fueron usados.

2.6 Descripción de sockets

2.6.1 Concepto de socket.

Para que las comunicaciones de datos se lleven a cabo de forma efectiva se necesita traspasar los niveles físico, de enlace de datos, de red y, hasta cierto punto, de transporte, debido a que, aunque éstos facilitan la comunicación entre dos o más nodos, no representan por sí solos un esquema completo de comunicación. Por tanto, se hace necesaria la inclusión de las capas de nivel superior: sesión presentación y aplicación, los cuales se encuentran ligados a la programación.

Un socket es una interfaz de entrada y salida de datos que permite la intercomunicación entre procesos. Dichos procesos pueden estarse ejecutando en el mismo o en distintos sistemas, unidos mediante una red.

2.6.2 Estructura de dirección socket.

Para establecer las comunicaciones entre procesos utilizando sockets, es necesario definir primero la familia o dominio al que pertenecerá. La familia o domino especifica el formato de las direcciones que se podrán dar a los sockets y los diferentes protocolos soportados por la comunicación entre sockets.

Todo dominio posee su propia estructura especialmente creada conocida como estructura de dirección socket. Los nombres de dichas estructuras inician con las letras sockaddr_ y finalizan con un sufijo único para cada conjunto de protocolos. Por ejemplo, para el dominio *IPV4 (AF_INET)* se utiliza la estructura de dirección *sockaddr_in;* para el dominio *IPV6 (AF_INET6)* se utiliza la estructura de dirección *sockaddr_in6.*

2.6.3 Conversión de direcciones.

Las direcciones de Internet vienen dadas en un formato fácilmente comprensible por las personas, pero incomprensibles de interpretar para una

computadora. Para realizar la conversión de las direcciones hacia un formato compresible por las computadoras se requiere de algunas funciones que convierten direcciones de Internet entre cadenas ASCII y valores binarios en ordenamiento de red. Estas funciones son:

- **inet_aton:** Convierte una cadena numérica con formato de dirección IPV4 a un entero de 32 bits ordenado en formato de red.
- **inet_addr:** Realiza exactamente la misma conversión que inet_aton, pero devuelve el entero de 32 bits ordenado en formato de red si la conversión tiene éxito o la constante INADDR_NONE en caso de error.
- **inet_ntoa:** Convierte un entero de 32 bits ordenado en formato de red a cadena numérica en formato IPV4.
- **inet_pton:** Convierte una dirección de red en formato de presentación (ASCII) a numérico con ordenamiento de red.
- **inet_ntop:** Realiza la conversión de formato numérico con ordenamiento de red a cadena ASCII.

2.6.4 Comunicación no orientada a la conexión

La comunicación no orientada a la conexión no ofrece la confiabilidad que se puede obtener mediante el uso del protocolo TCP, a pesar de esa desventaja, este tipo de comunicación, y sobre todo el protocolo UDP, es utilizada debido a que es de mayor velocidad y más simple que el orientado a la comunicación el cual se detallará posteriormente.

En la comunicación no orientada a la conexión no se establece un enlace directo, es decir que simplemente se comienzan a enviar datagramas.

Algunos protocolos que utilizan preferiblemente UDP en lugar de TCP son los servicios DNS, NFS y SNMP debido a que UDP ofrece un factor de carga en la transmisión mucho menor al de TCP.

2.6.5 Comunicación orientada a la conexión

Esta tipo de comunicación es completamente diferente a la no orientada a la conexión. Aquí debe establecerse una conexión entre ambas partes de la comunicación y al finalizar la transferencia de los datos realizar una finalización de la conexión.

Esta comunicación tiene una analogía a la comunicación telefónica, cuando una persona llama a otra utilizando el medio telefónico, primero espera un tono de marcado, después marca el numero de la persona con la que se desea comunicar y espera una respuesta, después de eso se ha generado la conexión y ambas personas ya pueden intercambiar información. Al momento de finalizar la conversación ambas personas se despiden, finalizan la conversación y se desconectan al colgar el teléfono. En la comunicación orientada a la conexión ocurren todos esos distintos momentos, es decir que no se fía de la capacidad de la red para el envío y la recepción de datos, más bien asegura la recepción de los datos, de manera que se convierte en una opción segura y tolerante a diversos fallos que se pueden dar en la red.

Después de analizar la funcionalidad de cada uno de los tipos de comunicación, puede hacerse una reflexión acerca de la utilidad de ellas en el presente proyecto. La comunicación no orientada a la conexión resulta de gran utilidad para la finalidad de la aplicación debido a sus características, principalmente por dos de ellas: la primera es, que el nodo que opere el programa no tiene necesariamente que establecer una conexión con cada nodo dentro de la red. La segunda es que, debido precisamente a la primera característica antes mencionada, la latencia que pueda generar la aplicación se ve disminuida por no entablar la conexión lo cual puede resultar en un mejor rendimiento para el sistema.

2.7 ANCHO DE BANDA.

El ancho de banda puede definirse como la cantidad de información que puede transmitirse a través de una conexión de red por unidad de tiempo[7]. Se suele medir en bits por segundo (bps).

2.7.1 Importancia del ancho de banda.

El ancho de banda es un factor muy importante a tener en cuenta dentro de una empresa debido a muchos factores. Uno de ellos es que el ancho de banda es un recurso limitado, tanto por el tipo de medios que se utilice para implementar la red como por el tipo de conexión que se tenga hacia redes publicas como Internet.

Otro factor a tener en cuenta es que el ancho de banda no es gratuito cuando se quiere conectar a redes de área amplia (WAN). Se hace necesario comprar el ancho de banda a un proveedor de servicios de comunicaciones. Aunado a esto, debe tenerse en cuenta que a medida que se diversifican las aplicaciones que se ejecutan en entorno de red, la demanda de ancho de banda aumenta. Un ejemplo de esto es el uso de telefonía basada en IP, lo que forzosamente hace necesario contar con un ancho de banda mayor.

2.7.2 Medición del ancho de banda.

Aunque la unidad básica del ancho de banda son los bits por segundo (bps), generalmente se utilizan múltiplos de bps para representarlo. Es por eso que comúnmente se habla de kilo bits por segundo (Kbps), que corresponde a mil bits por segundo; Mega bits por segundo (Mbps), que corresponde a un millón de bits por segundo; Giga bits por segundo (Gbps), que corresponde a mil millones de bits por segundo y Tera bits por segundo (Tbps), que corresponde a un billón de bits por segundo.

[7] En línea http://www.definicion.org/ancho-de-banda. Consultado el 19 de julio de 2006.

Muchas veces tiende a confundirse ancho de banda de una conexión con la velocidad de la misma. El ancho de banda no es exactamente lo mismo que la velocidad de conexión. Por ejemplo se podría decir que si se tiene una conexión T3 a 45Mbps y una T1 a 1,544Mbps entonces la conexión a T3 operará a una velocidad mayor que T1, sin embargo, si se transfiere una cantidad pequeña de datos cada uno de estos tipos de conexión transportará datos a aproximadamente la misma velocidad. Sin embargo si sería totalmente correcto afirmar que la conexión T3 posee un mayor ancho de banda que la conexión T1, ya que la primera puede transportar más datos por unidad de tiempo que la segunda.

2.7.3 Limitaciones del ancho de banda.

El ancho de banda puede verse limitado por los tipos de medios de la red, así como por las tecnologías LAN ó WAN que se utilicen. Las diferencias físicas en las formas en que se transmiten las señales son las que generan las limitaciones fundamentales en la capacidad que posee un medio dado para transportar información. Sin embargo, el verdadero ancho de banda de una red está determinado por la combinación de los medios físicos y las tecnologías seleccionadas para señalizar y detectar señales de red. [8]

En la siguiente tabla siguiente se muestra algunos tipos comunes de medios de networking, su ancho de banda y su distancia máxima teórica.

Medios Típicos	Ancho de Banda	Distancia Máxima Teórica
Cable coaxial de 50 ohmios (Ethernet 10BASE2, Thinnet)	10 Mbps	185 m
Cable coaxial de 50 ohmios (Ethernet 10BASE5, Thicknet)	10 Mbps	500 m
Cable de par trenzado no blindado de categoría 5 (UTP) (Ethernet 10BASE-T)	10 Mbps	100 m
Cable de par trenzado no blindado de categoría 5 (UTP) (Ethernet 100BASE-TX)	100 Mbps	100 m

[8] Programa de la academia de Networking de CISCO. CCNA 1: Conceptos básicos sobre Networking v3.1. Cisco Systems Inc. 2003

Cable de par trenzado no blindado de categoría 5 (UTP) (Ethernet 1000BASE-TX)	1000 Mbps	100 m
Fibra Óptica Multimodo (62.5/125µm) (100BASE-FX Ethernet)	100 Mbps	2000 m
Fibra Óptica Multimodo (62.5/125µm) (1000BASE-SX Ethernet)	1000 Mbps	220 m
Fibra Óptica Multimodo (50/125µm) (100BASE-FX Ethernet)	1000 Mbps	550 m
Fibra Óptica Monomodo (9/125µm) (100BASE-LX Ethernet)	1000 Mbps	5000m

Tabla 2. Tipos comunes de medios de networking y sus respectivos anchos de banda y distancia máxima teórica

La siguiente tabla muestra los servicios WAN más comunes y el ancho de banda asociado con cada servicio.

Servicio WAN	Ancho de banda
Módem	56 kbps
DSL	128 kbps a 6.1 Mbps
IDSN	128 kbps
Frame Relay	56 kbps a 44,736 Mbps
T1	1,544 Mbps
E1	2,048 Mbps
T3	44,736 Mbps
E3	34,368 Mbps
STS-1 (OC-1)	51,840 Mbps
STM-1	155,520 Mbps
STS-3 (OC-3)	155,251 Mbps
STM-3	466,560 Mbps
STS-48 (OC-48)	2,488,320 Mbps

Tabla 3. Tecnologías WAN comunes y sus respectivos anchos de banda.

2.7.4 La tasa de transferencia.

La tasa de transferencia se refiere a la medida real del ancho de banda, en un momento dado del día, usando rutas de Internet específicas, y al transmitirse un conjunto específico de datos. Desafortunadamente, por varios motivos, la tasa de transferencia a menudo es mucho menor que el ancho de banda digital máximo posible del medio utilizado. A continuación se mencionan algunos de los factores que determinan la tasa de transferencia:

- Dispositivos de Internetworking.
- Tipo de datos que se transfieren.
- Topología de la red.

- Cantidad de usuarios en la red.
- La computadora del usuario.
- La computadora servidor.
- Estado de la alimentación

2.7.5 Cálculo del tiempo de transferencia de datos

Un cálculo sencillo de la tasa de transferencia puede hacerse aplicando la fórmula: tiempo de transferencia = tamaño del archivo / ancho de banda $T = Tm \div AB$. Si se conoce el tamaño típico de un archivo para una aplicación dada, al dividir el tamaño del archivo por el ancho de banda de la red, se obtiene una estimación del tiempo más rápido en el cual se puede transferir el archivo.

Mejor descarga: $T = \frac{Tm}{AB}$

Donde:

T = Tiempo en el que debe producirse la transferencia.

Tm = Tamaño del archivo en bits.

AB = Máximo ancho de banda teórico del enlace mas lento entre el host origen y el host destino, medido en bps.

Sin embargo hay que considerar lo siguiente a partir de ese valor calculado:

- El resultado no es más que un estimado, porque el tamaño del archivo no incluye el gasto agregado por el encapsulamiento.
- Es probable que el resultado sea el tiempo de transferencia en el mejor de los casos, ya que el ancho de banda disponible casi nunca está en el máximo teórico para el tipo de red. Se puede obtener un estimado más preciso sustituyendo el ancho de banda por la tasa de transferencia en la ecuación.

Descarga Típica: $T = \frac{Tm}{P}$

Donde:

T = Tiempo en el que debe producirse la transferencia.

Tm = Tamaño del archivo en bits.

P = Tasa de transferencia real en el momento de la transferencia del archivo, medida en bps.

CAPÍTULO III: Calidad de Servicios (QoS)

3.1 Concepto de QoS

El término Calidad de Servicio o QoS (Quality of Service) tiene sus origen en el modelo OSI, y se refiere a la capacidad de una red de proveer un servicio cualificado al tráfico de una red cubriendo distintas tecnologías, incluyendo Frame Relay, ATM, Ethernet y redes 802.1, SONET y redes con enrutamiento IP[9], solventando diversos problemas que distorsionan o retrasan el tráfico de paquetes a través de la red y proporcionando ciertos parámetros que predigan el comportamiento de dichos paquetes. Se le da tratamiento a cada paquete del flujo en los nodos, para que cumplan con una serie de políticas específicas para cada flujo.

Fundamentalmente, QoS permite proveer un mejor servicio a diferentes flujos de datos. Esto se hace incrementando la prioridad de un flujo o limitando la de otro flujo. Cuanto se utilizan herramientas para administración de congestionamiento, se intenta incrementar la prioridad de un flujo encolando y dando servicio a las colas de distintas formas. El administrador de la cola utilizada para evitar la congestión incrementa la prioridad descartando los flujos de poca prioridad antes que los flujos de alta prioridad. Las políticas de prioridad brindan prioridad a los flujos limitando la salida de otros flujos. Las herramientas que proveen eficiencia en el enlace limitan los flujos largos para permitir el paso a los flujos más cortos.

Las herramientas de QoS ayudan a aliviar los problemas de congestión en las redes. Sin embargo, muchas veces existe demasiado tráfico para el ancho de banda disponible. Para estos casos, QoS actúa como una venda para frenar el problema.

Para proveer un trato preferencial a un tipo de tráfico, primero este tráfico debe ser identificado. Después, los paquetes deben o no ser marcados,

[9] Cisco Systems. *Cisco IOS Quality of Service Solutions Configuration Guide.* Cisco System Inc. USA.

dependiendo de la configuración que se quiera dar. Estas dos tareas conforman la *clasificación*. Cuando el paquete se ha identificado pero no ha sido marcado, se dice que el paquete se encuentra en un estado de clasificación básica, la clasificación pertenece solamente al dispositivo en que se encuentra, y no se pasa al siguiente router. Esto ocurre con las colas de prioridad (PQ) y las colas personalizadas (CQ). Cuando se marcan los paquetes para utilizarse en toda la red, los bits de precedencia del paquete IP pueden configurarse y manipularse. Los métodos más comunes para identificar los tipos de flujo incluyen las listas de control de acceso (ACL), ruteo basado en políticas, tasa de acceso comprometido (CAR) y reconocimiento de aplicaciones basadas en red (NBAR).

Otra de las bondades que ofrece QoS es que permite que los flujos que circulan a través de la red en la que se encuentra implementado puedan ser analizados con cierto nivel de detalle, lo que posibilitaría “predecir” los tipos de paquetes que pasan por ciertos sectores en la red, o el tráfico que se encuentra con mayor frecuencia circulando en la red.

La forma en que QoS implementa el servicio de “redes más predecibles” se da a través de:

- Ancho de banda dedicado
- Implementación de características perdidas
- Evasión y manejo de congestiones en la red
- Personalización del tráfico de la red
- Establecimiento de políticas de prioridad en la red

El tráfico en una red puede ser priorizado para adecuarse a los objetivos específicos de una empresa; esto es, por ejemplo, para asegurarse que el personal que navega a través de Internet no está reduciendo los recursos para aplicaciones críticas como comercio electrónico, aplicaciones estratégicas y servidores de información clasificada. Esto es especialmente crítico en oficinas externas en donde el ancho de banda es un recurso costoso y limitado.

Otra situación en que actúa QoS es cuando existe exceso de tráfico y congestión, que se priorizan los servicios principales y se entrega a ellos la mayor disponibilidad del ancho de banda.

En cuanto a políticas de seguridad se refiere, se pueden utilizar una gran variedad de parámetros, como lo son: usuario, grupo de trabajo, hora del día, tipo de servicio, dirección de origen, dirección de destino, puerto de origen, puerto de destino, protocolo de comunicación, entre otros.

3.2 Historia de QoS

Antes de que Internet pasara al uso comercial, la mayoría de los usuarios eran del sector académico (un reducido sector) y sus transacciones comprendían la transferencia de archivos, correo electrónico y acceso remoto. Al generalizarse el uso del Internet en la sociedad y las empresas, el número de usuarios creció exponencialmente así como el tipo de aplicaciones que transitan hoy por la Internet se ha diversificado considerablemente.

En los comienzos de Internet como una red comercial pública, no existía la necesidad de manejar o priorizar el tráfico que circulaba en dicha red, por lo que no se percibía aún el concepto de QoS. Esto hacía que toda la Internet funcionara bajo un sistema de “mejor esfuerzo” (best effort).

Por su naturaleza, el protocolo IP trabaja de una manera simple. Solo ofrece el servicio de direccionamiento. El encabezado de los paquetes incluye la dirección de quien originó el paquete, y la dirección del nodo destino. La red sólo se encarga de trasladar los paquetes a través de ella sin importar si existen los recursos disponibles para hacer la entrega con éxito. En caso de congestión, se pueden descartar los paquetes sin dar aviso al nodo transmisor de que ha descartado el paquete. IP se apoya de los protocolos de capas superiores como TCP para el reenvío de los paquetes perdidos. Por lo tanto, la responsabilidad en el control y “entendimiento” de los paquetes resida en los extremos de la red, o

sea, en los nodos que fungen como transmisor y receptor. Puede existir una certeza de que la información va hacer entregada exitosamente, sin embargo, no se puede garantizar que se haga en los tiempos adecuados. Además, existían solamente cuatro bits para tipo de servicio y tres bits de precedencia en cada mensaje, aunque no se utilizaban con mucha frecuencia. A los paquetes que viajan desde el origen al destino pueden ocurrirle muchas cosas en el camino, por lo que se vuelve un problema muchas veces la pérdida de integridad de la información. Entre los problemas que se dan están:

- **Pérdida de paquetes:** los router pueden fallar al entregar paquetes si estos llegan cuando los búferes de los routers están llenos. Algunos o todos los paquetes pueden perderse, dependiendo del estado de la red, y resulta imposible determinar qué ocurrió en el camino. Por esto, la aplicación que está recibiendo esa información debe "pedir" que la información perdida sea retransmitida, lo que causa retrasos en la transmisión.
- **Retraso:** el retraso es una propiedad impredecible: puede transcurrir mucho tiempo para que un paquete enviado alcance su destino, debido a que es retenido en largas colas esperando, o toma rutas más largas para evitar congestiones, o toma la ruta más rápida y directa.
- **Jitter:** los paquetes que van desde el origen alcanzan su destino con diferentes retrasos. Esta diferencia de retrasos es conocida como jitter y afecta especialmente la calidad de videos.
- **Entregas desordenadas:** cuando un conjunto de paquetes son enviados a través de Internet, los paquetes pueden tomar diferentes rutas, teniendo distintos retrasos para cada uno también. El resultado es que los paquetes llegan en un orden distinto al que fueron enviados originalmente. Esto requiere de protocolos especiales que reordenen los paquetes cuando llegan al destino.
- **Errores:** algunas veces los paquetes son mal diseccionados, o van erróneamente combinados, o son corrompidos mientras van en su ruta. El receptor debe detectar esto, además de solicitar a la fuente los paquetes que fueron descartados en la ruta durante el viaje.

3.3 Arquitecturas de QoS.

Se configura QoS en una red para proveer entregas seguras entre dos puntos. Existen elementos que son necesarios para la entrega segura a través de una red heterogénea:

- QoS en un solo elemento en la red, lo que incluye planificaciones de encolamiento y diseño de tráfico.
- Técnicas de señalización de QoS para la coordinación de las entregas entre dos puntos entre elementos de la red.
- Manejo de políticas de QoS y funciones de administración para controlar y administrar el tráfico entre dos puntos en una red.

No todas las técnicas son apropiadas para los routers en una red. Debido a que los routers de frontera y de backbone no realizan las mismas operaciones, las funciones QoS que realizan difieren un poco. Por ejemplo, si se quisiera configurar una red IP para tráfico de voz en tiempo real, se necesitarían las funciones tanto en el router de frontera como en el de backbone.

Las funciones de QoS que realizan los routers de frontera son: clasificación de paquetes, control de entradas y administración de configuración. Las funciones que realizan los routers de backbone son: administración de congestión y aplacamiento del mismo.

3.4 Utilidad de QoS

Los servicios que ofrece QoS son aplicados a distintos tipos de tráfico de red que circula, por ejemplo:

- Las aplicaciones multimedia requieren un enlace dedicado para su correcto flujo.
- La telefonía IP o voz sobre IP (VoIP) requiere límites estrictos en cuanto al retraso.
- Las videoconferencias requieren retrasos muy bajos

- Las emulaciones de enlaces dedicados requieren garantía en el flujo y límites estrictos de retraso.
- Las aplicaciones críticas como cirugías a larga distancia o aplicaciones con movimientos bancarios requieren enlaces dedicados.

QoS se encarga de dar los servicios necesarios que requieren los casos planteados, permitiendo tener enlaces con ancho de banda "dedicado", disminuyendo considerablemente los porcentajes de retraso y aprovechando al máximo el ancho de banda que se encuentre disponible en la red, sin la preocupación que aplicaciones menos importantes abarquen el ancho de banda y éste sea desaprovechado por las aplicaciones que realmente requieren una conexión segura y constante.

Todas las redes podrían beneficiarse de las bondades que ofrece QoS, ya sea si la red se encuentra en un negocio pequeño, una empresa o un proveedor de servicio de Internet (ISP). Los requerimientos en cuanto al tipo de QoS que utilicen los distintos usuarios pueden cambiar, dependiendo de las necesidades que éstos posean y las prioridades establecidas en sus empresas. Sin embargo, en muchas áreas, estos requerimientos son comunes para muchas de las necesidades implicadas.

Las redes de las empresas, por ejemplo, deben proveer soluciones QoS de punto a punto a través de distintas plataformas, el proveer conectividad para plataformas diferentes implica a menudo hacer diversas configuraciones de QoS para cubrir cada tecnología. A medida que la red se hace más compleja la red, que incrementan las aplicaciones críticas y el tráfico de aplicaciones multimedia, QoS prioriza el tráfico para asegurar que las aplicaciones obtengan el servicio que se les ha asignado.

Los ISP requieren aseguramiento de calidad de trabajo y escalabilidad. Por ejemplo, un ISP que ofrezca conectividad IP de "mejor esfuerzo" se encuentra en

algún momento dado transfiriendo voz, video y otros datos utilizados por aplicaciones críticas en tiempo real. QoS provee las facilidades necesarias para la escalabilidad y la calidad de trabajo que solicitan estas ISP para distinguir los distintos tipos de tráfico, permitiéndoles ofrecer servicios diferenciados a sus clientes.

En las empresas pequeñas y medianas, los administradores experimentan el rápido crecimiento de la empresa en el mundo de Internet. Las redes de estas empresas deben, además, manejar aplicaciones complejas de negocios. QoS permite a la red utilizar la conexión –de alto precio- de la WAN de la forma más eficiente posible para las aplicaciones de negocios.

3.5 Protocolos de QoS

- SMB: siglas de Server Message Block (Bloqueo de Mensajes de Servidor), es un formato de mensajes utilizado por DOS y Windows para compartir archivos, directorios y dispositivos. NetBIOS está basado en este formato, y muchos productos de red lo utilizan. SMB permite también compartir archivos entre iguales y diferentes sistemas operativos. Por ejemplo, Samba permite que UNIX y Windows compartan directorios y archivos.

- DiffServ o servicio diferenciado: es un método que trata de garantizar la calidad de servicio (QoS) en redes grandes como Internet. Diffserv tiene que tratar con flujos enormes de datos y no con flujos aislados y pequeños. Esto quiere decir que una sola negociación se hará para todos los paquetes de un ISP, por ejemplo, o una universidad o centro educativo. Los resultados de esta negociación son llamados Acuerdos de Nivel de Servicio (Service Level Agreements o SLA). Estos SLA especifican las clases de tráfico que se proveerán, las garantías a necesitar para cada clase y la cantidad de datos que se enviarán para cada clase. Una de las ventajas que proporciona es la habilitación de clasificación y políticas que controlen el tráfico en la red especificada.

- IntServ o servicios integrados: se trata de una arquitectura que especifica los elementos que garantizan la calidad de servicio en las redes. Puede ser utilizado para transmisiones de multimedia sin que existan interrupciones. IntServ especifica un sistema muy definido y depurado de QoS, que difiere muchas veces con el sistema de control utilizado por DiffServ. En IntServ cada router tiene implementada esta arquitectura, y cada aplicación que requiera algún tipo de garantías debe hacer una "reservación" individual de su espacio. Posee dos configuraciones: cantidad de flujo (Flow Specs) y reservación de recursos (RSVP). En IntServ una aplicación solicita un tipo específico de servicio a la red antes que esta envíe los datos. La solicitud se envía a través de señales, la aplicación detalla a la red el tipo de tráfico que debe tratar y a partir de esto solicita el servicio específico que necesita, facilitando a la red la configuración de parámetros como ancho de banda y tiempo en los retrasos ajustados a las necesidades de la aplicación. La aplicación enviará los datos hasta que reciba la confirmación de la red que los parámetros han sido ajustados. La red realiza también un control de admisiones, basada en la información que le proporciona la aplicación y los recursos disponibles que la propia red posea, además de asegurar los requerimientos detallados de dicha aplicación siempre y cuando el tráfico cumpla también con esos detalles. En esencia son dos las políticas que toma en cuenta IntServ: ¿qué tipo de tráfico es el que se estará manejando? Y ¿qué tipo de garantías necesita este tráfico de llegar a su destino final? Uno de los protocolos que utiliza IntServ para verificar estas políticas es el protocolo RSVP (Resource Reservation Protocol, Protocolo de Reservación de Recursos). Los dispositivos capaces de trabajar con QoS se encuentran enviando un mensaje "de ruta" cada cierto intervalo de tiempo (por lo general se envían cada 30 segundos) el cual se esparce por toda la red. Los hosts con aplicaciones que deseen obtener atención envían un mensaje de reservación el cual es rastreado por el dispositivo que emite los mensajes "de ruta", obteniendo del mensaje de reservación el detalle del

tipo de servicio que se está solicitando. Si el dispositivo es capaz de manejar el tráfico de acuerdo a los parámetros solicitados, realiza el trabajo encomendado asegurando las condiciones acordadas; de no poseer las capacidades necesarias para transportar de manera eficiente el tráfico con las condiciones habladas, se envía un mensaje al solicitante que el trabajo no se realizará, y la aplicación que solicitó el servicio continuará escuchando mensajes de dispositivos que estén ofreciendo el servicio y se realizará de nuevo el proceso de envío de especificaciones y detalles del tipo de tráfico a tratar. Sin embargo, existen ciertas desventajas respecto a IntServ. Una de ellas es que se deben de configurar múltiples estados en cada dispositivo que ofrezca este servicio, por lo que su escenario óptimo de trabajo se reduce a redes a pequeña escala, y, mientras la red comience a requerir mayor escalabilidad, se vuelve difícil llevar el control de todos los mensajes de reservación que se realizan. Esta es la causa principal por lo que IntServ no es demasiado recomendado ni implementado.

- MPLS: es uno de los desarrollos más importantes de las tecnologías de Internet, que permite agregar un número de capacidades esenciales a las redes IP que funcionan bajo el principio del “mejor esfuerzo” como conexiones con ATM y posibilidad de VPN. MPLS se considera que opera en una capa del modelo OSI que se encuentra entre las capas 2 y 3, y es llamado usualmente “protocolo de capa 2.5”. Se diseñó para proveer un servicio de transporte unificado de datos tanto para clientes basados en circuitos y clientes de conmutación de paquetes que proveen modelos de servicios de datagrama. Utiliza mecanismos que emulan algunas propiedades de redes de conmutación de circuitos sobre redes de conmutación de paquetes.

CAPITULO IV: Servicios Diferenciados

Actualmente existen principalmente 2 estrategias para dar un trato preferente al tráfico que circula en una red: Una de ellas esta orientada a la reservación y asignación de recursos basándose en flujos de tráfico. Alternativamente a este modelo se encuentra otra estrategia que se orienta a la priorización de un determinado tráfico clasificándolo en base a tipos. En este tipo de arquitectura los flujos de datos individuales se van agrupando en grandes agregados de tráfico de acuerdo a la "clase de servicio" a la que pertenezcan, y dependiendo de esa clase de servicio recibirán un distinto trato en los diferentes elementos de la red.

4.1 Definición de Servicios Diferenciados.

El modelo de Servicios Diferenciados (Conocido como DiffServ en idioma ingles) se basa en tráfico sin reservación. Este modelo clasifica los paquetes en un número pequeño de tipos de servicios y utiliza mecanismos de prioridad, para proporcionar una calidad de servicio adecuada al tráfico.

El objetivo principal de este mecanismo es asignar el ancho de banda de Internet a diferentes usuarios en una forma controlada durante periodos de congestión. Este mecanismo se aplica igualmente a aplicaciones tradicionales basadas sobre TCP, tales como transferencia de archivos, accesos a bases de datos o Servidores de Web, y nuevas clases de aplicaciones tales como audio o video en tiempo real.

Los Servicios Diferenciados pueden proveer a los usuarios, una expectativa predecible del servicio que la Internet le proveerá en tiempos de congestión, y permite que diferentes usuarios obtengan diferentes niveles de servicio de la red. [10]

[10] Raúl Rivera Rodríguez. *Protocolos y Arquitecturas para QoS.* PROPUESTA DEL GRUPO DE TRABAJO QCUDI. Ensenada, Baja California. Estados Unidos de América. 2000

4.2 Definición del campo de servicios diferenciados (DS)

Como se explicó anteriormente, el campo DS utiliza 6 bits del campo de tipo de servicio (ToS) de IPv4 y el Clase de tráfico (Traffic class) de IPV6, los últimos 2 bits son ignorados por los nodos que tienen implementada una arquitectura DiffServ. La estructura del campo se puede ilustrar en la siguiente figura:

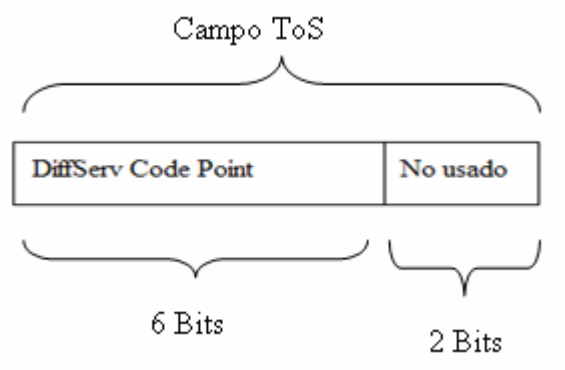

Figura 7. Definición del campo DS.

El Per Hop Behavior, mencionado más adelante, estará definido de acuerdo al valor que tomen esos seis bits del DiffServ Code Point, y será vital para determinar el tipo de tratamiento que se realizará en cada uno de los paquetes IP que sean recibidos.

El campo DS será modificado por cada nodo de acceso de la red para que los nodos más internos sepan que tratamiento debe dársele a un determinado paquete en el momento de su reenvío o su circulación en el interior de la red de acuerdo a políticas establecidas.

Version	Lon.Cab.	DS	Longitud total			
Identificación			X	D F	M F	Desplazamiento fragmento
Tiempo de vida		Protocolo	Checksum			
Dirección de origen						
Dirección de destino						
Opciones						

Figura 8. Cabecera IPv4 con DiffServ (RFC2474, 12/1998)

Los host o los routers que envían tráfico a una red con arquitectura diffserv marcan cada paquete transmitido con el valor DSCP. Los routers de una red diffserv utilizan DSCP para clasificar paquetes y para aplicar un comportamiento de cola específico basado en los resultados de la clasificación. El tráfico de varios flujos con requisitos de QoS parecidos se marca con el mismo DSCP, al agregar el flujo a una cola común o al programar el comportamiento.

4.3 Arquitectura de Servicios Diferenciados

En la arquitectura definida por Diffserv aparecen 2 tipos de nodos: nodos extremos DS de entrada y salida, y los nodos DS internos. Este conjunto de nodos definen un dominio Diffserv y presenta un tipo de políticas y grupos de comportamiento por salto (PHB), el cual determina el tratamiento que recibirán los paquetes dentro de la red.[11]

- Nodos extremos DS: Los nodos extremos DS realizan diferentes funciones como el acondicionamiento de tráfico entre los dominios Diffserv interconectados. En estos nodos se clasifican y establecen las condiciones de ingreso de los flujos de tráfico en función de diversos parámetros, tales como: dirección IP y puerto (origen y destino), protocolo de transporte y DSCP. Este clasificador se conoce como MF (Multi-Field Classifier). Una vez que los paquetes han sido marcados adecuadamente, los nodos internos deberán seleccionar el PHB definido para cada flujo de datos. Los nodos DS de entrada son los responsables de asegurar que el tráfico de entrada cumple los requisitos de algún TCA (Traffic Conditioning Agreement), entre los dominios interconectados. Los nodos DS de salida deberán realizar funciones de acondicionamiento de tráfico o TC (Traffic Conformation) sobre el tráfico transferido a otros dominios DS conectados.

[11] Escribano Salazar, Jorge, et al. *Diffserv como solución a la provisión de QoS en Internet.* Departamento de Ingeniería Telemática Universidad Carlos III de Madrid. España.

- Nodos internos DS: Los nodos internos, son los que se encargan de realizar las funciones de reenvío de paquetes de acuerdo a las políticas de calidad de servicio que se tengan especificadas. Los nodos internos realizan funciones limitadas de TC, tales como remarcado de DSCP. Estos nodos solo se conectan a otros nodos internos o a nodos externos de su propio dominio. A diferencia de los nodos externos, para la selección del PHB solo se tendrá en cuenta el campo DSCP, conocido como clasificador BA (Behavior Aggregate Classifier).

Los flujos de tráfico que ingresan a un dominio por los nodos de frontera del dominio, son sometidos a un proceso de acondicionamiento, y se les incorpora a alguna clase de tráfico definida en el dominio de acuerdo a su etiqueta en el campo DSCP.

Por el conocimiento que en cada nodo de frontera se tiene sobre la existencia, el paso y la subsistencia de cada flujo que ingresa por ese nodo al dominio, se dice que en esos nodos se lleva un control de estado por flujo. En contraste, en los nodos interiores se lleva un control de estado por clase, lo que hace mucho más fácil el manejo del tráfico para los nodos interiores [12].

Un servicio en DS se crea con la combinación de comportamientos por dominio, que incluyen los PHB y los acondicionamientos en los nodos extremos. La diferenciación de servicios se extiende a través de un dominio de red estableciendo un SLA (Service Level Agreement) en el que se especifica la clasificación y reglas de etiquetado para cada paquete. También se especifican los perfiles de tráfico y las acciones a tomar sobre los flujos de tráfico que se encuentran fuera de los perfiles establecidos.

[12] Mateos Papis, Alfredo Piero, et al. *Control de admisión en redes con arquitectura de diferenciación de servicios. Algunas tendencias*. Instituto Tecnológico Autónomo de México. México.

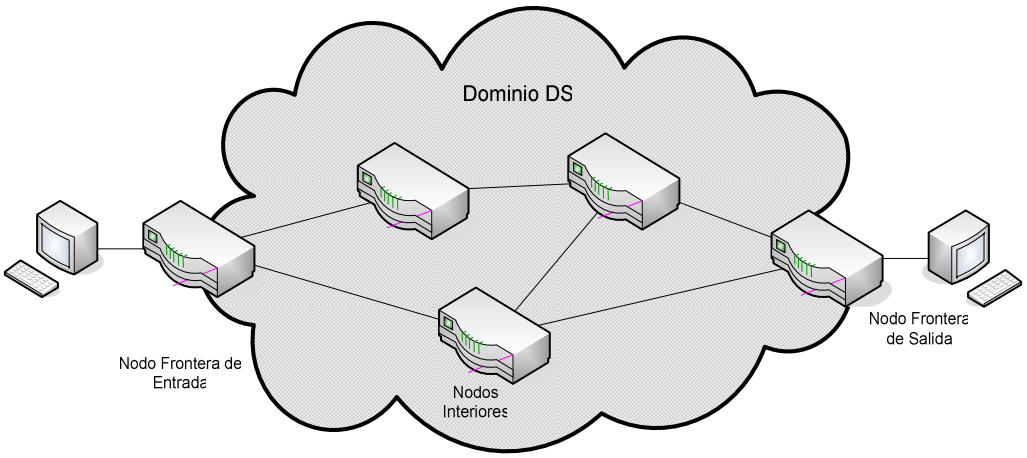

Figura 9. Dominio DS

4.3.1 Per Hop Behavior.

Cada uno de las 64 combinaciones que se pueden formar con el campo DS mapea un PHB determinado. De momento se han dividido en 3 grupos:

CodePoints	Valores	Uso
xxxyy0	32	Estándar
xxxx11	16	Local / Experimental
xxxx01	16	Reservado

Tabla 4. Grupos de valores del campo DS

En el grupo estándar los tres primeros bits (ccc) corresponden a la clase de servicios, y los dos siguientes (dd) corresponden al marcado intraclase (mayor o menor precedencia de descarte).

CodePoint	Uso
111110	Reservado (routing y control)
111100	Reservado (routing y control)
111010	Reservado (routing y control)
111000	Reservado (routing y control)
110110	Reservado (routing y control)
110100	Reservado (routing y control)
110010	Reservado (routing y control)
110000	Reservado (routing y control)
101110	Expedited (Premium)

CodePoint	Uso
011110	Assured Clase 3 Preced. Alta
011100	Assured Clase 3 Preced. Media
011010	Assured Clase 3 Preced. Baja
011000	Configurable por el usuario
010110	Assured Clase 2 Preced. Alta
010100	Assured Clase 2 Preced. Media
010010	Assured Clase 2 Preced. Baja
010000	Configurable por el usuario
001110	Assured Clase 1 Preced. Alta

101100	Configurable por el usuario	001100	Assured Clase 1 Preced. Media
101010	Configurable por el usuario	001010	Assured Clase 1 Preced. Baja
101000	Configurable por el usuario	001000	Configurable por el usuario
100110	Assured Clase 4 Preced. Alta	000110	Configurable por el usuario
100100	Assured Clase 4 Preced. Media	000100	Configurable por el usuario
100010	Assured Clase 4 Preced. Baja	000010	Configurable por el usuario
100000	Configurable por el usuario	000000	Best Effort (default)

Tabla 5. Valores de CodePoint Estándar en el campo DSCP

Actualmente existen 4 PHB especificados para ser usados dentro de una red de servicios diferenciados:

- Comportamiento por omisión (Default Behavior)
- Selector de clase
- Tránsito asegurado (Assured forwarding)
- Tránsito expedito (Expedited forwarding)

CAPITULO V: Mecanismos de regulación de tráfico

5.1 Control de congestión

Existen varios niveles en los cuales se puede proveer de calidad de servicio en una red IP. Uno de ellos es contar con una estrategia para el manejo de los paquetes en los períodos en los que se presenten congestiones, o estrategias que eviten que la red alcance este estado, descartando paquetes a medida que estos ingresan a la red.

Una congestión es una situación en la que el rendimiento de la red, o una parte de ella, se degrada debido a la presencia de tráfico excesivo. Una de las posibles consecuencias del control de congestión es ejercer control de flujo sobre él o los nodos que están produciendo la congestión.

El manejo de congestión es un término general usado para referirse a los distintos tipos de estrategias que se utilizan para manejar situaciones donde la demanda de ancho de banda requerida por las aplicaciones excede el ancho de banda disponible de la red, controlando la inyección de tráfico a la red, para que ciertos flujos importantes tengan prioridad sobre otros.

5.1.1 Principios generales del control de congestión

Para el control de la congestión caben dos planteamientos:

- Diseñar las cosas desde el principio para que la congestión no pueda llegar a ocurrir.
- Tomar medidas que permitan detectar la congestión y adoptar medidas correctoras en su caso.

La primera técnica es más segura, pero puede provocar ineficiencias si se aplican las limitaciones con demasiada severidad. La segunda permite aprovechar mejor la red, pero en caso de congestión puede ser difícil controlar la situación.

Entre los parámetros que permiten detectar la presencia de congestión, a nivel de red, se encuentran:

- Porcentaje de paquetes descartados.
- Longitud media de las colas en las interfases de los routers.

En cambio, a nivel de transporte se tiene:

- Retardo medio por TPDU (Transport Protocol Data Unit).
- Desviación media del retardo por TPDU (jitter).
- Número de TPDUs que se pierden o llegan al timeout y se retransmiten (se supone que esto no se debe a errores).

Para informar sobre situaciones de congestión el receptor puede utilizar paquetes de alerta y el emisor enviar paquetes de sondeo para averiguar el estado de la red. Para resolver la congestión solo hay dos posibles medidas:

- Reducir el tráfico solicitando al emisor que pare de transmitir, o que busque rutas alternativas.
- Aumentar la capacidad.

5.2 Algoritmos de control de congestión

5.2.1 Algoritmos de manejo de congestión basados en colas

Existen diferentes mecanismos ó estrategias de encolamiento basados en algoritmos que van desde los mas simples hasta los sumamente complejos. Algunas de estas estrategias se describen a continuación

5.2.1.1 Firs-In-First-Out, FIFO

Es el tipo más simple de encolamiento, y funciona de la siguiente forma: el primer paquete en entrar a la interfaz, es el primero en salir. Esta estrategia es útil para interfaces de alta velocidad, sin embargo, no es una de las mejores cuando se trabaja con interfaces de baja velocidad.

FIFO es capaz de manejar cantidades limitadas de ráfagas de datos. Si llegan más paquetes cuando la cola está llena, éstos son descartados. No tiene mecanismos de diferenciación de paquetes.

5.2.1.2 Fair Queuing, FQ.

Conocido también como WFQ (Weighted Fair Queuing), es un método automatizado que provee una justa asignación de ancho de banda para todo el tráfico de la red, utilizado habitualmente para enlaces de velocidades menores a 2048 Mbps.

WFQ clasifica el tráfico en flujos. Para lograrlo hace uso de una combinación de parámetros, tales como, dirección IP fuente, dirección IP destino, puerto de origen, puerto destino, etc. Una vez identificados estos flujos, el router determina cuáles son de uso intensivo o sensibles a los retrasos, priorizándolos y asegurando que los flujos de alto volumen (aquellos que consumen un elevado ancho de banda) sean empujados al final de la cola, y los volúmenes bajos, sensibles al retardo, sean empujados al principio de la cola.

WFQ es apropiado en situaciones donde se desea proveer un tiempo de respuesta consistente ante usuarios que generen altas y bajas cargas en la red, ya que WFQ se adapta a las condiciones cambiantes del tráfico en ésta. Sin embargo, la carga que significa para el procesador en los equipos de enrutamiento, hace de esta metodología poco escalable, al requerir recursos adicionales en la clasificación y manipulación dinámica de las colas. [13]

[13] Álvarez Moraga, Sebastián y González Valenzuela, Agustín. *Estudio y configuración de calidad de servicio para protocolos ipv4 e ipv6 en una red de fibra óptica wdm*. Revista de la Facultad de Ingeniería, Universidad. Tarapacá, vol. 13 Nº 3, 2005. En línea: www.scielo.cl/pdf/rfacing/v13n3/art15.pdf

5.2.1.3 Priority Queuing, PQ.

El Encolamiento de Prioridad consiste en un conjunto de colas, clasificadas desde alta a baja prioridad. Cuando se recibe un paquete en una interface es asignado a una de estas colas, las cuales son atendidas en un estricto orden de prioridad.

Las colas de mayor prioridad son siempre atendidas primero, luego la siguiente de menor prioridad y así sucesivamente. Si una cola de menor prioridad está siendo atendida, y un paquete ingresa a una cola de mayor prioridad, ésta es atendida inmediatamente.

Esta estrategia se utiliza en condiciones donde existe un tráfico importante, pero posee la desventaja que puede darse el caso en el cual, las colas de menor prioridad sean totalmente desatendidas.

5.2.1.4 Custom Queuing, CQ.

El Encolamiento Personalizado permite priorizar el tráfico que circula en la red evitando que las colas de menor prioridad queden sin atenderse completamente. Para lograr esto, se especifica el número de paquetes o bytes que deben ser atendidos para cada cola. Se pueden crear hasta 16 colas para categorizar el tráfico, donde cada cola es atendida siguiendo el método de planificación Round-Robin[14].

CQ ofrece un mecanismo más refinado de encolamiento, pero no asegura una prioridad absoluta como PQ. Este tipo de colas se utiliza para proveer ancho de banda garantizado a tráficos particulares en un punto de

[14] Round - Robin es un método para seleccionar todos los elementos en un grupo de manera equitativa y en un orden racional, normalmente comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento. Cada elemento del grupo se atiende durante a un pequeño periodo de tiempo, y luego se da la oportunidad a otro elemento para que sea atendido y así sucesivamente

posible congestión, asegurando para este tráfico una porción fija del ancho de banda y permitiendo al resto del tráfico utilizar los recursos disponibles.[15]

5.2.1.5 Class-Based WFQ.

Surgió debido a las limitaciones de escalabilidad del algoritmo de WFQ, ya que la implementación de éste se ve afectada a medida que el tráfico por enlace aumenta. Puede darse el caso que colapse debido a la cantidad excesivamente grande de flujos que debe analizar. CBWFQ fue desarrollada para evitar estas limitaciones, tomando como base el algoritmo de WFQ y expandiéndolo, permitiendo la creación de clases definidas por el usuario, permitiendo un mayor control sobre las colas de tráfico y asignación del ancho de banda.

Algunas veces es necesario garantizar una determinada tasa de transmisión para cierto tipo de tráfico, lo cual no es posible mediante WFQ, pero sí con CBWFQ. Las clases que son posibles implementar con CBWFQ pueden ser determinadas según listas de control de acceso (ACL), valor DSCP, o interfaz de ingreso. Cada clase posee una cola separada, y todos los paquetes que cumplen el criterio definido para una clase en particular son asignados a dicha cola. Una vez que se establecen los criterios para las clases, es posible determinar cómo los paquetes pertenecientes a dicha clase serán manejados.

Si una clase no utiliza su cuota de ancho de banda, otras pueden hacerlo. Se pueden configurar específicamente el ancho de banda y límite de paquetes máximos (o profundidad de cola) para cada clase. El peso asignado a la cola de la clase es determinado mediante el ancho de banda asignado a dicha clase.

[15] Álvarez Moraga, Sebastián y González Valenzuela, Agustín. Op Cit.

5.2.1.6 Low Latency Queuing, LLQ.

El Encolamiento de Baja Latencia es una mezcla entre Priority Queuing y Class-Based Weighted-Fair Queuing. Es actualmente el método de encolamiento recomendado para Voz sobre IP (VoIP) y Telefonía IP. Asimismo se adapta muy bien al tráfico de videoconferencias. LLQ consta de colas de prioridad personalizadas, basadas en clases de tráfico, en conjunto con una cola de prioridad, la cual tiene preferencia absoluta sobre las otras colas. Si existe tráfico en la cola de prioridad, ésta es atendida antes que las otras colas de prioridad personalizadas. Si la cola de prioridad no está encolando paquetes, se procede a atender las otras colas según su prioridad. Debido a este comportamiento es necesario configurar un ancho de banda límite reservado para la cola de prioridad, evitando que las colas de menor prioridad queden sin ser atendidas. La cola de prioridad que posee LLQ provee de un máximo retardo garantizado para los paquetes entrantes en esta cola, el cual es calculado como el tamaño del MTU dividido por la velocidad de enlace. [16]

Cada mecanismo de encolamiento tiene sus ventajas y desventajas, así como escenarios dónde es más recomendable aplicar ese mecanismo en particular. La elección del mecanismo de encolamiento a utilizar depende de lo que se quiera lograr.

5.2.2 Traffic Policing (políticas de tráfico).

Traffic policing permite controlar la tasa máxima del tráfico que se envía o recibe en una interfaz, además de clasificar la red en varios niveles de prioridad o clases de servicios.

Uno de los sistemas más utilizados para establecer perfiles de tráfico es el algoritmo de Leaky Bucket. El host puede enviar ráfagas que son almacenadas en un buffer de la interfaz, la que envía a la red un flujo constante

[16] Álvarez Moraga, Sebastián y González Valenzuela, Agustín. Op Cit.

de salida. Si la ráfaga es de tal intensidad o duración que el buffer se llena, los paquetes excedentes son descartados, o bien son enviados a la red con una marca especial que les identifica como de segunda clase. Estos paquetes serán los primeros candidatos a descarte en caso de congestión. Para definir el algoritmo se utilizan dos parámetros, el flujo r con que sale el flujo a la red, y la capacidad del buffer C.

El algoritmo traffic policing se configura usualmente en interfaces que están en el borde de una red, esto con la finalidad de limitar la tasa de tráfico que entra o que sale de la red. Una configuración común hecha con traffic policing es permitir la transmisión de paquetes en la categoría de conformación, y los paquetes en la categoría de exceso se envían con una prioridad menor o son descartados. En todo caso, las configuraciones las define el usuario de acuerdo a sus necesidades.

5.2.2.1 Beneficios de traffic policing.

- Manejo del ancho de banda limitando la tasa: traffic policing permite controlar la cantidad de tráfico enviado o recibido en una interfaz.
- Esto se configura en las interfaces que se encuentran al borde de la red, para limitar el tráfico entrante y/o saliente. El tráfico permitido, de acuerdo a los parámetros configurados, se envía, mientras que el tráfico que excede dichos parámetros es descartado o se envía con una prioridad diferente.
- Marcado de paquetes: el marcado de paquetes permite dividir la red en múltiples niveles de prioridad o clases de servicio. Esto da la posibilidad de, por ejemplo, calcular la probabilidad que un paquete, de acuerdo a la precedencia especificada, sea enviado o descartado. Además permite establecer prioridades a los paquetes y darles el tratamiento adecuado según las reglas configuradas.

5.2.2.2 Restricciones.

El algoritmo de traffic policing no puede ser configurado en interfaces fast ethernet, túnel (para VPN) y PRI.

5.2.3 Traffic Shaping (Modelado de tráfico)

El algoritmo de modelado de tráfico o traffic shaping está basado en el control de los paquetes que viajan por la red en el nodo que envía el paquete, de forma que permite enviar ciertos paquetes de forma "estratégica" hacia la red lo cual pude generar una especie de manejo de ancho de banda.

En general la función del algoritmo traffic shaper es recibir los paquetes de forma desordenada para luego enviarlos de una forma ordenada y con un retardo de tiempo que permita que el paquete sea transmitido a un ancho de banda especificado.

La aplicación de este algoritmo es básicamente sencilla, el traffic shaper toma los paquetes de la cola y los regresa a la red con diferencias de tiempo entre paquetes sucesivos, de manera que se define como R como la tasa a la que se desea dicho paquete.

Los principales objetivos de un modelo traffic shaper son dos:

- Que la tasa promedio sea lo más cercana posible a R.
- Que el tiempo de retraso que se genere en el paquete sea el menor posible.

Para controlar la tasa de la transmisión se utilizan tiempos "muertos" del procesador los cuales se denominarán de ahora en adelante sleeps. En el caso del algoritmo si se analiza el proceso en una línea de tiempo se vería similar a la siguiente:

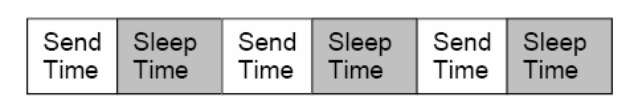

Figura 10: Ciclo de envío y tiempo muerto[17]

El tiempo de envío representa el tiempo que toma enviar un paquete y envuelve algunas otras tareas que debe realizar el procesador para finalizar esa acción. En este caso a la constante de tiempo de envío se denominara T, mientras que al tamaño del paquete se le denominara S, el tiempo para el sleep se podría calcular así:

$$Sleep = \max\left(\frac{S}{R} - T, 0\right)$$

De manera que la representación más sencilla en pseudocódigo para este algoritmo se detalla a continuación:

```
hecho = falso
TiempoEnvío = 0
inicio = getTime()
mientras (!hecho){
    paquete = queue.getPacket()
    socket.send(paquete)
    tamaño = paquete.getSize()
    sleepTime = tamaño/tasa - TiempoEnvío
    si sleepTime > 0 entonces
            sleep(sleepTime)
    fin si
    final = getTime()
    TiempoEnvío = final - inicio - sleepTime;
    si (TiempoEnvío < 0)
            TiempoEnvío = 0
    fin si
    inicio = final
fin mientras
```

La duración es medida una vez por ciclo y el tiempo de envío se determina restándole el sleepTime al ciclo de duración, por esa razón se obtienen los tiempos inicio y final, generando de esa forma el cálculo del

[17] Ming Chiu, Dah; Kadansky, Miriam; Provino, Joe; Wesley, Joseph. *Experiences in programming a traffic shaper*. Sun Microsystems Inc. Septiembre, 1999.

tiempo aproximado necesario para alcanzar la tasa de transferencia deseada en el reenvío del paquete.

Por supuesto este algoritmo puede también generar algunos inconvenientes excediendo los tiempos muertos del procesador lo cual es denominado como “Oversleeping”.[18]

En general el oversleeping es la diferencia entre el tiempo que el procesador requiere para llevar a cabo el tiempo muerto solicitado (tiempo muerto teórico) y el tiempo en que ejecuta el tiempo muerto (tiempo muerto real). En este caso el oversleeping puede variar de acuerdo a distintos patrones, los más importantes: tamaño del paquete, la tasa de transferencia y en muchos casos hasta la velocidad del procesador en donde se ejecute un código de traffic shaping.

$$Oversleeping = Tiemporeal - TiempoTeórico$$

Una de las formas en las que se puede corregir el problema del oversleeping es efectuando un envío de múltiples paquetes entre cada tiempo muerto del procesador. Para ilustrar mejor este planteamiento la siguiente figura muestra el ciclo de envío:

Figura 11: El ciclo de envío de múltiples paquetes para resolver el problema de oversleeping[19]

Aunque en la ilustración solamente aparecen tres paquetes, cabe recalcar que la cantidad de paquetes enviados entre los tiempos muertos del procesador pueden ser n paquetes.

[18] Ming Chiu, Dah; Kadansky, Miriam; Provino, Joe; Wesley, Joseph. Op Cit..
[19] Ming Chiu, Dah; Kadansky, Miriam; Provino, Joe; Wesley, Joseph Op. Cit.

De manera que si se puede enviar más de un paquete entre dos tiempos muertos de procesador, se puede descartar el tiempo de envío de cada paquete, ya que no tendrá mayor importancia para cada paquete.

Si al algoritmo anterior se le intentase corregir el oversleeping tendría la siguiente forma:

```
finalizado = falso
inicio = getTime()
tiempo_perdido = 0
tamaño = 0
mientras (!finalizado){
paquete = queue.getPacket()
socket.send(paquete)
tamaño += paquete.getSize()
sleepTime = tamaño/tasa
si tiempo_perdido > 0 entonces
        si tiempo_perdido > sleepTime
                tiempo_perdido = tiempo_perdido – sleepTime
                sleepTime = 0
        sino
                sleepTime = sleepTime – tiempo_perdido
                tiempo_perdido = 0
        fin si
        fin si
        si sleepTime > 0
        sleep(sleepTime)
        final = getTime()
        tiempo_perdido = final - inicio - sleepTime;
        si (tiempo_perdido < 0)
                tiempo_perdido = 0
        fin si
        inicio = final
fin si
fin mientras
```

De esa manera se evitara generar un tiempo muerto de procesador por cada paquete que sea enviado.

Esa es una de las formas en las que se puede generar administración de ancho de banda en mediante los modelos básicos de algoritmos diferenciales de tiempo para obtener un modelado de tráfico.

5.2.3.1 Algoritmo Token Bucket

El algoritmo token bucket es una versión mejorada del algoritmo leaky bucket. Este algoritmo compensa al host que alterna intervalos de tráfico con otros de inactividad frente al que esta siempre transmitiendo.

Al manejar los paquetes que entran o salen de la máquina encargada de administrar dichos paquetes, se producen ciertos tipos de congestiones o ráfagas de envíos que deben "ordenarse" antes de permitirles avanzar en la red. El algoritmo de token bucket es utilizado para este fin, ya que permite controlar la cantidad de datos que se dejan pasar a la red, permitiendo que las ráfagas de datos puedan avanzar sin problemas.

Token bucket es un mecanismo de control que marca los tiempos en que se puede transmitir tráfico, esto basado en la presencia de ciertas señales que son las que permiten avanzar o detener el paso en la red. Este algoritmo contiene señales que pueden representar unidades de bytes. Se especifica cuántas señales se necesitan para transmitir un determinado número de bytes y permitir que el flujo de datos continúe su camino. Si no existen señales en el depósito de señales no se transmiten datos. De esta forma, se puede controlar la cantidad máxima de datos que pueden estar fluyendo en ráfaga y así controlar el tráfico en la red, siempre y cuando se hayan hecho las configuraciones apropiadas.

A grandes rasgos el algoritmo conlleva los siguientes pasos[20]:

[20] En línea. Wikipedia. *Token Bucket*. http://en.wikipedia.org/wiki/Token_bucket

- Se agrega una señal o símbolo al depósito de paquetes cada 1/*r* segundos, donde *r* representa la tasa promedio a la que los paquetes son enviados.
- El depósito puede contener un máximo de *b* símbolos. Si un símbolo llega cuando el depósito está lleno, será descartado.
- Cuando llegue un paquete de n bytes de tamaño, se quitarán n símbolos del depósito y se enviarán los paquetes a la red.
- Si hay disponibles menos de n símbolos, no se removerá nada del depósito, y el paquete que ha llegado se considerará que no pertenece al resto de paquetes.

Los paquetes que no pertenecen al grupo de los demás paquetes que han sido tratados, se pueden procesar de distintas formas:

- Pueden descartarse.
- Pueden encolarse y guardarse para transmisiones posteriores cuando se hayan acumulado suficientes símbolos y se tenga que vaciar el depósito.
- Pueden transmitirse, pero marcados como “no conformantes” del grupo que se ha enviado, con el riesgo que puede ser descartado en el camino si la red está saturada.

5.2.4 Rate limiting.

Rate limiting, o limitación de tasa, se utiliza para controlar la tasa o el ancho de banda del tráfico enviado o recibido en una interfaz de red. Si la tasa de tráfico que está pasando por la interfaz de red es menor o igual a la especificada, dicho tráfico se envía, pero si sobrepasa la tasa especificada, los paquetes en ese tráfico se descartan o se demoran intencionalmente, a fin de que se ajuste el tráfico a la tasa máxima fijada. La limitación de tasa es uno de los componentes de la clase de servicio de control de políticas. Existen dos tipos de clase de servicio en el control de políticas, y cada una utiliza una forma

distinta de limitar el tráfico, estas formas son: límite de tasa basado en prioridades y límite de tasa basado en el rol.

5.2.4.1 Límite de tasa basado en prioridades.

El límite de tasa basado en prioridades está asociado con uno o más de los algoritmos de límite de tasa descritos en 802.1p. El ancho de banda especificado se escribe directamente al puerto que está ligado a la transferencia del tráfico a menos que dicho puerto esté en la lista de exclusión de puertos.

Cuando se implementa el límite de tráfico basado en prioridades, el total del ancho de banda de todo el tráfico perteneciente al puerto que coincide con la prioridad asociada con el ancho de banda límite especificado no puede sobrepasar dicha barrera. Si la tasa excede el límite configurado, algunos paquetes son descartados hasta que el tráfico regrese al límite impuesto. Se impondrá este límite hasta que este se modifique en el controlador de políticas. El límite de tasa no solo puede ser asociado a un puerto, sino también a una dirección de red o a ambos (puerto y dirección de red), así como también a un tipo de protocolo específico.

La configuración del límite de tasa basado en prioridades incluye:

- Límite de tasa: fijación de la tasa de transmisión máxima a la que puede pasar el tráfico.
- Dirección: la dirección o sentido al que se le aplicará el límite (entrante o saliente). Si es para un mismo puerto o dirección, se configuran dos límites, uno para tráfico entrante y otro para tráfico saliente.
- Prioridad: prioridad o importancia que se le asigna a los paquetes en el tráfico.
- Precedencia: la regla o conjunto de reglas que definen las tasas que se aplicarán primero o después.

5.2.4.2 Límite de tasa basado en el rol.

El límite de tasa basado en el rol está íntimamente ligado a las funciones y reglas establecidas. Cuando este tipo de límite se implementa, el tráfico en el puerto en que se ha configurado se limita a la tasa especificada. Al igual que con el límite de tasa basado en prioridades, si la tasa excede el límite impuesto, los paquetes son descartados o, para este caso, repriorizados hasta que la tasa vuelve a su límite o debajo de él.

Este tipo de límite provee la capacidad de crear grupos de puertos o direcciones para limitar, además de la dirección o el sentido que se limitará a dichos grupos.

La configuración del límite de tasa basado en el rol incluye:

- Límite de tasa: la tasa máxima de transmisión a la que los paquetes entran o salen de una interfaz.
- Evento: acción a tomar cuando se excede el límite de tráfico impuesto (descartar los paquetes o asignarles una nueva priorización).

En resumen, la limitación de la tasa se produce mediante políticas (descartando paquetes excesivos), encolamiento (retrasando paquetes en el tráfico) o control de la congestión (se manipula el mecanismo de congestión del protocolo). La aplicación de políticas y encolamiento pueden realizarse sobre cualquier protocolo de red, pero el control de congestión solo puede aplicarse a protocolos con esta misma característica, como el protocolo de control de transmisión (TCP).

5.2.5 Evasión de Congestión

Las metodologías de evasión de congestión se basan en la manera que los protocolos operan, con el fin de no llegar a la congestión de la red.

Las técnicas de RED (Random Early Detection) y WRED (Weighted Random Early Detection) evitan el efecto conocido como Sincronización Global. Cuando múltiples conexiones TCP operan sobre un enlace común, todas ellas incrementaran el tamaño de su ventana deslizante a medida que el tráfico llega sin problemas. Este aumento gradual consume el ancho de banda del enlace hasta congestionarlo. En este punto las conexiones TCP experimentan errores de transmisión, lo que hace que disminuyan su tamaño de ventana simultáneamente. Esto conlleva a una sincronización global, donde todos los flujos comienzan a incrementar su tasa de transmisión nuevamente para llegar a otro estado de congestión. Este ciclo es repetitivo, creando picos y valles en la utilización del ancho de banda del enlace. Es debido a este comportamiento que no se utiliza los máximos recursos de la red.

Los métodos de evasión de congestión tratan con este tipo de situación, descartando paquetes de forma aleatoria. RED fuerza a que el flujo reduzca el tamaño de su ventana de transmisión, disminuyendo la cantidad de información enviada. A medida que se alcanza el estado de congestión en la red, más paquetes entrantes son descartados con el fin de no llegar al punto de congestión en el enlace.

Lo que limita a estas técnicas de evasión de congestión es que sólo sirve para tráfico basado en TCP, ya que otros protocolos no utilizan el concepto de ventana deslizante.

CAPÍTULO VI: Diseño de Odisea Bandwidth Manager.

6.1 Requerimientos de software

El software planteado en este proyecto como solución al problema de administración de ancho de banda, cuenta con ciertos requerimientos de software, los cuales son detallados a continuación:

- Sistema Operativo: Microsoft Windows XP (O versiones superiores orientadas al trabajo con redes como Windows 2003 Server Edition, entre otros)
- Entorno .NET framework 2.0
- Gestor de Bases de Datos MySQL: Utilizado tanto para la creación del software como para su normal funcionamiento. Este gestor será empleado para almacenar la información de los usuarios del sistema así como las reglas de administración del ancho de banda.

6.2 Requerimientos de Hardware.

Los requerimientos para el funcionamiento óptimo del sistema son los siguientes:

- Procesador: 3.2 GHz,
- Memoria RAM: 1 GB.
- Espacio en disco: 500 MB.
- Interfaces de Red; 2 Dispositivos Fast Ethernet (uno conectado a la red de alta velocidad y otro conectado a la red de baja velocidad).

6.3 Diseño de pantallas

Debido a que el software contará con una interfaz gráfica, se hace necesario diseñar los distintos formularios que se tendrán en la aplicación. La función principal de los formularios será ofrecer un entorno gráfico para la captura de información referente al manejo de las reglas que regularán el tráfico de la red, sin embargo, el manejo del tráfico se hará de forma interna en el sistema, de manera que no será necesario mostrar como se maneja dicha información.

- **Formulario de validación de usuarios.**

  Este formulario contará con 2 cuadros de texto, el primero será para introducir u nombre de usuario de la aplicación y el segundo será para introducir la contraseña. Si el usuario y la clave son válidos entonces se tendrá acceso al formulario principal de la aplicación, de lo contrario se mostrará un mensaje de error.

Figura 12. Pantalla de verificación de usuario.

- **Formulario principal de la aplicación.**

  La pantalla principal de la aplicación estará formada por el logo de la aplicación y 6 menús.

Figura 13. Pantalla principal de la aplicación

1. **Menú Usuarios.** En este menú se podrá administrar la cuenta del usuario que ha ingresado al sistema, es decir que permitirá cambiar la contraseña de acceso cuando se requiera. Para ello require que el usuario ingrese de nuevo su contraseña, la nueva contraseña y una confirmación de la nueva contraseña.

Figura 14. Menú Usuarios de la aplicación

2. **Menú Reglas.** En este menú se podrá administrar las tareas principales del manejador de reglas. Esto incluye la adición, modificación y eliminación de reglas. También se podrán activar o desactivar todas las reglas y se contará con las opciones de eliminación de todas las reglas. Contará con las opciones de Administración de reglas, en la cual se podrá agregar, modificar, activar, desactivar o eliminar una regla por separado, así mismo con las opciones Habilitar todas las reglas y Deshabilitar todas las reglas los cuales manejarán todas las reglas a la vez, y por último la opción Eliminar todas las reglas el cual borrará todas las reglas que hayan sido creadas.

Figura 15. Menú Reglas.

3. **Menú Grupos de IP o MAC.** Este menú posee 2 opciones: La primera de ellas es la de Administración de grupos, en la cual se especifican las direcciones IP o direcciones MAC que pertenecerán a cada grupo. Así mismo permite agregar, modificar o eliminar un grupo. La segunda opción es la de eliminar todos los grupos que hayan sido creados.

Figura 16. Menú Grupos IP o MAC

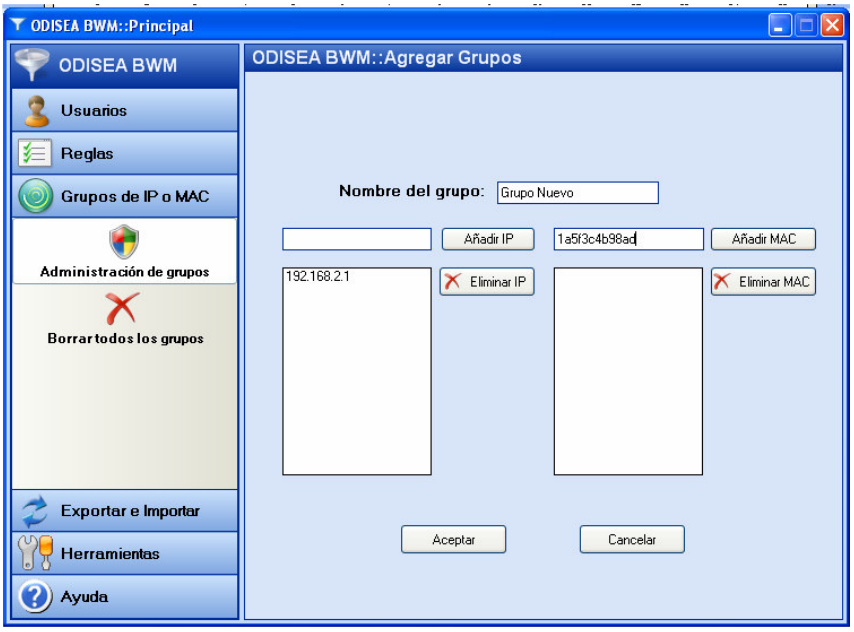

Figura 17 Adición de grupos de direcciones IP o MAC

4. **Menú Exportar e Importar.** En este menú se permitirá al usuario exportar las reglas del sistema a un archivo o importar reglas desde un archivo al sistema. El archivo es de tipo xml el usuario debe especificar la ubicación de dicho archivo tanto en el proceso de exportación como en el de importación.

Figura 18. Menú Exportar e Importar.

5. **Menú Herramientas.** Este menú contiene opciones administrativas del sistema para el monitoreo del desempeño del sistema. Sus opciones son:
    a. **Visor de eventos.** El cual contiene un historial de los eventos ocurridos en el sistema
    b. **Estadísticas.** Esta opción muestra las estadísticas del comportamiento de las reglas a medida que los paquetes de datos son procesados por la aplicación
    c. **Configuración de la Aplicación.** En esta opción se establecen parámetros del sistema como la designación de las interfaces utilizadas para la conexión entre la red LAN y la red WAN, y el ancho de banda del que dispondrá el sistema como máximo.

Figura 19. Menú Herramientas. Visor de eventos.

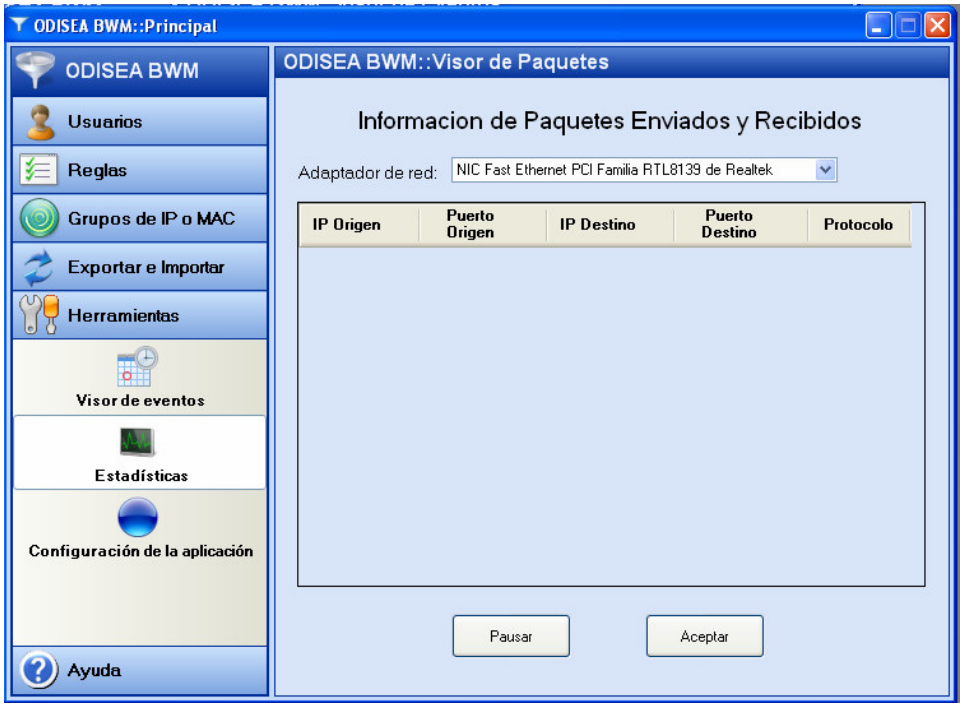

Figura 20. Menú Herramientas. Estadísticas

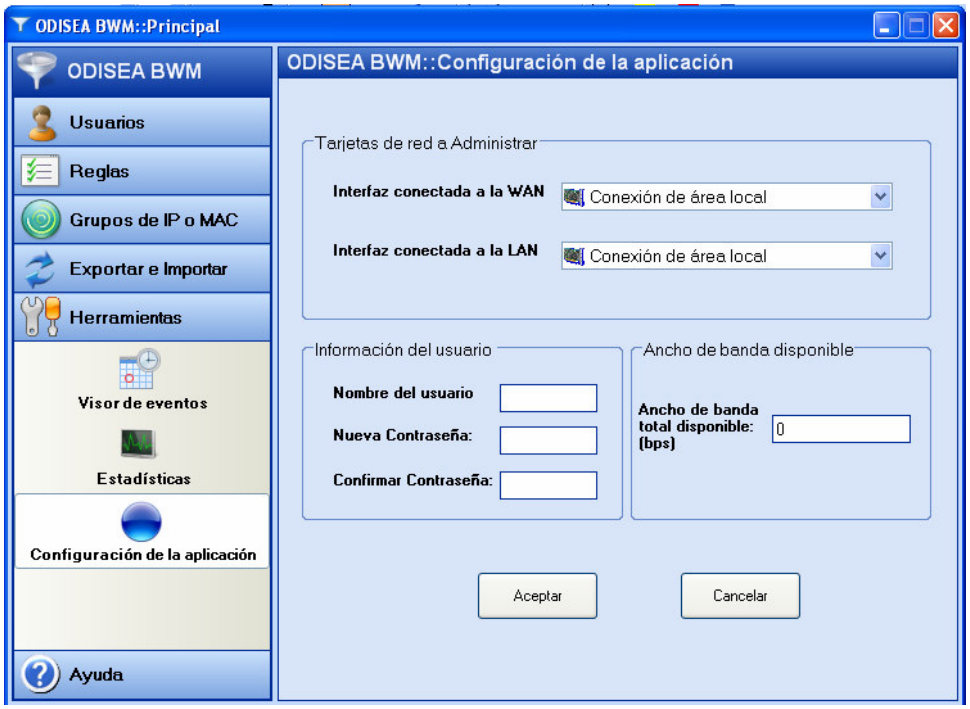

Figura 21. Menú Herramientas. Configuración de la Aplicación

6. **Menú Ayuda.** Este menú contiene un minucioso tutorial acerca del uso de la aplicación y la función de cada una de sus opciones. Las opciones de

este menú son: Contenido, en el cual está desglosado el contenido del manual de la aplicación, y la opción Acerca de…, en la cual se muestra el nombre de la aplicación, la versión y los nombres de los programadores del sistema.

6.4 Diseño de la base de datos.

El diseño propuesto para la base de datos se define en el siguiente diagrama entidad-relación:

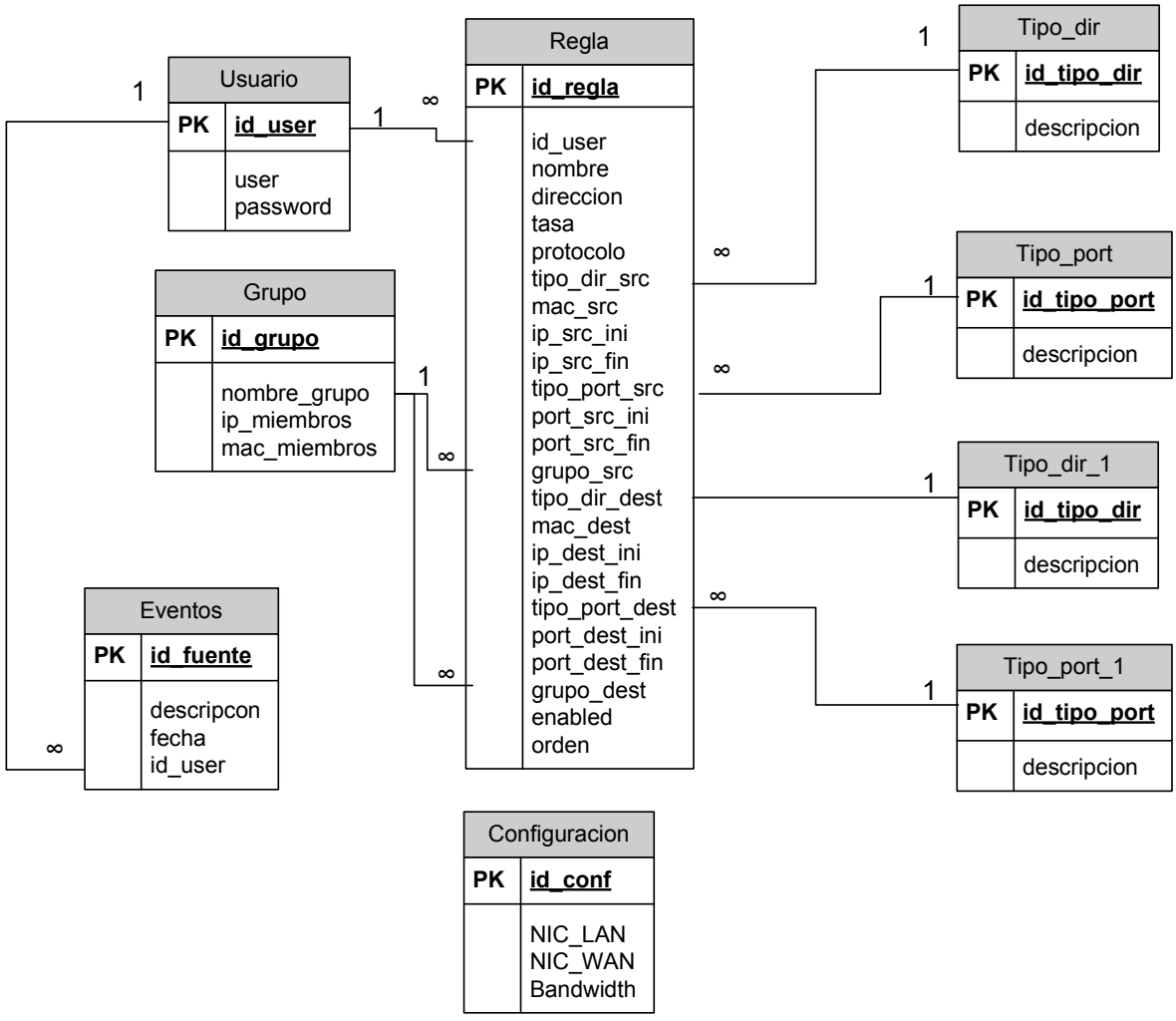

Figura 22: Diagrama Entidad-Relación de la base de datos del proyecto

La base de datos del sistema estará compuesta por 7 tablas: Evento, Grupo, Regla, tipo_dir, tipo_port, usuario y est_seg. En el diagrama Entidad-

Relación mostrado en la figura 22 se observan 2 tablas más: tipo_dir_1 y tipo_port_1. Estas 2 tablas son tablas auxiliares, aunque representan las entidades tipo_port y tipo_dir respectivamente.

La tabla principal de la base de datos, como se puede apreciar en la figura 22, es la tabla Regla. Ésta contiene información general de las reglas para filtrar y administrar el tráfico. Los campos que la componen son los siguientes:

- id_regla. Contiene un número para identificar a la regla. Este campo es autonumérico.
- id_user. Contiene el id del usuario que creó la regla.
- tipo_dir_src. Contiene el tipo de dirección origen Los valores que puede tomar son los siguientes:
    1. 1 = Se usará una dirección MAC.
    2. 2 = Se usará una dirección IP única.
    3. 3 = Se usará un rango de direcciones IP's
    4. 4 = Se usará cualquier IP.
    5. 5 = Se usará un grupo de usuarios.

- tipo_port_src. Guarda el tipo de puerto origen Los valores que puede tomar son los siguientes:
    1. 1 = Se usarán cualquier puerto.
    2. 2 = Se usará un puerto único.
    3. 3 = Se usará un rango de puertos.
- port_src_ini. Guarda el número de puerto origen o puerto inicial si se trata de un rango.
- port_src_fin. Guarda el número de puerto final para un rango de puertos origen.
- tipo_dir_dest = Guarda el tipo de dirección destino Los valores que puede tomar son los mismos que tipo_dir_src.
- tipo_port_dest. Guarda el tipo de puerto destino Los valores que puede tomar son los mismos que tipo_port_src.

- port_dest_ini. Guarda el número de puerto destino o puerto inicial si se trata de un rango.
- port_dest_fin. Guarda el número de puerto final para un rango de puertos destino.
- Nombre. Almacena el nombre de la regla.
- Dirección. Guarda la dirección de trafico que se regulara (entrante, saliente o ambos).
- tasa. Almacena la tasa de transferencia asignada al tráfico que cumpla con esta regla.
- protocolo. Contiene el protocolo que se manejará.
- mac_src. Almacena la dirección MAC origen en el caso que se utilice este parámetro.
- ip_src_ini. Guarda la dirección IP origen, o dirección IP inicial si se trata de un rango.
- ip_src_fin. Guarda la dirección IP final origen en el caso de un rango.
- grupo_src. Si se manejará en base a grupos se almacena el nombre del grupo acá.
- mac_dest. Almacena la dirección MAC destino en el caso que se utilice este parámetro.
- ip_dest_ini. Guarda la dirección IP destino, o dirección IP inicial si se trata de un rango.
- ip_dest_fin. Guarda la dirección IP final destino en el caso de un rango.
- grupo_dest. Si se manejará en base a grupos se almacena el nombre del grupo acá.
- enabled. Regla habilitada (true o false).
- Orden. Indica el orden de prioridad que tiene la regla en el listado de reglas activas

La tabla usuario almacena la información de los usuarios que están habilitados para ingresar a la aplicación, administrar reglas y grupos. Los campos que la conforman son los siguientes:

- id_user. Número de identificación de usuario. Este campo es autonumérico.
- user. Almacena el nombre del usuario.
- password. Guarda la contraseña del usuario.

En la tabla evento se lleva un registro de las acciones que han ocurrido sobre alguna de las reglas o grupos que se hayan definido en el programa. Se puede saber cuándo fue creada una regla o grupo, el usuario que lo creó, y si ha tenido modificaciones o inclusive si ha sido eliminado. Los campos que la forman son los siguientes:

- id_evento. Número de identificación del evento. Este campo es autonumérico.
- fecha. Almacena la hora y fecha a la que ocurrió el evento.
- descripción. Guarda la descripción del evento que ocurrió.
- id_user. Guarda el número de identificación del usuario que generó el evento.

La tabla grupo almacena la información de los grupos de usuarios que hayan sido creados. Estos grupos se crean en base a direcciones IP o direcciones MAC. Los campos que la forman son los siguientes:

- id_grupo. Número de identificación del grupo. Este campo es autonumérico.
- nombre_grupo. Almacena el nombre del grupo.
- ip_miembros. Guarda una lista de las direcciones IP que forman parte del grupo.
- mac_miembros. Guarda una lista de las direcciones MAC que forman parte del grupo.

La tabla tipo_dir se encarga de manejar el tipo de direcciones que se va a utilizar. Los datos de esta tabla son predefinidos y se de muestran a continuación:

Id_tipo_dir	descripción
1	Se usará una dirección MAC.
2	Se usará una dirección IP única.
3	Se usará un rango de direcciones IP's
4	Se usará cualquier IP.
5	Se usará un grupo de usuarios.

Tabla 6. Valores de la tabla tipo_dir.

La tabla tipo_port se encarga de manejar el tipo de direcciones que se va a utilizar. Los datos de esta tabla son predefinidos y se de muestran a continuación:

Id_tipo_port	descripción
1	Se usarán cualquier puerto.
2	Se usará un puerto único.
3	Se usará un rango de puertos.

Tabla 7. Valores de la tabla tipo_port.

La tabla est_seg será la encargada de almacenar los datos que corresponden a las estadísticas de las reglas que están siendo utilizadas en un determinado momento en el sistema. Esta tabla se encuentra conformada por los campos:

- id_est. Número de identificación de uno de los datos estadísticos. Es autonumérico.
- Id_regla. Indica el identificador de la regla a la que ese dato pertenece.
- Total_Bytes_Acum_Entrante. Indica el total de bytes que han ingresado al sistema bajo una determinada regla en un momento específico.
- Total_Bytes_Acum_Saliente. Indica el total de bytes que han salido de la aplicación para una determinada regla en un momento específico.
- Tasa_Entrante. Expresa la tasa con la cual entra cierto tráfico en el sistema en un determinado momento para una determinada regla.

- Tasa_Saliente. Expresa la tasa con la cual sale cierto tráfico del sistema en un determinado momento para una determinada regla.
- Paquetes_en_Cola_Entrantes. Muestra la cantidad de paquetes entrantes al sistema que se encuentran en la cola de una regla.
- Paquetes_en_Cola_Salientes. Muestra la cantidad de paquetes que salen del sistema que se encuentran en la cola de una regla.

La tabla Configuración almacena la información de la configuración de la aplicación, la cual es consultada para evaluar las reglas que rigen la restricción del ancho de banda. Los campos que la componen son:

- Id_conf. Número que identifica una configuración y la distingue de otra configuración
- NIC_LAN. Almacena el nombre de la interfaz de red que se encuentra conectada a la red LAN.
- NIC_WAN. Almacena el nombre de la interfaz de red que se encuentra conectada a la red WAN.
- Bandwidth. Guarda el total de ancho de banda disponible en el acceso WAN, es decir el ancho de banda máximo al que la red LAN puede acceder por medio del enlace WAN.

6.4.1 Enlace con MySQL

Los usuarios registrados para utilizar el sistema, las reglas de administración del ancho de banda y las ocurrencias de los distintos eventos que pasen en relación al sistema, serán almacenados en una base de datos. El gestor de bases de datos seleccionado para el sistema es MySQL. Las principales razones por las que se ha decidido seleccionar dicho gestor, es debido a sus características como software gratuito, a pesar de eso MySQL es un gestor lo suficiente potente como para almacenar la información relacionada al sistema de administración de ancho de banda.

Para que el sistema logre interactuar con la base de datos creada en MySQL es necesario establecer una conexión utilizando código en lenguaje C, aunque tal conexión se encuentra facilitada por la librería MySQL.h, incluida en el gestor de bases de datos. De esta manera se pretende mantener un control de la información procesada en el sistema en una base de datos.

6.5 Algoritmo de Administración Empleado en la Aplicación

Como se pudo observar en el capítulo anterior existen muchos algoritmos que pueden ser empleados para regular el tráfico de la red. En el desarrollo de este proyecto se ha decidido emplear el algoritmo de modelado de tráfico (Traffic Shaper), la razón por la que se empleará dicho algoritmo es debido a que no es muy complejo si se le compara con otros algoritmos, es de fácil programación, y a pesar que no ofrece una administración óptima del ancho de banda, se considera que la tolerancia que el algoritmo pueda tener con respecto al valor de ancho de banda deseado es aceptable para fines prácticos.

En el numeral 5.2.3 de este documento se describe el funcionamiento del algoritmo el cual se encuentra basado en generar tiempos muertos en el procesador al momento de hacer el reenvío de los paquetes. De hecho el algoritmo implementado en la función debe tomar el paquete de las colas, en las que los paquetes han sido previamente clasificados de acuerdo a las reglas de priorización de tráfico establecidas por el usuario de la aplicación, y enviarlo a la red, después de eso se genera el retrazo de tiempo generado como resultado de la ecuación:

$$Sleep = \max\left(\frac{S}{R} - T, 0\right)$$

Sin embargo dado que la regulación del tráfico se da de manera que se puedan generar *n* reglas, el algoritmo debe ejecutarse para cada una de ellas

es decir que debe haber *n* procesos ejecutando el mismo algoritmo pero cada uno regulando un tipo diferente de tráfico.

6.6 Algoritmo de Encolamiento en el Sistema

El sistema de administración de ancho de banda utilizará una cola para ubicar los paquetes que se reciban en la interfaz de red mientras son enviados con el ancho de banda asignado. En este caso de los algoritmos de encolamiento listados en el numeral 5.2.1 se explican los algoritmos de encolamiento, de entre ellos el que se ha seleccionado es el de First-In-Fist-Out (FIFO) descrito con detalle en el numeral 5.2.1.1.

Las razones por las que se ha decidido la implementación de este sistema de encolamiento son: Dado que la administración de ancho de banda estará regulada mediante el algoritmo de modelado de tráfico el cual necesita simplemente tomar un paquete de los que están en las colas, no es necesario que en el encolamiento se realice un proceso de ordenamiento de paquetes, por lo cual para acelerar el proceso de administración el algoritmo FIFO resulta ser eficiente en cuanto a tiempo y complejidad, así mismo por su fácil programación.

6.7 Modelado del Sistema

El funcionamiento básico del sistema se puede representar en Lenguaje de Modelado Unificado (UML) mediante casos de uso. Estos casos de uso involucrarían dos actores, el primero el que interactúa con el sistema para realizar su configuración y determinar la forma en la que este trabajará, el cuál será denominado Administrador de la Red. El segundo actor que interviene en el funcionamiento del sistema es el tráfico en la red que atraviesa el nodo que contiene el sistema y que es clasificado y regulado de acuerdo a la serie de reglas que hayan sido establecidas.

De manera gráfica los casos de uso que representan al sistema se muestran a continuación:

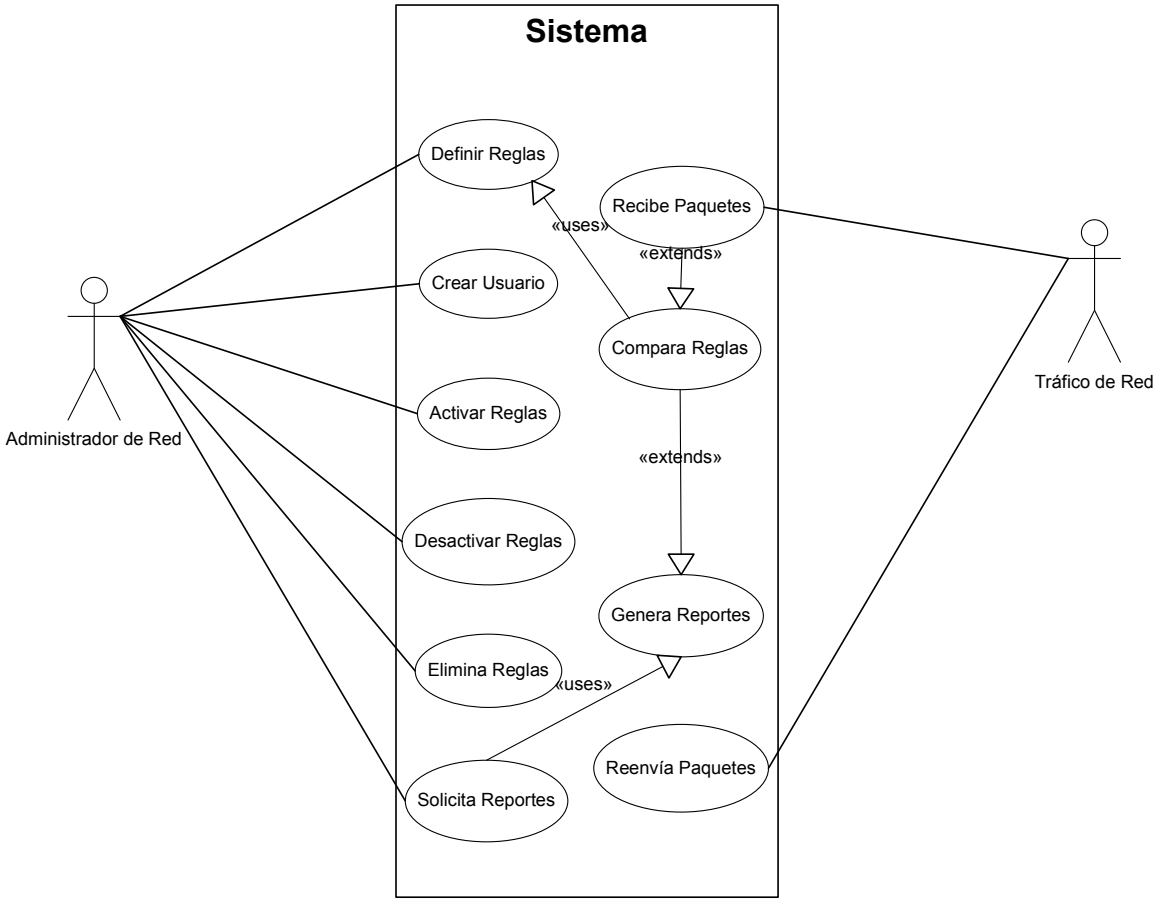

Figura 23. Casos de uso del Odisea Bandwidth Manager.

Cuando el sistema recibe un paquete el sistema “extiende” la función del caso de uso a comparar el contenido del paquete con las reglas definidas, entonces se verifica o se “usa” el caso de la definición de las reglas, así mismo el caso de comparar las reglas se extiende a almacenar la información en el historial del sistema para generar reportes posteriormente.

6.8 Clases utilizadas en la aplicación

Las clases que se han utilizado en la aplicación son presentadas y detalladas a continuación:

6.8.1 Clase Rule

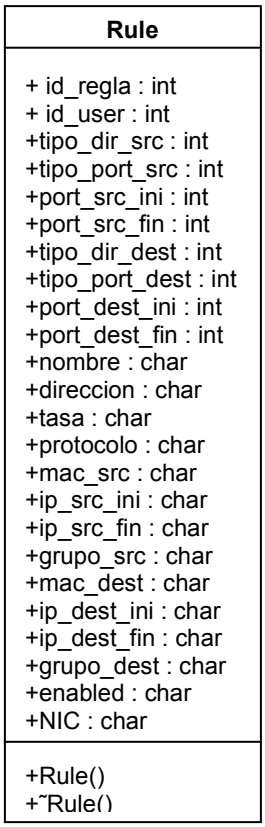

Figura 24. Detalle de la clase Rule

La clase Rule contiene la definición en el programa de todos los parámetros utilizados para formar una regla la cual será validada y almacenada en la base de datos, y con la que se comparará con las propiedades de los paquetes que sean evaluados por el programa y, de acuerdo a los resultados que presente, se le dará el tratamiento especificado por la regla que cumpla.

Los atributos utilizados por la clase Rule son:

- Id_regla: contiene el identificador único de cada una de las reglas que sean creadas.
- Id_user: contiene el identificador del usuario que ha creado la regla. El identificador del usuario se toma en base al usuario que ha ingresado al programa y ha realizado configuraciones en su sesión.

- Tipo_dir_src: Contiene el tipo de dirección origen. Puede tomar los siguientes valores: 1: se usará una dirección MAC; 2: se usará una dirección IP única; 3: se usará un rango de direcciones IP; 4: se usará cualquier IP; 5: se usará un grupo de usuarios.
- Tipo_port_src: Guarda el tipo de puerto origen. Puede tomar los siguientes valores: 1: se usará cualquier puerto; 2: se usará un puerto único; 3: se usará un rango de puertos.
- port_src_ini: guarda el número de puerto origen o puerto inicial (si se trata de un rango).
- port_src_fin: guarda el número de puerto final para un rango de puertos origen.
- tipo_dir_dest: guarda el tipo de dirección destino. Los valores que puede tomar son los mismos que tipo_dir_src.
- tipo_port_dest: guarda el tipo de puerto destino. Los valores que puede tomar son los mismos que tipo_port_src.
- port_dest_ini: guarda el número de puerto destino o puerto inicial (si es rango).
- port_dest_fin: guarda el número de puerto final para un rango de puertos destino.
- Nombre: contiene el nombre que se le asigne a la regla.
- Dirección: contiene el sentido del tráfico que se estará regulando (tráfico entrante, tráfico saliente, ambas direcciones).
- Tasa: contiene la tasa de transferencia que se ha asignado al tráfico que cumpla con la regla que se está creando.
- Protocolo: contiene el protocolo que se manejará.
- mac_src: contiene la dirección MAC origen (en caso de utilizarse este parámetro).
- ip_src_ini: contiene la dirección IP origen, o dirección IP inicial si se trata de un rango.
- ip_src_fin: guarda la dirección IP final del origen (en el caso de un rango).

- grupo_src: se almacena el nombre del grupo de inicio si el tráfico se manejara en base a grupos.
- mac_dest: almacena la dirección MAC destino en el caso que se utilice este parámetro.
- ip_dest_ini: guarda la dirección IP destino, o dirección IP inicial si se trata de un rango.
- ip_dest_fin: guarda la dirección IP final destino (en caso de un rango).
- grupo_dest: se almacena el nombre del grupo final si el tráfico si se manejará en base a grupos.
- Enabled: almacena el estado inicial de la regla (habilitada o deshabilitada, con true o false).
- NIC: guarda la interfaz de red en la cual la regla será aplicada.

Los métodos utilizados en la clase Rule son:

- Rule: constructor utilizado por la clase.
- ~Rule: destructor de la clase.

6.8.2 Clase WSockServer

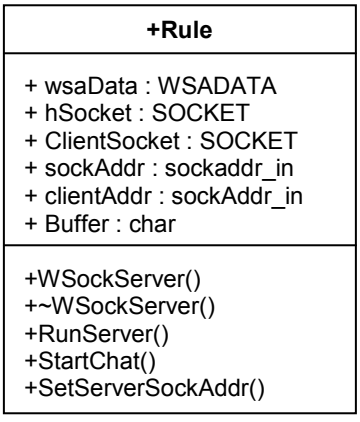

Figura 25. Detalle de la clase WSockServer

La clase WSockServer contiene las variables necesarias para iniciar, negociar y mantener una comunicación entre el servidor y los clientes mediante el uso de sockets (interfaces lógicas para comunicación entre máquinas), proporcionando las herramientas necesarias para el envío y recepción de información entre ambos.

Los atributos utilizados por la clase WSockServer son:

- wsaData: esta variable indica la versión de socket que se utilizará para la comunicación y evaluará si la versión es la correcta para dicho fin.
- hSocket: contiene el identificador del socket que se utilizará en la comunicación.
- ClientSocket: variable que contiene la información relacionada con el cliente con el cual se está comunicando el servidor a través del socket creado.
- sockAddr: en esta variable se almacena la dirección y el puerto del servidor en donde se ha creado el socket para establecer la comunicación con el cliente.
- ClientAddr: variable que almacena la dirección y el puerto del cliente en donde se ha creado el socket para establecer la comunicación con el servidor.
- Buffer: variable que almacena, tanto en el cliente como en el servidor, la información que dicho cliente envía al servidor.

Los métodos utilizados en la clase WSockServer son:

- WSockServer: constructor utilizado por la clase.
- ~WSockserver: destructor utilizado por la clase.
- RunServer: habilita el programa para comunicación con los clientes, creando el socket respectivo.
- StartChat: en el cliente, inicia el envío de la información respectiva; en el servidor, inicia la recepción de los clientes y sus datos.
- SetServerSockAdr: establece el socket y la dirección a la que los clientes estarán enviando los datos en el servidor.

6.9 Pruebas preliminares

Se han realizado pruebas preliminares del código y los algoritmos desarrollados para funciones específicas como bloqueo de paquetes y administración de ancho de banda basado en direcciones IP, puertos y protocolos.

En los comienzos del desarrollo de la aplicación se contaba con la opción de utilizar la herramienta de programación en entorno gráfico QT, sin embargo al crearse la primera versión de la aplicación se determinó que el uso que esta herramienta hacía de la memoria de la computadora, generaba un desempeño ineficiente de la aplicación. Específicamente ocurría cuando la aplicación se había estado ejecutando por tres horas continuas o más, lo cual conllevaba a que la aplicación dejara de administrar el ancho de banda hasta que el programa se reiniciara.

Debido a ese problema encontrado en la aplicación se decidió utilizar una herramienta distinta para el desarrollo del entorno gráfico de la aplicación, una que no requiriera consumir posiciones de memoria de forma dinámica como QT lo hacía y fue así como se determino crear la Interfaz gráfica del sistema en Visual Basic.net.

Sin embargo se decidió incluir las pruebas realizadas en la primera versión que demuestran el funcionamiento de la misma, y se muestran las pruebas realizadas en la versión 2.0.2 en la cual se diferencia el cambio a Visual Basic .net.

Para el desarrollo de la aplicación se utilizó una librería para el manejo de paquetes llamada winpkfilter. Esta librería proporciona funciones que permiten manipular en una mejor forma los paquetes o el flujo de datos que se maneja en el dispositivo de red. Winpkfilter permite a los usuarios (desarrolladores) filtrar de

forma transparente los paquetes de red sin causar algún impacto negativo a la actividad normal de la red.[21]

En winpkfilter, el paso o restricción de paquetes se controla mediante banderas definidas en la librería, las cuales son:

- MSTCP_FLAG_SENT_TUNNEL: encola todos los paquetes que han enviado desde el módulo de TCP de Microsoft hacia la interfaz de red, y los paquetes originales son descartados.
- MSTCP_FLAG_RECV_TUNNEL: encola los paquetes que han sido indicados por la interfaz de red hacia el módulo de TCP, y los paquetes originales son descartados.
- MSTCP_FLAG_SENT_LISTEN: encola los paquetes enviados desde el módulo de TCP hacia la interfaz de red. Se envían los paquetes originales.
- MSTCP_FLAG_RECV_LISTEN: encola los paquetes indicados por la interfaz de red, hacia el módulo de TCP. Se envían los paquetes originales.

6.9.1 Prueba de bloqueo de paquetes.

La prueba consistió en bloquear el paso de paquetes a través del dispositivo de red de la máquina. Como resultado de esta acción, toda aplicación que requería el uso, ya sea directo o indirecto, de la interfaz de red, tuvo como respuesta un error, ya sea de falta de respuesta de otra máquina o notificación de fracaso al querer alcanzar algún objetivo.

Se tiene primero la prueba de conectividad con otro host. Para esto se utiliza el comando ping, con el que se verificará si se encuentra conexión con otra máquina; de ser así, se obtendría una respuesta de la máquina a la que se le está haciendo la solicitud de respuesta, en caso contrario se obtendría un mensaje de no conexión o no respuesta.

[21] NT Kernel Resources. http://www.ntkernel.com/w&p.php?id=7

La dirección de la máquina con la que se hará la prueba es la 192.168.1.4, como se muestra en la figura 27.

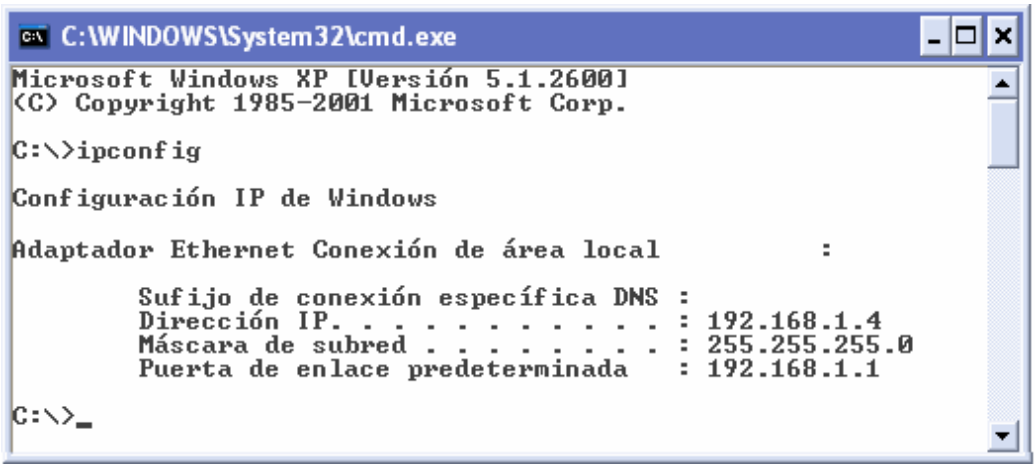

Figura 26. Información de red del host de prueba.

Al hacer ping hacia el gateway por defecto muestra que si hay conectividad como se muestra en la siguiente figura. Asimismo se hizo una prueba de conectividad con un sitio web como se muestra en la siguiente figura:

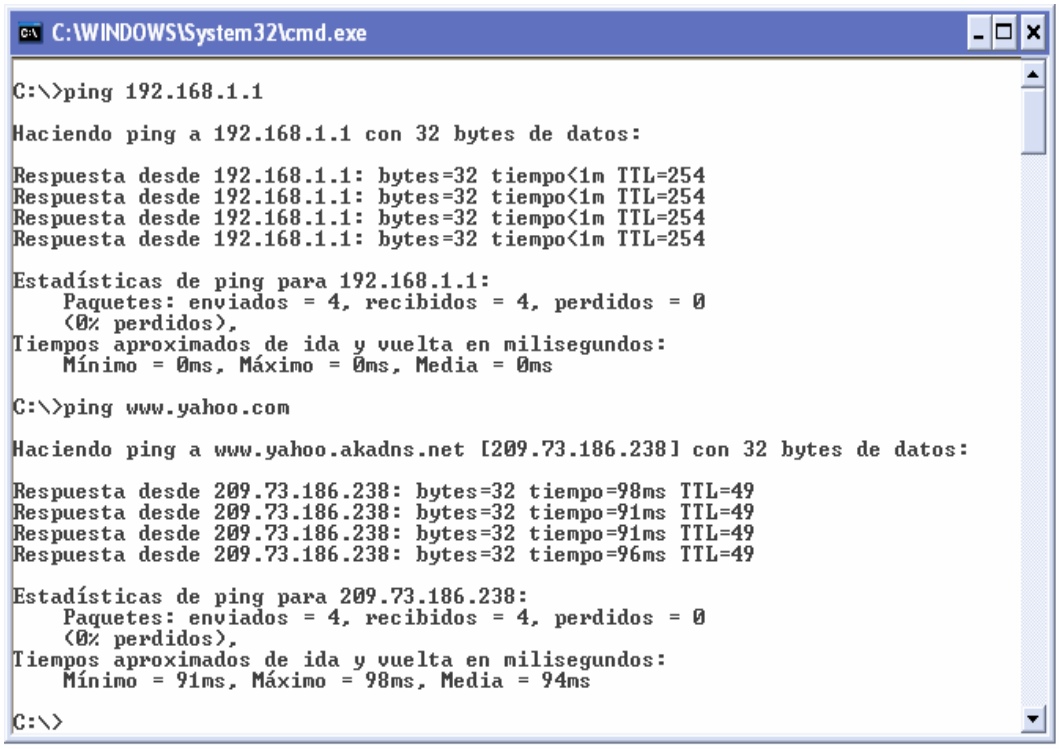

Figura 27. Pruebas de conectividad antes del bloqueo de paquetes ICMP.

Sin comenzar a ejecutar el programa de bloqueo de paquetes, se puede observar que existe total conectividad con el host de prueba, por lo que se está seguro que el paso de paquetes es completamente libre, lo que permite establecer la conexión con la otra máquina a la cual pedimos una respuesta en un mensaje ICMP.

Una vez que se ha comprobado que existe conectividad se procede a crear una regla para evitar que los paquetes ICMP lleguen a su destino. De esta forma se evitará que el host que tiene la dirección IP 192.168.1.4 pueda hacer ping a otro host de la misma red o de una red pública. Para esto debe crearse una regla con los siguientes parámetros:

- Nombre de la regla: Bloqueo ICMP.
- Dirección: Entrante y saliente.
- Tasa: Bloquear.
- Protocolo: ICMP.
- Dirección Origen: 192.168.1.4.

- Puerto Origen: Cualquiera.
- Dirección Destino: Cualquier dirección IP.
- Puerto Destino: Cualquier puerto.

Una vez que se ha creado la regla se procede a verificarla en el formulario de administración de reglas (Versión 1.0 del software):

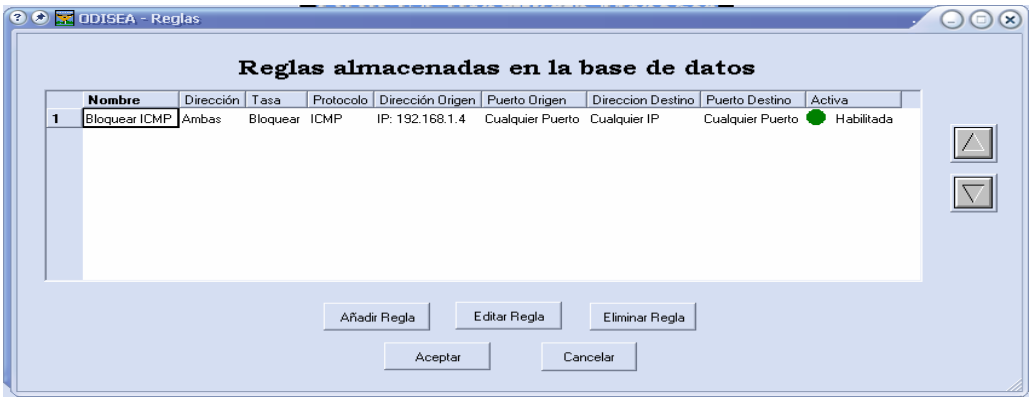

Figura 28. Formulario de administración de reglas donde se muestra la regla creada.

Para comprobar que el tráfico ICMP ha sido bloqueado se procede a hacerse las pruebas de conectividad con el comando ping.

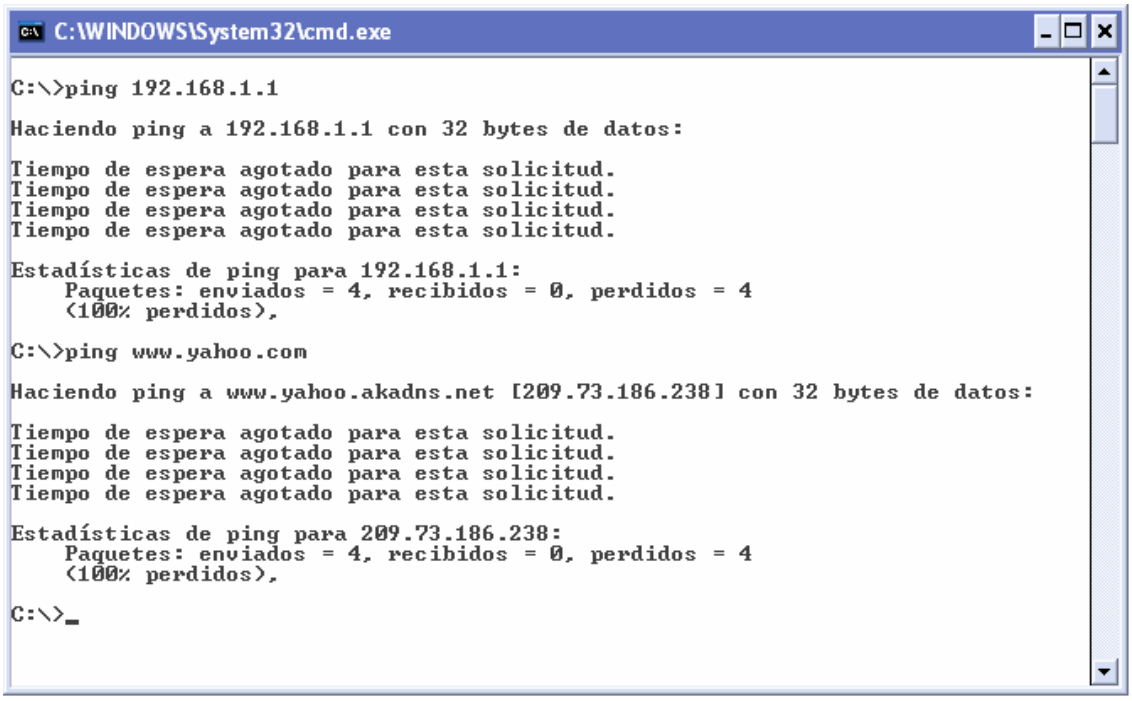

Figura 29. Pruebas de conectividad después del bloqueo de paquetes ICMP.

Se observa que se han perdido todos los paquetes que intentaban establecer comunicación con el host con la dirección 192.168.1.4, con lo que se comprueba que los paquetes han sido bloqueados al no obtener respuesta de las peticiones ICMP enviadas.

Como resultado de la prueba puede decirse que aunque físicamente existe la conectividad apropiada entre los host, los paquetes son retenidos antes de enviarse a la red, por lo que no puede completarse la petición de respuesta Ping.

6.9.2 Pruebas de la versión 2.0.2

También se realizó una prueba de limitación del ancho de banda de una descarga de un servidor ftp y de un servidor http, los resultados ambas pruebas cumplieron con las expectativas. Sin embargo después de 2 horas continuas de funcionamiento del sistema se determino que la aplicación degradaba su funcionamiento, debido a que las funciones del entorno gráfico utilizaban memoria dinámica, lo cuál causaba que mientras el tiempo transcurría el programa tomaba memoria utilizada para las colas de los paquetes.

Fue por eso que se procedió a mudar el entorno gráfico de la aplicación de QT a Visual Basic .net. Cuando el entorno gráfico fue terminado e integrado con el núcleo de la aplicación se realizaron las mismas pruebas con el mismo escenario.

6.9.2.1 Pruebas de bloqueo a internet

Cuando se verificó que el bloqueo de paquetes era realizado propiamente, se procedió a verificar la regulación del ancho de banda en una conexión local a Internet. La topología que se utilizó para esa prueba fue la siguiente:

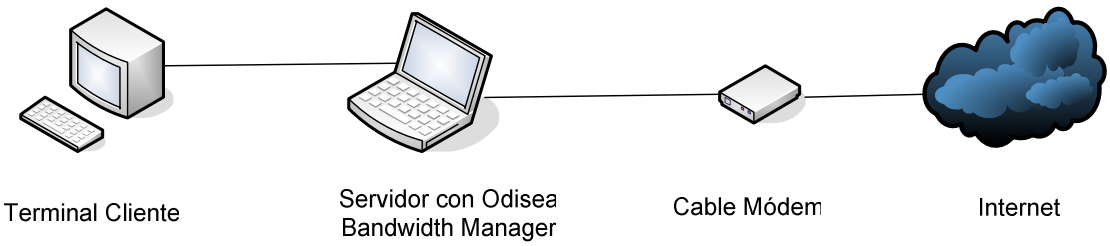

Figura 30. Topología de Prueba

En esta prueba el objetivo era limitar la tasa de navegación o descarga web que la Terminal cliente tendría al acceder a Internet. Para esto en el servidor que contiene la aplicación se utilizaron simultáneamente dos adaptadores de red, uno que lo conectaría a la Terminal cliente y la otra que lo mantendría conectado al nodo de acceso a Internet.

En el servidor de la aplicación también se configuró, mediante el uso del asistente para la conexión de Windows, que otros equipos se conectaran a Internet mediante ese equipo, configuración que fue complementada con el mismo asistente en la Terminal cliente como acceso a Internet por medio de otro equipo de la red.

La primera prueba incluyó bloqueo del tráfico en el protocolo http en la Terminal cliente, utilizando la siguiente regla en el administrador de ancho de banda:

- Nombre de la regla: Bloqueo Http.
- Dirección: Entrante y saliente.
- Tasa: Bloquear.
- Protocolo: TCP.
- Dirección Origen: Cualquier dirección IP.
- Puerto Origen: 80.
- Dirección Destino: Cualquier dirección IP.
- Puerto Destino: Cualquier puerto.

Al ingresar la regla al sistema en la opción administración de reglas, tiene la apariencia siguiente:

Figura 31. Formulario de administración de reglas (versión 2.0.2)

La prueba fue exitosa, ya que el tráfico de protocolo http fue bloqueado completamente en la Terminal cliente y al desactivar el administrador de ancho de banda, la Terminal cliente era capaz de acceder a Internet.

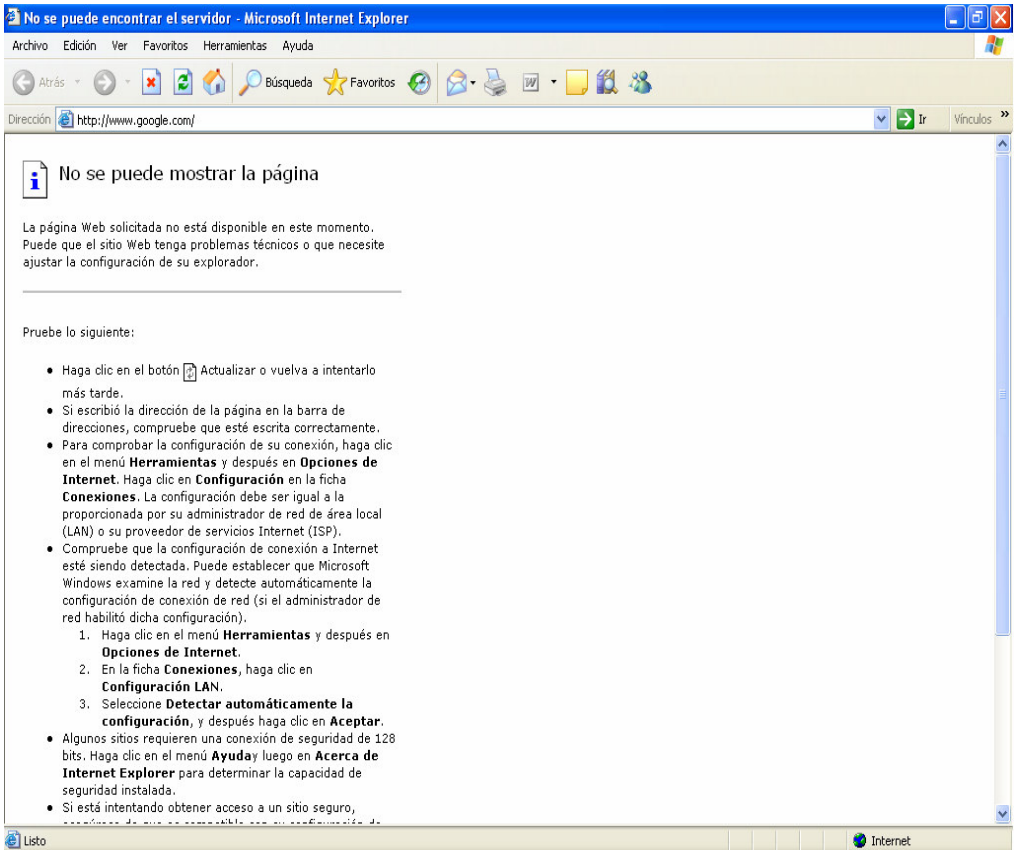

Figura 32. Terminal Cliente intentando acceder a Internet con el administrador de ancho de banda ejecutándose.

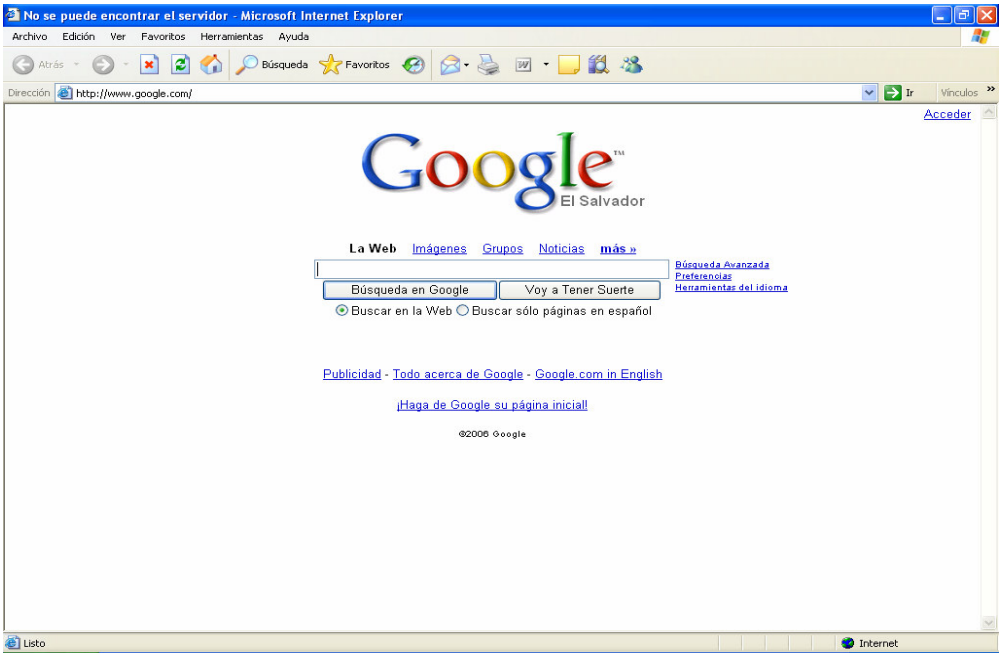

Figura 33. Terminal cliente accediendo a Internet con el administrador de ancho de banda desactivado.

6.9.2.2 Pruebas de ancho de banda limitado

Después se procedió a limitar la tasa de transferencia en la Terminal cliente para el tráfico del protocolo http, se le asigno una tasa de 128 Kbps y se probó una descarga http de un sitio web:

Figura 34. Formulario de administración de reglas. RestHttp es la regla que limitará el tráfico http.

La prueba nuevamente fue exitosa ya que el tráfico de protocolo http fue limitado como estaba esperado a una tasa de descarga máxima aproximada de 16 KBps.

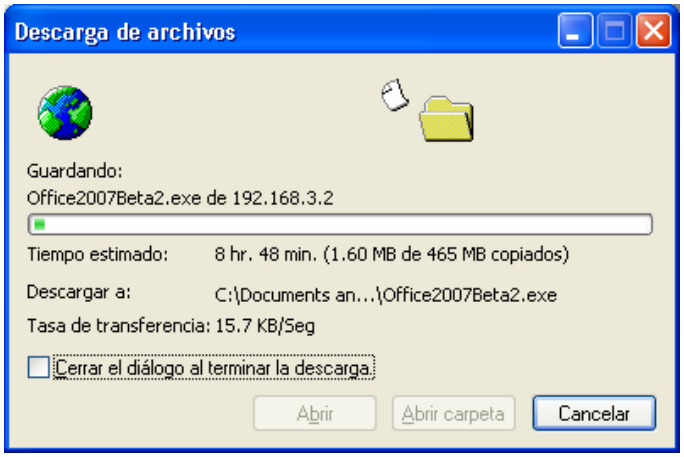

Figura 35. Descarga http con tráfico limitado a 128 Kbps.

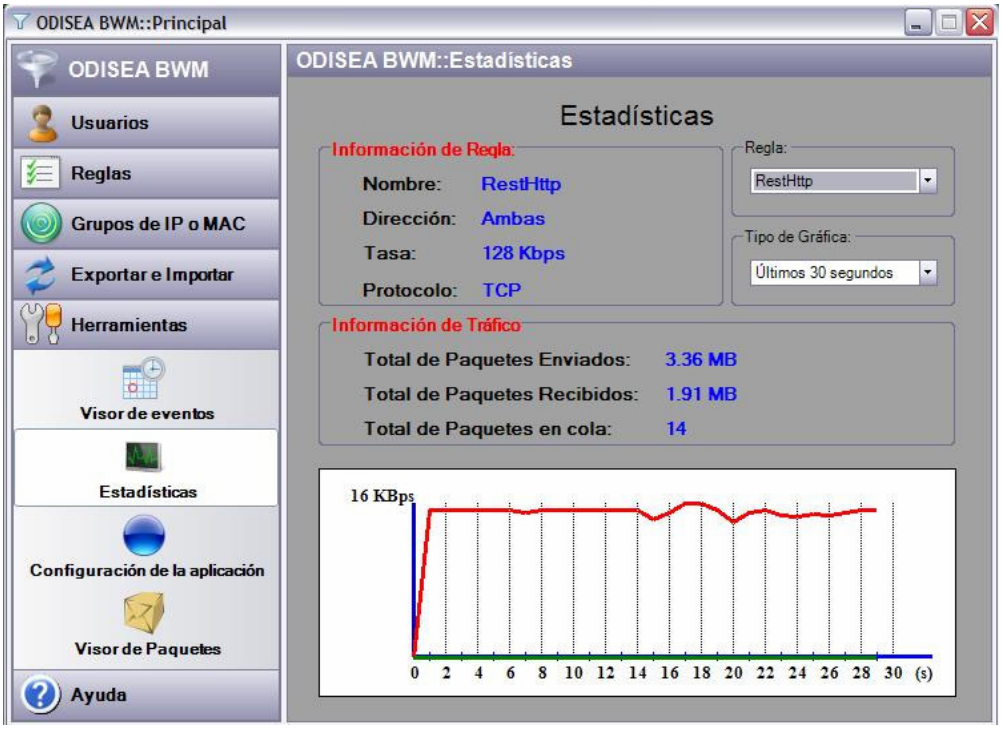

Figura 36. Estadísticas de la regla de limitación de tráfico http.

6.9.2.3 Pruebas de descarga desde servidores remotos

Una vez que fue comprobado el funcionamiento de la aplicación con una red publica como Internet, se decidió generar un escenario un poco más personalizado en el cual se incluyera una red LAN que recibiera y enviara distintas clases de tráfico con servidores ubicados en una red distinta utilizando equipo de ruteo.

Fue entonces que se realizó la prueba empleando la topología siguiente:

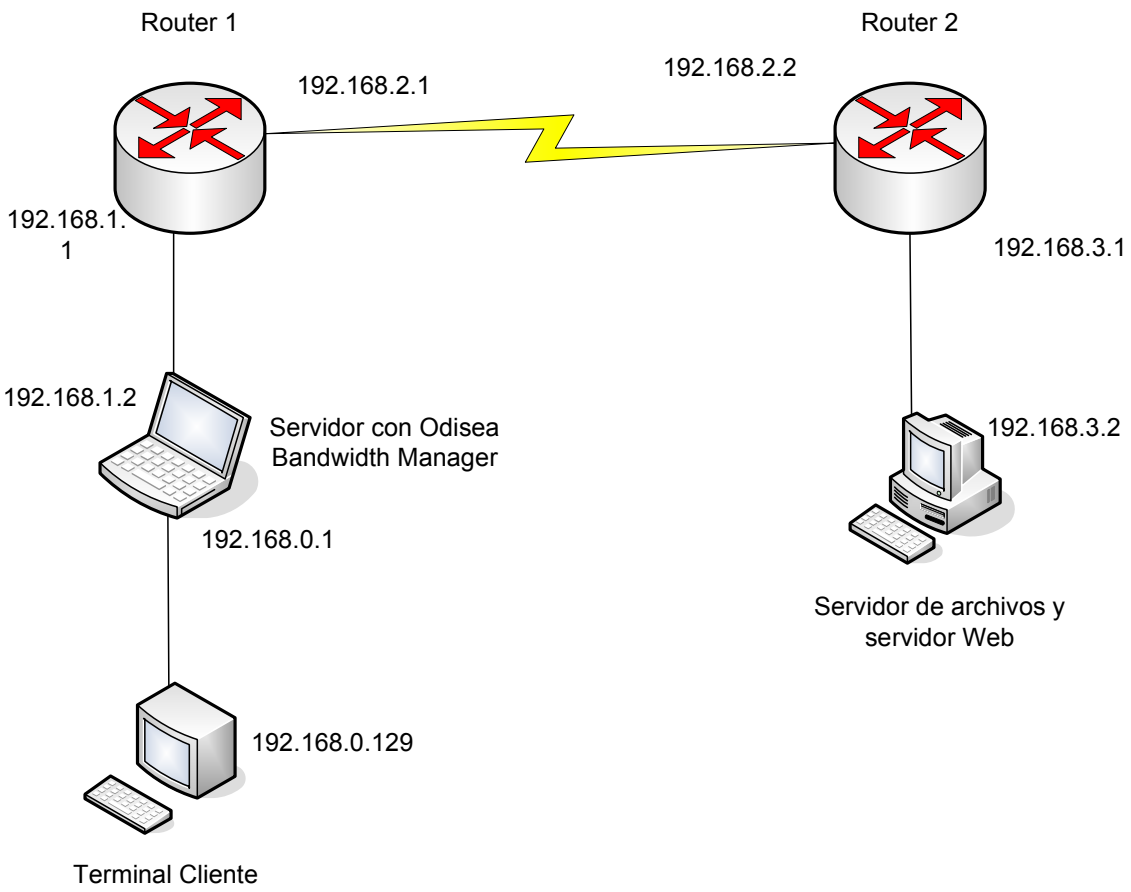

Figura 37. Topología de prueba empleada para distintos tipos de tráfico.

En esta prueba el objetivo era utilizar diversos tipos de tráfico simultáneamente con reglas limitando cada tipo de tráfico para verificar el comportamiento del sistema. En este caso la Terminal cliente intentaría realizar descargas de un servidor ftp instalado en el servidor de archivos, al mismo tiempo realizar descargar archivos desde el servidor web utilizando el protocolo http.

Las reglas utilizadas fueron las siguientes:

Nombre de la Regla: RestFTP

Dirección: Ambas

Tasa de transferencia: 56000 bps

Protocolo: TCP

IP Origen: Cualquier dirección IP

IP Destino: Cualquier dirección IP

Puerto Origen: 20

Puerto Destino: Cualquier puerto

Nombre de la Regla: RestHttp

Dirección: Ambas

Tasa de transferencia: 128000 bps

Protocolo: TCP

IP Origen: Cualquier dirección IP

IP Destino: Cualquier dirección IP

Puerto Origen: 80

Puerto Destino: Cualquier puerto

Los resultados de esta prueba fueron igualmente exitosos ya que la tasa de transferencia en descargas simultáneas tanto http como ftp fueron las esperadas (Aproximadamente 16 Kbps y 7 Kbps respectivamente). A continuación se muestra el comportamiento de la aplicación durante la descarga:

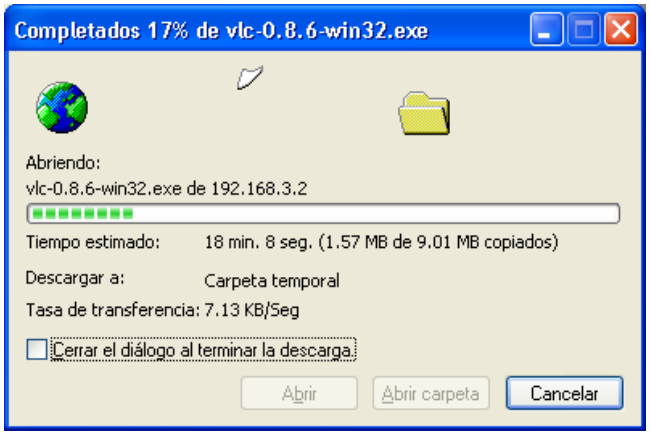

Figura 38. Descarga en la Terminal cliente desde el servidor FTP restringida por la regla RestFTP.

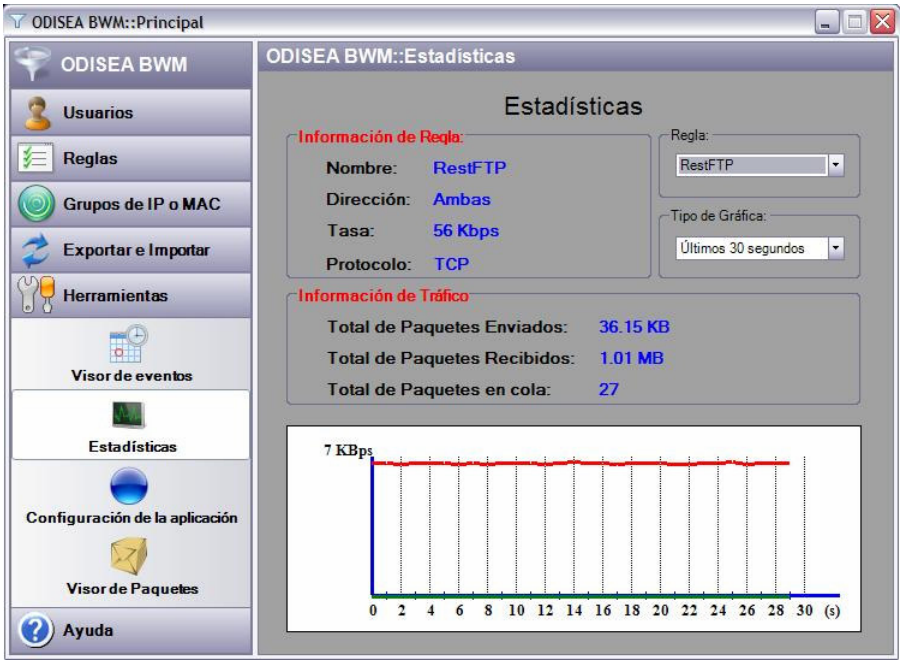

Figura 39. Estadísticas en la aplicación de la regla RestFTP mientras se produce la descarga en la Terminal cliente desde el servidor FTP.

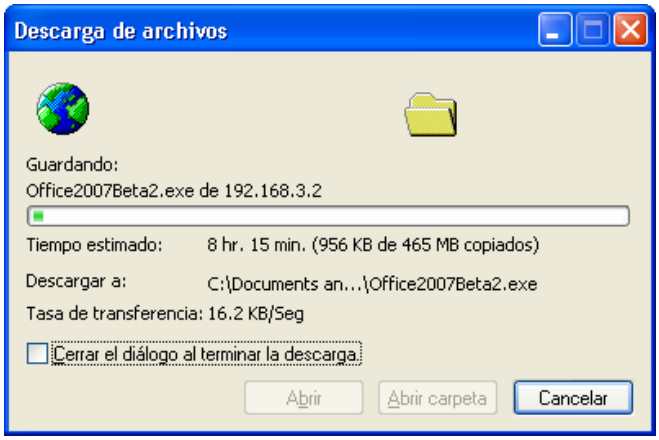

Figura 40. Descarga en la Terminal cliente desde el servidor Web mediante el protocolo http. Esta descarga se está produciendo mientras se descarga por el protocolo FTP y la aplicación se ejecuta en el servidor de acceso.

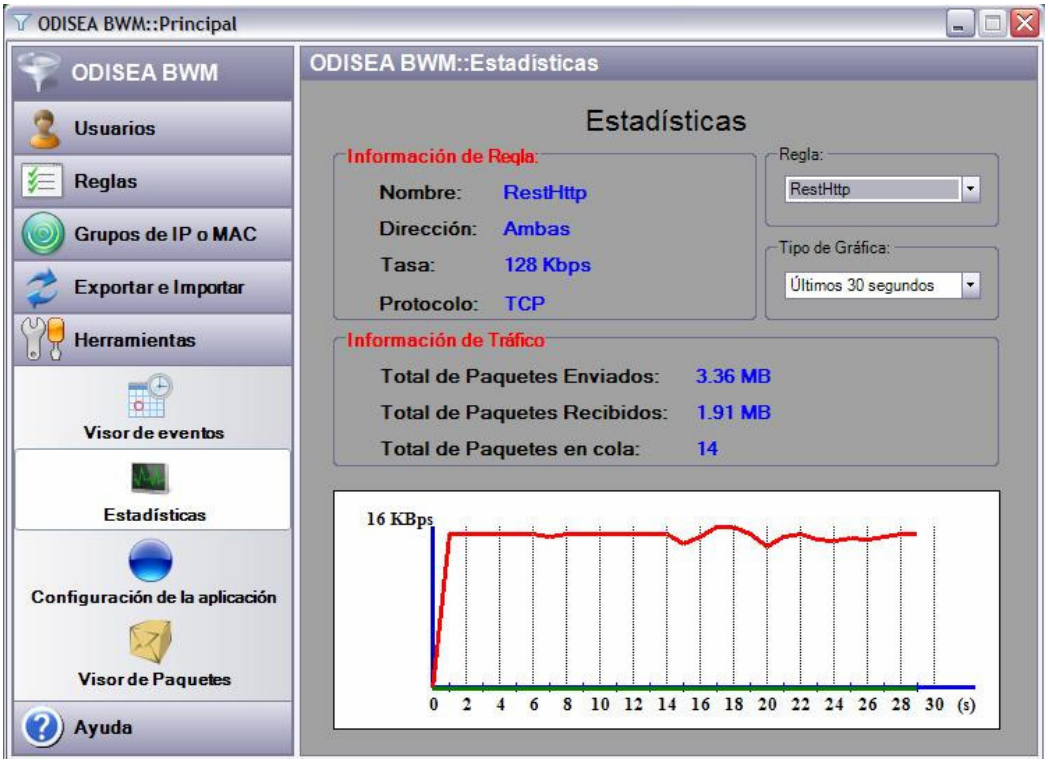

Figura 41. Estadísticas de la regla RestHttp en el administrador de ancho de banda mientras se ejecutan otras reglas al mismo tiempo.

Una vez finalizadas las pruebas se procedió a concluir con su respectiva documentación y a realizar pruebas con un tráfico más intenso, con más nodos accediendo al servidor simultáneamente. Utilizando prácticamente la misma topología pero añadiendo dos terminales cliente más en la red LAN, se realizaron múltiples descargas en cada una de las terminales y todas cumplían con las reglas de regulación utilizadas en el escenario anterior, con lo cual se puede concluir que el administrador de ancho de banda Odisea cumplió con las expectativas durante las pruebas.

6.9.2.4 Pruebas de flujo de videos.

Se realizó la prueba de flujo utilizando la misma topología planteada, solo que esta vez, en lugar de descargar los archivos vía web o FTP, se descarga el video en tiempo real.

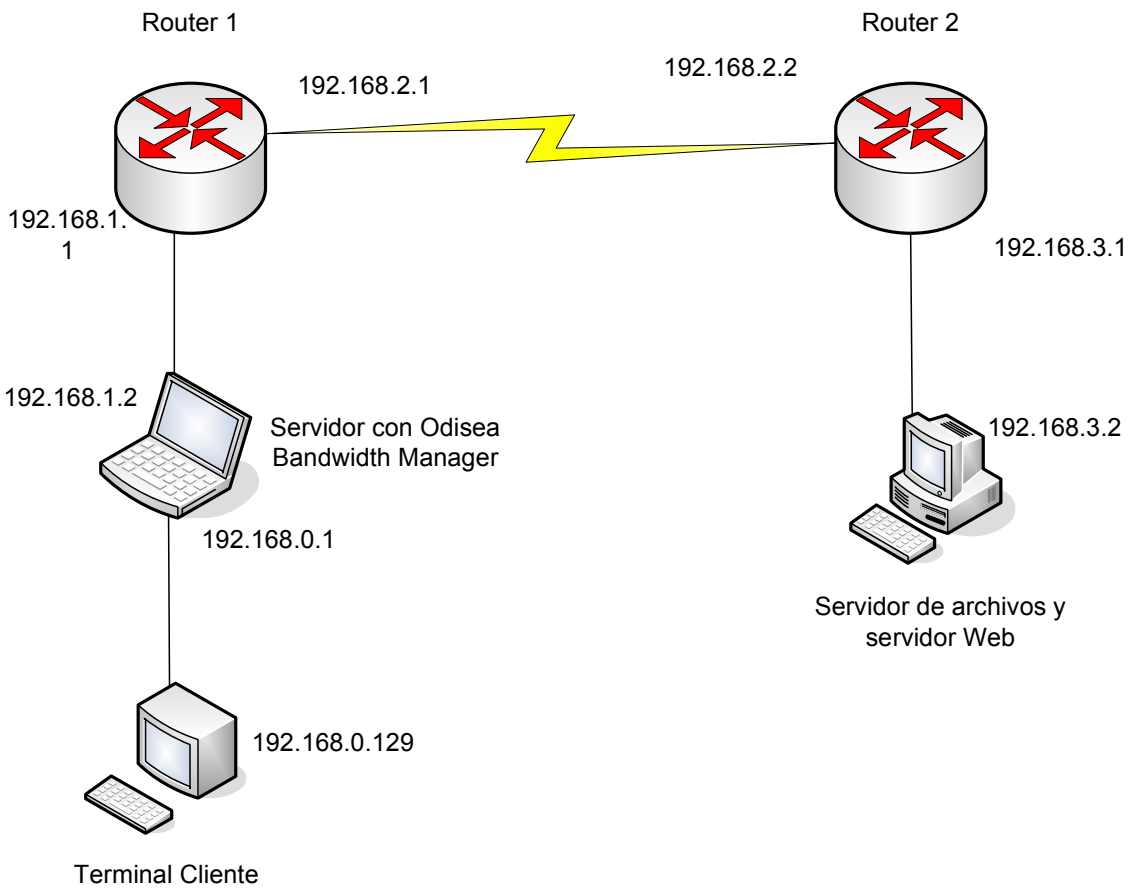

Figura 42. Topología de prueba empleada para el flujo de video.

La prueba consistió en reproducir un archivo de video, ubicado en el servidor web, utilizando el reproductor de Windows. Se utilizaron videos en formato mpg, wmv y wm.

La regla utilizada fue:

Nombre de la Regla: video

Dirección: Ambas

Tasa de transferencia: 256000 bps

Protocolo: TCP

IP Origen: 192.168.3.2

IP Destino: 192.168.0.129

Puerto Origen: Cualquier puerto

Puerto Destino: Cualquier puerto

Se utilizaron anchos de banda de 256 kbps, 512 kbps, 1 Mbps, 2 Mbps, 5 Mbps y 10 Mbps.

La prueba de flujo devolvió los resultados esperados: la calidad en la reproducción de los videos utilizados era muy pobre en las reglas de 256 y 512 kbps, era aceptable en la regla de 1 Mbps y de muy buena calidad en los de 2 y 5 Mbps. Al intentarlo con la regla de 10 Mbps, la aplicación tuvo un sobreflujo en la cola y colapsó. Esto sirvió para medir la capacidad que la aplicación puede alcanzar al regular tráficos bastante pesados.

6.9.2.5 Pruebas de jitter.

El objetivo de esta prueba era medir la latencia o jitter que la aplicación agregaba a los paquetes en una transmisión de datos. Para esto se utilizó un software llamado Iperf, el cual permite medir el jitter entre un nodo que actúa como cliente y uno que actúa como servidor. La topología utilizada es la misma que se ha establecido anteriormente.

La terminal cliente es donde se ejecuto la versión cliente de Iperf y en el servidor de archivos se ejecutó la versión servidor de Iperf. Para medir la latencia que la aplicación agregaba primero se midió la latencia que los paquetes tenían sin la aplicación corriendo, y después con la aplicación corriendo, de manera que la diferencia entre ambas latencias daría por resultado la latencia añadida por el sistema:

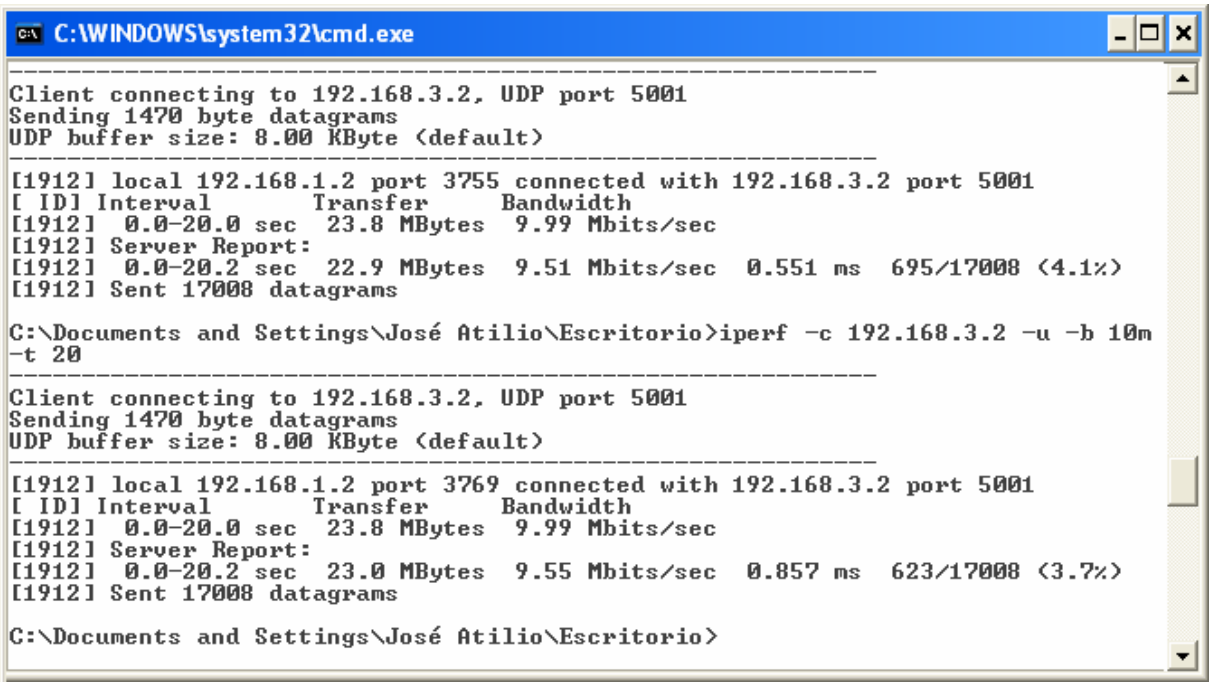

Figura 43. Ejecución de Iperf en la terminal cliente primero con la aplicación desactivada y después con la aplicación ejecutándose.

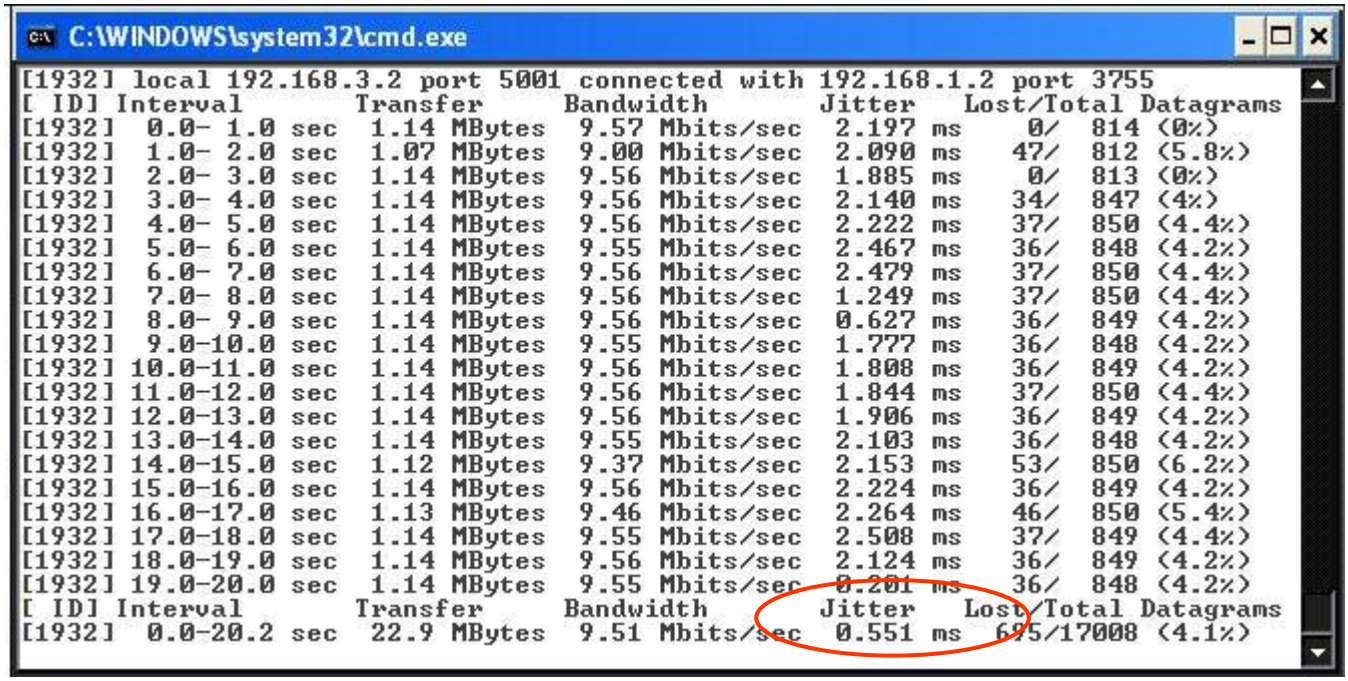

Figura 44. Resultado de Iperf **sin** la aplicación funcionando.

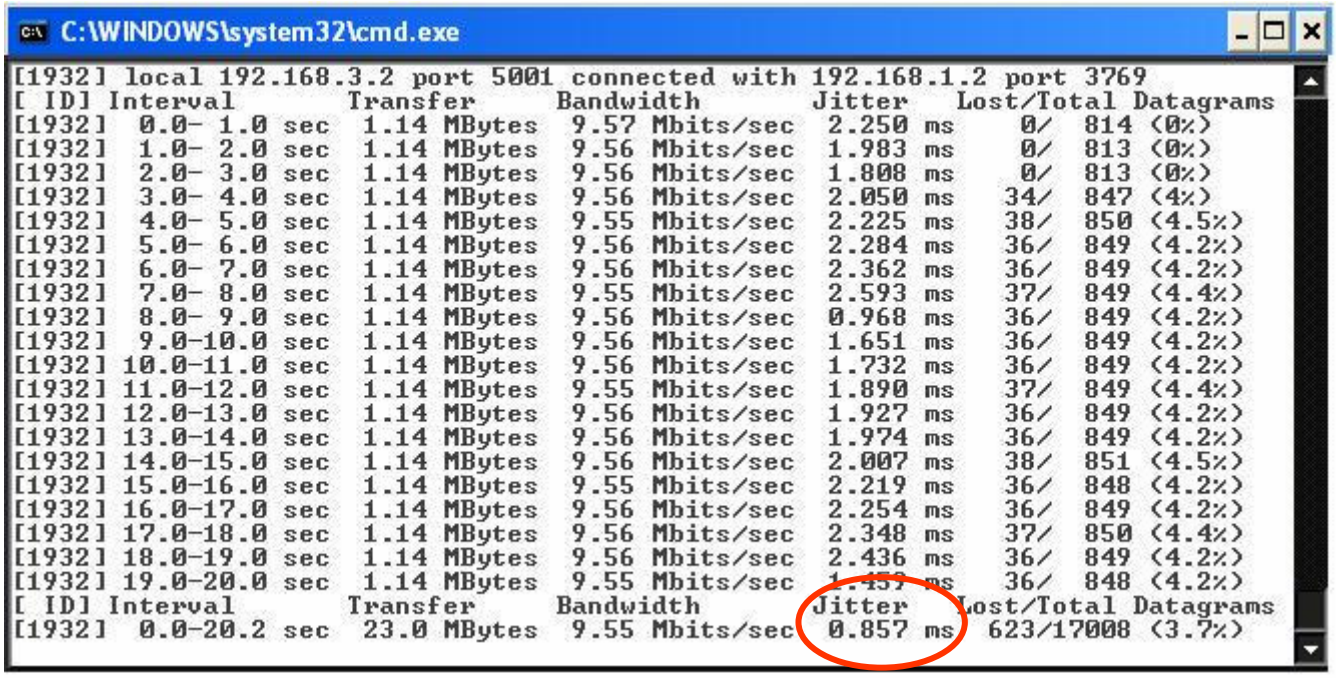

Figura 45. Resultado de Iperf **con** la aplicación funcionando.

De acuerdo a los resultados obtenidos en esta prueba, la latencia medida si la aplicación no se está ejecutando es en promedio de 0.551 ms y cuando la aplicación se ejecuta la latencia medida en promedio es de 0.857 ms. Por tanto:

Latencia Promedio de la Aplicación = 0.857ms -0.551ms= 0.306 ms

6.9.2.6 Pruebas con varias máquinas cliente.

Para determinar los requerimientos del sistema se realizaron pruebas con múltiples terminales generando tráfico en la aplicación. Para determinar cómo es afectado el uso del procesador con respecto al número de reglas utilizado, se decidió crear una regla por cada una de las terminales que utilizarían el tráfico.

La topología que se empleo fue la siguiente:

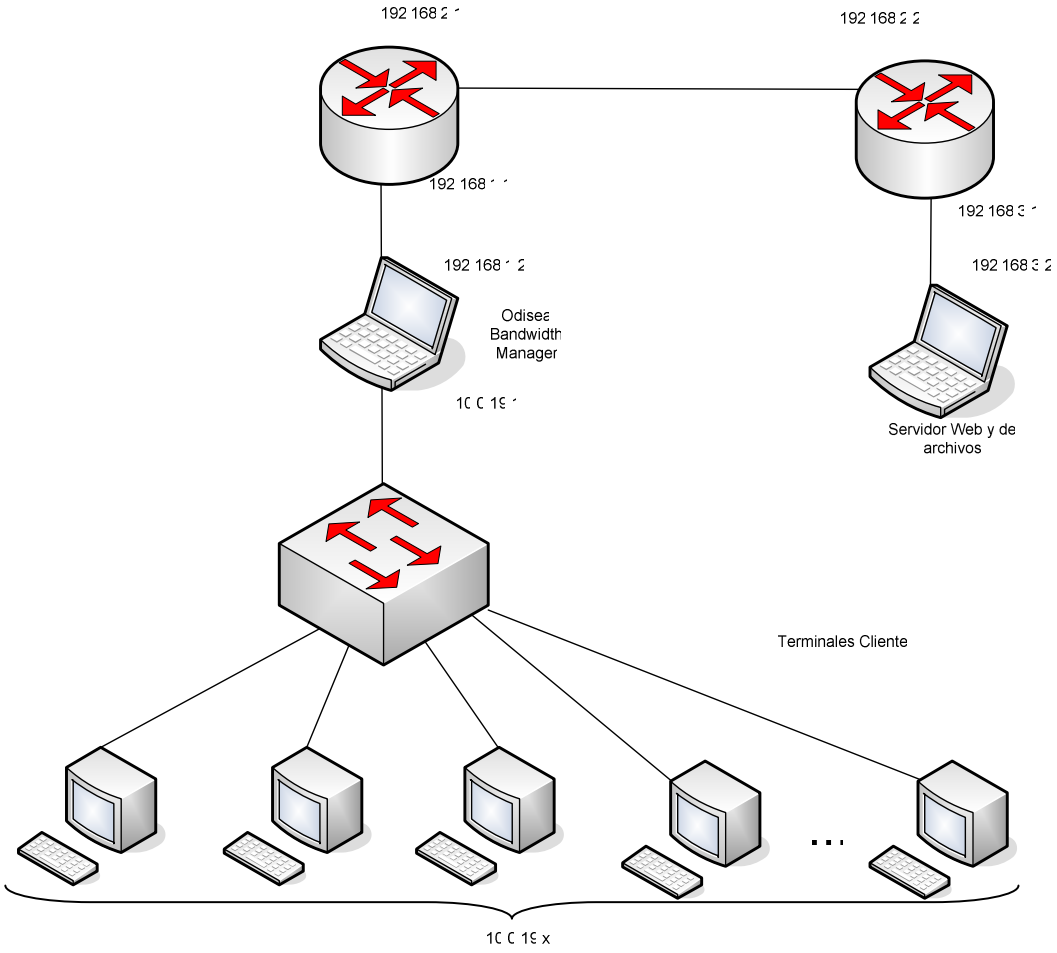

Figura 46. Topología de prueba con varios clientes.

Para esta prueba se utilizaron 36 terminales clientes, todas realizando descargas desde un servidor web utilizando el protocolo http.

La aplicación funcionó perfectamente administrando las 36 reglas incluidas en ella para cada una de las descargas. Así mismo el procesador del host que mantenía la aplicación se mantuvo oscilando entre 2 y 8% de uso.

Para someter la aplicación a un tráfico más intenso se realizó también una prueba de streaming de video en todas las terminales. Para esta prueba cada terminal tenía asignado un ancho de banda de 2 Mbps. El video tuvo una reproducción de muy buena calidad en las primeras 15 terminales que realizaron el streaming, sin

embargo las terminales 16-20 se comenzó a notar una degradación en la calidad del video y en el resto de terminales (21-36) el video tuvo una calidad bastante mal. Cabe recalcar que después de que la 17ª terminal inició el streaming el procesador del host que ejecutaba Odisea Bandwidth Manager alcanzó el 100% de su utilización.

CONCLUSIONES

De acuerdo a lo observado en las pruebas, se puede concluir que el sistema diseñado –Odisea Bandwidth Manager- administra efectivamente el tráfico de acceso de una red de alta velocidad a una de baja velocidad mediante reglas específicas para cada tráfico que se desea regular.

Durante el desarrollo de la aplicación se investigaron sobre las diferentes técnicas de calidad de servicios, entre ellas los métodos de encolamiento y los algoritmos de regulación de tráfico, tomándose los que, a criterio del grupo, eran los algoritmos que presentaban una efectividad aceptable y mayor facilidad de implementación.

Inicialmente se habían planteado las herramientas Qt y Visual C++ como las herramientas adecuadas para administrar el ancho de banda de la red, sin embargo, debido a ciertos problemas detectados durante el desarrollo del software, se decidió separar el módulo que gestiona el manejo de las políticas para la restricción del tráfico del módulo de regulación de tráfico. Para el primer módulo, se decidió utilizar la herramienta Visual Basic 2005, debido a que brinda más facilidades y una mayor estabilidad que la herramienta Qt. Para el módulo de administración de ancho de banda, se decidió implementarlo en Visual C++, utilizando la librería WinpkFilter para el manejo de los paquetes.

Se encontraron muchos inconvenientes a la hora de implementar la administración dinámica, debido a que no se podía saber con certeza el ancho de banda total del que se dispondría al momento de distribuirlo a las máquinas que se quisieran regular.

El sistema Odisea Bandwidth Manager es capaz de regular tráfico en base a los siguientes parámetros: direcciones de capa 3, aplicaciones, usuarios, protocolos y dirección MAC. Queda a criterio de cada usuario final de

la aplicación utilizar los criterios que más le convenga de acuerdo a la arquitectura de la red en la cual se implementará el sistema.

Basado las pruebas realizadas se puede concluir que Odisea Bandwidth Manager trabaja de forma adecuada en las siguientes condiciones:

- Límite de Reglas: El sistema ha sido probado con 36 Reglas, funcionando todas ellas de forma simultánea para tráfico HTTP Y FTP y no ha presentado problemas con estos tipos de tráfico. Asimismo, Odisea BWM ha sido probado con 15 host para los cuales es capaz de proveer una excelente regulación de tráfico de streaming.
- Número de host soportado: Odisea Bandwidth Manager es capaz de soportar más de 36 host cuando se trata de tráfico HTTP ó FTP y 15 hosts cuando se trate de tráfico de video.
- El procesador de 1.73 Mhz soporta perfectamente 15 reglas de streaming de video y 36 ó más reglas de tráfico HTTP y tráfico FTP.

RECOMENDACIONES

Para un funcionamiento óptimo del sistema pueden efectuarse ciertas mejoras, entre ellas:

Para fines prácticos, la aplicación fue realizada con herramientas que trabajan bajo el entorno de Windows, por lo que, para facilitar su arquitectura de implementación, podría realizarse una versión compatible con otros sistemas operativos como Linux o Mac, o estandarizar el sistema para que sea multiplataforma.

Como opción alternativa para la regulación del tráfico en base a la dirección de capa 2 (MAC) y en base a usuario, se podría realizar implementando un módulo que capte la información mediante SNMP, con lo que sólo bastaría habilitar el servicio en los clientes, y se obviaría el tener que instalar la aplicación extra del lado del cliente, tal como se encuentra actualmente.

Actualmente el sistema trabaja con el protocolo de capa de red IP versión 4, sin embargo puede añadírsele un módulo que le permita trabajar con la versión 6 de dicho protocolo, o bien permitir un modo híbrido en cuanto a versiones de IP.

El sistema está diseñado para trabajar sobre Ethernet, por lo que podría adicionársele la capacidad para trabajar sobre otras arquitecturas de red, como Token Ring, AppleTalk u otras.

HOJA TÉCNICA - ODISEA BANDWIDTH MANAGER.

Odisea Bandwidth Manager es un Administrador de ancho de banda que utiliza técnicas de encolado y Perfiles de Tráfico (Traffic Shaping) para optimizar el tráfico que circula entre una red de alta velocidad como una red LAN y una red de baja velocidad, por ejemplo Internet.

Odisea es ideal para empresas donde el mal uso de los recursos de Internet esté afectando el buen desempeño de las actividades normales de la empresa, ó esté influyendo de forma negativa en el desempeño de las actividades críticas que ahí se realizan. Este software supervisa el tráfico de red y limita el ancho de banda de acuerdo a los parámetros que el administrador de red le especifique. El resultado que proporciona es un aumento inmediato de la eficacia de la red en la que es implementado y una reducción en las necesidades de ancho de banda totales permitiendo a aplicaciones que utilizan Internet de forma crítica desempeñarse en una mejor manera.

Odisea Bandwidth Manager funciona en plataformas Windows, filtrando tráfico basado en reglas. Dichas reglas pueden especificar un límite de ancho de banda para cada tipo de tráfico y para cada usuario de Internet. Se pueden establecer reglas que limiten tráfico en base a puertos, direcciones IP, direcciones MAC, puertos y usuarios dentro de un domino, todo esto bajo una interfaz gráfica amigable y fácil de usar.

Tipo de aplicación:

- Aplicación de escritorio.

Plataforma de funcionamiento:

- Microsoft Windows NT / 2000 / XP / 2003.

Requisitos de software:

- Framework .Net 2.0
- Gestor de Base de Datos MySQL.
- Framework de WinpkFilter.
- El servidor donde se instalará la aplicación debe poseer como mínimo 2 Tarjetas de Interfaz de Red. Una conectada a la red de alta velocidad y otra conectada a la de baja velocidad.

Requerimientos de Hardware:

- Procesador: 3.2 GHz,
- Memoria RAM: 1 GB.
- Espacio en disco: 100 MB.
- Interfaces de Red; 2 Dispositivos Fast Ethernet (uno conectado a la red de alta velocidad y otro conectado a la red de baja velocidad).

Características principales:

- Configuración centralizada desde un punto único de la red, ya que trabaja en un servidor y no es necesario configurar las reglas en cada uno de los host que se va a regular.
- Permite exportar las reglas y grupos almacenados en la base de datos a un archivo XML.
- Incorpora un visor de encabezados de paquetes IP para monitorear el tráfico entrante y saliente de la aplicación.
- Uso de reglas bidireccionales para especificar el máximo de tasa de transferencia que se le asignará a cada host o grupo de hosts.
- Regulación de tráfico en base a puertos, direcciones IP, direcciones MAC, grupos de direcciones IP y MAC y usuarios dentro de un domino[22].
- La aplicación es transparente para el usuario final.

[22] Para la administración en base a direcciones MAC y usuarios dentro de un dominio es necesario instalar un programa cliente en cada una de los host que se deseen regular en esta forma.

- Ofrece un módulo de estadísticas detalladas para cada una de las reglas, así como un gráfico que muestra el comportamiento de cada regla en los últimos 30 segundos, 1 hora, último día, última semana y último mes.
- Límite de Reglas: El sistema ha sido probado con 36 Reglas, funcionando todas ellas de forma simultánea para tráfico HTTP y FTP. Asimismo, Odisea Bandwidth Manager ha sido probado con 15 host para los cuales es capaz de proveer una excelente regulación de tráfico de streaming.
- Número de host soportado: Odisea Bandwidth Manager es capaz de soportar más de 36 host cuando se trata de tráfico HTTP ó FTP y 15 hosts cuando se trate de tráfico de video.

Topología de instalación:

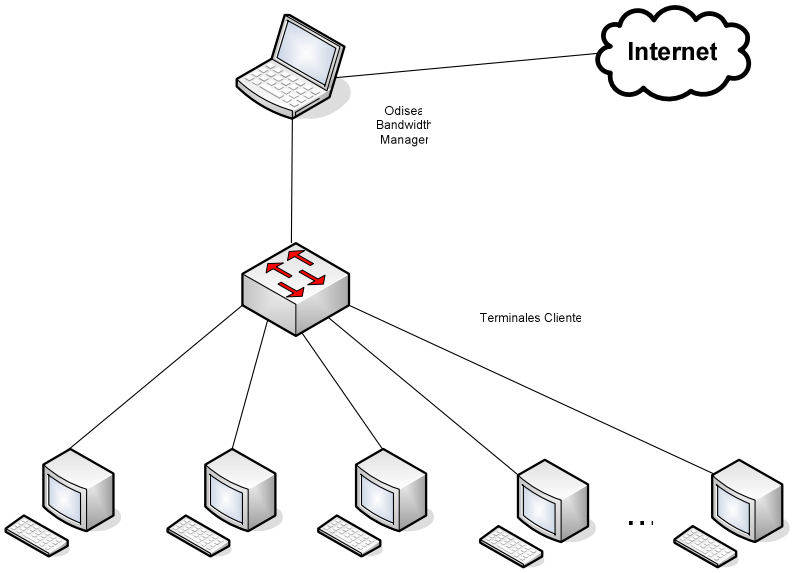

Figura 47. Topología de instalación de Odisea Bandwidth Manager.

Fuentes de Información

1 Bibliografía

1. CISCO SYSTEMS. CISCO IOS QUALITY OF SERVICE SOLUTIONS CONFIGURATION GUIDE. Cisco Systems, Inc. Corporate Headquarters. USA. 2001.
2. CISCO SYSTEMS. INTERNETWORKING TECHNOLOGIES HANDBOOK. Cisco Systems, Inc. Corporate Headquarters. USA. 2001.
3. ESCALANTE RENDÓN, José Isaac. "GUIA DE IMPLEMENTACIÓN DE REDES UTILIZANDO EL PROTOCOLO IPV6". Universidad Don Bosco, Escuela de Computación. 2005.
4. HERNÁNDEZ CERÓN, Francisco Alexander, Et al. "DISEÑO E IMPLEMENTACIÓN DE UN SIMULADOR DE REDES (SIR - 2003)". Universidad Don Bosco, Escuela de Computación. 2003.
5. ÁLVAREZ MORAGA, Sebastián y González Valenzuela, Agustín. ESTUDIO Y CONFIGURACIÓN DE CALIDAD DE SERVICIO PARA PROTOCOLOS IPV4 E IPV6 EN UNA RED DE FIBRA ÓPTICA WDM. Revista de la Facultad de Ingeniería, Universidad. Tarapacá, vol. 13 Nº 3, 2005. En línea: www.scielo.cl/pdf/rfacing/v13n3/art15.pdf
6. MING CHIU, et al. EXPERIENCES IN PROGRAMMING A TRAFFIC SHAPER. Sun Microsystems Inc. 1999.
7. LEE CHEN, Yng Jang. ALGORITMO DE PROCESAMIENTO EQUITATIVO DE FLUJOS DE DATOS EN ENRUTADORES INTERNET. Pontificia Universidad Católica de Chile Escuela De Ingeniería, 2005.

2 Revistas y artículos electrónicos.

1. Revista electrónica PC-News. VPN ó Redes Virtuales Privadas (Parte I). En línea: http://www.pc-news.com/detalle.asp ?sid=&id=11&Ida=1156. Julio 2003

2. Resumen de los mecanismos de QoS y cómo interoperan. Microsoft TechNet. En línea: http://www.microsoft.com/latam/technet/articulos/windows2k/qosmech/
3. MATEOS PAPIS, Alfredo Piero, et al. CONTROL DE ADMISIÓN EN REDES CON ARQUITECTURA DE DIFERENCIACIÓN DE SERVICIOS. ALGUNAS TENDENCIAS. Instituto Tecnológico Autónomo de México. México. En línea: www.ipv6.itam.mx/lafmi/admisionroc04.pdf
4. ESCRIBANO SALAZAR, Jorge, et al. DIFFSERV COMO SOLUCIÓN A LA PROVISIÓN DE QOS EN INTERNET. Departamento de Ingeniería Telemática Universidad Carlos III de Madrid. España. En línea: http://www.ist-mobydick.org/publications/cita2002.pdf
5. LEONARDO BALLIACHE. PRACTICAL QOS. OpalSoft, c.a. En línea: http://www.opalsoft.net/qos/index.html

3 Otros Sitios de Internet

1. http://www.glosarium.com/index.php. Web de diccionario y glosario de términos. Actualización: Diciembre de 2005.
2. http://millenniumindicators.un.org/. Base de Datos de indicadores de las naciones unidas. Actualización: Julio 2005
3. www.wikipedia.org. Enciclopedia en línea.

Glosario

A

Administración de red: Referencia al trabajo de administración, supervisión y control de una red. Los protocolos como el SNMP automatizan algunas de las tareas de supervisión y control.

Ancho de banda: Medida de capacidad de comunicación o velocidad de transmisión de datos de un circuito o canal

B

Backbone: Una línea de alta velocidad o una serie de conexiones que forman un mayor ancho de banda en una red. El término es relativo de un Backbone en una pequeña red, mucho más pequeña, que muchas líneas no backbones en una red grande.

Banda amplia: Ruta/circuito de comunicaciones de capacidad media. Suele indicar una velocidad de 64000 bps a 1544 Mbps.

Banda ancha: Ruta/circuito de comunicaciones de gran capacidad. Normalmente implica una velocidad superior a 1544 Mbps.

Bit: Cantidad de información más pequeña que puede transmitirse. Una combinación de bits puede indicar un carácter alfabético, un dígito, una señal, un modificador u otras funciones.

bits per second bps: Unidad de medida de la capacidad de transmisión de una línea de telecomunicación.

C

Cuello de botella: Límite en la capacidad del sistema que puede reducir el tráfico en condiciones de sobrecarga.

D

Dirección IP: Dirección de 32 bits del protocolo Internet asignada a un host. La dirección *IP* tiene un componente del host y un componente de la red.

E

Encapsulamiento: es el proceso por el cual los datos que se deben enviar a través de una red se deben colocar en paquetes que se puedan administrar y rastrear.

I

Internet Protocol (IP) (Protocolo Internet): Conjunto de reglas que regulan la transmisión de paquetes de datos a través de Internet. La versión actual es IPv4 mientras que en el proyecto Internet2 se intenta implementar la versión 6 (IPv6), que permitiría mejores prestaciones dentro del concepto QoS (Quality of Service).

L

Local Area Network (LAN): Red de Area Local. Red de datos para dar servicio a un área geográfica máxima de unos pocos kilómetros cuadrados, por lo cual pueden optimizarse los protocolos de señal de la red para llegar a velocidades de transmisión de Gbps (gigabits por segundo).

N

Network: Una red de computadoras es un sistema de comunicación de datos que conecta entre sí sistemas informáticos situados en lugares más o menos próximos. Puede estar compuesta por diferentes combinaciones de diversos tipos de redes.

Networking: Término utilizado para referirse a las redes de telecomunicaciones en general y a las conexiones entre ellas.

P

Peer-to-Peer (P2P): Comunicación bilateral exclusiva entre dos personas a través de Internet para el intercambio de información en general y de archivos en particular.

Protocol: Descripción formal de formatos de mensaje y de reglas que dos computadoras deben seguir para intercambiar dichos mensajes. Un protocolo puede describir detalles de bajo nivel de las interfaces máquina-a-máquina o intercambios de alto nivel entre programas de asignación de recursos.

Q

Quality of Service (QoS): Nivel de prestaciones de una red, basado en parámetros tales como velocidad de transmisión, nivel de retardo, rendimiento, horario, ratio de pérdida de paquetes.

Queue (cola): Conjunto de paquetes en espera de ser procesados.

R

Router: Dispositivo que distribuye tráfico entre redes. La decisión sobre a donde enviar los datos se realiza en base a información de nivel de red y tablas de direccionamiento.

S

Sniffer: Programa que busca una cadena numérica o de caracteres en los paquetes que atraviesan un nodo con objeto de conseguir alguna información. Normalmente se usa con fines ilegales.

Streaming video: Método de transmisión de imágenes en movimiento (por ejemplo, una película) a través de Internet. Las imágenes, que pueden ser pregrabadas o emitidas en directo y pueden ir acompañadas de sonido, se transmiten comprimidas para optimizar el tiempo de envío. El usuario, que debe contar con un programa de visualización de las mismas, normalmente integrado en su navegador, las recibe a medida que van llegando. Si las imágenes van con sonido, a este tipo de transmisión se le denomina streaming media.

T

TCP/IP: Protocolo de control de transmisiones/Protocolo Internet. Es el protocolo estándar de comunicaciones en red utilizado para conectar sistemas informáticos a través de Internet.

Tunneling: En Internet, este término se aplica al uso de la Red como parte de una red privada segura. El túnel es un conducto específico por el que viajan los mensajes o ficheros de una determinada empresa

V

Videoconferencia: Sistema de comunicación mediante el cual dos o mas personas situadas físicamente en distintos lugares pueden conversar y verse en vídeo a través de la Red.

W

Wide Area Network (WAN): Red de ordenadores conectados entre sí en un área geográfica relativamente extensa. Este tipo de redes suelen ser públicas, es decir, compartidas por muchos usuarios.