UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO MATEMATICAS

CALIDAD DE SERVICIO EN REDES DE COMPUTADORES

POR

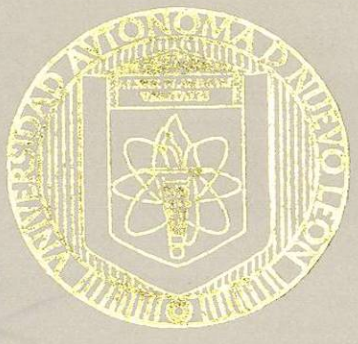
JUAN ANTONIO CASTILLEJA GARCIA

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE LICENCIADO EN CIENCIAS COMPUTACIONALES

MARZO 2003

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO MATEMATICAS



CALIDAD DE SERVICIO EN REDES DE COMPUTADORES

POR

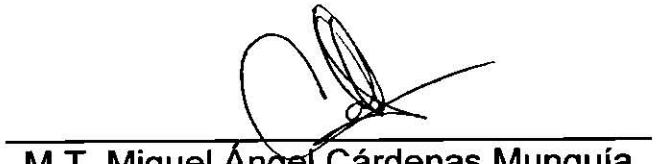
JUAN ANTONIO CASTILLEJA GARCIA

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE LICENCIADO EN CIENCIAS COMPUTACIONALES

MARZO 2003

CALIDAD DE SERVICIO EN REDES DE COMPUTADORES

Aprobación de la Tesis:



M.T. Miguel Ángel Cárdenas Munguía

PRESIDENTE

M.I. Irma Leticia Garza González

SECRETARIO

M.T. Jorge Hernández Baez

VOCAL

M.A. Carmen del Rosario de la Fuente García

DIRECTORA DE LA FACULTAD DE CIENCIAS FISICO-MATEMATICAS

RESUMEN

Juan Antonio Castilleja García **Fecha de de Graduación: Marzo, 2003**

Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Título del Estudio: CALIDAD DE SERVICIO EN REDES DE COMPUTADORES

Número de páginas: 243 **Candidato para el grado de Licenciado en Ciencias Computacionales**

Área de Estudio: Telecomunicaciones

Propósito y Método del Estudio: Las redes de computadores han ido evolucionado a pasos agigantados desde sus inicios; ésta evolución no ha sido solamente en usuarios conectados a ellas, si no también en los tipos de aplicaciones y servicios que se usan, lo cual ha traído a colación problemáticas en el rendimiento de las redes. La Calidad de Servicio es un concepto que intenta subsanar estos problemas mediante métodos y tecnologías que permitan una mejor utilización y aprovechamiento de los recursos finitos de las redes. Para lograr este fin, los mecanismos de encolamiento de los equipos de comunicaciones juegan un papel muy importante. En el presente trabajo se realizaron una serie de experimentos que nos permitieron medir parámetros que determinan la Calidad de Servicio, tales como el retraso, la variación en el retraso, el caudal eficaz y la pérdida de paquetes. Para la realización de este trabajo, se creó un ambiente de laboratorio para los diferentes escenarios de redes utilizados para llevar a cabo los experimentos; aunado a la experimentación se realizó un análisis formal de los datos obtenidos mediante un Análisis de Varianza, para demostrar la relación existente entre el retardo y los distintos mecanismos de encolamiento.

Conclusiones y Contribuciones: En el experimento realizado para el retardo, pudimos demostrar que al cambiar el mecanismo de encolamiento en los equipos de comunicaciones el retardo se ve afectado. Se pudo observar en los experimentos para los otros parámetros de medición cómo al aplicar distintos mecanismos de encolamiento los resultados se ven afectados, viendo que para ciertos escenarios los resultados obtenidos al modificar los mecanismos de encolamiento no son significativos. Se comprobó que es necesario implementar técnicas de Calidad de Servicio desde el primer equipo de comunicaciones y no sólo en el núcleo de la red.

Este trabajo contribuye en la parte didáctica ya que muestra una extensa explicación de la Calidad de Servicio y tecnologías para su implementación, así como un análisis de distintos mecanismos de encolamiento y sus implicaciones en las redes de cómputo.

En la parte experimental este trabajo representa una contribución al realizar un estudio formal sobre la relación entre el retraso y los mecanismos de encolamiento; este estudio aunque fue realizado en un ambiente de laboratorio, puede servir de base para la extrapolación de los resultados hacia un ambiente real.

FIRMA DEL ASESOR: _____

M.T. Miguel Ángel Cárdenas Munguía

AGRADECIMIENTOS

A mi asesor de tesis el M.T. Miguel Ángel Cárdenas Munguía. Gracias maestro, ya que gracias a lo interesante de sus clases me motivó a realizar este trabajo.

Al M.C. Sergio Arratia Dávila maestro de la Facultad, ya que sin su valioso apoyo en estadística y diseño de experimentos, parte de este trabajo no hubiera sido posible.

A la M.I. Irma Leticia Garza González directora de la academia de computación de la Facultad, por brindarme su apoyo, asesoría y sugerencias para la realización de este trabajo.

A todos los maestros que me impartieron clases, gracias por compartir su conocimiento. En especial a aquellos maestros que con sus cátedras además de adquirir conocimiento crecí enormemente como persona.

Al Ing. José Guadalupe Hernández E. encargado del Laboratorio de Interoperabilidad de la Dirección de Sistemas e Informática, por permitirme hacer uso de todo el equipo del Lab, además de su valiosa ayuda al realizar la revisión del texto y aportarme ideas con sus comentarios. Gracias además por dejarme "vivir" en la Lab durante todo el tiempo que estuve trabajando con mi tesis.

Al Dr. Roberto Mercado director de la Dirección de Sistemas e Informática de nuestra Universidad por el apoyo brindado.

Al Lic. Mario Rojas y al Lic. Rubén González que además de ser grandes amigos, sus comentarios fueron muy sustanciales para este trabajo.

A mi familia, a la cual descuide mucho durante el tiempo que estuve realizando mi trabajo, pero que sin embargo me comprendieron y me brindaron todo su apoyo. En especial a ti mamá que me aguantabas cuando llegaba malhumorado en las noches después de las largas jornadas de trabajo.

A mis amigos, los cuales compartieron tantos momentos buenos y malos conmigo a lo largo de mi carrera. En especial a Aline, César Yépez, Claudia, Lucy, Mario, Perla, Roberto, los cuales a lo largo de todo este trayecto siempre estuvieron ahí. Los puse en orden alfabético para que no haya sentimentalismos.

A los integrantes del Laboratorio de Interoperabilidad, por estar siempre al pendiente de mi trabajo, y por aguantarme todo este tiempo.

A la "Fuerza" por darme la salud y bienestar necesario para poder realizar este trabajo.

A todos aquellos que de una u otra manera contribuyeron con la realización de este trabajo.

DEDICATORIA

A mi familia, por siempre creer en mí y brindarme en todo momento su apoyo incondicional.

A mi padre, mi madre, mis hermanos Kelly y Ricardo, mis abuelos, mis tíos, primos, a todos ellos, ya que me han ayudado a crecer no sólo profesionalmente sino también como persona.

En especial a mi abuela Esperanza y a mi madre Margarita, ya que han sido un gran ejemplo para mí, y me han enseñado lo más preciado que puede aprender una persona: los valores.

TABLA DE CONTENIDO

Capítulo **Página**

Capítulo **Página**

Capítulo **Página**

Capítulo **Página**

LISTA DE TABLAS

Tabla **Página**

Tabla **Página**

Tabla **Página**

Tabla **Página**

Tabla **Página**

LISTA DE FIGURAS

Figura		Página

Figura **Página**

Figura **Página**

Figura **Página**

Figura **Página**

CAPITULO 1

INTRODUCCION

Antes de comenzar a hablar del término Calidad de Servicio, es muy importante el hacer una retrospectiva a las redes de computadores, el como nacieron y como han ido evolucionando a pasos agigantados hasta llegar a las redes actuales, que soportan múltiples servicios (comercio electrónico, educación en línea, etc.), tráfico de distintos tipos (voz, datos y video) y que además están soportados en ambientes heterogéneos.

Esto lo hago ya que probablemente, debido al rápido crecimiento de las redes de cómputo y telecomunicaciones, no nos hemos detenido a observar la monstruosa manera como han crecido, tanto en capacidad, como en servicios soportados, y no se diga en cantidad de usuarios conectados a ellas.

Así que demos un breve paseo por la historia y vayamos viendo como hemos llegado hasta el punto en que nos encontramos hoy en día, el cuál, gracias a los constantes avances tecnológicos, esto será historia el día de mañana.

1.1 Evolución de las Redes de Comunicaciones

Para ir viendo como han ido evolucionando las redes de comunicaciones, veamos una cronología [1].

1.1.1 Siglo XIX.

En el siglo XIX, comenzaron a surgir los inventos que revolucionaron las comunicaciones de esa época, por una parte en 1836, Cooke y Wheatstone patentaron el telégrafo, lo cual dío inicio a las telecomunicaciones.

Un suceso impactante, fue la instalación de un cable trasatlántico entre 1858 y 1866, lo cuál permitía comunicación directa a través del Océano Atlántico y unió a los dos mundos.

Por otro lado en 1860 Antonio Meucci inventó el teléfono [2], lo cual vino a complementar los servicios de comunicaciones de esa época. Todo esto sirvió como punta de lanza para que se iniciara el "boom" de las telecomunicaciones.

1.1.2 1950's.

Durante la década de los cincuentas, hubo dos sucesos importantes que marcaron el inicio de la investigación sobre ciencia y tecnología. En 1957 la URSS lanza al espacio el "Sputnik", que fue el primer satélite artificial de la tierra. En respuesta a esto el 7 de Febrero de 1958 los Estados Unidos crean la Advanced Research Project Agency (ARPA) en conjunto con el Departamento

de Defensa (DoD), cuya principal meta era ser líderes en ciencia y tecnología aplicable a lo militar.

La creación de esta agencia marca un paso importante de los Estados Unidos para convertirse en el líder, lo cual provoca que se formalicen los estudios científicos y beneficien en primera instancia a los proyectos militares, pero que tiempo después fueron marcando la pauta a las tecnologías que utilizamos hoy en día.

1.1.3 1960´s.

Durante la década de los sesenta hubo una serie de sucesos, que culminaron con el nacimiento de la Internet.

En 1961, Leonard Kleinrock del MIT (Massachussets Institute of Technology) presenta el primer documento sobre la teoría de conmutación de paquetes, denominado: "Information Flow in Large Communications Nets", lo cual inicia el estudio es esta área. Tres años más tarde Paul Baran publica "On Distributed Communications Networks", sobre las redes de conmutación de paquetes, lo cual confirma el uso de esta tecnología para las redes de computadores.

En 1966, Lawrence G. Roberts del MIT publica "Towards a Cooperative Network of Time-Shared Computers", lo que inicia el primer plan sobre ARPANET (Advanced Research Projects Agency Network). En Abril de 1967

durante la reunión de la ARPA en Ann Arbor, Michigan, Larry Roberts encabeza el grupo de discusión para el diseño de la ARPANET.

Durante 1968, se presenta la red conmutada por paquetes (PS Network) ante la ARPA. Mientras tanto, El Network Working Group (NWG, Grupo de trabajo de redes), encabezado por Steve Crocker, se organiza a fin de desarrollar protocolos a nivel terminal para establecer comunicaciones en ARPANET.

Después de muchos trabajos, en 1969 el Departamento de Defensa de los Estados Unidos designa al grupo ARPANET para la investigación sobre redes de computadores. Este mismo año se crea el primer nodo en la Universidad de California en Los Angeles (UCLA), seguidos por otros tres nodos en el Instituto de Investigación de Standford, la Universidad de California en Santa Bárbara y en la Universidad de Utha. Esta interconexión da inicio a lo que hoy conocemos como Internet.

1.1.4 1970's.

Durante está década, comenzaron a surgir los primeros programas de aplicación para la red. En 1971 Ray Tomlinson inventa un programa de correo electrónico, y al año siguiente se elige el símbolo "@" para representar el "en" en las direcciones de correo. Ese mismo año, se publica el RFC 318 el cual contiene la especificación para el Telnet.

El 23 de Marzo de 1972 la junta directiva del Departamento de Defensa cambia el nombre de la ARPA al nombre de Defense Advanced Research Projects Agency (DARPA). DARPA es creada como una agencia de defensa separada bajo la supervisión de la Oficina de la Secretaría de Defensa de los Estados Unidos.

En el año de 1973, se realizaron las primeras conexiones internacionales a la ARPANET, conectándose el University College of London de Inglaterra y el Royal Radar Establishment de Noruega. Este mismo año se publican más RFC con especificaciones para protocolos de aplicación, siendo los más importantes el RFC 454 que contiene la especificación para la transferencia de archivos, y el Protocolo de Voz en Redes (RFC 741), el cuál sugería hacer llamadas de conferencia a través de la ARPANET. Este protocolo marca la pauta y los intentos por no solo ofrecer servicios de datos a través de las redes.

Un estudio de la ARPA en 1973 demuestra que el 75% del tráfico total de la ARPANET lo genera el correo electrónico.

En 1974 Vinton Cerf y Bob Kahn publican el "Protocolo para Interconexión de Redes por paquetes" ("A Protocol for Packet Network Interconnection") que especifica en detalle el diseño del Protocolo de Control de Transmisión (TCP). Mas tarde, en 1978, este protocolo se divide a como lo conocemos hoy en día: TCP e IP.

Durante 1979 comienzan experimentos sobre redes de radio financiadas por la DARPA, surgiendo la red Packet Radio Network (PRNET).

Así, la década de los setentas transcurrió con avances principalmente en las aplicaciones, las cuales permitían darle un uso a la red recién nacida, siendo el correo electrónico el principal uso de la red.

1.1.5 1980´s.

Al comienzo de los ochentas, en 1981 se crea la CSNET (Computer Science Network, Red de las Ciencias de la Computación) se crea gracias a la colaboración de expertos en computación de la Universidad de Delaware, la Universidad Purdue, La Universidad de Wisconsin, RAND Corporation y BBN financiado por la NSF (Nacional Science Foundation) con el objetivo de prestar servicios de red (especialmente de correo electrónico) a los científicos que carecían de acceso a la ARPANET. Más tarde la CSNET se conocería como la Red de Computación y Ciencia.

En 1982 se establece el Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP) como el conjunto de protocolos -conocido comúnmente como TCP/IP- para ARPANET. Esto genera una de las primeras definiciones de "internet" como: "una serie de redes conectadas entre sí, específicamente aquellas que utilizan el protocolo TCP/IP" y se utiliza el término "Internet" como: "internets TCP/IP interconectadas". Debido a esto, el Departamento de Defensa establece que el protocolo TCP/IP será el estándar

para ese organismo, lo que provoca que cada dispositivo que se le quisiera vender a este departamento tuviera que incluir este juego de protocolos.

A principios de 1983 nace en Europa la Movement Information Net (MINET), conectándose a Internet en Septiembre. Este mismo año, la ARPANET se divide en ARPANET y en MILNET (Militar Network), esta última se integra con la Red de Información de Defensa. Un año después, se crean mas redes alrededor del mundo, estableciéndose en Japón la Japan Unix Network (JUNET), y en el Reino Unido se crea la Joint Acedemic Network (JANET).

Ese mismo año, la cantidad de anfitriones conectados superaba los mil, lo cual hizo que se introdujera el Sistema de Nombre de Dominio (DNS) debido a lo difícil que resultaba identificarlos a través de su dirección física por la cantidad de nodos conectados

En 1986 se crea la National Science Foundation Network (NSFNET), con una velocidad en el backbone de 56Kbps. NSFNET conectaba 5 centros de super cómputo para proveer un mayor poder de cómputo. Esto permitió una mayor oportunidad de conectividad, principalmente para las universidades. La red NSFNET se unió también a la ARPANET.

Para 1987 la cantidad de anfitriones superaba los 10,000. El backbone de la NSFNET crece a un T1 (1.544Mbps). Ya para 1989 la cantidad de anfitriones superaba los 100,000 lo cual es un incremento del 900% en tan solo dos años.

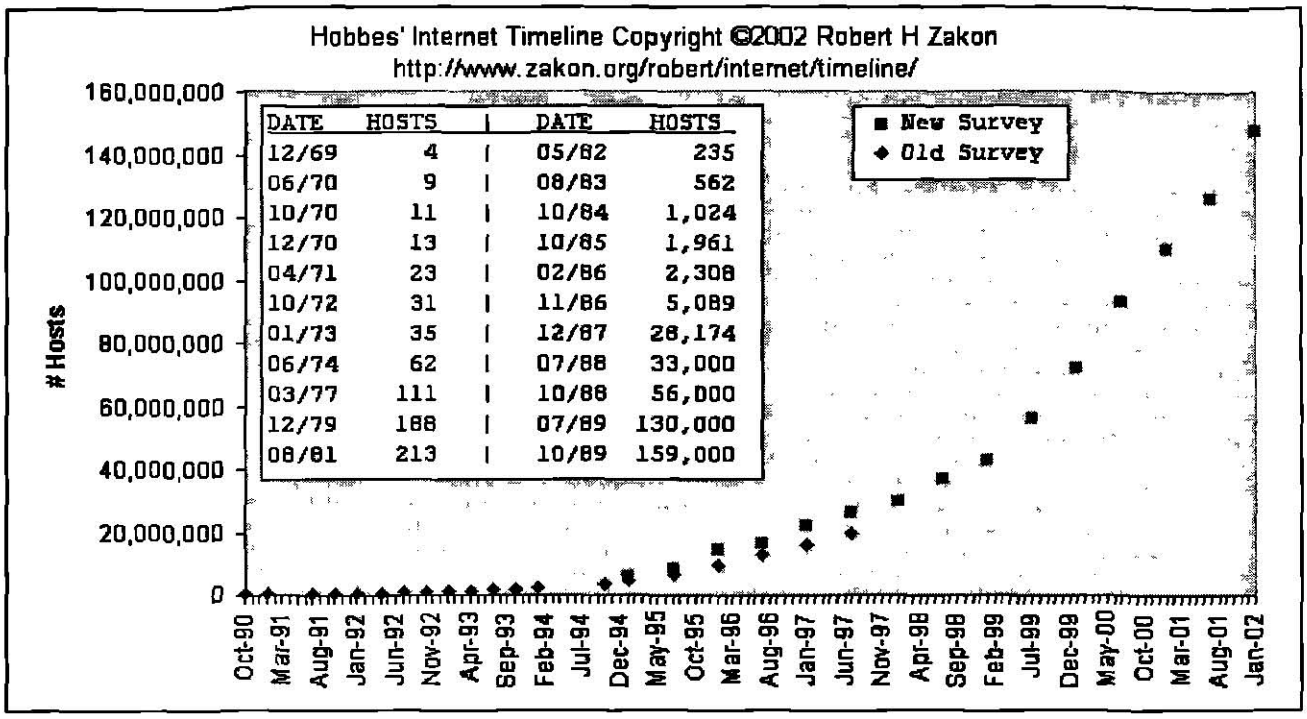
La historia del Internet en México empieza en el año de 1989 con la conexión del Instituto Tecnológico y de Estudios Superiores de Monterrey, en el Campus Monterrey hacia la Universidad de Texas en San Antonio (UTSA), *específicamente a la escuela de Medicina. Una línea privada analógica de 4* hilos a 9600 bits por segundo fue el enlace.

Así la década de los ochentas termina con grandes avances en la consolidación de redes a lo largo del mundo, lo cual permitió que la Internet se expandiera, hecho que fue notorio al ver la cantidad de nodos conectados a finales de la década y el impresionante ritmo al cual crecían.

1.1.6 1990´s.

Antes de la década de los 90´s, el tráfico que tenía la red era principalmente tráfico de correo electrónico y de transferencia de archivos, lo cual no significaba gran problema de ancho de banda, aunque los canales de comunicación eran limitados estos no se veían comprometidos por el tipo de tráfico que transportaban.

A medida que la tecnología fue evolucionando, se crean servicios como el World Wide Web (1991), que provoco que el número de usuarios creciera impresionantemente como muestra la Figura 1.



DATE	HOSTS	DATE	HOSTS
12/69	4	05/82	235
06/70	9	08/83	562
10/70	11	10/84	1,024
12/70	13	10/85	1,961
04/71	23	02/86	2,308
10/72	31	11/86	5,089
01/73	35	12/87	28,174
06/74	62	07/88	33,000
03/77	111	10/88	56,000
12/79	188	07/89	130,000
08/81	213	10/89	159,000

Figura 1. Crecimiento del número de conexiones a Internet en la década de los 90's.

Una vez que la tecnología WWW comenzó a evolucionar, empezaron a surgir nuevas y sofisticadas tecnologías que complementaban los servicios que podían ofrecerse en el WWW; así en 1995 la compañía Sun Microsystems lanza al mercado su lenguaje Java, el cual permite a los usuarios del WWW tener contenido con movimiento, desplegando texto, imágenes y sonido en una misma página. Esto causó revuelo por parte de los usuarios, pero a la vez trajo consigo problemas de ancho de banda, ya que los contenidos de las páginas cada vez contaban con más elementos multimedia. Ese mismo año, la compañía Real Networks lanza sus productos para transmitir audio (y más tarde video) a través de Internet, lo cual acrecentó el problema al incrementarse el consumo de recursos de las redes.

Como hemos visto, el tipo de contenido que se transmitía por las redes fue evolucionando, aunado al crecimiento explosivo de usuarios de la red, lo cual ha traído consigo problemas severos de ancho de banda, los cuales se van incrementando conforme pasa el tiempo, ya que como vemos en la figura 2, el número de usuarios en la red sigue creciendo minuto a minuto.

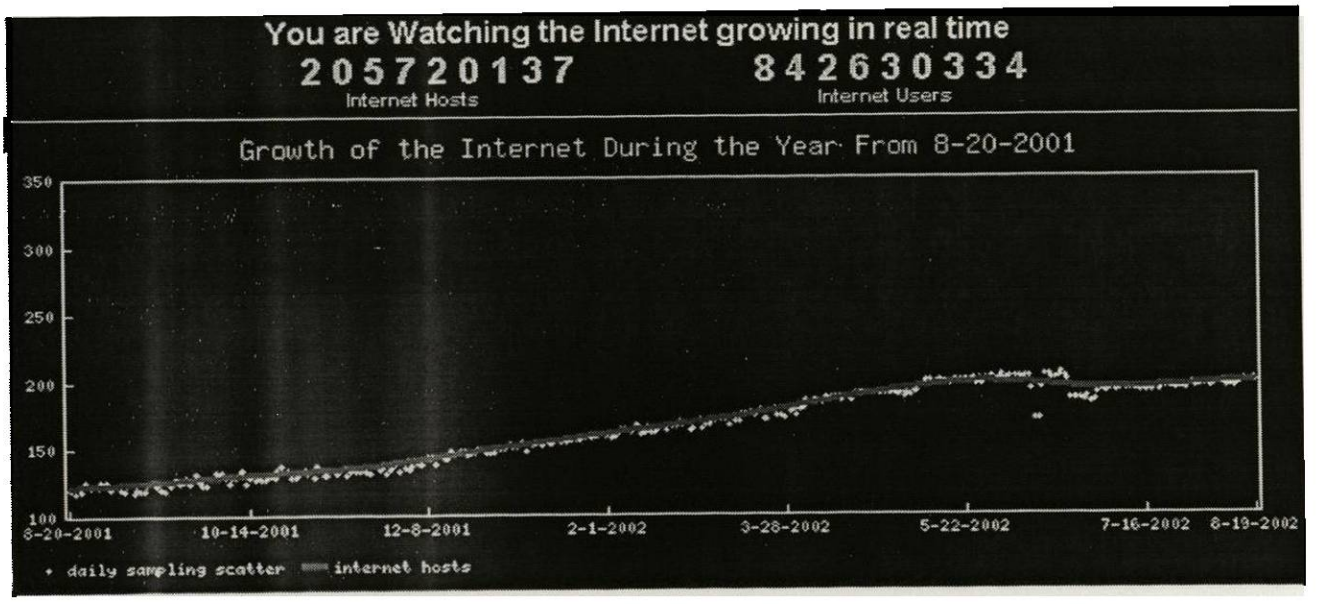


Figura 2. Equipos y usuarios conectados a Internet en la actualidad.

1.2 Panorama Actual (Conclusiones)

Hoy en día, las necesidades de comunicaciones en los usuarios han ido aumentado, hasta tal punto que no solo los grandes corporativos usan las redes de datos para comunicarse, ya que en algunos hogares es factible contar con servicios de voz sobre IP, y conforme las capacidades de ancho de banda en los hogares crezca se introducirán servicios como el video. Esto trae consigo

que las redes tienen que soportar el tránsito de tráfico isócrono, los cuales son sumamente sensibles a los retardos.

Así pues, cada vez hay más usuarios en la red, y cada vez se introducen nuevos servicios y tecnologías que requieren ancho de banda, el cual lamentablemente, no es infinito. Esto hace que no solo tenemos que pensar en crecer nuestro ancho de banda para poder soportar los servicios, si no mas bien, debemos de pensar en como administrar el tráfico que hacemos circular por nuestros enlaces de comunicaciones para que podamos tener un mejor control de los recursos que disponemos.

Esto ha preocupado a los grupos de trabajo de Internet, y desde la década de los 90´s se ha comenzado a trabajar en técnicas y mecanismos para la administración de nuestras redes, lo cuál nos beneficia ampliamente para permitirnos seguir visualizando el tipo de contenido al que estamos acostumbrados hoy en día con una Calidad de Servicio acorde a nuestras necesidades.

CAPITULO 2

MODELOS DE REFERENCIA

Para comenzar a hablar de Calidad de Servicio, y de las técnicas y protocolos involucrados para lograrla, es necesario dar una vista rápida a los modelos de referencia existentes, ya que sobre estos están basados los procesos de comunicaciones. También se presenta una discusión acerca de los dos modelos mas utilizados: OSI y TCP/IP.

Este vistazo es necesario, ya que los modelos mencionados no fueron diseñados originalmente para soportar Calidad de Servicio, pero el propio diseño de éstos permitió a los diseñadores ir integrando nuevos protocolos los cuales basan sus servicios en el modelo original.

Para poder establecer una comunicación entre dos equipos en una red, hay que establecer un modelo de comunicación, el cual nos permita que el proceso de comunicación se realice aún sobre ambientes heterogéneos.

Para este fin son de suma importancia los dos conceptos siguientes [3]:

- Los protocolos
- Las arquitecturas de comunicaciones entre computadoras

Para que dos entidades se comuniquen con éxito es necesario que "hablen" el mismo idioma. El qué se comunica, cómo se comunica y cuándo se comunica debe de seguir un una serie de convenciones mutuamente aceptadas por las entidades involucradas. Este conjunto de convenios se denominan protocolos, que se pueden definir como: el conjunto de reglas que gobiernan el intercambio de datos entre dos entidades.

Para que el proceso de comunicación se lleve a cabo de manera efectiva, es necesario que exista un alto grado de cooperación entre las computadoras. En lugar de implementar toda la lógica para llevar a cabo la comunicación en un único módulo, esta tarea se divide en subtareas, donde cada una se realiza por separado. Así definimos a una arquitectura de comunicaciones como: una estructura consistente compuesta por un conjunto de módulos que realizarán las funciones para llevar a cabo la comunicación.

El objetivo de crear modelos de comunicaciones basados en protocolos y arquitecturas de comunicaciones es obtener un esquema diseñado por capas el cual utiliza servicios jerárquicos. Esta estratificación por capas nos permite hacer una diferencia de servicios y funcionalidades, los cuales a su vez, nos permiten crear mecanismos de presentación de servicios al usuario final.

Ahora veremos un modelo sencillo de comunicaciones basado en tres capas principales, para después analizar los modelos OSI y TCP/IP, los cuales se basan en este modelo.

2.1 Modelo de Tres Capas.

Para realizar el proceso de comunicación son necesarios, de manera muy general, tres elementos involucrados: software de aplicación, computadoras y redes. Si nos basamos en esta premisa nos damos cuenta que se puede diseñar un modelo de comunicaciones el cual maneje de manera independiente cada uno de estos tres elementos, y en donde cada capa realiza las funciones y control necesario para llevar a cabo la tarea de ese elemento; además debe ser capaz de comunicarse con las demás capas para intercambiar parámetros, o bien, los datos. Bajo este concepto definimos nuestro modelo básico de tres capas.

Para ir analizando cada una de estas capas veremos el proceso basándonos en la figura 3.

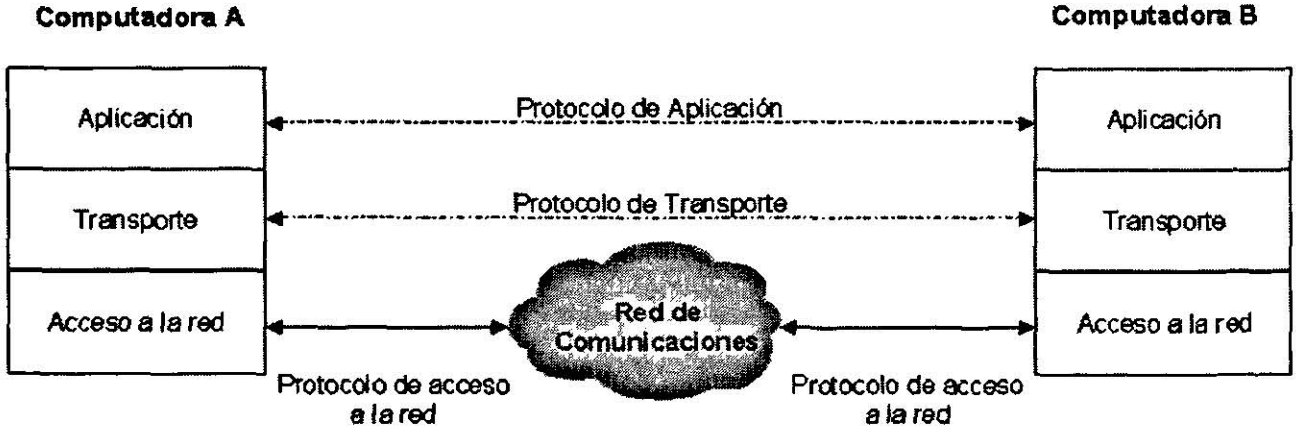


Figura 3. Modelo de tres capas.

Bajo este esquema, cada computadora contiene software en las capas de acceso a la red, en la capa de transporte y en la capa de aplicación. Para que la comunicación pueda realizarse se necesitan dos niveles de direccionamiento. La primera, un direccionamiento físico, en el cual cada computadora o elemento de la red debe tener una dirección única, que le permita a la red entregar los datos a la computadora apropiada. La segunda, es un direccionamiento lógico, en la cual cada aplicación debe tener una dirección única dentro de la computadora, que le permite a la capa de transporte entregar los datos a la aplicación apropiada. Estas últimas direcciones son denominadas Puntos de Acceso al Servicio (SAP, Service Access Point).

Las funciones que deben realizar cada capa de este modelo son las siguientes:

- *Capa de Aplicación.*- La capa de aplicaciones debe contener la lógica necesaria para soportar varias aplicaciones del usuario. Como estas aplicaciones pueden realizar distintas tareas, se debe tener un módulo independiente y con características bien definidas para cada una de las aplicaciones.

- *Capa de transporte.*- Esta capa se encarga del intercambio seguro de los datos, independientemente de las aplicaciones que estén intercambiando datos. Esto es, se implementan en este módulo los algoritmos y técnicas necesarias para la corrección y detección de errores en la transmisión para asegurarse que todos los datos

enviados lleguen a su destino, así como que lleguen en el orden enviado. Si esto no ocurre, este módulo debe tener los mecanismos necesarios para informar al emisor los errores ocurridos en la transmisión del flujo de datos para que este reenvíe los datos que causaron error.

- *Capa de acceso a la red.*- Esta capa es la encargada del intercambio de información entre la computadora y la red a la que se está conectado. Las funciones y características del software de esta capa dependen directamente del tipo de red que se esté utilizando, así se han desarrollado diversos estándares para conmutación de paquetes, conmutación de circuitos, redes de área local, etcétera.

El objetivo de esta capa es separar en un módulo independiente las funciones que tienen que ver con el acceso a la red, logrando con esto que el software de las capas superiores no tenga que preocuparse por que tipo de red se esta utilizando. En esta capa se pueden implementar servicios de acceso a la red y mecanismos de Calidad de Servicio tales como prioritización, los cuales veremos a detalle en el capítulo III.

En cualquier modelo de transmisión de datos es necesario agregar información de control adicional a los datos del usuario, esto para que los equipos puedan controlar la información que reciben. Para esto, se agregan datos de control llamados cabeceras, donde cada capa del modelo agrega una cabecera de control además de los datos del usuario. A la unión de los datos de

la cabecera de la capa actual con los datos de la capa superior se denomina Unidad de Datos del Protocolo (PDU, Protocol Data Unit). Como vemos en la figura 4 en la sección de datos de la capa actual está la información de la capa anterior, tanto de la cabecera como de los datos. A este concepto se le llama encapsulación.

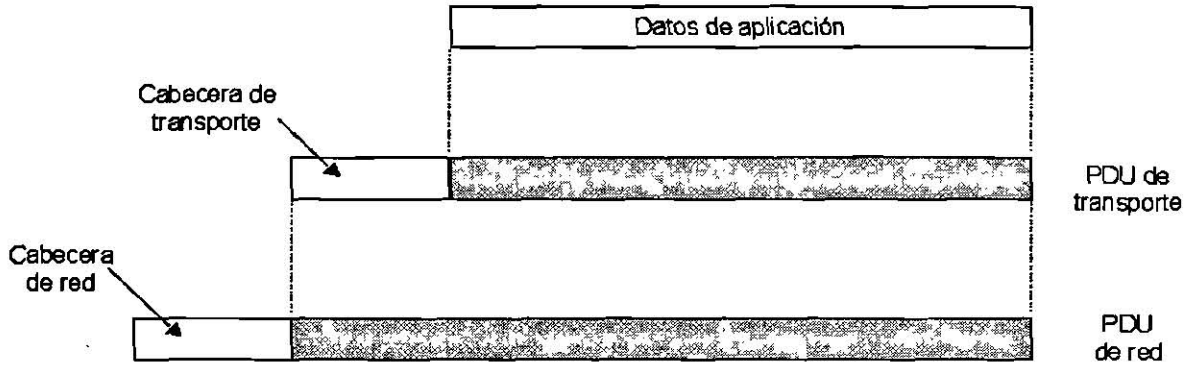


Figura 4. Encapsulación de datos entre capas.

Cuando el PDU de red llega a su destino el computador lee la cabecera de red, si esa es la máquina destino elimina la cabecera de red y pasa la sección de datos de esa capa a la capa superior, donde ésta lee su cabecera y realiza las acciones pertinentes para después pasar solo la sección de datos a la capa superior siguiente. A este proceso se le llama desencapsulación.

Cabe mencionar que cada capa en el emisor se comunica con su misma capa de la máquina destino en un proceso conocido como *comunicación igual-a-igual (peer-to-peer)*.

Este modelo de tres capas no esta diseñado para ser implementado, ya que solo contempla los aspectos básicos de la comunicación de datos, pero sirve de guía para el diseño de modelos más complejos e implementables.

A continuación veremos los dos modelos más populares, el modelo OSI y la arquitectura de protocolos TCP/IP, los cuales se basan en el modelo descrito anteriormente.

2.2 Arquitectura de Protocolos TCP/IP

La arquitectura de protocolos TCP/IP nace cuando en 1969 Vinton Cerf y Robert Kahn publican un documento llamado "Protocolo para la Interconexión de Redes de Paquetes" el cual describía el protocolo TCP; mas tarde se dieron cuenta que era mejor dividir el protocolo para poderlo implementar de manera más simple, por lo que decidieron separarlo en dos protocolos, el Protocolo de Internet (IP, Internet Protocol), y el Protocolo de Control de Transmisión (TCP, Transmission Control Protocol).

Estos protocolos surgen como una evolución del Protocolo de Control de Red (NCP, Network Control Protocol) el cual se utilizó en los inicios de la ARPANET. Debido a que otras redes comenzaron a conectarse con ARPANET era necesario adoptar un protocolo que pudiera mantener el orden aún con la creciente comunidad de usuarios. Ya que TCP era más rápido, fácil de usar y menos costoso de implementar que NCP, se eligió TCP/IP para este fin.

El hecho de que el Departamento de Defensa (DoD) de Estados Unidos haya establecido una política para que cualquier equipo o software de redes que se comprara por este departamento debería de incluir la familia de protocolos TCP/IP hizo que éstos protocolos hayan logrado erguirse como la arquitectura dominante, logrando imponerse a arquitecturas propietarias como SNA (Systems Network Architecture, Arquitectura de Sistemas de Red) de IBM.

Hoy en día TCP/IP es la arquitectura más utilizada, además que la Internet está construida sobre este conjunto de protocolos, lo cual hace que mayoría de los desarrollos de protocolos futuros se basan en esta arquitectura. Por esta razón la mayoría de los temas tratados en los capítulos posteriores estarán basados en TCP/IP.

2.2.1 Modelo TCP/IP.

A diferencia del modelo OSI, no existe un modelo de referencia oficial de TCP/IP, sin embargo basándonos en los protocolos que se han desarrollado bajo esta arquitectura, se puede esquematizar el proceso de comunicación en cinco capas relativamente independientes (figura 5).

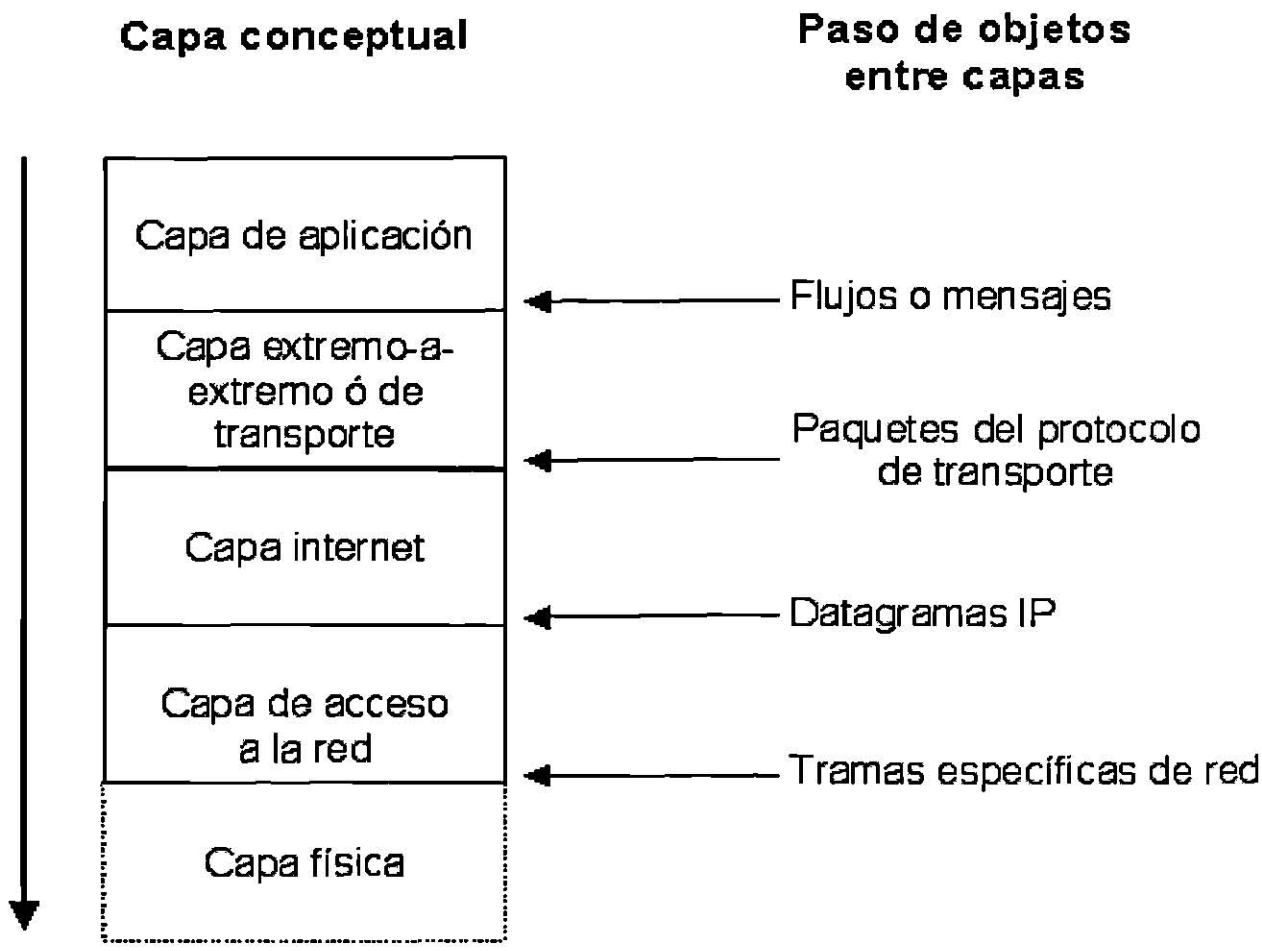


Figura 5. Modelo TCP/IP.

En términos conceptuales, la arquitectura TCP/IP está organizada en cuatro capas conceptuales que se soportan bajo una quinta capa física o de hardware. Veamos cada una de las capas de la arquitectura.

- *Capa de Aplicación.*- Al igual que en el modelo de tres capas, la capa de aplicación en TCP/IP tiene las mismas funcionalidades, solo que se adecua a los servicios de transporte proporcionados por las capas inferiores. Así, en el nivel más alto, los usuarios llaman a una aplicación que utiliza servicios disponibles a través de la red. Una aplicación interactúa con uno de los protocolos de nivel de transporte

para enviar o recibir datos. Cada programa de aplicación selecciona el tipo de transporte necesario, el cual puede ser una secuencia de mensajes individuales (UDP), o un flujo continuo de octetos (TCP). El programa de aplicación pasa los datos en el formato requerido hacia la capa de transporte para que sean entregados a su destino.

- *Capa de Transporte extremo-a-extremo.*- Esta capa es la responsable de la integridad de los datos de extremo a extremo. Los dos protocolos más importantes de esta capa son el Protocolo de Control de Transmisión (TCP, Transmission Control Protocol) y el Protocolo de Datagrama de Usuario (UDP, User Datagram Protocol); la diferencia radica en el tipo de servicio que se requiera, ya que mientras TCP brinda un servicio orientado a conexión con control y detección de errores, el UDP provee un servicio de datagramas poco confiable no orientado a conexión.

- *Capa Internet.*- Esta capa es la encargada de enrutar los mensajes entre las redes. El protocolo mas conocido de esta capa es el Protocolo de Internet (IP, Internet Protocol), el cual provee el servicio básico de entrega de paquetes entre las redes TCP/IP.

  Esta capa es utilizada por las capas superiores e inferiores del modelo, lo cual significa que todo el flujo de datos TCP/IP utiliza IP tanto cuando envía como cuando recibe datos, independientemente de su destino final.

Las funciones realizadas por esta capa son: Define el *datagrama* el cual es la unidad básica de transmisión en Internet. Define el esquema de direccionamiento de Internet. Mueve los datos entre las capas de acceso a la red y la capa de transporte. Enruta los datagramas a los anfitriones remotos. Fragmenta y reensambla datagramas.

- *Capa de Acceso a la Red.*- Esta capa consta de una interfaz de red responsable de aceptar los datagramas IP y transmitirlos a una red específica. En esta capa se define cómo usar la red para transmitir las "tramas", las cuales son la unidad de datos que se transmiten a través de una conexión física. En esta capa se define también el intercambio de información entre la computadora y la red física.

A diferencia de protocolos de nivel superior, los protocolos de la capa de acceso a la red deben entender los detalles subyacentes de la capa física, tales como la estructura de los paquetes, el tamaño máximo de la trama, etc. Entendiendo los detalles y limitaciones de la capa física, se asegura que estos protocolos pueden dar el formato correcto a los datos para que puedan ser transmitidos a través de la red.

El diseño de TCP/IP oculta las funciones de esta capa a los usuarios en lo que concierne a la transmisión de datos a un tipo específico de red física (como Ethernet, Token Ring, etc). Este diseño

reduce la necesidad de rescribir los protocolos superiores cuando se introducen nuevas tecnologías de redes físicas (tales como ATM, Frame Relay, etc).

- *Capa Física.*- Define las características del medio de transmisión, la tasa de señalización, y esquemas de codificación de señales.

2.2.2 Servicios.

De manera conceptual una red TCP/IP proporciona tres conjuntos de servicios y su distribución sugiere una dependencia entre ellos. Estos servicios se soportan en las capas de la arquitectura TCP/IP. En la figura 6 se muestra esta relación.

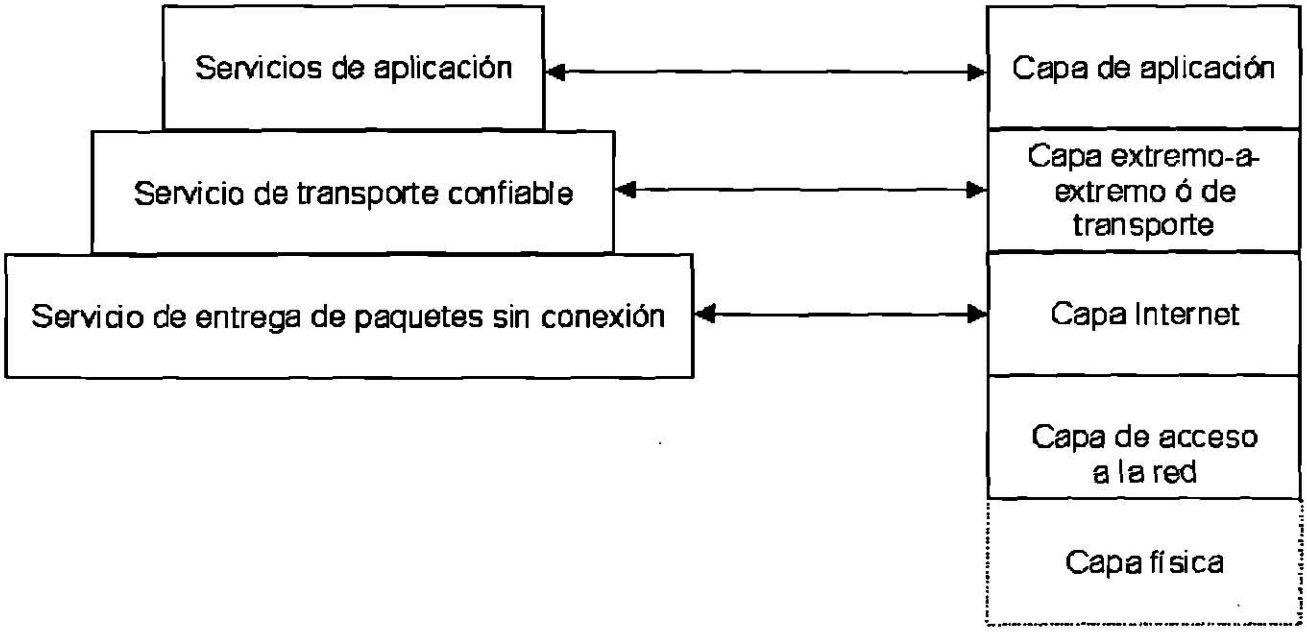


Figura 6. Relación entre los servicios y las capas en la arquitectura TCP/IP.

En el nivel inferior, un servicio de entrega sin conexión proporciona el fundamento sobre el que se apoya el resto del modelo. Un nivel superior, brinda un servicio de transporte confiable el cual proporciona una plataforma de alto nivel de la cual dependen las aplicaciones.

La razón de identificarlos como partes conceptuales es debido a que éstos constituyen un aspecto fundamental en la filosofía del diseño de la arquitectura. Una de las ventajas más significativas de esta separación de conceptos es que es posible reemplazar un servicio sin afectar a los otros.

2.2.2.1 Entrega de Paquetes.

El servicio más importante en esta arquitectura consiste en un sistema de entrega de paquetes. Técnicamente, el servicio se define como un sistema de entrega de paquetes sin conexión y con el mejor esfuerzo, análogo al servicio proporcionado por el hardware de red que opera bajo el paradigma de mejor esfuerzo. El servicio se conoce como *no confiable* porque la entrega no está garantizada. Los paquetes se pueden perder, duplicar, retrasar o entregar sin orden, pero el servicio no detectará estas condiciones ni informará al emisor o al receptor. El servicio es llamado sin conexión dado que cada paquete es tratado de manera independiente de todos los demás. Una secuencia de paquetes que se envían de una computadora a otra puede viajar por diferentes rutas, algunos de ellos pueden perderse mientras otros se entregan. Por último, se dice que el servicio trabaja con base a una *entrega con el mejor esfuerzo* porque el software hace un serio intento por entregar los paquetes. Esto es, la

red no descarta paquetes caprichosamente; la no confiabilidad aparece sólo cuando los recursos están agotados o la red subyacente falla.

2.2.2.2 Transporte.

En la arquitectura TCP/IP es posible transferir datagramas IP entre computadores, donde cada datagrama se enruta a través de la red basándose en la dirección IP del destino. En la capa del Protocolo de Internet, una dirección de destino identifica una computadora anfitrión, mas no se hace ninguna distinción respecto a que usuario o programa de aplicación recibirá los datos.

Debido a que la mayoría de los sistemas operativos son multitarea, varios programas de aplicación pueden estar ejecutándose al mismo tiempo. Entonces puede haber muchos procesos del sistema operativo los cuales necesiten utilizar los servicios de red. Puede parecer natural decir que un proceso es el destino final de un mensaje de la red, sin embargo, especificar que proceso en particular en una máquina en particular es el destino final para un datagrama resulta algo confuso. Esto debido a que los procesos dentro del sistema operativo se crean y se destruyen de manera dinámica, por lo cual resulta imposible estar informando a todos los anfitriones los cambios en los procesos.

En vez de pensar en un proceso como un destino final, pensemos que cada máquina tiene un grupo de puntos abstractos de destino, llamados *puertos de protocolo*. Cada puerto de protocolo se identifica por medio de un número

entero positivo. El sistema operativo local es el encargado de proporcionar un mecanismo de interfaz que los procesos utilizan para especificar o accesar un puerto.

Así pues para que una máquina pueda enviar información a un anfitrión debe enviar además de la dirección origen y la dirección destino, el número de puerto destino de la máquina a la que se envía, así como el número de puerto de la máquina fuente a la que se deben direccionar las respuestas.

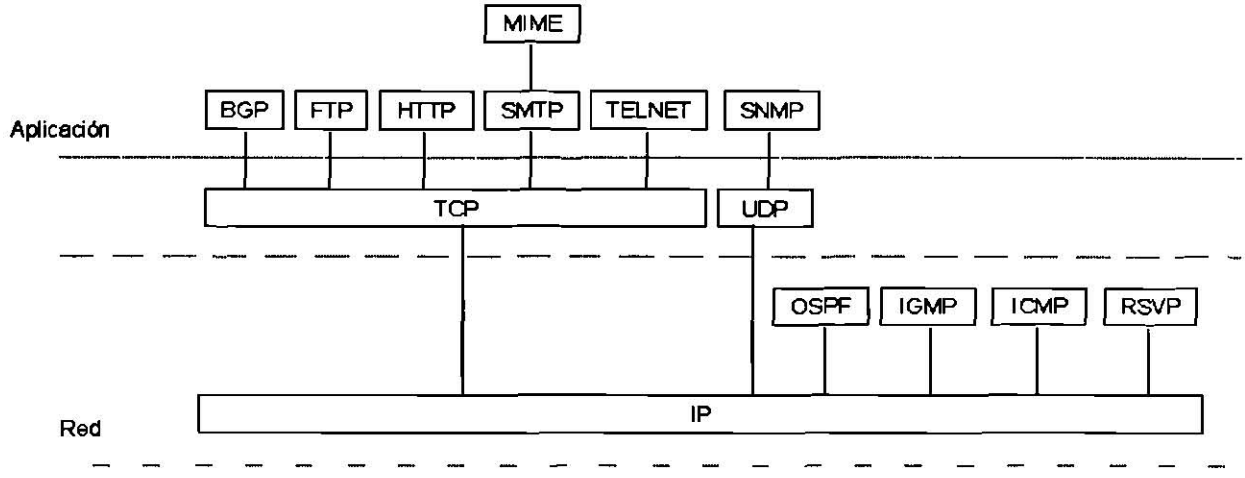
Además, como ya hemos visto, en el nivel más bajo las redes de comunicaciones proporcionan una entrega de paquetes no confiable, es decir, los paquetes se pueden perder o destruir cuando los errores de transmisión interfieren con los datos, cuando falla el hardware de red o cuando las redes se sobrecargan demasiado. Adicionalmente, las redes que enrutan dinámicamente los paquetes pueden entregarlos en desorden, con retraso o duplicados.

La capa de transporte en la arquitectura de protocolos TCP/IP, implementa los módulos de control necesarios proporcionando los servicios de transporte adecuados, los cuales resuelven la problemática descrita en los párrafos anteriores.

Entonces, el término "transporte" con el cual está bautizada la capa 4 del modelo TCP/IP no se refiere a el llevar los datos de un anfitrión origen a un anfitrión destino a través de una red de redes, lo cual comúnmente se confunde. Si no más bien se refiere a proporcionar servicios los cuales permitan a múltiples programas en un anfitrión comunicarse con un anfitrión destino,

brindándole servicios de transporte confiable mediante los cuales el anfitrión origen se asegure de que los datos le llegarán de manera correcta al anfitrión destino. Por lo tanto, el nombre correcto para esta capa debe ser "Capa de Transporte extremo a extremo", ya que asegura la correcta comunicación entre anfitriones en una red de redes.

La figura 7 muestra la familia de protocolos TCP/IP.



IP. Protocolo de Internet (Internet Protocol).
OSPF. Protocolo Abierto del Primer Camino más Corto(Open Shortest Path First).
IGMP. Protocolo de Gestión de Grupos de Internet (Internet Group Management Protocol).
ICMP. Protocolo de Mensajes de Control de Internet (Internet Control Message Protocol).
RSVP. Protocolo de Reservación de Recursos (Resource Reservation Protocol).
TCP. *Protocolo de Control de Transmisión (Transmition Control Protocol).*
UDP. Protocolo de Datagrama de Usuario (User Datagram Protocol).
BGP. Protocolo de Pasarela Frontera (Border Gateway Protocol).
FTP. Protocolo de Transferencia de Archivos (File Transfer Protocol).
HTTP. Protocolo de Transferencia de HyperTexto (HyperText Transfer Protocol).
SMTP. Protocolo Sencillo de Transferencia de Correo (Simple Mail Transfer Protocol).
SNMP. Protocolo Sencillo de Gestión de Redes (Simple Networ Management Protocol).
MIME. *Extensiones Multipropósito de Correo Electrónico en Internet (Multi-Purpose Internet Mail Extension).*

Figura 7. Familia de Protocolos TCP/IP.

2.3 Modelo de Referencia OSI

La historia del desarrollo del modelo OSI es por alguna razón una historia conocida. Mucho del trabajo de diseño de OSI fue realizado por un grupo del Honeywell Information Systems liderado por Mike Canepa, con Charlie Bachman principal miembro técnico.

A principios y mediados de los 70's el interés del grupo de Canepa fue principalmente el diseño de bases de datos y después en el diseño de bases de datos distribuidas. A mediados de los 70's le quedo claro al grupo que para soportar máquinas con bases de datos, acceso distribuido y cosas por el estilo se necesitaba de una arquitectura de comunicaciones estructurada y distribuida. El grupo estudió algunas de las soluciones existentes, incluyendo SNA (Systems Network Architecture, Arquitectura de Sistemas de Red) de IBM, el trabajo de protocolos recién terminados para ARPANET, y algunos conceptos de servicios de presentación que habían sido desarrollados para la estandarización de sistemas de bases de datos. El resultado de este esfuerzo da como resultado el desarrollo en 1977 de una arquitectura de siete capas conocida internamente como la Arquitectura de Sistemas Distribuidos (DSA, Distributed System Architecture).

Mientras tanto, en 1977 el Instituto Británico de Estándares (British Standard Institute) propuso a la Organización Internacional de Estandarización (ISO, International Organization Standardization) que se necesitaba una arquitectura estándar que definiera la infraestructura de comunicaciones para el

procesamiento distribuido. Como resultado de esta propuesta la ISO formó el subcomité de Interconexión de Sistemas Abiertos (OSI, Open Systems Interconnection) como Comité Técnico 97, Subcomité 16. El Instituto Nacional de Estándares Americanos (ANSI, American Nacional Standards Institute) fue el encargado de desarrollar propuestas de avance de la primera reunión formal del subcomité.

Bachean y Canepa participaron en las primeras reuniones de la ANSI y presentaron su modelo de siete capas. Este modelo fue seleccionado como la única propuesta para ser presentado al subcomité de la ISO. El equipo de Honeywell presentó su solución un una reunión de la ISO en Washington DC en Marzo de 1978. En esta reunión se llego a un consenso de que esta arquitectura en capas satisfacía la mayoría de los requerimientos del OSI, y tenía la capacidad de ser expandida después para satisfacer nuevos requerimientos. Se publicó una versión provisional del modelo en Marzo de 1978; la siguiente versión, con algunas modificaciones pequeñas, se publicó en Junio de 1979 y finalmente quedó estandarizado. El modelo OSI resultante es esencialmente el mismo modelo DSA desarrollado en 1977 [4].

2.3.1 El *Modelo OSI*.

Este modelo de siete capas utiliza también una técnica de estructuración muy utilizada: la jerarquización por capas. En la figura 8 se muestra el modelo OSI conceptualizado bajo este paradigma de estratificación por capas.

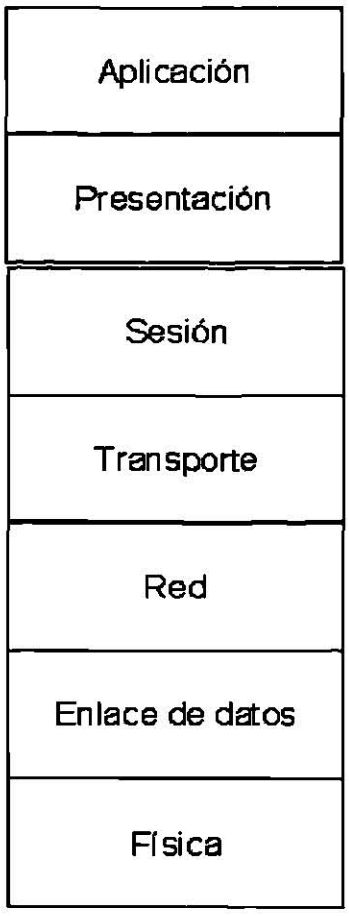


Figura 8. Capas del modelo OSI.

Veamos cada una de las capas de este modelo.

- *Capa de Aplicación.*- La capa de aplicación proporciona a los programas un medio para que accedan al entorno OSI. Esta capa incluye las funciones de administración y en general los mecanismos necesarios en la implementación de aplicaciones distribuidas.

- *Capa de Presentación.*- Esta capa definen los aspectos y las reglas usadas en la especificación formal de los datos a intercambiar entre aplicaciones. Esta capa provee una

representación común de los datos transferidos entre aplicaciones y proporciona los medios para seleccionar y modificar la representación utilizada. Algunos ejemplos, son la compresión y cifrado de datos.

- *Capa de Sesión.-* El propósito de la capa de sesión es proporcionar el medio necesario para la cooperación de las entidades de presentación, así como organizar y sincronizar su diálogo y manejar el intercambio de datos entre ellos. Para hacer esto, la capa de sesión provee servicios para establecer las conexiones de sesión entre dos entidades de presentación para soportar ordenadamente las interacciones de intercambio de datos, así como liberar la conexión de una manera ordenada.

Las cuatro capas inferiores del modelo OSI proporcionan un medio para el intercambio seguro de datos y proporciona a su vez distintos niveles de calidad de servicio.

- *Capa de Transporte.-* Esta capa proporciona un mecanismo transparente para transferir datos entre las entidades de sesión, y las libera de todos los detalles que tienen que ver con el costo de una entrega confiable de los datos

  El servicio de transporte orientado a conexión asegura entregas libres de errores, en orden, y sin pérdidas ni duplicaciones. La capa de transporte también puede estar

involucrada en la optimización del uso de los servicios de red proporcionando la calidad de servicio solicitada. Por ejemplo, se puede solicitar una tasa de error determinada, un retardo mínimo, una prioridad y un nivel de seguridad dado.

Debido al tamaño y complejidad del protocolo de transporte, ISO ha desarrollado una familia de cinco estándares de protocolos de transporte, cada uno de ellos especificado para un determinado servicio subyacente. El Protocolo de Transporte (TP, Transport Protocol) tiene cinco clases distintas:

- Clase 0. Clase Simple
- Clase 1. Clase Básica de Restauración de Error.
- Clase 2. Clase de Multiplexado.
- Clase 3. Clase de Restauración de Error y Multiplexado.
- Clase 4. Clase de Detección y Restauración de Errores

Cada una de estas clases de protocolo tiene su propia estructura de Unidad de Datos de Protocolo (PDU).

- *Capa de Red.*- La capa de red realiza la transferencia de información entre sistemas finales a través de algún tipo de red de comunicación. Libera a las capas superiores de la necesidad de tener conocimiento sobre la transmisión de datos subyacente y las tecnologías de conmutación utilizadas para conectar los sistemas. En esta capa el computador establecerá un diálogo con la red

para especificar la dirección destino y solicitar ciertas facilidades, como por ejemplo la gestión de prioridades.

Esta capa proporciona el medio para establecer, mantener y terminar las conexiones de red entre sistemas que contienen módulos de aplicación comunicándose. El principal servicio proporcionado por esta capa es el de proveer una transferencia de datos transparente entre módulos de transporte.

El principal protocolo para esta capa en el modelo OSI, es el Protocolo de Red Sin Conexión (CLNP, Connectionless Network Protocol). Otros protocolos de enrutamiento complementan la suite de protocolos OSI, como se ve en la figura 9.

- *Capa de Enlace de Datos.*- Uno de los objetivos de esta capa es intentar hacer que el enlace físico sea seguro, proporcionando los medios para activar, mantener y desactivar el enlace físico. El principal servicio proporcionado por esta capa a las capas superiores es el de detección y control de errores en la transmisión. De esta manera, si nuestro protocolo de enlace funciona a la perfección, la capa de red puede suponer que la transmisión de datos está libre de errores. Sin embargo, si la transmisión se realiza entre dos sistemas que no están directamente conectados, si no que se conectan a través de una serie de enlaces en serie cada uno operando de manera

independientemente, entonces, la capa de red no estará libre de la responsabilidad del control de errores.

- *Capa Física.-* La capa física proporciona las funciones y procedimientos necesarios para activar, mantener y desactivar las conexiones físicas para la transmisión de bits entre módulos de enlace.

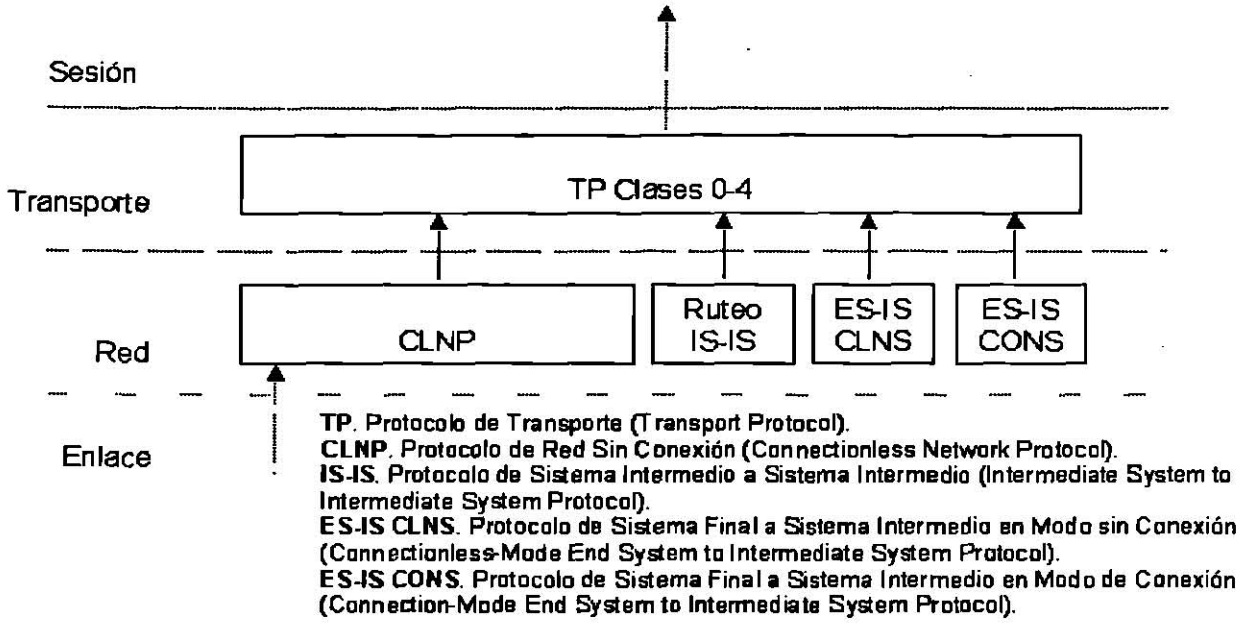


Figura 9. Protocolos de la familia OSI.

2.4 OSI vs TCP/IP (Conclusiones)

El objetivo de esta sección no es confrontar estas dos importantes arquitecturas, ya que cada una de ellas tiene sus ventajas y desventajas en cuanto al diseño e implementación de las mismas; mas que nada es para hacer

una clara distinción entre ellas ya que muchas veces por razones pedagógicas se llegan a mezclar conceptos muy propios de alguna arquitectura en particular.

Así pues, escuchamos decir frases como "¿A qué capa del modelo OSI pertenece el protocolo IP?", o bien, "El protocolo característico en la capa cuatro del OSI es el TCP", o nunca falta aquel que cree que el modelo OSI es el amo y señor de las comunicaciones. No trato de demeritar ni hacer menos al modelo OSI, pero si me gustaría dejar los conceptos muy en claro para evitar confusiones al momento de hablar sobre estos modelos.

De esta forma, el *modelo* OSI -y lo pongo en letra Itálica con toda intención-, es como bien lo dice, un modelo de referencia para las comunicaciones, que aunque la ISO ha diseñado protocolos para algunas de sus capas, la popularidad mas bien ha sido por la manera de conceptualizar cada uno de los términos, así como la facilidad de entender el proceso de comunicaciones a través de sus siete capas, cada una con funciones y tareas muy específicas.

Entonces, para fines didácticos el modelo de referencia OSI resulta ideal, mas no así para la implementación, lo cual se ha visto en el mercado de las comunicaciones donde la familia de protocolos TCP/IP se ha erguido como el líder en el mercado.

Resulta importante recalcar que la arquitectura TCP/IP solo define protocolos para sus tres capas superiores, por lo que es muy fácil el hecho de confundir y tratar de relacionarlos con el modelo OSI. Esto no es malo, ya que

como se había mencionado, el modelo OSI es conocido por todo el mundo involucrado en el área de las comunicaciones. Así bien, es posible tratar de "mapear" los protocolos de TCP/IP a su capa equivalente en OSI, pero solo con fines demostrativos.

Para darnos una idea de cómo podemos referenciar los protocolos de múltiples tecnologías de comunicaciones con el modelo OSI, la figura 10 muestra las pilas de protocolos en relación al modelo OSI.

<table>
<tr><th>Capa OSI</th><th>Sistemas Banyan</th><th>IBM SNA</th><th>Novell Netware</th><th>TCP/IP</th><th>Xerox XNS</th><th>Protocolos OSI</th></tr>
<tr><td>Aplicación</td><td colspan="6">Programas y protocolos de aplicación</td></tr>
<tr><td>Presentación</td><td rowspan="2">Llamada a Procedimientos Remotos (Net RPC)</td><td>Servicios de Transacción y Servicios de Presentación</td><td>Netware Core Protocols (NCP)</td><td rowspan="2">TELNET, FTP SMTP, HTTP, etc.</td><td rowspan="2">Interacción de Control y Procesos</td><td>ISO 8823</td></tr>
<tr><td>Sesión</td><td>Control de Flujo de Datos</td><td>Network Basic Input/Output System (NetBIOS)</td><td>ISO 8327</td></tr>
<tr><td>Transporte</td><td>Comunicación Interprocesos VINES (VIPC)</td><td>Control de Transmisión</td><td>Sequenced Packet Exchange (SPX)</td><td>Protocolo de Control de Transmisión (TCP) Protocolo de Datagrama de Usuario (UDP)</td><td>equenced Packet Protocol (SPP)</td><td>TP Clases 0-4</td></tr>
<tr><td>Red</td><td>VINES Internet Protocol (VIP)</td><td>Control de Ruta</td><td>Internet Packet Exchange (IPX)</td><td>Protocolo de Internet (IP)</td><td>Internet Datagram Protocol (IDP)</td><td>CLNP</td></tr>
<tr><td>Enlace</td><td colspan="6">Tecnologías de enlace de datos
Ethernet, Token Ring, ATM, FDDI, etc.</td></tr>
<tr><td>Física</td><td colspan="6">Medio de Transmisión
Partrenzado, fibra óptica, Coaxial, Inalámbrico, etc.</td></tr>
</table>

~~Figura~~ 10. Pilas de protocolos en relación al modelo OSI.

Tabla

CAPITULO 3

CALIDAD DE SERVICIO

Hoy en día las redes de telecomunicaciones están cambiando para converger en redes capaces de integrar servicios de voz, datos y video dentro de una misma infraestructura. Esta convergencia implica que el modelo original de servicio bajo el mejor esfuerzo no sea suficiente para poder asegurar un servicio oportuno; por ejemplo, el transporte de tráfico isócrono o en tiempo real, donde parámetros tales como el retraso y la variación del retraso tienen que minimizarse. Sin perder de vista las aplicaciones convencionales tales como la transferencia de archivos, mensajería, correo electrónico, etcétera, que si bien no requieren que el retraso sea mínimo, si requieren que los datos lleguen libres de errores.

Hagamos la siguiente analogía, imaginemos que tenemos una avenida muy amplia, de ocho carriles de circulación, donde el tráfico fluye tranquilamente. Esta avenida sirve además como avenida de desahogo para múltiples calles pequeñas donde todo su tráfico converge hacia nuestra gran avenida. Pero tanto en la mañana como en la tarde cuando la gente entra o sale de su trabajo esta gran avenida experimenta congestionamientos debido a la cantidad de vehículos que circulan en ella, provocando grandes caos viales. Y aunque se supone que los carriles internos están reservados para vehículos de

alta velocidad, los conductores no respetan estas reglas y se adueñan de cualquier carril que este disponible, aunque no corresponda con su velocidad. Los vehículos oficiales como ambulancias y patrullas tienen además prioridad para circular. Y no falta un choque que paralice el tráfico.

Bueno, tal vez esta situación la experimentemos a diario cuando nos dirigimos a nuestros trabajos, pero quizá no hemos pensado que esto mismo sucede en nuestras redes de comunicaciones. Probablemente nos sentimos muy confiados porque tenemos enlaces de alta velocidad, pero los congestionamientos en las redes son muy frecuentes en la mayoría de los casos. Imaginemos que nuestro enlace principal es nuestra gran avenida, donde el tráfico fluye de manera correcta siempre y cuando no estemos en horas pico donde todo mundo trata de accesar recursos sobre este enlace. Nuestras diferentes aplicaciones no pueden tener las mismas prioridades y recursos, además debemos asignar los recursos necesarios a cada una para un funcionamiento adecuado, pero siempre teniendo nosotros el control, ya que no se requiere el mismo trato cuando fluye tráfico de correo electrónico que tráfico sensitivo como nuestras bases de datos o aplicaciones de misión crítica.

Aquí es donde tenemos que idear e implementar mecanismos que nos permitan controlar los picos en el flujo de datos, así como también debemos de ser capaces de distinguir los diferentes tipos de tráfico que circulan por nuestra red para poder asegurarles los recursos necesarios para el adecuado funcionamiento de las aplicaciones que nuestros usuarios requieren. Es aquí donde entra el concepto de la Calidad en el Servicio.

3.1 ¿Qué es Calidad de Servicio?

El término calidad puede resultar muy trillado, ya que lo escuchamos en muchas partes. Calidad de vida, calidad total, calidad de persona, etcétera, son términos que escuchamos a diario y por lo mismo puede que se piense en nuestro término de Calidad de Servicio como un slogan publicitario para vender más.

Así pues, la Real Academia Española define Calidad como [5]:

Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.

No vamos a adentrarnos en etimologías ni nada por el estilo, pero sí esta definición nos puede ayudar, ya que podemos transformarla para darle un sentido en el área tratada en este trabajo. Entonces bajo este preámbulo me puedo atrever a definir Calidad en el Servicio en el sentido estricto de las redes de comunicaciones como:

Propiedad o conjunto de propiedades inherentes a las redes de comunicaciones, que permiten juzgar el valor del servicio prestado.

La calidad en el servicio puede ser buena o mala, por supuesto que el sentido que le queremos dar a nuestras definiciones es de una buena Calidad en el Servicio. Lógicamente, las definiciones oficiales del término distan mucho de parecerse a esta definición semántica. Así que veamos como diferentes autores definen la calidad de servicio en las redes de computadores.

Crawley en el RFC 2386 define Calidad de Servicio como [6]:

Un conjunto de requerimientos que deben ser satisfechos por la red mientras se transporta un flujo de datos.

Deborah S Bakin del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, Institute of Electrical and Electronics Engineers) da la siguiente definición [7]:

Es la capacidad de un elemento de la red (aplicación, anfitrión o enrutador) de proveer un nivel de aseguramiento para que ese tráfico y los requerimientos de servicio puedan ser satisfechos.

Podemos complementar los conceptos definiendo de manera general:

Calidad de Servicio se refiere a la capacidad de una red de proveer el mejor servicio al tráfico de red seleccionado sobre diferentes tecnologías, incluyendo Frame Relay, ATM, Ethernet y redes 802.1, SONET, etcétera, así como redes IP que utilizan una o todas de estas tecnologías subyacentes.

3.2 Porqué se Necesita

Podría sonar lógico que para resolver las problemáticas presentadas anteriormente la solución sería incrementar nuestros canales de comunicaciones, y en efecto, esta solución sería un mecanismo primitivo de Calidad de Servicio. Pero Calidad de Servicio no crea ancho de banda. Es imposible pedirle a la red que proporcione algo que no tiene.

Además, el problema es que el tráfico no solo aumenta en volumen, sino también en naturaleza, ya que existen varios tipos nuevos de tráfico provenientes de nuevas aplicaciones, las cuales varían tremendamente en sus *requerimientos.*

Por esta razón el incrementar nuestros canales de comunicaciones no resuelve el problema, es por ello que requerimos agregar inteligencia a la red, la cual permita brindar los servicios que los usuarios requieren aprovechando al máximo los recursos disponibles.

Podemos tener tres escenarios los cuales determinan si realmente nos es útil implementar mecanismos de calidad de servicio:

- Escenario A.- En este escenario, nuestro enlace está subutilizado, aún en las horas pico que existen ráfagas de tráfico nuestro enlace no se ve comprometido. En este caso no es necesario implementar mecanismos de Calidad de Servicio, ya que no tiene ninguna ventaja costo-beneficio. Sería necesario implementar otros mecanismos, por ejemplo, para hacer segura nuestra red, pero ese tema esta fuera del alcance de este trabajo.

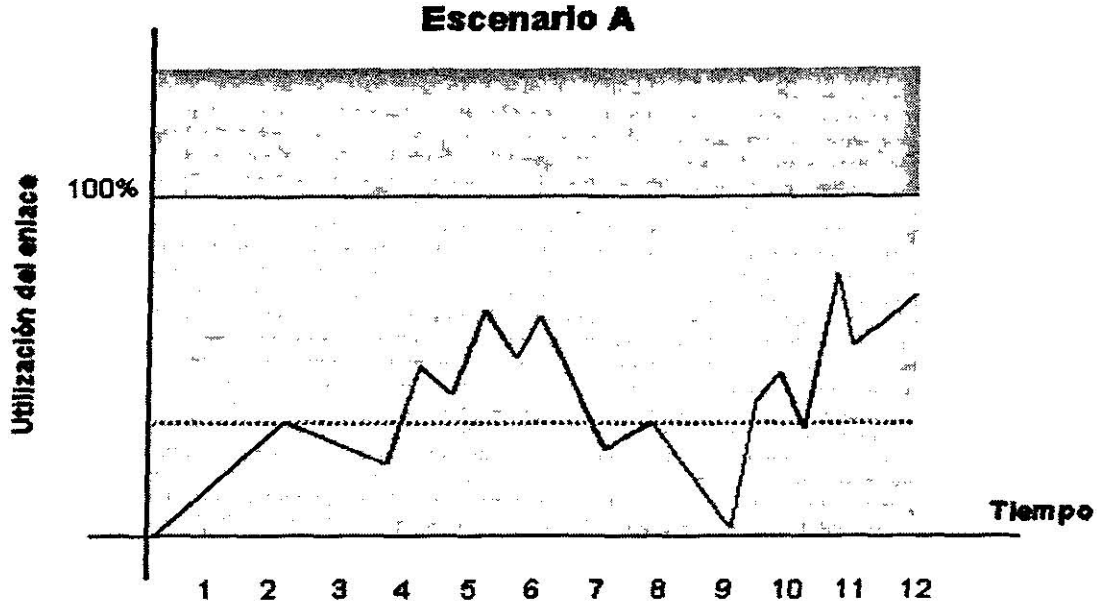


Figura 11. Escenario A.

- Escenario B.- En este escenario existen congestiones temporales, lo que provoca que nuestro enlace de comunicaciones tenga que "tirar" paquetes durante estos periodos de congestión, provocando pérdida de paquetes en nuestra red haciendo que nuestras aplicaciones no se ejecuten de manera correcta o con los parámetros de calidad deseados. Este escenario es el ideal para implementar mecanismos de Calidad de Servicio, ya que como la sobre-utilización es temporal, con una buena administración de nuestros recursos, apoyados de los mecanismos de Calidad de Servicio podemos resolver este problema.

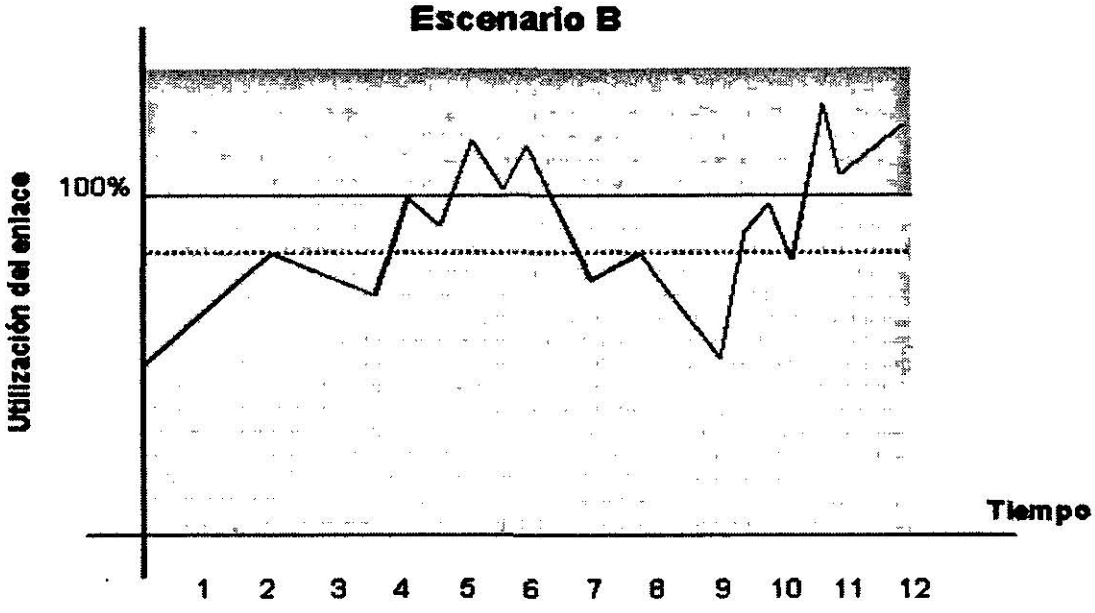


Figura 12. Escenario B.

- Escenario C.- En este último escenario, vemos que nuestro enlace se encuentra sobre-utilizado y solamente en algunos momentos nuestro tráfico se encuentra dentro de las capacidades del enlace. En este caso implementar mecanismos de Calidad de Servicio pueden solucionarnos ciertos problemas, como por ejemplo que los recursos del enlace sean utilizados preferentemente por aplicaciones críticas, pero esto no resuelve totalmente el problema, ya que es necesario incrementar el ancho de banda de nuestro canal de comunicaciones.

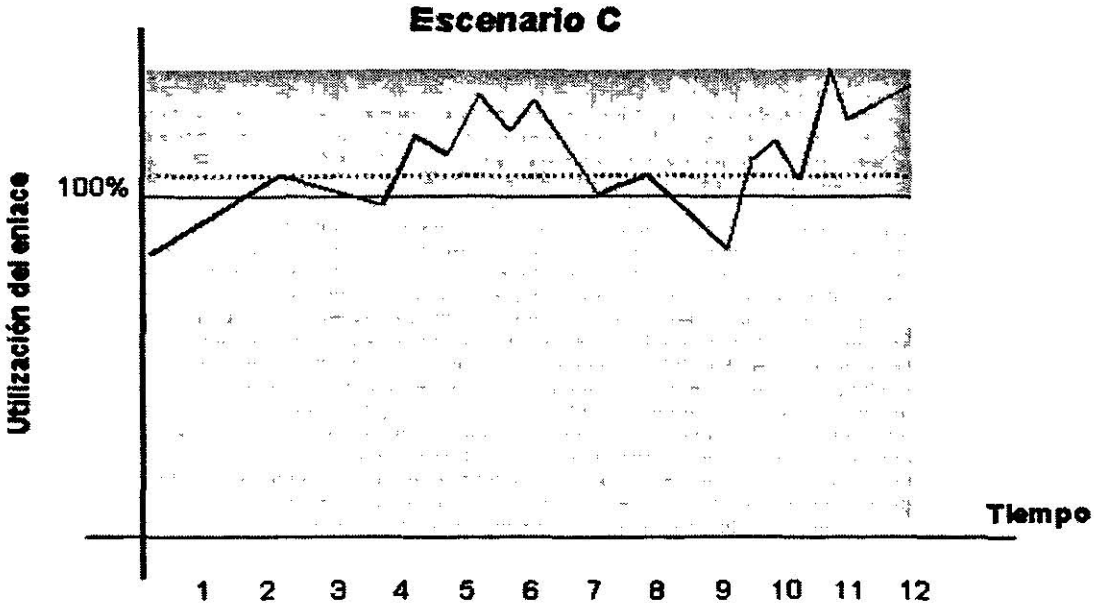


Figura 13. Escenario C.

3.3 Dónde se Implementa

La gran pregunta sería: *¿Dónde tengo que implementar Calidad de* Servicio?; existe un principio fundamental en el cual se basa el diseño de la mayoría de las *arquitecturas de Calidad de Servicio, este es:* "Deja la complejidad en los 'bordes' y mantén simple el 'núcleo' de la red" [8].

Este principio se basa en la premisa de que la implementación debe ser lo más cercano a nuestro dominio de administración, esto para permitirnos tener el control.

Pero la pregunta original no ha sido resuelta, entonces ¿dónde?; el ofrecer Calidad de Servicio de extremo a extremo implica que cada elemento en la red –conmutadores, enrutadores, cortafuegos (firewalls), anfitriones, clientes,

aplicaciones y así sucesivamente- entreguen su parte de QoS, por lo que será difícil implementar mecanismos en redes en las cuales no tenemos nosotros la administración. Así, si nuestro tráfico cruza a través de redes de algún proveedor de servicios nosotros no podemos realizarles ninguna configuración, mucho menos si queremos mantener la Calidad de Servicio cuando nuestro tráfico cruza la Internet.

Entonces proveer Calidad de Servicio de extremo a extremo resulta un tanto problemático, y el reto es desarrollar robustos mecanismos de señalización de QoS que puedan operar extremo a extremo sobre infraestructuras de red heterogéneas. No obstante, muchas soluciones viables proveen QoS en algunas partes de la infraestructura, aunque su alcance está limitado a una parte de la red.

La respuesta a la pregunta sería: ofrecer QoS extremo a extremo en nuestra red es posible siempre y cuando nosotros tengamos la posibilidad de configurar todos los elementos de la red; de lo contrario la implementación solo será posible en los sectores de la red en la cual tengamos nosotros el control, o bien si cruza por redes de un proveedor de servicios debemos de tener un contrato de Acuerdo de Nivel de Servicio (SLA, Service Level Agreements).

En la sección posterior nos centraremos en analizar las técnicas, mecanismos y protocolos para ofrecer Calidad de Servicio en ciertas partes de nuestras redes donde los problemas de congestiones se agudizan, dejando a

un lado el estudio de mecanismos para ofrecer Calidad de Servicio extremo a extremo.

3.4 Modelos de Implementación de Calidad de Servicio

Si volvemos a nuestra analogía inicial en la que comparábamos la Calidad de Servicio con el tráfico vehicular, podríamos decir que estos modelos de implementación serían como los agentes de tránsito y los señalamientos viales los cuales se encargan de dar fluidez al tráfico y evitar congestionamientos, así como mantener informado a los conductores sobre las reglas que deben de seguir cuando circulan por ciertas avenidas.

Existen diversos y muy variados modelos para implementar Calidad de Servicio, algunos de ellos ya estandarizados y otros sólo son propuestas, por lo que es un tanto difícil conjugar todos los elementos dentro de una misma red, ya que por lo general ésta presenta heterogeneidad en los equipos de comunicaciones. Veamos ahora la clasificación de los diferentes modelos de Calidad de Servicio, así como una explicación de los mismos [9].

3.4.1 Etiquetado de Prioridad Relativa.

En este modelo las aplicaciones, los anfitriones o los nodos seleccionan una prioridad relativa o "precedencia" para un paquete, y los nodos de la red por la que atraviesa el paquete aplican el comportamiento apropiado para el reenvío del paquete correspondiente con la prioridad que contiene el Campo de

Prioridad en la cabecera del paquete. Veamos ahora algunos modelos y tecnologías que se basan en este concepto.

3.4.1.1 Etiquetado de Precedencia en IPv4.

La precedencia en IP provee la habilidad de clasificar los paquetes de la red en la capa de Internet de la arquitectura TCP/IP. La precedencia es un esquema para la asignación de recursos en la red basados en la importancia de los diferentes flujos de datos.

Este esquema fue propuesto por Jon Postel en la especificación del Protocolo de Internet [10] en 1981 donde define el campo Tipo de Servicio (TOS, Type of Service) en la cabecera del protocolo IP versión 4 (Figura 14).

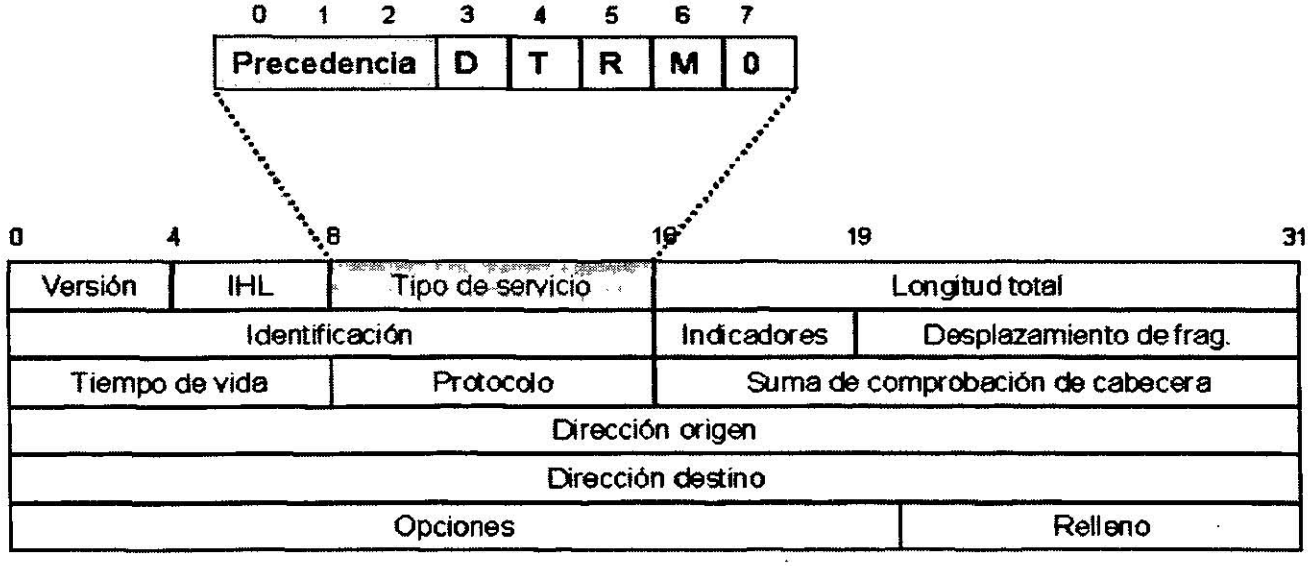


Figura 14. Campo Tipo de Servicio en la cabecera de IPv4.

La Precedencia IP (IP Precedence) utiliza tres de los ocho bits que tiene el campo de Tipo de Servicio. Estos tres bits pueden representar ocho clases de servicio mediante la combinación de sus bits, tal como se muestra en la tabla I [10]:

TABLA I

TIPOS DE TRÁFICO UTILIZANDO PRECEDENCIA IP

Bits	Prioridad	Tipo de tráfico
111	7	Control de Red
110	6	Control Entre Redes
101	5	Tráfico crítico/Control de Errores
100	4	Anulación de Flash
011	3	Flash
010	2	Inmediato
001	1	Prioritario
000	0	Rutinario

Como vemos, el rango de clasificación va del 0-7 donde el cero es la prioridad mas baja y el siete la mas alta. La configuración de estas prioridades es realizada por el administrador de la red según el tipo de tráfico que poseen.

Los equipos de comunicaciones o las máquinas anfitriones necesitan "etiquetar" o marcar este campo de Precedencia IP con el valor correspondiente al tipo de tráfico solicitado.

El mecanismo básico para el procesamiento de la precedencia es en la asignación preferencial de recursos, incluyendo el servicio de colas de espera y control de congestión [11].

Los equipos de comunicaciones deben poder soportar estas implementaciones que permitan el tratamiento al tráfico "etiquetado" con alguna prioridad. Para esto, pueden tener mecanismos de encolamiento especiales los cuales utilizan los valores de precedencia en todos los puntos de procesamiento de los paquetes para la asignación de recursos finitos, tales como memorias temporales (buffers) o conexiones con la capa de enlace. Estas implementaciones son independientes en cada caso y dependen de las necesidades propias de cada red.

Estas mismas funcionalidades de la precedencia IP pueden ser aprovechadas en el Protocolo de Internet versión 6 (IPv6) utilizando el campo Clase de Tráfico, tal como se muestra en la figura 15. Este campo todavía se encuentra en fase experimental [12] por lo que no ahondaremos en el tema.

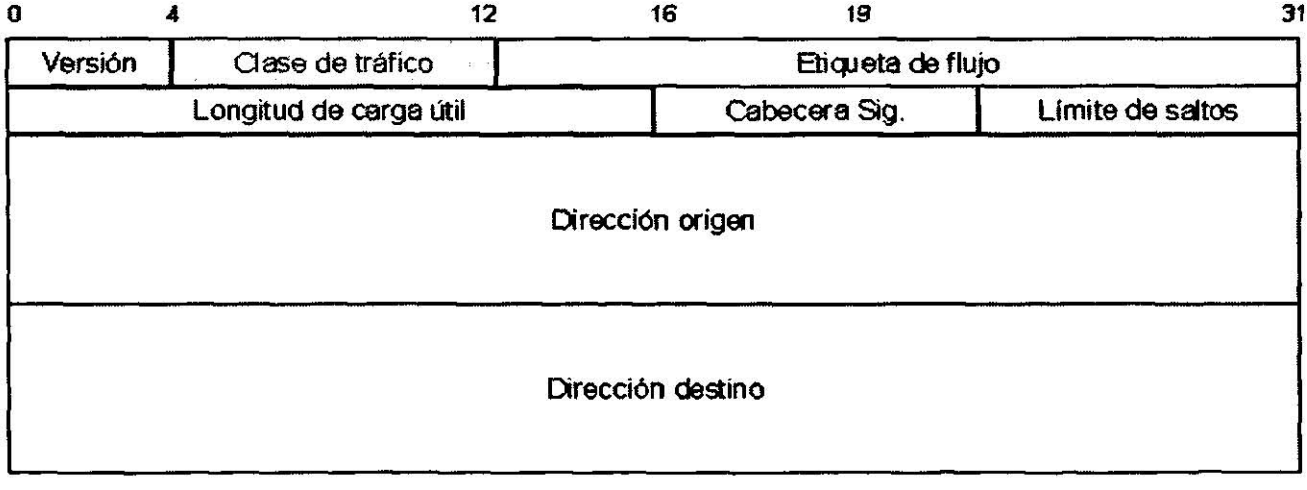


Figura 15. Campo de Clase de Tráfico en la cabecera de IPv6.

3.4.1.2 IEEE 802.1p

IEEE 802.1p es una técnica de señalización del Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronic Engineering, IEEE), la cual permite priorizar tráfico en la capa de enlace de datos (capa 2 del modelo OSI) utilizando la tecnología Ethernet.

Cabe aclarar que el estándar IEEE 802.1p es parte del estándar IEEE 802.1D, el cual define la agilización de las clases de tráfico y el filtro dinámico de multicast que son parte de los puentes de Control de Acceso al Medio (Medium Access Control, MAC). El estándar 802.1q define la arquitectura para redes locales virtuales (Virtual Local Area Networks, VLAN's) el cual también es parte del estándar 802.1D [13].

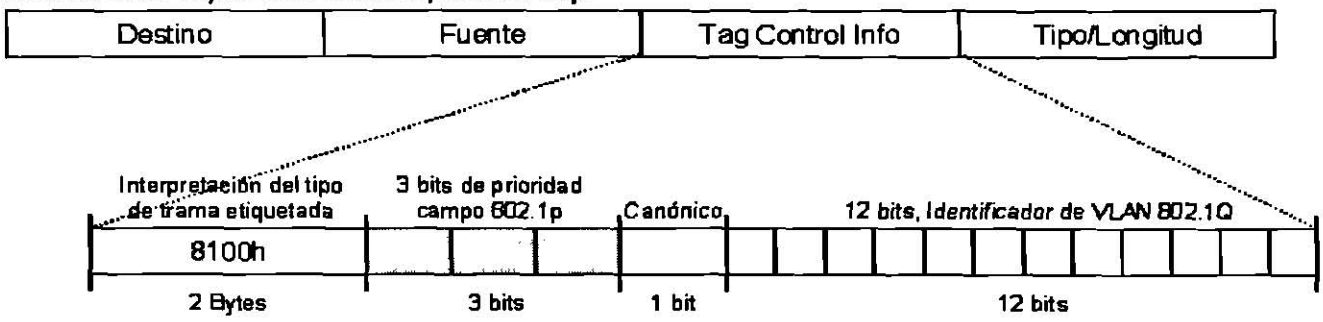
Como ya hemos mencionado el estándar 802.1p es un esquema de prioritización, cuyo paradigma es: Los paquetes de prioridad más baja no son enviados mientras haya paquetes de prioridad más alta en las colas de transmisión.

Bajo 802.1p el campo Etiqueta de Control de Información (Tag Control Info, TCI) de 4 bytes es insertado en la cabecera de la capa 2 entre la Dirección Destino y el campo de Tipo/Longitud del cliente como se muestra en la figura 16.

Cabecera de capa 2 de Ethernet, sin 802.1p

Destino	Fuente	Tipo/Longitud

Cabecera de capa 2 de Ethernet, con 802.1p



Campo de Etiqueta de Control	Descripción
Interpretación del tipo de trama etiquetada	Siempre establecido a 8100h en tramas Ethernet
Campo de prioridad de 3 bits (802.1p)	Valores del 0-7 representando los niveles de prioridad del usuario (7 es el mayor)
Canónico	Siempre es 0
Identificador de VLAN de 12 bits (802.1Q)	Numero de identificador de LAN Virtual (VLAN)

Figura 16. Campo de Etiqueta de Control de Información en la cabecera de Ethernet.

Similar a la Precedencia IP, el IEEE 802.1p establece ocho niveles de prioridades tal como se muestran en la tabla II.

TABLA II

TIPOS DE TRÁFICO EN 802.1p.

Bits	Prioridad	Tipo de tráfico	Tipo de aplicaciones
111	7	Control de Red	Control de red
110	6	Voz	Voz
101	5	Video	vides
100	4	Carga Controlada	Bases de datos, SAP, etc.
011	3	Excelente Esfuerzo	Usuarios importantes
010	2	Mejor Esfuerzo	Tráfico ordinario
001	1	Sobrante	Trasferencias grandes
000	0	Segundo plano	Juegos, mensajería, etc.

Los dispositivos que cumplen con el estándar 802.1p leen los 3 bits del campo de Prioridad y envían la trama hacia el buffer/cola que le corresponde según su prioridad. Los dispositivos que soportan 802.1p por lo general tienen al menos dos buffers/colas, aunque pueden tener más. Esto lo trataremos a mas detalle en el capitulo 4.

El servicio de prioridad en la cola corresponde con el nivel de prioridad indicado por el usuario. Las colas de mayor prioridad son atendidas antes que las colas de menor prioridad. Los paquetes no etiquetados o con prioridad cero son atendidos con el servicio de "mejor esfuerzo", el cual comúnmente corresponde a la cola de menor prioridad.

Los administradores de red asignan las prioridades en base a políticas. Dependiendo de las políticas y de la infraestructura el etiquetado puede realizarse en las tarjetas de red que cumplen con el estándar 802.1p, en los equipos de comunicaciones, o bien, en las aplicaciones de los usuarios.

El estándar 802.1p no describe ningún método de control de admisión, por lo que es posible dar prioridad de "Control de la Red" a todos los paquetes lo cual provocaría un caos. El estándar por si mismo no limita la cantidad de recursos utilizados por cada usuario.

Aunque el 802.1p es ya un estándar muchos dispositivos antiguos no lo soportan, lo cual trae consigo problemas. Debido que a la trama "etiquetada" se le agregan 4 bytes adicionales del campo TCI, los dispositivos que no soportan este estándar reconocen la trama como pasada de tamaño (oversized) y tiran

los paquetes con estas características [14]. En el mejor de los casos, enviará los paquetes sin tomar en cuenta la prioridad establecida.

3.4.2 Marcación de Servicios.

El ejemplo más claro de este modelo es el Tipo de Servicio en IPv4 (ToS, Type of Service). En este modelo cada paquete es marcado por una solicitud de "tipo de servicio", el cual puede incluir "mínimo retraso", "máximo caudal eficaz", "maximizar la fiabilidad" o "minimizar el costo". Los nodos en la red deben seleccionar rutas o comportamientos de reenvío las cuales se basan en ingeniería de ruteo para satisfacer los servicios solicitados. El etiquetado ToS es muy genérico y no abarca todo el rango de posibles servicios, además el servicio solicitado es asociado individualmente con cada paquete, mientras que algunas semánticas del servicio pueden depender de agregar comportamientos de reenvío de una secuencia de paquetes. Este modelo de marcación de servicios no se acopla fácilmente al crecimiento en el número y tipo de futuros servicios, e involucra configuraciones del "comportamiento de reenvío ToS" en cada nodo del núcleo de nuestra red. Veamos con más detalle este modelo.

3.4.2.1 IPv4 ToS.

Como ya hemos mencionado, las rutas en Internet varían ampliamente en la Calidad de Servicio que pueden proporcionar, debido a que no se tiene conocimiento directo de cómo optimizar determinada ruta para alguna aplicación o usuario en particular. Además el caudal eficaz y el retraso pueden variar enormemente. Es aquí donde encontramos algunas disyuntivas: La ruta

que nos proporciona el mayor caudal eficaz puede no ser la que proporciona el mínimo retraso o el más bajo costo monetario. Así pues, la ruta "optima" para un paquete que pasa por la Internet depende de las necesidades de las aplicaciones y de sus usuarios.

El protocolo IP proporciona (aunque muy limitado) facilidades a las capas superiores para que proporcionen indicios a la capa de Internet acerca de la interrelación de servicios para cada paquete en particular.

Este mecanismo de Tipo de Servicio fue definido originalmente en el estándar del protocolo IP (RFC 791), para luego ser complementado y descrito más a detalle en el RFC 1349 titulado Tipo de Servicio (Type of Service).

El servicio de ToS es una de las características del octeto Tipo de Servicio de la cabecera del datagrama IP. Este octeto lo podríamos definir [15] como se muestra en la Figura 17.

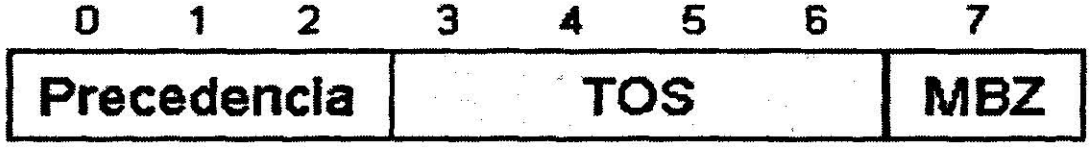


Figura 17. Octeto Tipo de Servicio de la cabecera de IPv4.

Como vemos, este campo es el mismo mostrado en la figura 14, solamente que ahora definimos el uso de los demás bits. El campo "Precedencia" tiene la misma funcionalidad definida en la sección 3.4.1.1.

El campo "TOS" denota cómo la red debe interrelacionar los servicios entre caudal eficaz, retraso, fiabilidad y costo.

El campo "MBZ" (Debe ser cero, Must Be Zero) no es usado actualmente, pero puede usarse para fines experimentales en la definición de nuevos servicios.

3.4.2.1.1 Especificaciones del Campo TOS. Originalmente en el estándar del protocolo IP [10] el campo TOS solo definía los bits 3-5, siendo los primeros tres los bits de precedencia, y los últimos dos eran ceros; en esta especificación solo se definían comportamientos para retraso, caudal eficaz y fiabilidad, quedando la distribución de los bits como se muestra en la tabla III.

TABLA III

DEFINICIÓN ORIGINAL DEL CAMPO TOS

Bits	Definición
0-2	Precedencia
3	0 = Retraso normal, 1 = Mínimo retraso
4	0 = Caudal Eficaz normal, 1 = Máximo Caudal Eficaz
5	0 = Fiabilidad normal, 1 = Máxima fiabilidad
6-7	0, Reservados para uso futuro

En el RFC 1349 esta definición se modifica y se agrega un servicio más: el costo monetario. La tabla IV define la semántica para la combinación de los bits 3-6 del campo Tipo de Servicio.

TABLA IV

SEMÁNTICA PARA EL CAMPO TOS UTILIZANDO LOS BITS 3-6

Combinación de bits	Semántica	Significado
1000	Minimizar el retraso	Indica que una entrega rápida es importante para esos datagramas.
0100	Maximizar el caudal eficaz	Indica que una alta taza de datos es importante para esos datagramas.
0010	Maximizar la fiabilidad	Indica que un alto nivel de esfuerzo para asegurar la entrega es importante para esos datagramas
0001	Minimizar el costo monetario	Indica que el costo monetario de los enlaces por donde se atraviesa es importante para esos datagramas.
0000	Servicio normal	Indica que esos datagramas no requieren ningún tratamiento especial.

Cabe aclarar que en esta definición el significado de los bits es expresado en números binarios y no tiene ningún significado si por ejemplo se desactiva el bit de minimización de retraso, ya que estos se toman como un conjunto y no individualmente.

Los valores usados en el campo TOS son llamados "valores TOS", mientras que el valor del campo TOS en el paquete IP es llamado "TOS solicitado". Si el valor del campo TOS es 0000 entonces se le llama "TOS por defecto" [15].

Aunque la semántica de otros valores además de los cinco aquí descritos no ha sido definida, estos son perfectamente valores TOS legales y esta abierto para usos experimentales en la definición de nuevos servicios.

De acuerdo al RFC 1060 [16] ninguna de las aplicaciones comunes TCP/IP establecen múltiples valores para el campo TOS, por lo que claramente, establecer todos los bits equivale a no establecer ninguno de ellos, ya que al hacer esto se indica que ninguno de los tipos de optimización solicitados tiene mayor importancia que cualquiera de los otros.

La regla fundamental en esta especificación de Tipo de Servicio es que un anfitrión nunca debe ser penalizado por usar ToS. Además, si un anfitrión hace un uso apropiado de esta facilidad el servicio obtenido de la red debe ser al menos tan bueno (esperando que sea mejor) que si no se utilizara esta especificación.

Es importante aclarar que el uso de las palabras "minimizar" y "maximizar" usadas en las definiciones anteriores no indican por ejemplo, que si se establece el valor de TOS en 1000 (minimizar retraso) se garantice que el datagrama tenga un retraso que el usuario considere mínimo, más bien la red intenta seleccionar la ruta con menor retraso disponible; además la red no descartará paquetes simplemente porque cree que el retraso de las rutas disponibles es "muy alto". Por lo tanto, la interpretación de las palabras "minimizar" y "maximizar" pueden diferir entre lo que el usuario considera y lo que la red puede proporcionar.

3.4.3 Servicios Diferenciados.

Este método surge de la clara necesidad de contar con un método robusto y relativamente simple de proveer clases de servicios diferenciados al tráfico de Internet, que soporten diversos tipos de aplicaciones y requerimientos específicos de los usuarios. Servicios Diferenciados proporciona Calidad de Servicio en las redes empleando un pequeño y bien definido conjunto de bloques mediante los cuales pueden ser construidos una gran variedad de "*comportamientos de grupo*". Un pequeño modelo de bits en el octeto TOS de IPv4 o en el octeto Clase de Tráfico en IPv6 es utilizado para marcar un paquete para que reciba un tratamiento particular en el reenvío, o un "comportamiento por salto" en cada nodo de la red. Un común entendimiento acerca del uso y la interpretación de este modelo de bits es requerido para su uso entre dominios, interoperabilidad entre múltiples vendedores, y un razonamiento consistente acerca del comportamiento esperado en una red [17].

Para esto se estandarizó una distribución común para un campo de seis bits de ambos octetos (IPv4 e IPv6), llamado campo DS (*Differentiated Service*, Servicios Diferenciados). El RFC 2474 y el RFC 2475 definen la arquitectura y el uso general de los bits en el campo DS, sustituyendo la definición del octeto ToS en IPv4 del RFC 1349.

Un "servicio" se define como alguna característica significativa de la transmisión del paquete en una sola dirección a través de un conjunto de una o mas rutas en una red [18]. Estas características pueden ser especificadas

cuantitativamente o estadísticamente en términos de caudal eficaz, retraso, variación en el retraso, pérdida de paquetes, o bien, puede ser especificado en términos de alguna prioridad relativa en el acceso a los recursos de la red.

Esta arquitectura esta compuesta de un número de elementos funcionales implementados en los nodos de la red, incluyendo un pequeño conjunto de comportamientos de reenvío por salto, funciones de clasificación de paquetes, además de funciones de condicionamiento del tráfico incluyendo medición, marcado, configuración y políticas. Esta arquitectura logra escalabilidad implementando complejas funciones de clasificación y condicionamiento solo en los nodos frontera de la red, y aplicando comportamientos por salto a grupos de tráfico los cuales han sido apropiadamente marcados usando el campo DS de las cabeceras de IPv4 e IPv6.

Con el objetivo de ofrecer Calidad de Servicio extremo-a-extremo, esta arquitectura tiene dos componentes principales: Marcado de paquetes usando el byte TOS de IPv4, y los Comportamientos por Salto (PHB, Per Hop Behaviors). Veamos cada uno de ellos.

3.4.3.1 Marcado de Paquetes.

A diferencia de la Precedencia IP, el octeto de TOS ha sido totalmente redefinido (Como se muestra en la Figura 18), ahora son utilizados seis bits para la clasificación de paquetes. Este campo ha sido llamado Campo de Servicios Diferenciados (DS, Differentiated Services), en el cual dos de sus bits no son utilizados [19]. Los seis bits reemplazan los tres bits de Precedencia IP,

y son ahora llamados Punto de Código de Servicios Diferenciados (DSCP, Differentiated Services Code Point). Con DSCP pueden ser soportadas más de 64 diferentes clases/grupos (2^6) por nodo [20].

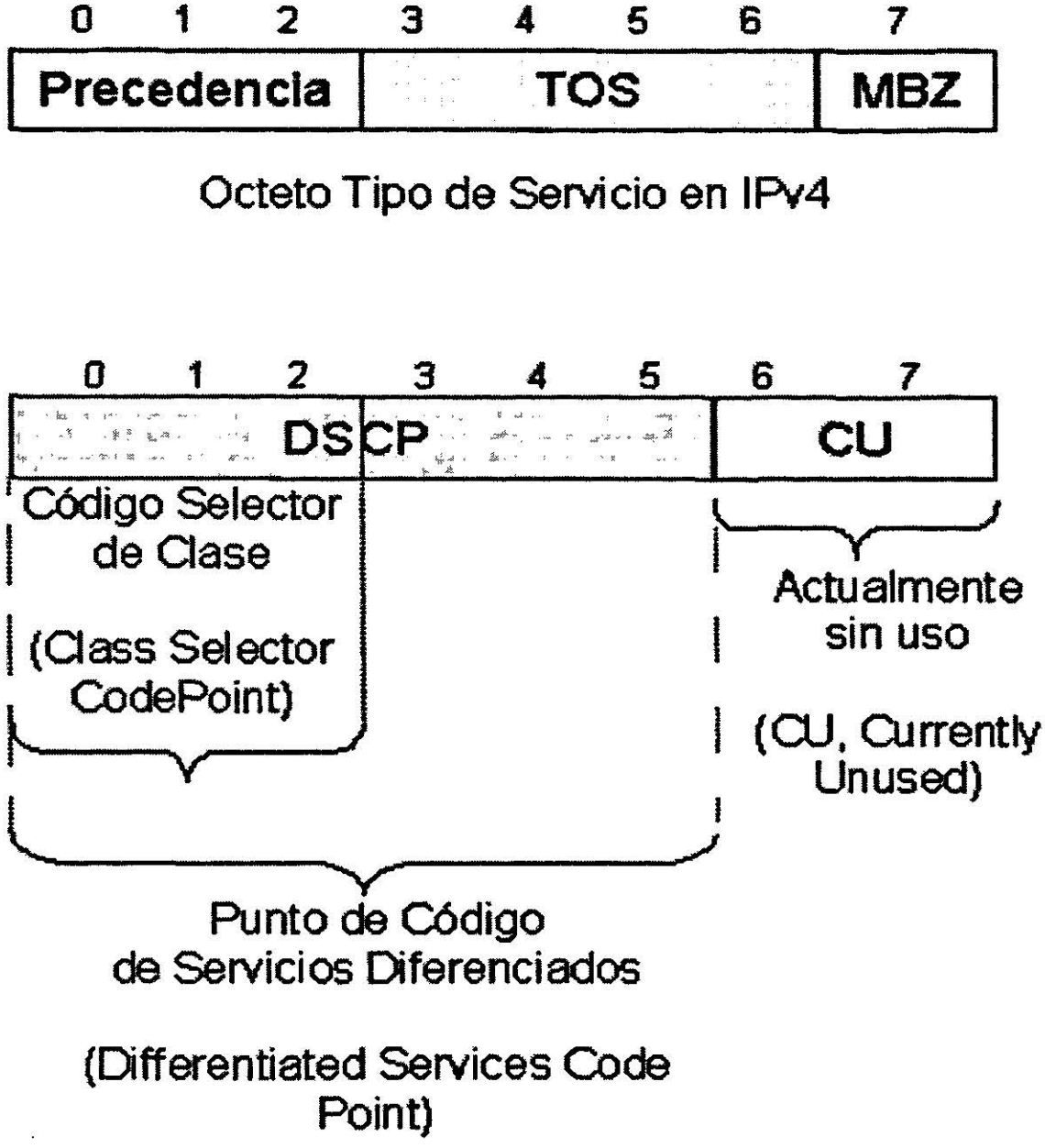


Figura 18. Redefinición del octeto TOS de IPv4.

3.4.3.2 Comportamientos por Salto.

Una vez que los paquetes han sido marcados con DSCP surge la siguiente pregunta: ¿Cómo podemos proporcionar una clasificación significativa sobre los flujos y brindar la Calidad de Servicio que se requiere?

Veamos algunas definiciones que nos responderán esta pregunta. La colección de paquetes que tienen el mismo valor DSCP entre ellos y cruzan por una dirección en particular son llamados Grupo de Comportamiento (BA, Behavior Aggregate); de esta manera paquetes de múltiples aplicaciones/fuentes pueden pertenecer a un mismo BA. Definamos un Comportamiento Por Salto (PHB, Per Hop Behavior) como "el comportamiento de reenvío externo observable aplicado por un nodo que cumple con Servicios Diferenciados a un Grupo de Comportamiento DS" [18]. En otras palabras, un PHB se refiere a los comportamientos en cuanto a calendarización de los paquetes, encolamiento, políticas o configuración por un nodo sobre cualquier paquete perteneciente a un BA, y es configurado por medio de un Acuerdo de Nivel de Servicio (SLA) u otras políticas.

Existen cuatro PHB estándares disponibles para construir una red habilitada con Servicios Diferenciados (DiffServ) y lograr una Calidad de Servicio de una gran granularidad.

3.4.3.2.1 Comportamiento por Salto por Omisión. El Comportamiento por Salto por Omisión (Default PHB) definido en el RFC 2474 [19] especifica esencialmente que un paquete marcado con un valor DSCP de '000000' obtiene el tradicional servicio de "mejor esfuerzo" en un nodo que cumple con Servicios

Diferenciados (esto es, un nodo que cumple con todas las especificaciones y requerimientos de Servicios Diferenciados). Además, si un paquete llega a un nodo que cumple con Servicios Diferenciados y el valor que tiene su DSCP no puede ser mapeado a cualquier otro PHB, este será mapeado al PHB por omisión.

3.4.3.2.2 Comportamiento por Salto Selector de Clase. El Comportamiento por Salto Selector de Clase (Class-Selector PHB) definido en el RFC 2474 [19] fue creado para mantener la compatibilidad con el esquema de Precedencia IP, en el que son definidos los valores DSCP de la forma 'xxx000', donde x puede ser 1 o 0. Estos códigos son llamados Puntos de Código de Selección de Clase (Class-Selector CodePoint). Note que el código por omisión ('000000') es un código de selección de clase. El PHB asociado con un código de selección de clase es un PHB selector de clase.

Estos PHBs mantienen la mayoría de los comportamientos de reenvío que los nodos que implementan Precedencia IP basados en la clasificación y reenvío. Por ejemplo, paquetes con un valor DSCP '110000' o Precedencia IP '110' tienen un trato preferencial en el reenvío comparados con paquetes con un valor DSCP '100000' o Precedencia IP '100'. Estos PHBs aseguran que nodos que cumplen con Servicios Diferenciados puedan coexistir con nodos que implementan Precedencia IP.

3.4.3.2.3 Comportamiento por Salto de Reenvío Acelerado. El *Comportamiento por Salto de Reenvío Acelerado* (EF PHB, Expedited Forwarding PHB) definido en el RFC 2598 [21] y redefinido en el RFC 3246 [22] es el elemento clave en Servicios Diferenciados para proveer una baja pérdida, baja latencia y asegurar el ancho de banda a los servicios. Aplicaciones como voz sobre IP (VoIP), video, y entrenamiento en línea, requieren un tratamiento robusto en la red. EF puede ser implementado usando prioritización de colas en conjunto con una limitación de tasas de envío para las clases de tráfico. Aunque cuando se implementan Servicios Diferenciados con EF PHB se pueden lograr servicios "premium" para algunos usuarios, el objetivo de estos es el uso en aplicaciones críticas, debido a que si existe una congestión es imposible dar un tratamiento de alta prioridad a todo el tráfico. EF PHB es especialmente adecuado para aplicaciones como voz sobre IP que requieren una pérdida de paquetes muy baja, garantizar el ancho de banda, un retraso bajo y una variación en el retraso mínima.

3.4.3.2.4 Comportamiento por Salto de Reenvío Asegurado. El Comportamiento por Salto de Reenvío Asegurado (AF_{xy} PHB, Assured Forwarding PHB) definido en el RFC 2597 [23] y actualizado en el RFC 3260 [24] define un método mediante el cual los Comportamientos por Grupo pueden obtener diferentes aseguramientos en el reenvío. Por ejemplo, el tráfico puede ser dividido en clase oro, plata, bronce, donde la clase oro tiene asignado el 55% del ancho de banda del enlace, la clase plata tiene el 30%, y la bronce el 15%. Los Comportamientos Por Salto AF_{xy} definen cuatro clases AF_x llamadas

AF_1, AF_2, AF_3, AF_4. A cada clase le es asignado una cierta cantidad de espacio en la memoria temporal o buffer y ancho de banda dependiendo del Acuerdo de Nivel de Servicio (SLA) o políticas.

En cada clase AF_x es posible especificar tres valores de precedencia para "tirar" los paquetes. Esto es, si hay una congestión en cierto nodo con Servicios Diferenciados y los paquetes de cierta clase AF_x necesitan ser tirados (digamos la clase AF_1), los paquetes de las clases AF_{xy} serán tirados bajo la siguiente regla: $dP(AFx_1) \leq dP(AF_{x2}) \leq dP(AF_{x3})$, donde $dP(AF_{xy})$ es la probabilidad de que los paquetes de la clase AF_{xy} sean tirados. Entonces el subíndice "y" de la notación AF_{xy} denota la precedencia para tirar un paquete de la clase AF_x. Por ejemplo, paquetes de la clase AF_{13} van a ser tirados antes que los de la clase AF_{12}, antes que la clase AF_{11}. Este concepto es muy útil para penalizar flujos de tráfico que exceden el ancho de banda asignado.

Los valores del DSCP para estos PHB´s se denotan bajo la regla 'xyzab0', donde 'xyz' representan las cuatro clases AF_x 001, 010, 011, 100; mientras que 'ab' representan los bits de precedencia de tirado de paquetes. Por ejemplo, para una clase de tráfico AF_1 con una precedencia media, su DSCP sería representado por los bits 001100.

3.4.3.3 Integración de todos los componentes

La integración de todos los componentes de este modelo podría parecer una receta de cocina. De este modo, una región de Servicios Diferenciados (DS-Region) puede estar compuesta por uno o más Dominios DS, los cuales

posiblemente se encuentren bajo autoridades administrativas diferentes. El secreto de esta receta son las políticas o el Acuerdo de Nivel de Servicio.

Un dominio de Servicios Diferenciados lo podríamos definir formalmente como [18]: un conjunto contiguo de nodos los cuales operan con un conjunto común de servicios, proporcionando políticas y definiciones de saltos por nodos (PHB). Mientras que una Región de Servicios Diferenciados la definiríamos como: un conjunto contiguo de dominios DS los cuales pueden ofrecer Servicios Diferenciados sobre rutas a través de los dominios DS.

La figura 19 muestra un pequeño vistazo a esta arquitectura. Para que haya una verdadera Calidad de Servicio la ruta IP completa por la cual atraviesan los paquetes debe ser capaz de brindar Servicios Diferenciados. Por ejemplo, una política de servicio puede definir que el trafico EF (Reenvío Acelerado) obtenga el 10%, el tráfico oro el 40%, el plata el 30%, el bronce el 10%, y el restante 10% para el trafico con servicio de mejor esfuerzo (PHP por omisión). El tráfico oro, plata y bronce pueden ser mapeados a las clases AF, por ejemplo AF_1, AF_2, AF_{13}. Esto puede ser forzado en cualquier parte de la nube de comunicaciones.

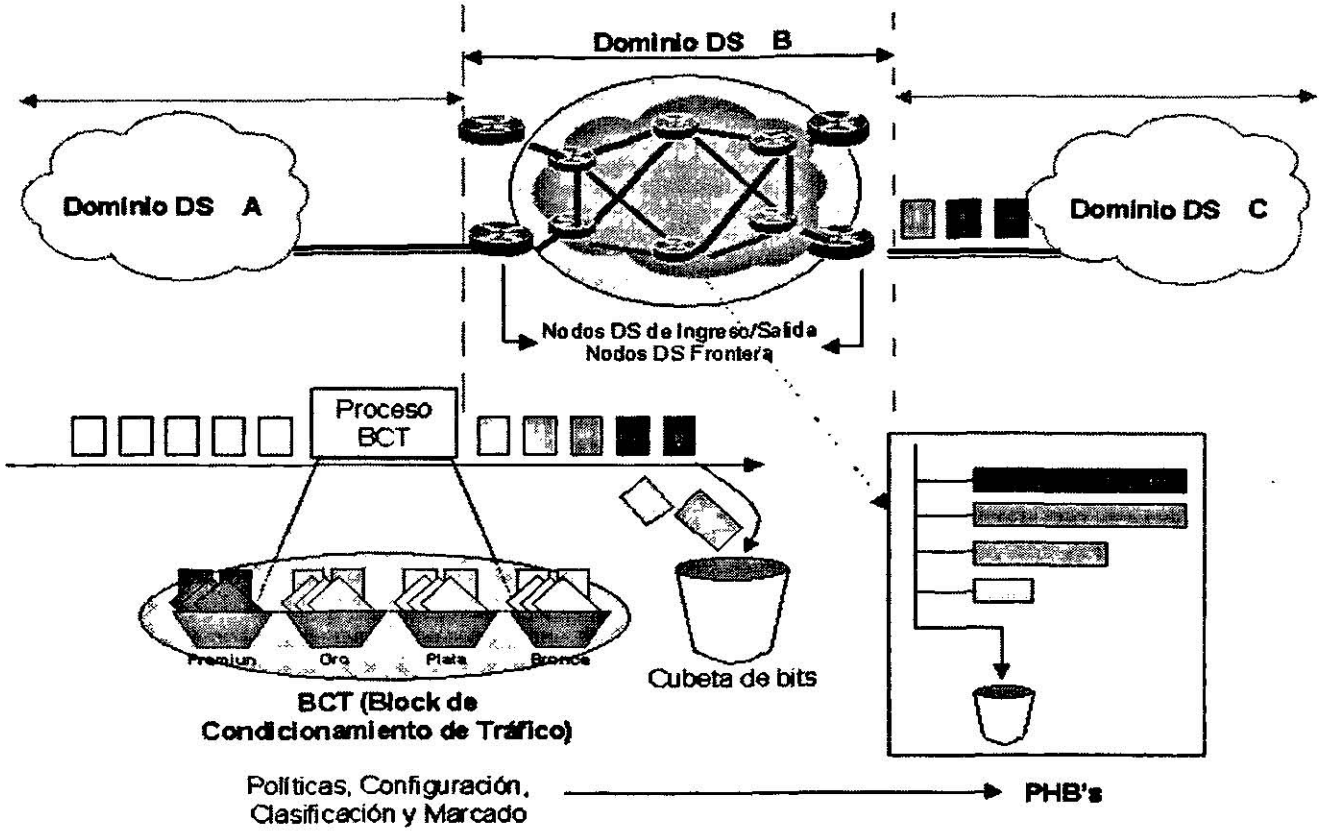


Figura 19 Vistazo al modelo de Servicios Diferenciados.

Un Dominio DS por sí mismo está constituido por nodos de Ingreso DS, nodos DS Interiores (en el núcleo) y nodos de DS de Salida. Los nodos de Ingreso DS son nodos frontera cuya tarea es manejar el tráfico que entra a un Dominio DS; un nodo Interior DS es un nodo que no es un nodo frontera y cuya tarea corresponde a emplear los PHB para reenvío de los paquetes; mientras que un nodo DS de Salida es un nodo frontera cuya tarea es manejar el trafico que abandona el Dominio DS.

Además un nodo de Ingreso o de Salida es un nodo DS Frontera el cual conecta un Dominio DS a otro Dominio DS, o bien a un dominio que no tiene capacidad de brindar Servicios Diferenciados. Típicamente un nodo DS

Frontera realiza condicionamiento de tráfico; un acondicionador de tráfico (Traffic Conditioner) es una entidad que realiza funciones de condicionamiento de tráfico las cuales pueden contener medidores, marcadores, desechadores o tiradores y configuradores de tráfico. Un acondicionador de tráfico puede además remarcar un flujo de tráfico o puede descartar paquetes para alterar las características temporales del flujo de datos y que éste pueda acoplarse con un perfil de datos definido.

Los nodos DS Interiores fuerzan el Comportamiento Por Salto (PHB) apropiado empleando políticas o técnicas de configuración de tráfico, y algunas veces remarcando los perfiles de los paquetes, dependiendo de la política o del Acuerdo de Nivel de Servicio. La figura 20 muestra un acondicionador de tráfico típico en las orillas de un Dominio DS.

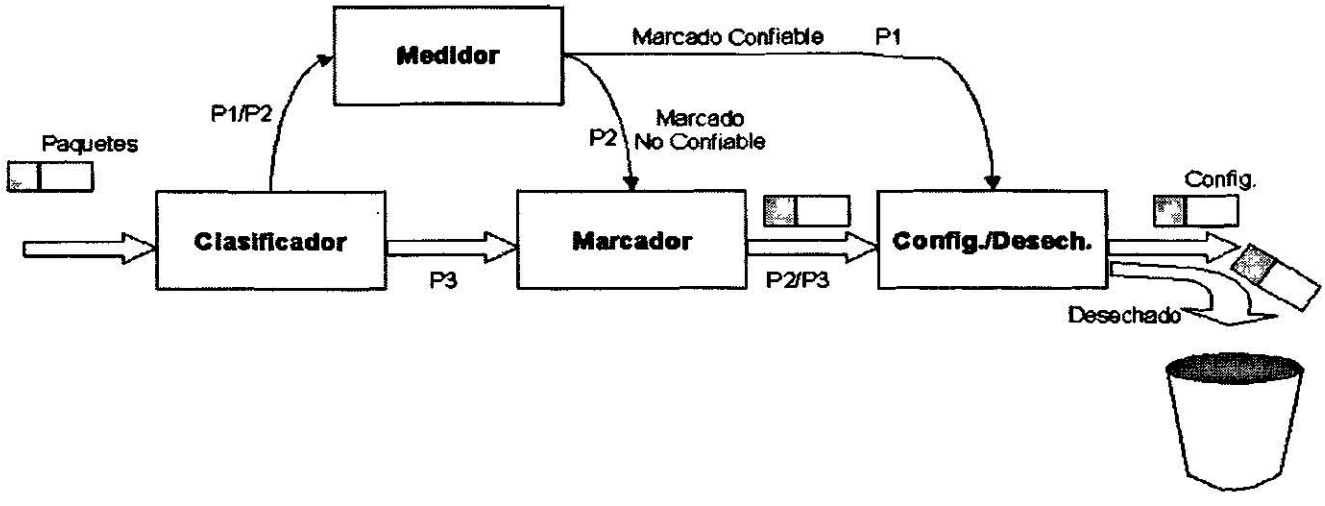


Figura 20. Bloque de Acondicionamiento de Tráfico de Servicios Diferenciados (TCB, *Traffic Conditioner Block*).

Basándonos en la figura 20 y en el RFC 2475 [18] definimos un Clasificador (Classifier) como una entidad que selecciona paquetes basándose en el contenido de las cabeceras de los paquetes y de acuerdo a reglas definidas; un Medidor o Contador (Meter) es una entidad que lleva a cabo procesos de medición de propiedades temporales (por ejemplo, tasa de transferencia) de un flujo de tráfico seleccionado por un Clasificador, el cual puede ser usado para afectar las operaciones de Marcado, Configuración, Desecho, o puede se usado solamente con propósitos de contabilidad y medición.

El Marcador (Marker) es una entidad que establece los valores del código DS (DSCP) sobre un paquete basándose en reglas definidas. Por último el proceso de Configurar/Tirar; donde el Configurador o Modelador (Shaping) es una entidad que se encarga del proceso de retrasar los paquetes de un flujo de tráfico con la finalidad de que se acople a algún perfil de tráfico definido; mientras que un Desechador o Tirador de paquetes (Dropper) es una entidad que se encarga de el proceso de descartar paquetes basado es reglas o políticas definidas.

3.4.4 Servicios Integrados.

El modelo de servicios integrados se basa en el modelo tradicional de entrega de datagramas, pero permite a las fuentes y destinos intercambiar mensajes de señalización los cuales establecen clasificación adicional a los paquetes así como tratamiento específico en los reenvíos sobre cada nodo a lo

largo de toda la ruta entre ellos. Estas políticas son actualizadas en tiempos determinados y no responden instantáneamente a la actividad de la red.

Este modelo se basa en el Protocolo de Reservación de Recursos (RSVP, Resource ReSerVation Protocol), que es un protocolo de señalización que proporciona configuración y control en la reservación para habilitar los Servicios Integrados (IntServ, Integrated Services), el cual trata de hacer una aproximación lo más cercana a la emulación de circuitos sobre redes IP [8]. RSVP es la más compleja de todas las tecnologías de QoS, tanto para las aplicaciones (anfitriones) como para los elementos de la red (conmutadores y enrutadores). Este modelo proporciona el más alto nivel de Calidad de Servicio en términos de servicios garantizados, mayor granularidad en la asignación de recursos y una detallada retroalimentación de las aplicaciones y los usuarios que hacen uso de la Calidad de Servicio. Veamos un vistazo de cómo funciona este protocolo mediante la figura 21.

Primeramente, el emisor caracteriza el tráfico en base a los límites superiores e inferiores del ancho de banda, retraso y variación del retraso. RSVP envía un Mensaje de Ruta del emisor que contiene su información de Especificación de Tráfico (TSpec, Traffic Specification) hacía la dirección destino. Cada enrutador habilitado con RSVP a lo largo de la ruta de "bajada" establece un "Estado de Ruta" que incluye la dirección destino previa del Mensaje de Ruta (esto es, el siguiente salto de "subida" hacia el emisor).

Para realizar la reservación de recursos, el receptor envía un mensaje de "subida" RESV (Solicitud de Reservación, Reservation Request). Además del TSpec el mensaje RESV incluye una Especificación de Solicitud (RSpec, Request Specification) que indica el tipo de Servicios Integrados requeridos (puede ser Carga Controlada o Garantizado), además de una Especificación de Filtro (Filter Spec, Filter Specification) que describe los paquetes para los cuales se ha hecho la reservación de recursos (por ejemplo, el protocolo de transporte o el número de puerto). Juntos, el RSpec y la Especificación de Filtro representan un Descriptor de Flujo (Flow-descriptor) que el enrutador utiliza para identificar cada reservación (conocido también como "flujo" o una "sesión").

Cuando cada enrutador RSVP a través de la ruta de "subida" recibe un Mensaje RESV, este utiliza el proceso de control de admisión para autenticar la solicitud y asignar los recursos necesarios. Si la solicitud no puede ser cumplida (debido a falta de recursos o falla en la autorización) el enrutador envía un error al receptor. Si es aceptado, el enrutador envía el mensaje RESV hacia el siguiente enrutador.

Cuando el último enrutador recibe el Mensaje RESV y acepta la solicitud, este envía un mensaje de confirmación al receptor (Nota: el "último enrutador" es el más cercano al emisor).

Existe un proceso explícito de demolición para la reservación cuando el emisor o el receptor terminan la sesión RSVP.

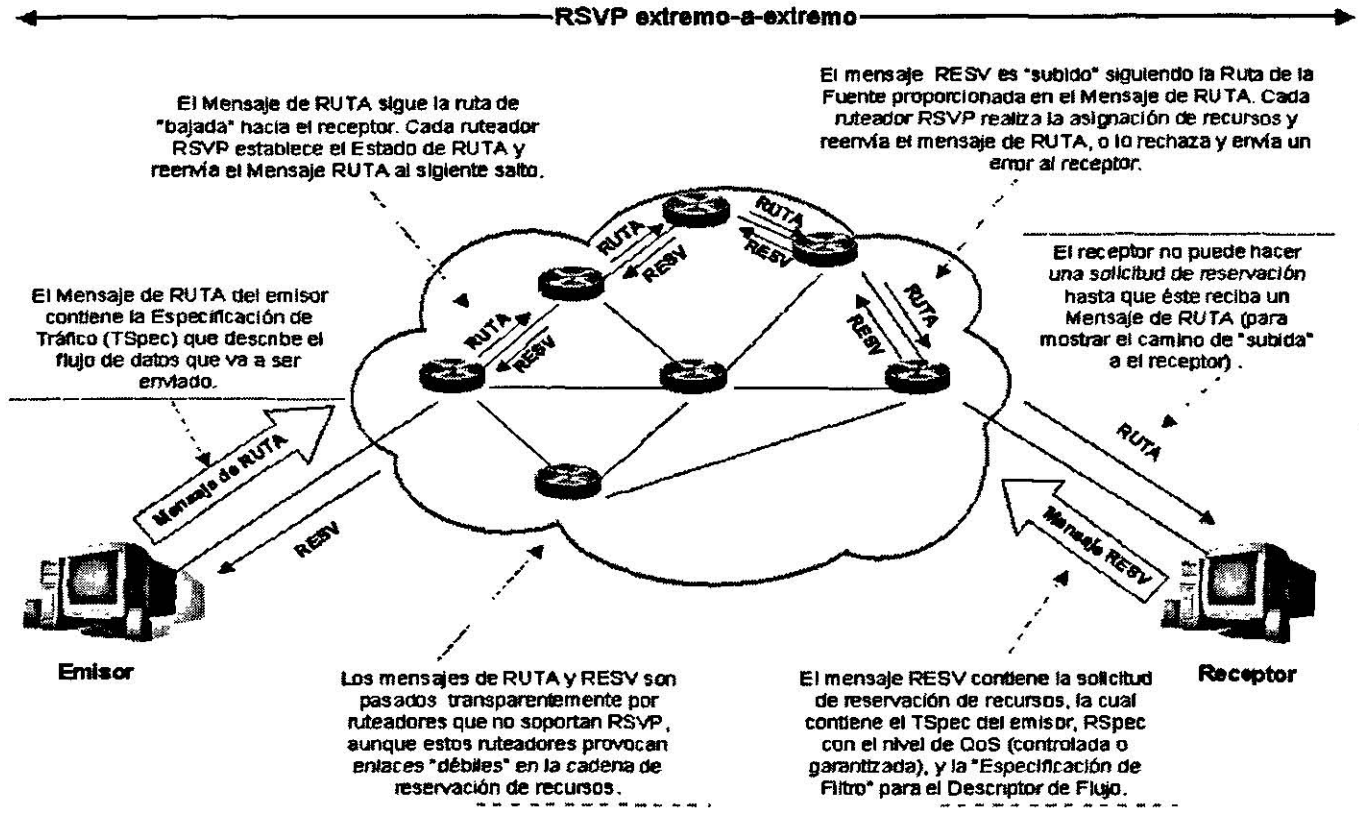


Figura 21. Servicios Integrados utilizando RSVP.

Como ya hemos mencionado RSVP proporciona Servicios Integrados, de los cuales existen fundamentalmente dos diferentes tipos:

- Garantizado.- Este tipo trata de acercarse lo más posible a la emulación de un circuito virtual dedicado. Éste provee un consistente (matemáticamente probable) límite en los retrasos de las colas extremo-a-extremo por la combinación de parámetros de varios elementos de la red sobre una ruta, además de asegurar la disponibilidad del ancho de banda de acuerdo a los parámetros de la Especificación de Tráfico [25].

- Carga Controlada.- Este tipo es el equivalente al "servicio de mejor esfuerzo bajo condiciones de baja carga". Por lo tanto, es "mejor que el servicio de mejor esfuerzo", pero no puede asegurar los estrictos límites en el servicio que el servicio Garantizado promete [26].

3.5 Conclusiones

Como hemos visto, la implementación de Calidad de Servicio no es una varita mágica que viene a resolvernos todos nuestros problemas de transporte de tráfico, ya que no es posible en todos los casos aplica su implementación, sin embargo si resulta de mucha utilidad en muchas de las situaciones.

El implementarlo o no es una decisión de los administradores de la red y debe de llevar un estudio previo de sus redes. Este estudio comprende desde la utilización de los enlaces, tipos de tráfico que circulan en ellos, prioridad de las aplicaciones, usuarios, etcétera, por lo que, como en todo proyecto, el gran trabajo se encuentra en las fases de análisis y planeación, y Calidad de Servicio no es la excepción.

En este capítulo vimos también algunas de las propuestas que existen para la implementación de Calidad de Servicio, donde cada una de ellas tiene sus ventajas y limitantes; aunque estos no son los únicos trabajos que existen sobre este tema, ya que es posible implementar Calidad de Servicio utilizando algunas tecnologías como MPLS (MultiProtocol Label Switching, Conmutación

de Etiquetas MultiProtocolo) [27], ATM (Asynchronous Transfer Mode, Modo de Transferencia Asíncrono) [28] o algunas otras iniciativas [29] las cuales están fuera del alcance de este trabajo.

Así pues vemos que es posible implementar desde mecanismos básicos de priorización tanto de capa 2 como de capa 3, hasta complejos sistemas como Servicios Diferenciados ó Servicios Integrados el cual asegura una Calidad de Servicio extremo-a-extremo.

La decisión de qué tecnología emplear tiene que ver con diversos factores tales como la granularidad de Calidad de Servicio que se quiere lograr, los equipos de comunicaciones con los que se cuenta, los recursos económicos y tecnológicos para realizar implementaciones de este estilo, hasta el control que tenemos en la administración de las redes bajo las cuales queremos implementar estas tecnologías.

Los equipos de comunicaciones juegan un rol muy importante en estos esquemas, ya que ellos son los encargados de analizar todo el tráfico que pasa por las redes y tomar decisiones sobre el tratamiento que se le tiene que dar a los paquetes. Es por eso que en el siguiente capítulo trataremos temas de gran importancia dentro de los equipos de comunicaciones como lo son los mecanismos de encolamiento, ya que en gran medida estos determinan la eficiencia en los tiempos en la entrega de la información.

CAPITULO 4

COMO MEDIR LA CALIDAD DE SERVICIO

Hemos podido observar a lo largo de este trabajo cómo la Calidad de Servicio puede beneficiar en ciertos ambientes, así como los diferentes modelos o arquitecturas de implementación, pero ningún método es válido si no tenemos una forma de medir o "censar" el comportamiento de nuestro ambiente o del nivel deseado de funcionalidad.

El objetivo de este capítulo es presentar un panorama general de los elementos involucrados en la medición de la Calidad de Servicio, así como mostrar la forma en que estos elementos influyen en el comportamiento general de la red.

Para esto hablaremos primero de algunos parámetros esenciales para la medición de la Calidad de Servicio, describiendo el papel que juegan en el modelo general y como la variación de los mismos afecta en el desempeño.

Otro factor a considerar en este capítulo son los diferentes mecanismos de encolamiento existentes, los cuales afectan el desempeño de los equipos de comunicaciones, para esto revisaremos algunos de estos mecanismos dando una perspectiva general de los mismos, para por último mostrar como todos

estos factores se interrelacionan y pueden afectar el comportamiento de nuestra red.

4.1 Parámetros a Medir

El tener parámetros de medición es de suma importancia y cobran mayor relevancia, ya que éstos son algunas de nuestras variables de las cuales podemos nosotros tener el control. Estos parámetros nos proporcionan un valor cuantitativo del funcionamiento de nuestro ambiente, y con esa información podemos tomar decisiones mas acertadas.

En esta sección analizaremos cuatro parámetros los cuales, si bien no son los únicos, si nos permiten lograr un mejor entendimiento de nuestro ambiente en base a ellos.

4.1.1 Retraso.

El retraso, también conocido como latencia (en inglés delay y latency respectivamente) es una expresión que determina cuanto tiempo toma un paquete de datos en ir de un punto de origen a un punto destino.

El retraso es medido en unidades de tiempo, es decir segundos. Existen dos tipos de mediciones del retraso, el retraso en una sola dirección (one-way delay) [30] y el retraso de viaje de ida y vuelta (round-trip delay) [31].

La diferencia primordial es que el retraso de ida y vuelta, como el nombre lo indica, mide el tiempo que tarda un paquete en ir de una dirección origen a un

destino y regresar a la dirección origen. El problema con esta medición es que debido a que en las redes TCP/IP los datagramas tomas rutas distintas para alcanzar sus destinos, la ruta de ida no siempre en la misma ruta que de regreso, proporcionando un valor que aunque es muy aproximado, no es real.

La medición en un solo sentido proporciona solamente el tiempo que le lleva a un paquete ir de una dirección origen a una dirección destino. Aunque existen propuestas para que se utilicen estas técnicas de medición, la más utilizada por simplicidad sigue siendo la medición de retraso de ida y vuelta, por lo que una manera aproximada de obtener el tiempo que tarda en ir en una sola dirección es:

$$T_{ida} = T_{ida\text{-}vuelta} / 2$$

4.1.2 Variación en el Retraso.

La variación en el retraso (jitter) es la variación en los tiempos entre llegadas de los datos a un destino final. Para variaciones mínimas es llamado jitter (menos de 10 Hz), mientras que para variaciones más grandes (menos de un día) esta variación es llamada "wander" [32].

Una definición similar indica que la variación en el retraso es la variación (en valor absoluto) de la diferencia D en espacio de los paquetes en el receptor comparado con el emisor para un par de paquetes dados [33].

Esto es equivalente a decir que es la diferencia en el "tiempo relativo de tránsito" para dos paquetes. Este tiempo relativo de tránsito es la diferencia

entre el sello de tiempo (time stamp) y el reloj del receptor en el momento de la llegada del paquete, medidos en las mismas unidades.

Entonces, si S_i es el sello de tiempo para un paquete *i*, y R_i es el tiempo de arribo en unidades de marca de tiempo para un paquete *i*, entonces podemos decir que para dos paquetes *i* y *j*, D puede ser expresado como:

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_i) - (R_j - S_i)$$

Cabe aclarar que la pérdida de paquetes es ignorada en el cálculo de la variación del retraso, debido a que los paquetes perdidos pueden considerarse como paquetes con retraso infinito, lo cual hace impráctico su uso para los cálculos de la variación del retraso.

4.1.3 Pérdida de Paquetes.

La tasa de pérdida de paquetes es la fracción de todos los paquetes que no llegaron a su destino. Esto puede ser debido a varias causas, por ejemplo, la red puede "tirar" paquetes en cualquier momento si la suma de comprobación no es correcta, debido a fallas en el medio físico de transmisión. Los paquetes con formato no definido también son "tirados" e incrementan la pérdida de paquetes.

En otros casos los paquetes son descartados debido a congestión en los equipos de comunicaciones, esto debido a que no hay suficiente memoria temporal en los equipos provocando un desbordamiento en las colas. Este caso es el que mas nos interesa y se tratará mas adelante.

El medir la tasa de paquetes perdidos puede resultar simple con la siguiente formula:

$$\Delta_{\text{paquetes perdidos}} = [(P_{\text{enviados}} - P_{\text{recibidos}}) / P_{\text{enviados}}] * 100$$

Lo mas difícil es medir los paquetes perdidos o bien los paquetes recibidos para poder hacer nuestro cálculo. Los equipos de comunicaciones proporcionan mecanismos para obtener estos datos y su estudio esta fuera del alcance de este trabajo. Existen algunas iniciativas para estas mediciones las cuales pueden revisarse a detalle en el RFC 2680 [34].

4.1.4 Caudal Eficaz.

El caudal eficaz (throughput) es la tasa total de datos transmitidos entre nodos; expresa una fracción de la capacidad y puede ser interpretado como la utilización [3]. Este parámetro es expresado en unidades de datos por periodo de tiempo, por ejemplo, bits por segundo.

Cabe aclarar que el caudal eficaz aquí mencionado se refiere a la tasa de paquetes de datos exitosamente transmitidos, los cuales incluyen también los bits de cabecera y de arrastre, adicionalmente a los datos; existe otro parámetro llamado Carga Ofrecida a la red, el cual se refiere al total de paquetes ofrecidos a la red incluyendo paquetes de control, colisiones, etc. Este parámetro esta fuera del alcance de este trabajo, pero puede consultar [3] para mayor referencia.

Para medir el caudal eficaz de extremo a extremo nos basaremos en la siguiente formula:

Caudal Eficaz = Datos recibidos / Tiempo total

Donde:

Datos recibidos = Datos correctamente recibidos por el destinatario (medidos en bits).

Tiempo total = El tiempo que tardaron esos datos en ir desde el origen hasta el destino (en segundos).

4.2 Colas de Espera

Una cola de espera se forma cuando un servidor se encuentra ocupado atendiendo una solicitud, por lo que las nuevas solicitudes que entran al sistema tienen que esperar a ser atendidas. En los equipos de comunicaciones y en las computadoras no es la excepción, ya la mayoría de las veces se forman colas de espera para ser atendidos por determinado proceso.

En los equipos de comunicaciones (conmutadores y enrutadores) las colas se forman cuando la tasa de llegada de datos es mayor a la capacidad del canal de salida, por lo que los paquetes entrantes tienen que almacenarse en una memoria temporal o buffer esperando a ser procesados. Debido a que la memoria temporal es un recurso finito, las colas de espera tienen un límite en el tamaño según la capacidad de la memoria temporal, por lo que cuando esta memoria llega a su límite todos los paquetes entrantes a ese puerto de comunicaciones son "tirados" debido a que no hay recursos para almacenarlos.

Esto afecta al rendimiento debido a que en algunos casos (por ejemplo si usa TCP como protocolo de transporte) el emisor al enterarse que sus paquetes fueron "tirados" tiene que reenviar la misma información; esto hace que el control de las colas de espera sea un aspecto muy importante.

En la mayoría de los casos, los equipos de comunicaciones implementan memorias temporales (y por lo tanto colas) en la interfaz de entrada al puerto, una memoria temporal para los procesos internos del equipo de comunicaciones, y una más en la interfaz de salida. Las menciones a colas de espera que hagamos en este trabajo nos estaremos refiriendo a la cola que se forma en la interfaz de salida.

A continuación haremos una revisión a algunos de los diferentes mecanismos de encolamiento que existen.

4.2.1 Primero en Entrar Primero en Salir (FIFO).

Este es el algoritmo básico de colas, se basa en la premisa el "primero en entrar" es el "primero en salir", de ahí su nombre (FIFO, First In First Out). En la forma más simple, las colas FIFO guardan los paquetes cuando la red se congestiona y cuando la red comienza a descongestionarse los reenvía en el orden en que llegaron. FIFO es el algoritmo de colas por omisión y no requiere ninguna configuración (figura 22).

Entre sus beneficios se encuentran:

- El mecanismo FIFO representa una carga computacional muy baja en el sistema.
- El comportamiento de FIFO es muy predecible, debido a que los paquetes no son reordenados y el máximo retardo de espera esta determinado por el tamaño de la cola.

Algunas de sus limitantes son:

- Una sola cola FIFO no permite organizar los paquetes almacenados en la memoria temporal, por lo que no se puede diferenciar entre diferentes clases de tráfico.
- Durante periodos de congestión FIFO beneficia a los flujos de datos UDP sobre los flujos TCP; esto debido a que cuando existe pérdida de paquetes debido a una congestión, TCP reduce su tasa de transmisión, mientras que como UDP hace caso omiso de la pérdida de paquetes mantiene su misma tasa de transmisión. Esto provoca que se incremente el retraso y el jitter en los flujos TCP, además de la reducción de la cantidad de recursos en el canal de salida.
- Una ráfaga de tráfico puede consumir totalmente la memoria temporal de una cola FIFO causando que se les niegue el servicio a los demás flujos entrantes hasta que esa ráfaga de tráfico sea servida, incrementando el retraso, jitter y pérdida de paquetes en las aplicaciones.

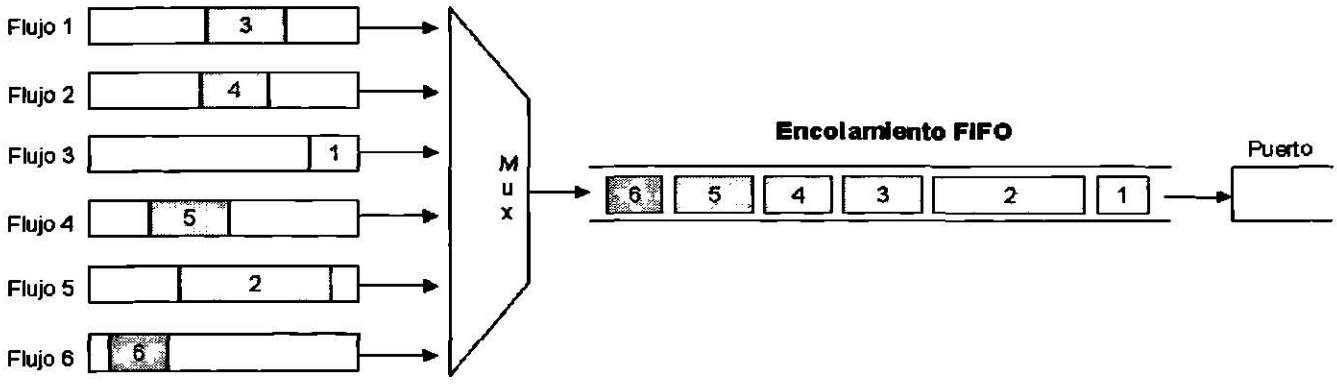


Figura 22. Mecanismo de encolamiento FIFO.

Otro punto importante es que las colas FIFO no toman decisiones sobre la prioridad de los paquetes; el orden de llegada es el que determina el ancho de banda necesario, la rapidez con que se atiende, y la asignación en el buffer. Tampoco provee protección contra aplicaciones con comportamientos extraños (en cuanto a requerimiento de recursos). Las ráfagas de tráfico pueden causar grandes retrasos en la entrega de tráfico de aplicaciones sensitivas o de señales de control de la red. El algoritmo FIFO es un buen comienzo para control de tráfico, pero la necesidad de redes inteligentes de hoy en día necesita algoritmos más sofisticados [35].

4.2.2 Encolamiento Priorizado (PQ).

El mecanismo Encolamiento Priorizado (PQ, Priority Queuing) es la base para los algoritmos de encolamiento programado, los cuales son diseñados para proporcionar un método relativamente simple para soportar servicios diferenciados. En este mecanismo, primeramente son clasificados los paquetes y asignados a las diferentes colas de prioridad.

Permite definir tipos de tráfico (por ejemplo: alto, medio, bajo) para cada interfaz. Una vez que el tráfico ha sido asignado a cada una de las colas (según la prioridad de los paquetes) los paquetes en la cola de prioridad alta son transmitidos primero, cuando la cola de alta prioridad esté vacía se transmiten los paquetes de la cola de siguiente de mayor prioridad, y así sucesivamente (figura 23).

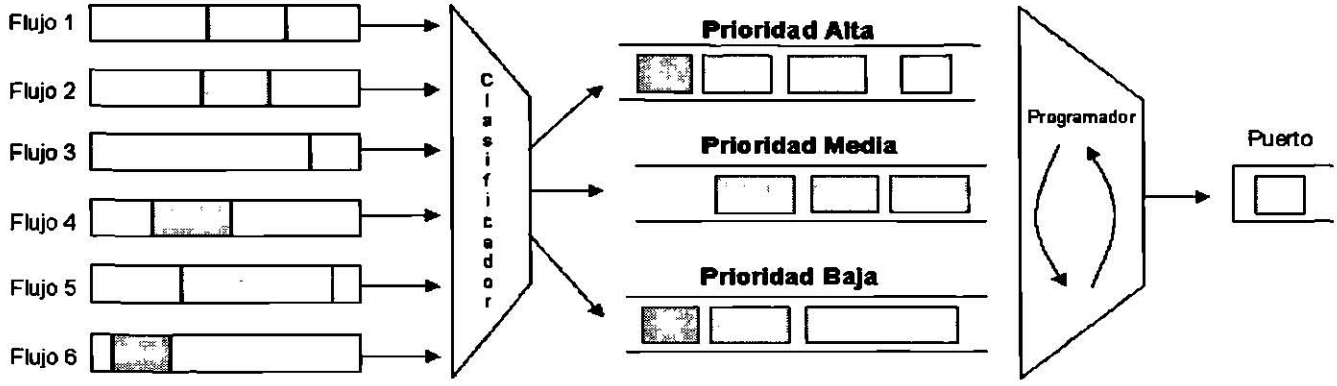


Figura 23. Mecanismo de encolamiento PQ.

Entre sus beneficios se encuentran:

- El mecanismo PQ representa una carga computacional muy baja en el sistema.
- PQ permite a los equipos organizar los paquetes almacenados en memoria temporal para después servir cada una de las clases de tráfico de manera independiente.

Algunas de sus limitantes son:

- Si la cantidad de tráfico de alta prioridad no es condicionada en los extremos de la red, las colas de prioridad baja pueden experimentar

un excesivo retraso esperando a que la cola de alta prioridad sea servida.

- Si el volumen de tráfico de alta prioridad se vuelve excesivo, el tráfico de baja prioridad puede ser descartado debido a que puede existir un desbordamiento en el espacio en la memoria temporal para esa cola.
- Algún flujo de datos con comportamiento extraño puede causar el incremento en el retraso y el jitter para otros flujos de datos de alta prioridad en la misma cola, esto debido a que ya estando en la cola de alta prioridad se atienden los paquetes en un mecanismo FIFO.
- El mecanismo PQ no es la solución para resolver la limitación de las colas FIFO con respecto a que los paquetes UDP son favorecidos sobre los paquetes TCP durante periodos de congestión; si se pretende asignar los flujos de TCP a la cola de alta prioridad, el manejo de ventanas TCP y los mecanismos de control de flujo del protocolo provocarán que este consuma todos los recursos de ancho de banda disponibles en el puerto de salida, lo cual causaría una privación en el servicio a los flujos UDP.

Comúnmente los fabricantes implementan este mecanismo para operar en dos modos diferentes: encolamiento con prioridad estricta y encolamiento con prioridad de tasa controlada. El mecanismo PQ estricto asegura que los paquetes en la cola de alta prioridad siempre sean servidos antes que las colas de baja prioridad. El problema con este método es, como ya se había

mencionado, que una excesiva cantidad de tráfico de alta prioridad puede provocar una "hambruna" de ancho de banda para las colas de baja prioridad.

Con el mecanismo PQ de tasa controlada se permite a los paquetes en las colas de alta prioridad ser atendidos antes que los paquetes de las colas de prioridad mas baja solamente sí la cantidad de tráfico en la cola de alta prioridad está por debajo de un valor configurable. Por ejemplo, pensemos que la cola de alta prioridad tiene una tasa límite de 35% del total del ancho de banda del puerto de salida, conforme la cola de alta prioridad consuma menos del 35% los paquetes de esta cola serán atendidos antes que los paquetes de las colas de baja prioridad. Mientras que si la cola de alta prioridad consume más del 35% del ancho de banda del puerto de salida, los paquetes de las colas de baja prioridad pueden ser servidos antes que las colas de alta prioridad. No existe un estándar para este método y cada fabricante implementa su propio mecanismo para decidir cuando servir los paquetes de las colas de baja prioridad con respecto a los de alta prioridad [36].

Sin embargo, implementar mecanismos de encolamiento priorizado para diferentes tipos de servicio requiere que se lleve a cabo un efectivo trabajo de condicionamiento de tráfico en los extremos de las redes para prevenir que las colas de alta prioridad se adueñen de los recursos. Si no se cuida esta parte al realizar nuestro diseño de la red, estos servicios se pueden volver imposibles de soportar.

4.2.3 Encolamiento Justo (FQ).

El mecanismo de Encolamiento Justo (FQ, Fair Queuing) fue propuesto por John Tagle en 1987. FQ es la base para los tipos de colas que son diseñadas para asegurar que cada flujo de datos tenga un acceso justo a los recursos de la red y prevenir que ráfagas de tráfico consuman más de lo que justamente les corresponde del ancho de banda del puerto de salida.

En este mecanismo, los paquetes primeramente son clasificados en flujos por el sistema y después asignados a la cola correspondiente dedicada específicamente para ese flujo. Después, las colas son servidas un paquete por turno en un orden round-robin; las colas vacías son descartadas (figura 24). Este mecanismo es también conocido como encolamiento por flujo o basado en flujos.

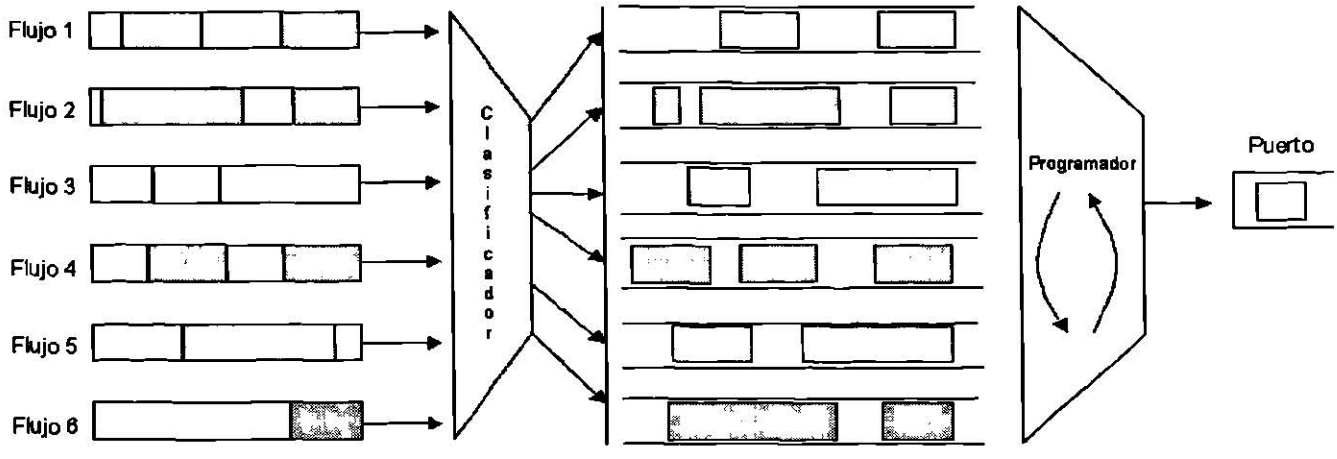


Figura 24. Mecanismo de encolamiento FQ.

El principal beneficio de FQ es que aún en casos de ráfagas de tráfico o flujos con comportamientos extraños no se degrada la calidad en el servicio

entregado para otros flujos, debido a que cada flujo es aislado en su propia cola. Si un flujo pretende consumir más de lo que justamente le corresponde de ancho de banda, solamente esta cola es afectada y no impacta el rendimiento de las otras colas que comparten el puerto de salida.

Este método tiene algunas limitantes, como:

- Los fabricantes solamente implementan FQ en sus equipos mediante software y no sobre hardware, lo cual afecta en el rendimiento limitando su aplicación en interfaces de baja velocidad en los extremos de las redes.
- El objetivo de FQ es asignar la misma cantidad de ancho de banda a cada flujo en un periodo de tiempo. Este mecanismo no está diseñado para soportar flujos con distintos requerimientos de recursos.
- El mecanismo FQ proporciona cantidades de recursos iguales a cada flujo de datos sí y solo sí todos los paquetes de todas las colas son del mismo tamaño; esto es, si existe un flujo cuya mayoría de los paquetes son de gran tamaño, este obtiene un tiempo mas grande en la repartición del ancho de banda del puerto de salida con respecto a flujos que contienen predominantemente paquetes pequeños.
- FQ es sensitivo al orden de llegada de los paquetes, esto es, si un paquete llega a una cola vacía inmediatamente después de que la cola ha sido visitada en el orden del round-robin, el paquete tendrá que esperar en la cola hasta que las demás colas sean atendidas.

- El Encolamiento Justo no proporciona mecanismos que permitan un fácil soporte a aplicaciones en tiempo real como Voz y Video por IP.

4.2.4 Encolamiento Justo Ponderado (WFQ).

El Encolamiento Justo Ponderado (WFQ, Weighted Fair Queuing) resuelve algunas de las limitantes de FQ ya que soporta flujos de datos con diferentes requerimientos de ancho de banda dando a cada cola una ponderación que le asigna diferente porcentaje del ancho de banda del puerto de salida. Además, WFQ soporta paquetes de longitud variable ya que los flujos con paquetes largos no se les asignan mayor ancho de banda que los flujos de paquetes pequeños.

El mecanismo WFQ aplica prioridades o ponderaciones para identificar el tráfico y clasificarlo para determinar cuanto ancho de banda le corresponde. Típicamente cada prioridad tiene su propia cola; las ponderaciones son asignadas a cada cola para determinar la asignación del ancho de banda. Cada que un nuevo paquete es encolado se le calcula un tiempo de finalización virtual o el tiempo de espera para desencolar o transmitir ese paquete (figura 25). El tiempo de finalización virtual F(k,i) para el k-ésimo paquete en una sesión i (asumiendo que el paquete va a una sola y única cola i) es [37]:

$$F(k,i) = max(F(k-1,i), V(a(k,i))) + L(k,i)/ponderación(i)$$

Donde:

F(0,i) = 0

a(k,i) = Tiempo de llegada

L(k,i) = Longitud del paquete

Ponderación(i) = Ponderación de la cola *i*

V(t) = Función de tiempo virtual, llamada Número Aleatorio

La progresión del tiempo virtual esta definida por:

$$dV(t)/dt = 1 / \sum \text{ponderaciones}(i)$$

Donde la sumatoria es tomada de todas las sesiones activas en la interfaz.

En otras palabras, la ecuación anterior determina que el tiempo de finalización virtual se mueve más rápido cuando hay pocas sesiones activas que cuando hay muchas sesiones activas. El mecanismo WFQ comparte el ancho de banda de salida encontrando la cola con el menor tiempo de finalización virtual y seleccionándola para el desenconamiento.

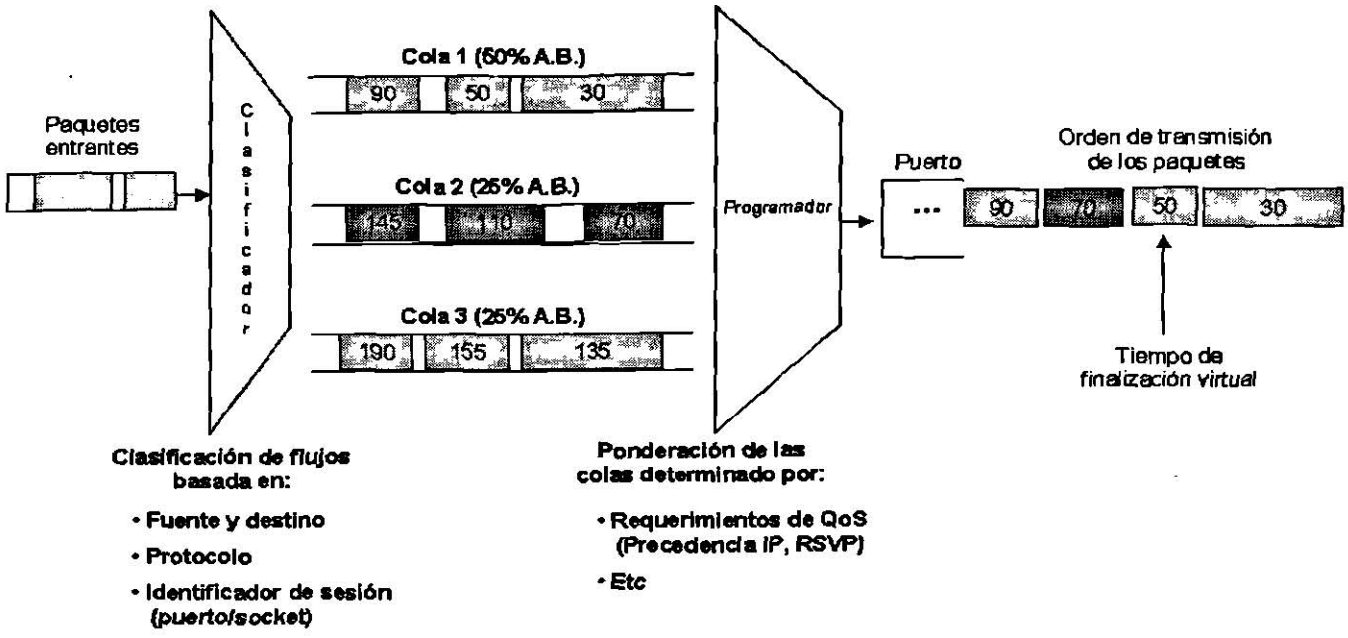


Figura 25. Mecanismo de encolamiento WFQ.

En el mecanismo WFQ los flujos con comportamientos extraños (aquellos que envían numerosos y/o enormes paquetes sin ningún control de congestión) no afectan el rendimiento de otras sesiones debido a que su tiempo virtual de finalización crecerá; por lo tanto, estos flujos no serán transmitidos hasta que se transmita el último flujo con tiempo de finalización virtual menor, por lo que esto es una forma de castigo a este tipo de flujos, debido a que WFQ siempre selecciona la cola con menor tiempo de finalización virtual para la transmisión. Sin embargo, esto provocará un gran retraso para este tipo de colas.

Cabe mencionar que la implementación interna del algoritmo de WFQ en los equipos de comunicaciones depende del fabricante del equipo, pero la esencia se mantiene.

Este mecanismo tiene dos ventajas principales:

- WFQ proporciona protección a cada clase de servicio asegurándole un nivel mínimo del ancho de banda del puerto de salida independientemente del comportamiento de las otras clases de tráfico.
- Cuando se combina con condicionamiento de tráfico en los extremos de la red, WFQ garantiza una justa repartición ponderada del ancho de banda del puerto de salida para cada clase de servicio con un retraso definido.

Y por supuesto, tiene algunas limitantes:

- Las implementaciones de WFQ por parte de los fabricantes son basadas en software, lo cual limita la aplicación en interfaces de baja velocidad en los extremos de las redes.
- WFQ implementa un complejo algoritmo el cual requiere el mantenimiento de una gran cantidad de estados de clases por servicio y un rastreo interactivo para cada paquete que llega y sale.
- La complejidad computacional impacta la escalabilidad de WFQ cuando se pretende soportar un gran número de clases de servicio sobre interfaces de alta velocidad.

4.2.5 Otros Mecanismos de Encolamiento.

Existen otros mecanismos de encolamiento mas complejos como puede ser el Encolamiento Round Robin Ponderado (WRR, Weighted Round Robin) o también conocido como Encolamiento Basado en Clases (CBQ, Class-based Queuing), el Encolamiento Round Robin de Déficit Ponderado (DWRR, Deficit Weighted Round Robin). Algunos otros como el Encolamiento Customizado (CQ, Custom Queuing) son implementaciones propietarias de los fabricantes (en este caso Cisco), por solo mencionar algunos. El estudio de estos mecanismos está fuera del alcance de este trabajo pero puede consultar [36], [38] para mayor referencia.

4.3 Uso de los Campos de QoS en los Mecanismos de Encolamiento

Ya vimos que la mayoría de los mecanismos de encolamiento utilizan un mecanismo de clasificación de paquetes para determinar la cola a la cual enviarán un determinado paquete, pero, ¿en que se basan para hacer esta clasificación?

Primordialmente se basan en el valor de QoS contenido en el campo, esto puede variar según la capa del modelo de comunicaciones a la que pertenezca el dispositivo en cuestión, ya que si es de capa dos (del modelo OSI) se basará en el campo 802.1p (Vea capítulo 3.4.1.2), o bien, si es un equipo de ruteo puede basarse en la Precedencia IP (Vea capítulo 3.4.1.1), o en el valor del Tipo de Servicio (Vea capítulo 3.4.2.1), etc. Cada mecanismo de

encolamiento puede utilizar los valores de QoS en su algoritmo para tomar decisiones en la asignación de recursos.

Lo interesante es realizar un mapeo de la Prioridad del Usuario a la Clase de Tráfico. Entendamos por Prioridad del Usuario el valor contenido en los campos mencionados en el párrafo anterior, y Clase de Tráfico a la cola de espera desde donde va a transmitirse el paquete. Este mapeo dependerá principalmente de las características del equipo de comunicaciones, ya que cada fabricante implementa sus propias reglas para realizar estas configuraciones en sus equipos por lo que no hay un estándar definido para esto.

La figura 26 muestra de manera general un diagrama de un mapeo de prioridades de usuario a su clase de tráfico correspondiente, independientemente del mecanismo de encolamiento a utilizar.

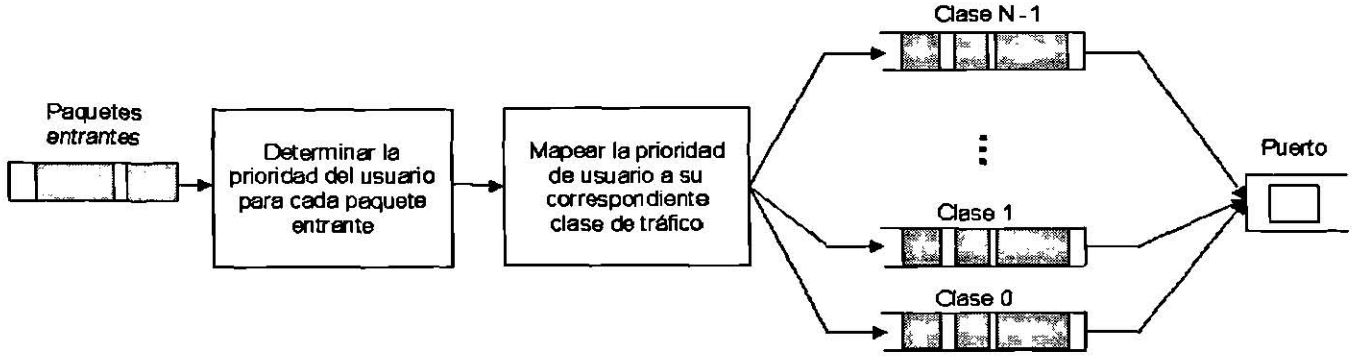


Figura 26. Mapeo de prioridades a la clase de tráfico.

En el caso del estándar 802.1p la IEEE recomienda realizar el mapeo de prioridades según la tabla V, aunque esta solo es una recomendación y el

usuario puede realizar la configuración como a él más le convenga. Esta tabla muestra como realizar el mapeo según el número de colas que tengamos en nuestro puerto. La primera fila en la tabla muestra la asignación cuando solo hay una cola y todas las clases de tráfico tienen que ser asignadas a esta cola. Si hay dos colas la IEEE recomienda asignar el tráfico de control de red, voz, video y carga controlada a la cola de mayor prioridad, y el demás trafico a la cola de menor prioridad, y así sucesivamente según incremente el número de colas del dispositivo.

TABLA V

RECOMENDACIÓN DE LA IEEE PARA EL MAPEO DE PRIORIDADES EN 802.1p.

Número de colas	Tipos de Tráfico							
1	**ME** (EE, SP, VO, CC, VI, CR)							
2	**ME** (EE,SP)				**VO** (CC, VI, CR)			
3	**ME** (EE, SP)				**CC** (VI)		**VO** (CR)	
4	**SP**		**ME** (EE)		**CC** (VI)		**VO** (CR)	
5	**SP**		**ME** (EE)		**CC**	**VI**	**VO** (CR)	
6	**SP**		**ME**	**EE**	**CC**	**VI**	**VO** (CR)	
7	**SP**		**ME**	**EE**	**CC**	**VI**	**VO**	**CR**
8	**SP**	–	**ME**	**EE**	**CC**	**VI**	**VO**	**CR**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	Prioridad del usuario							

SP = Segundo Plano
ME = Mejor Esfuerzo
EE = Excelente Esfuerzo
CC = Carga Controlada

VI = Video (<100 ms de retraso y jitter)
VO = Voz (<10 ms de retraso y jitter)
CR = Control de Red

Nota: En cada registro, la letra negrita es el tipo de tráfico al que esta orientado la asignación del tipo de clase.

Estos mapeos son de suma importancia, ya que es una base de datos que utilizará el equipo de comunicaciones al momento de tomar decisiones en la transmisión con respecto a los paquetes entrantes. El realizar mapeos nos permite también lograr que la prioridad del usuario pueda mantenerse aún y cuando los datos fluyan a través de diferentes capas del modelo de comunicaciones.

4.4 Interrelación de los Parámetros de Medición de Calidad de Servicio (Conclusiones).

Como se indicó en la sección 4.1.3 la pérdida de paquetes es provocada por dos razones: la primera, el campo de suma de comprobación es incorrecto debido a fallas en el medio de transmisión; la segunda, y mas importante para nosotros en este trabajo, es la perdida de paquetes provocada por congestiones en los equipos de comunicaciones por el desbordamiento de las colas en los puertos.

La variación en el retraso es provocada porque muchas veces existen tiempos de espera variables en las colas de transmisión para diferentes paquetes pertenecientes a un mismo flujo, lo que provoca que al llegar al receptor la diferencia de los tiempos de llegada en los paquetes sea diferente, provocando el jitter; otra causa es cuando existen ráfagas de tráfico que provocan saturación en los recursos disponibles en las memorias de almacenamiento temporal.

El caudal eficaz, como indica la fórmula de la sección 4.1.4, es inversamente proporcional al retraso, por lo que a mayor retraso de los paquetes el caudal eficaz disminuye y viceversa.

Entonces vemos que el retraso juega un papel de suma importancia ya que su comportamiento afecta a los demás parámetros. El retraso de extremo a extremo es causado por diversos factores:

- El retraso provocado en las computadoras, que es el tiempo que tardan los datos en ir desde la aplicación del usuario, hasta entrar al medio de transmisión, y se debe a:
  - Las aplicaciones
  - Los protocolos
  - La velocidad de transmisión
  - Etc.

  Aquí están involucrados todos los elementos de la computadora del emisor y del receptor, tales como el diseño de la aplicación, el sistema operativo utilizado, la velocidad del procesador, cantidad de memoria, velocidad de la tarjeta de red, etcétera, los cuales en suma forman este retraso, y todos en mayor o menor medida afectan el retraso final.

- El retraso provocado por el enlace físico de comunicaciones, el cual es llamado retraso de propagación, y se refiere a la cantidad de

tiempo que tarda en viajar la señal a través del medio. Este tiempo esta definido por la siguiente fórmula:

$$R_p = \frac{d}{v}$$

Donde:
R_p = Retraso de propagación
d = Distancia del canal de comunicaciones (en metros)
v = Velocidad de la luz en el medio

Este retraso se ve afectado por el tipo de medio utilizado, ya sea cobre, fibra óptica, o bien, el aire.

- Y por último, el retraso provocado por los equipos de comunicaciones a través de los cuales atraviesan nuestros datos, debido a:
  - Tiempo de señalización
  - Tiempo de procesamiento
  - Tiempo en colas
  - Tiempo de paquetización
  - Tiempo de reenvío
  - Protocolos utilizados
  - Etc.

Este retraso se ve afectado por el poder de procesamiento y almacenamiento de los equipos de comunicaciones, ya que estos tiempos son debido a que el equipo tiene que procesar cada paquete que entra y que tiene que reenviar. Otro factor que determina el tiempo de serialización (tiempo de inserción) -el tiempo que tarda el equipo de comunicaciones en poner los datos

en el medio de transmisión-es la velocidad de transmisión de los puertos de comunicaciones.

Medir cada uno de los tiempos que provocan el retraso final extremo a extremo puede llegar a ser demasiado complicado debido a la gran cantidad de variables involucradas en todo el sistema (figura 27), por lo que en este trabajo mediremos solamente el retraso total de extremo a extremo.

Figura 27. Modelo de red extremo a extremo.

Si nos basamos en la figura 27 podemos decir que en general, el tiempo total que tardan los datos en ir de la aplicación del usuario emisor a la aplicación del usuario destino sería la suma de todos los tiempos que tardan los datos en atravesar toda la ruta de comunicaciones, lo cual podemos representar en la siguiente ecuación:

$$R_{extremo\text{-}a\text{-}extremo} = \sum_{i=0}^{j} r_i$$

Donde:

r_i = El retraso en cada elemento de la ruta

j = Cantidad de elementos en la ruta de comunicaciones.

CAPITULO 5

PRUEBAS EXPERIMENTALES PARA LA MEDICION DE PARAMETROS QUE DETERMINAN LA CALIDAD DE SERVICIO

El objetivo de este capítulo es mostrar cómo a través de una serie de experimentos en condiciones de laboratorio podemos medir parámetros útiles que nos ayudarán a determinar el comportamiento real de nuestras redes de computadores, y saber si realmente es justificable la implementación de mecanismos de Calidad de Servicio.

Se presentan tres experimentos diferentes, cada uno con objetivos y metodologías diferentes, pero buscando el mismo objetivo: Medir el comportamiento de nuestras redes.

5.1 Experimento A: Medición del Retraso.

Este experimento tiene como objetivo demostrar que mediante el uso de diferentes mecanismos de encolamiento se puede afectar el tiempo de respuesta entre dos entidades.

Para esto, nos basamos en el planteamiento de la sección 4.4, en la que mencionábamos que el tiempo total empleado para que los datos fluyan de la aplicación del usuario emisor a la aplicación del usuario destino sería la suma

de todos los tiempos que tardan los datos en atravesar toda la ruta de comunicaciones, por lo que si disminuimos el tiempo que tardan los paquetes en cola, entonces el tiempo total disminuye.

Los datos recabados fueron los tiempos de respuesta (retraso) entre dos computadores, para después utilizar diseño factorial para demostrar la hipótesis mediante Análisis de Varianza.

5.1.1 Descripción del Experimento A.

El objetivo del experimento es demostrar que utilizando diferentes mecanismos de encolamiento se puede afectar el tiempo que tardan los paquetes en viajar de una entidad a otra. Para lograr este objetivo, esta prueba se diseño de tal manera que se tomaran en cuenta la mayoría de los factores que afectan al tiempo de respuesta.

En este experimento se midió el tiempo de respuesta entre dos computadores bajo diferentes ambientes, para esto se identificaron tres factores primordiales que afectan el resultado, por lo que se realizó una prueba para cada una de las combinaciones de los factores.

5.1.1.1 Factores que Intervinieron en el Experimento.

Los factores que intervinieron en este experimento son: Método de encolamiento, tamaño de paquete y carga de tráfico. Cada uno de ellos tiene diferentes niveles, los cuales se resumen en la tabla VI.

TABLA VI

FACTORES ESTUDIADOS EN EL EXPERIMENTO A

Factores		
Factor A Método de encolamiento	Factor B Tamaño de Paquete	Factor C Carga de tráfico
Nivel 1.- Primero en Entrar Primero en Salir (FIFO) Nivel 2.- Encolamiento Priorizado (PQ) Nivel 3.- Encolamiento Justo Ponderado (WFQ)	Nivel 1.- 64 Bytes Nivel 2.- 128 Bytes Nivel 3.- 256 Bytes Nivel 4.- 512 Bytes Nivel 5.- 1024 Bytes Nivel 6.- 1280 Bytes Nivel 7.- 1518 Bytes	Nivel 1.- 0 Mbps (0%) Nivel 2.- 2.5 Mbps (25%) Nivel 3.- 5 Mbps (50%) Nivel 4.- 7.5 Mbps (75%) Nivel 5.- 10 Mbps (100%)

El factor método de encolamiento se refiere al algoritmo de encolamiento que se utilizó en el enrutador. Este es el factor principal para nuestro estudio, ya que lo que se quiere demostrar es la variación del tiempo de respuesta cuando varía el método de encolamiento. Los tres algoritmos de encolamiento utilizados ya fueron descritos en la sección 4.2.

El factor tamaño de paquete se refiere al tamaño (en bytes) de los paquetes que se enviaron para medir el tiempo de respuesta. Este factor consta de siete niveles, cada uno con tamaño de paquete distinto. El tamaño máximo utilizado fue de 1518 bytes que corresponde al tamaño máximo de la trama de Ethernet; si se emplearan tamaños de paquetes mas grandes estos tendrían que ser fragmentados lo cual afecta el tiempo total, por lo que con esta distribución de tamaños de paquetes utilizada se evitó este fenómeno. Esta distribución de tamaños de paquete está basada en las recomendaciones hechas en el RFC 2544 [39].

El tercer factor, carga de tráfico, se refiere a la cantidad de tráfico existente en el modelo de pruebas debido a tráfico proveniente de otras entidades distintas a nuestras entidades de prueba. Este factor, al controlarlo de manera sistemática satura los enlaces de comunicaciones a diferentes cargas según el nivel del factor, con la finalidad de crear un cuello de botella para que haya congestionamientos y lograr que el modelo se asemeje a la realidad.

5.1.1.2 Diagrama del Experimento.

Para realizar este experimento se utilizó un modelo de conectividad aislado utilizando cuatro computadores, dos conmutadores (switches) y un enrutador interconectados como se muestra en la figura 28.

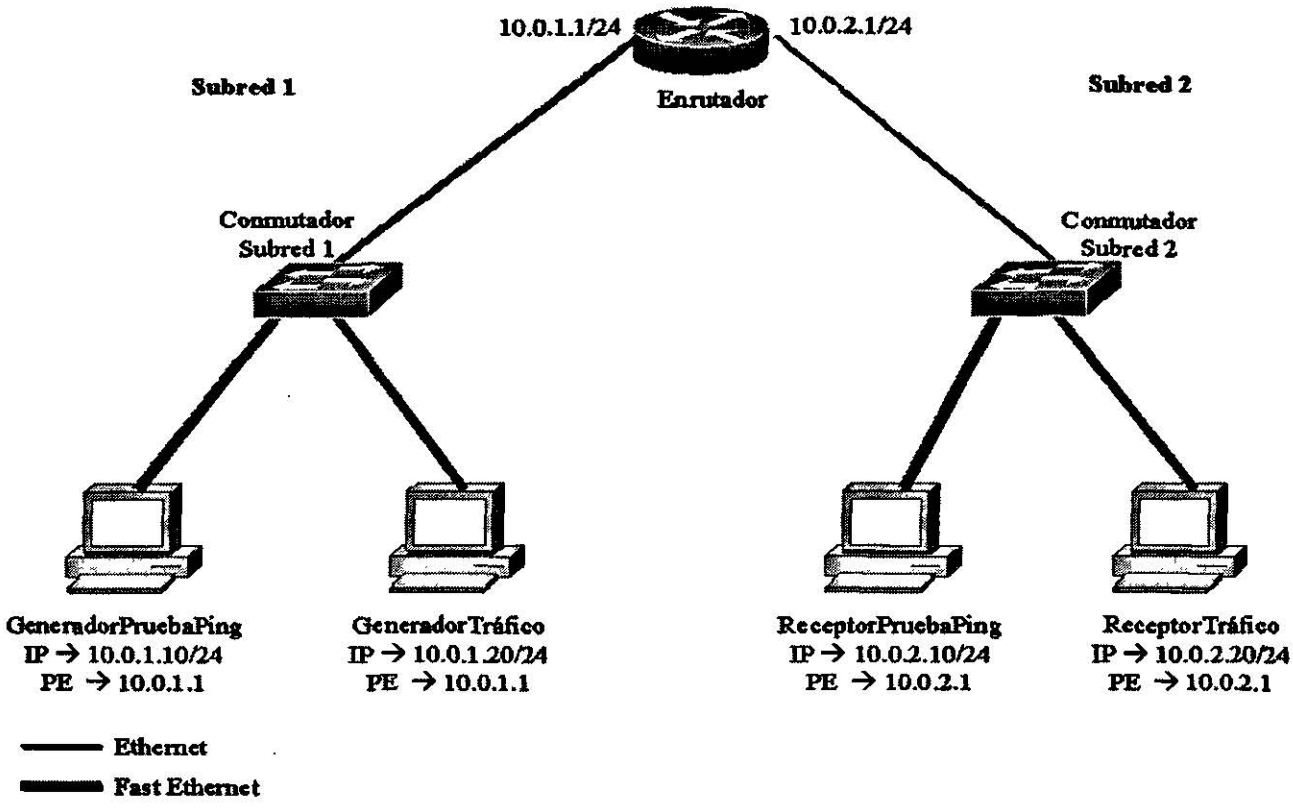


Figura 28. Esquema de conectividad del experimento A.

Se crearon dos subredes con el objetivo de estudiar un comportamiento cercano al ambiente real. Los computadores de cada subred se conectan a su conmutador de acceso a través de un enlace Fast Ethernet (100Mpbs), mientras que la conectividad de los conmutadores al enrutador es a través de un enlace Ethernet (10Mbps), esto con la finalidad de crear el cuello de botella.

Cada elemento de este modelo cumple tareas específicas, en cada subred hay la misma cantidad de elementos y ambas subredes se comunican a través del enrutador.

El enrutador realizó las funciones para interconectar ambas subredes, además de que es este equipo donde se implementaron los diferentes algoritmos de encolamiento descritos en el Factor A.

Los conmutadores simplemente brindaron la conectividad a los computadores para que estos puedan intercambiar datos entre ellos. Estos conmutadores no tienen ninguna configuración adicional a la proveniente de fábrica, por lo que su único rol en este modelo es la conectividad.

Los cuatro computadores del modelo cuentan con el sistema operativo Linux RedHat 8.0 (Kernel 2.4.18), y cada uno de ellos realiza actividades específicas. En la subred 1 hay dos computadores, los cuales llamaremos Generador de Pruebas Ping y Generador de Tráfico, mientras que en la subred 2 les llamaremos a los computadores Receptor de Pruebas Ping y Receptor de Tráfico.

El Generador de Pruebas Ping es el encargado de enviar los paquetes de prueba al Receptor de Pruebas Ping, y se medirá el tiempo que tarda el paquete en ir de Generador de Pruebas Ping a Receptor de Pruebas Ping y regresar a Generador de Pruebas Ping. Es en el Generador de Pruebas Ping donde se modificó el tamaño del paquete de acuerdo a los diferentes niveles del Factor B.

El Generador de Tráfico se encarga de generar flujos constantes de tráfico hacia el Receptor de Tráfico, con la finalidad de crear una carga de tráfico en nuestro modelo simulando un ambiente de uso real. Es en el

Generador de Tráfico donde se modificó el tamaño del flujo de paquetes de carga de acuerdo a los diferentes niveles del Factor C.

5.1.1.3 Procedimiento.

Este experimento consistió en la realización de 105 pruebas, abarcando todas las combinaciones de los niveles de los tres factores. Para cada prueba se realizaron 30 réplicas, lo cual es el mínimo necesario de pruebas para garantizar un buen resultado del Análisis de Varianza. Estas pruebas fueron realizadas en orden aleatorio para reducir el error experimental [40].

Se utilizó la herramienta PING (Packet InterNet Groper, Buscador entre redes de paquetes) para obtener el tiempo de respuesta entre el computador Generador de Pruebas Ping y el Receptor de Pruebas Ping. Para mayor referencia sobre la herramienta puede consultar el RFC 1739 [41].

En las pruebas que era necesario generar tráfico se utilizó la herramienta Traffic Generator versión 2, con la cual se crearon flujos constantes de tráfico UDP [42].

Se utilizaron flujos constantes de tráfico UDP ya que, aunque el comportamiento real de carga de tráfico de una red es con tráfico a ráfagas este no puede ser controlado de manera adecuada para motivos experimentales. Se eligió tráfico UDP debido a que el tráfico TCP al momento de existir perdidas de paquetes éste retransmite los paquetes perdidos, lo cual no era necesario para nuestro experimento porque lo que nos interesaba era que en determinado instante de tiempo hubiera una determinada carga de tráfico en la red, y no

precisamente que se transmitieran los paquetes de carga sin importar el tiempo que tardasen.

Era necesario medir también los paquetes perdidos por el generador de tráfico, ya que el computador Generador de Tráfico enviaba los paquetes a Receptor de Tráfico pero había la posibilidad de que estos fueran desechados debido a congestionamiento de la red. Para esto se utilizó la herramienta IPTraf versión 2.7.0 [43].

Para cada prueba se guardaron los tiempos de respuesta de cada una de las 30 replicas, así como los paquetes perdidos por el generador de tráfico para su posterior análisis.

Las configuraciones de los equipos de comunicaciones, así como los scripts de las herramientas utilizadas en este experimento pueden ser vistas a detalle en el Apéndice A.

5.1.1.4 Factores No Controlados.

Las configuraciones de los equipos de comunicaciones fueron únicamente en el enrutador, y el acceso de los computadores al conmutador no se controló, por lo que el método de acceso al medio fue un factor no controlado.

Debido a que se utilizó Ethernet como tecnología de red local, existen colisiones al momento en que diferentes computadores tratan de acceder al medio, por lo que puede existir la posibilidad de que algunos de los paquetes

perdidos no fueron debido a congestionamiento en los enlaces del enrutador a los conmutadores, si no ocasionado por colisiones en el conmutador debido a la solicitud de acceso al medio de los dos computadores. Este factor, como ya se mencionó, no estuvo controlado en este experimento.

5.1.2 Resultados del Experimento A.

Como ya se había mencionado, el análisis de los datos de este experimento se llevó a cabo realizando un análisis de varianza, los cuales se muestran a detalle en el Apéndice B.

El resultado que más nos interesa, ya que es el que responde al planteamiento original de este experimento, es saber si al variar el mecanismo de encolamiento el tiempo de respuesta disminuye.

Basándonos en los resultados del análisis de varianza podemos concluir con un 95% de certeza que efectivamente, al modificar el mecanismo de encolamiento en los equipos de comunicaciones el tiempo de respuesta entre el computador Generador de Pruebas Ping y el Receptor de pruebas Ping sí varía.

Todos los factores analizados en este experimento, así como la interacción de estos factores afectan significativamente el tiempo de respuesta entre los computadores; a excepción de la combinación de los factores A y B, la cual si bien no es significativa, estadísticamente tampoco puede decirse que no afecta. Aunque esto suena a un juego de palabras, quiere decir que cuando se varía el mecanismo de encolamiento y el tamaño del paquete, el tiempo de

respuesta no se ve afectado significativamente, esto no quiere decir que el tiempo no varía, simplemente que estadísticamente no es posible afirmar que debido a la variación de los factores mencionados el tiempo de respuesta se ve afectado (ver Apéndice B).

El afirmar que al modificar la carga de tráfico en la red el tiempo de respuesta se ve afectado puede resultar lógico, sin embargo los datos recabados en este experimento pueden servirnos para visualizar de manera cuantitativa como afectan realmente estas modificaciones.

Las tablas VII, IX, XI, XIII y XV muestran los tiempos de respuesta promedio obtenidos de las muestras de datos recabadas del experimento, así como el intervalo de confianza para todas las combinaciones de los factores involucrados en este experimento. Las tablas VIII, X, XII, XIV y XVI muestran el porcentaje de paquetes perdidos por el generador de tráfico para cada una de las combinaciones de los factores.

Para las 3150 réplicas de este experimento, solamente hubo 11 paquetes perdidos por el ping, los cuales no se tomaron en cuenta para realizar los cálculos aquí mostrados, sin embargo sí se menciona bajo que circunstancias ocurrió este fenómeno.

Como se puede apreciar en la tabla VII, los tiempos promedio de respuesta al igual que los intervalos de confianza son semejantes entre los diferentes mecanismos de encolamiento para todos los tamaños de paquete. Esto quiere decir que es irrelevante el utilizar diferentes mecanismos de

encolamiento cuando no hay carga de tráfico en la red. De las 630 réplicas de esta sección de pruebas, hubo 3 paquetes perdidos por el ping al utilizar el mecanismo de encolamiento WFQ con paquetes de 1024 bytes.

TABLA VII

TIEMPOS DE RESPUESTA PROMEDIO E INTERVALOS DE CONFIANZA PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 0 MBPS.

		Mecanismo de encolamiento					
		FIFO		PQ		WFQ	
		Tiempo de respuesta promedio (*milisegundos*)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (*milisegundos*)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (*milisegundos*)	Intervalo de Confianza (95%)
Tamaño paq. Bytes	64	0.529800	0.000879	0.532567	0.001869	0.528033	0.001213
	128	0.781133	0.002358	0.749400	0.001202	0.751300	0.002302
	256	1.202667	0.001864	1.201000	0.002174	1.203667	0.002737
	512	2.154000	0.002224	2.135667	0.002605	2.122667	0.002809
	1024	3.951000	0.001720	3.989667	0.001754	4.269259	0.098073
	1280	4.867667	0.002034	4.856667	0.002713	4.865000	0.002441
	1518	5.717333	0.002289	5.720333	0.002393	5.720333	0.002894

TABLA VIII

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 0 MBPS.

		Mecanismo de encolamiento		
		FIFO	PQ	WFQ
		% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico
Tamaño Paq. Bytes	64	0	0	0
	128	0	0	0
	256	0	0	0
	512	0	0	0
	1024	0	0	0
	1280	0	0	0
	1518	0	0	0

Como se muestra en la tabla IX, al haber una carga de tráfico del 25% en la red los tiempos de respuesta entre los diferentes mecanismos de encolamiento son semejantes, sin embargo podemos apreciar con los intervalos de confianza que existe menor dispersión entre los tiempos de respuesta cuando se utiliza PQ y WFQ, sobre todo con los paquetes de mayor tamaño. Como se muestra en la tabla X, bajo estas condiciones de tráfico no hubo paquetes perdidos por el generador de tráfico.

En estas pruebas hubo 3 paquetes perdidos en el ping, los cuales fueron al utilizaron FIFO con paquetes de 1024 bytes. Esta perdida de paquetes es despreciable, ya que sólo son 3 paquetes de 630 replicas.

TABLA IX

TIEMPOS DE RESPUESTA PROMEDIO E INTERVALOS DE CONFIANZA PARA LAS *DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO* Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 2.5 MBPS.

		Mecanismo de encolamiento					
		FIFO		PQ		WFQ	
		Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)
Tamaño paq. Bytes	64	0.585900	0.035509	1.420667	0.248187	1.007133	0.176497
	128	1.130467	0.135276	1.013833	0.228699	0.936500	0.095980
	256	1.644333	0.217763	1.728000	0.178485	1.728333	0.237321
	512	2.285333	0.122669	2.347333	0.165518	2.190333	0.050724
	1024	4.048519	0.101812	3.995667	0.048827	4.176667	0.103333
	1280	4.986667	0.122696	4.988667	0.091592	4.905667	0.069405
	1518	5.943000	0.116701	5.752333	0.049625	5.947667	0.109408

TABLA X

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 2.5 MBPS.

		Mecanismo de encolamiento		
		FIFO	PQ	WFQ
		% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico
Tamaño Paq. Bytes	64	0	0	0
	128	0	0	0
	256	0	0	0
	512	0	0	0
	1024	0	0	0
	1280	0	0	0
	1518	0	0	0

Al haber una carga de tráfico del 50%, vemos (tabla XI) que para paquetes de tamaño grande los tiempos de respuesta son mejores con el mecanismo de encolamiento PQ que con WFQ o FIFO, lo mismo sucede con la dispersión, la cual es menor al utilizar PQ.

Bajo estas condiciones de tráfico en la red, vemos que el generador de tráfico comenzó a tener mínimas pérdidas de paquetes (tabla XII) cuando se utilizaron los mecanismos de encolamiento PQ y WFQ.

TABLA XI

TIEMPOS DE RESPUESTA PROMEDIO E INTERVALOS DE CONFIANZA PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 5 MBPS.

		Mecanismo de encolamiento					
		FIFO		PQ		WFQ	
		Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)
Tamaño paq. Bytes	64	1.450567	0.258407	1.656967	0.345121	1.361067	0.266674
	128	1.584000	0.278780	1.500800	0.270172	4.705633	2.129675
	256	1.770667	0.225474	2.082333	0.283420	4.677667	1.856726
	512	2.817000	0.252664	2.754667	0.223329	2.717000	0.282758
	1024	4.151333	0.161819	4.447000	0.210048	4.334000	0.169620
	1280	5.254000	0.237221	5.347000	0.203197	5.515333	0.246246
	1518	6.099333	0.187103	5.800333	0.067970	5.951333	0.135343

TABLA XII

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 5 MBPS.

		Mecanismo de encolamiento		
		FIFO	PQ	WFQ
		% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico
Tamaño Paq. Bytes	64	0	0	0
	128	0	0	0
	256	0	0	0
	512	0	0	0.006857
	1024	0	0	0
	1280	0	0	0.057143
	1518	0	0.006857	0.002286

Como vemos en la tabla XIII, con carga de tráfico del 75% no hubo variaciones significativas de los tiempos de respuesta entre los diferentes mecanismos de encolamiento. La tasa de paquetes perdidos por el generador de tráfico se incremento un poco, tal como se muestra en la tabla XIV.

TABLA XIII

TIEMPOS DE RESPUESTA PROMEDIO E INTERVALOS DE CONFIANZA PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 7.5 MBPS.

		Mecanismo de encolamiento					
		FIFO		PQ		WFQ	
		Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)
Tamaño paq. Bytes	64	2.236333	0.523023	2.076700	0.350673	2.487167	0.469646
	128	1.947967	0.315808	2.389333	0.339253	2.194600	0.390145
	256	2.596333	0.423949	2.580333	0.365574	2.429333	0.390439
	512	3.137667	0.313695	3.084667	0.304843	3.569667	0.288914
	1024	5.176667	0.313906	4.997000	0.234197	5.136667	0.293303
	1280	5.670333	0.269929	5.795667	0.302962	5.422000	0.223431
	1518	6.836667	0.415115	6.925333	0.312623	7.486333	0.310445

TABLA XIV

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 7.5 MBPS.

		Mecanismo de encolamiento		
		FIFO	PQ	WFQ
		% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico
Tamaño Paq. Bytes	64	0.415714	0.048571	0.535714
	128	0.088571	0.505714	0.534286
	256	0.040000	0.510000	0.234286
	512	0.255714	0.561429	0.480000
	1024	0.174286	0.771429	0.030000
	1280	0.431429	0.131429	0.321429
	1518	0.324286	0.068571	0.088571

Podemos apreciar en la tabla XV, que con una carga de tráfico en la red del 100%, en general los tiempos de respuesta fueron mas bajos utilizando el mecanismo de encolamiento PQ; se puede observar también que bajo estas condiciones de tráfico y al utilizar tamaños de paquetes grandes, el mecanismo de encolamiento FIFO es mejor bajo estas circunstancias que el mecanismo WFQ.

La tasa de paquetes perdidos por el generador de tráfico se incrementó considerablemente (tabla XVI), sin embargo no existe mucha variación de esta tasa al utilizar diferentes mecanismos de encolamiento.

De las 630 réplicas, hubo 4 paquetes perdidos por el ping al utilizar el mecanismo de encolamiento FIFO con paquetes de 1518 bytes; así mismo, al utilizar el mecanismo WFQ con paquetes de 1518 bytes, hubo una pérdida de 1 paquete del ping.

TABLA XV

TIEMPOS DE RESPUESTA PROMEDIO E INTERVALOS DE CONFIANZA PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 10 MBPS.

		Mecanismo de encolamiento					
		FIFO		PQ		WFQ	
		Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)	Tiempo de respuesta promedio (milisegundos)	Intervalo de Confianza (95%)
Tamaño paq. Bytes	64	23.521933	9.457747	17.209567	8.557614	29.147567	9.409865
	128	23.567967	8.040268	14.113100	8.405752	17.049800	8.354692
	256	23.947000	8.590271	20.292000	8.067393	13.330000	6.618897
	512	13.422667	6.672544	19.434667	7.947378	16.633667	8.080766
	1024	29.538000	8.723152	18.687333	7.828454	32.165000	9.166242
	1280	26.739000	7.982068	27.327000	9.064422	32.784333	8.737060
	1518	22.642692	7.747281	19.692667	8.128943	39.770690	9.332804

TABLA XVI

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE PAQUETE DEL EXPERIMENTO A CON CARGA DE TRÁFICO DE 10 MBPS.

		Mecanismo de encolamiento		
		FIFO	PQ	WFQ
		% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico
Tamaño Paq. Bytes	64	28.300571	28.573714	34.553143
	128	37.777143	34.310857	32.521143
	256	30.752000	33.362286	34.618286
	512	35.450286	33.859429	33.562286
	1024	24.817143	33.240000	34.922286
	1280	34.593143	32.664000	34.318857
	1518	36.372571	36.806857	34.938286

Con los datos mostrados en estas tablas, podemos determinar con un 95% de confianza que si se utiliza determinado mecanismo de encolamiento, con un cierto tamaño de paquete y con una carga de tráfico determinada, el tiempo de respuesta de los paquetes bajo un esquema de conectividad como el de la figura 28 estaría entre:

$$t_p - IC < t < t_p + IC$$

Donde:
t es el tiempo de respuesta.
t_p es el tiempo promedio obtenido de la tabla.
IC es el intervalo de confianza obtenido de la tabla.

Supongamos por ejemplo, que queremos saber el tiempo de respuesta entre dos computadores conectados bajo un esquema idéntico al de la figura 28, utilizando el método de encolamiento priorizado con un tamaño de paquete de 1024 bytes y una carga de tráfico en la red de 7.5 Mbps; con un 95% de confianza y basándonos en la tabla XIII diríamos que el tiempo estará entre los 4.7628 y los 5.2312 milisegundos, y la pérdida de paquetes para el generador de tráfico será del 0.7714% (tabla XIV).

5.2 Experimento B: Mediciones de Variación en el Retraso, Pérdida de Paquetes y Caudal Eficaz.

El experimento anterior mostraba el retardo extremo a extremo entre dos computadores, sin embargo, este tiempo se obtuvo utilizando el protocolo ICMP (Protocolo de Mensajes de Control de Internet, Internet Control Message Protocol) el cual se "cuelga" directamente del protocolo IP (figura 7), por lo que

no utiliza ningún protocolo de transporte; esto no ocurre cuando la comunicación es entre aplicaciones de usuarios, ya que estas necesitan de algún protocolo de transporte, por lo que también es necesario realizar mediciones las cuales se asemejen a una comunicación entre aplicaciones utilizando protocolos de transporte.

Este experimento tiene como objetivo mostrar el comportamiento extremo a extremo de la variación en el retraso, la pérdida de paquetes y el caudal eficaz en diferentes escenarios para comprender el comportamiento extremo a extremo de nuestra red, utilizando para ello flujos de paquetes UDP.

5.2.1 Descripción del Experimento B

En este experimento se midieron tres parámetros de suma importancia: la variación en el retraso, la pérdida de paquetes y el caudal eficaz. Estas mediciones se realizaron utilizando el protocolo de transporte UDP, ya que es el que nos permite obtener este tipo de mediciones. El uso del protocolo TCP no esta incluido en este trabajo, ya que utilizando este protocolo solamente podríamos obtener mediciones del caudal eficaz.

Al igual que en el experimento anterior, se identificaron tres factores que afectan los resultados de nuestros parámetros. Igualmente, se realizaron pruebas para cada una de las combinaciones de estos factores.

5.2.1.1 Factores que Intervinieron en el Experimento

Los factores que intervinieron en este experimento son: Método de encolamiento, tamaño de flujo de prueba y carga de tráfico. Cada uno de ellos tiene diferentes niveles, los cuales se resumen en la tabla XVII.

TABLA XVII

FACTORES ESTUDIADOS EN EL EXPERIMENTO B

Factores		
Factor A Método de encolamiento	Factor B Tamaño de Flujo de Prueba	Factor C Carga de tráfico
Nivel 1.- Primero en Entrar Primero en Salir (FIFO) Nivel 2.- Encolamiento Priorizado (PQ) Nivel 3.- Encolamiento Justo Ponderado (WFQ)	Nivel 1.- 1 Mbps Nivel 2.- 2 Mbps Nivel 3.- 3 Mbps Nivel 4.- 4 Mbps Nivel 5.- 5 Mbps Nivel 6.- 6 Mbps Nivel 7.- 7 Mbps Nivel 8.- 8 Mbps Nivel 9.- 9 Mbps Nivel 10.- 10 Mbps	Nivel 1.- 0 Mbps (0%) Nivel 2.- 2.5 Mbps (25%) Nivel 3.- 5 Mbps (50%)

El factor método de encolamiento se refiere al algoritmo de encolamiento que se utilizó en el enrutador. Los tres algoritmos de encolamiento utilizados ya fueron descritos en la sección 4.2.

El factor tamaño de flujo de prueba se refiere a la cantidad de datos enviados por las entidades de prueba para llevar a cabo las mediciones. Los niveles de este factor (medidos en mega bits por segundo) van desde 1 Mbps hasta 10 Mbps que es el valor máximo teórico del enlace de comunicaciones entre el enrutador y los conmutadores.

El tercer factor, carga de tráfico, se refiere a la cantidad de tráfico existente en el modelo de pruebas debido a tráfico proveniente de otras entidades distintas a nuestras entidades de prueba. En este experimento los niveles de este factor van desde una carga de tráfico del 0% hasta el 50% (5 Mbps). Solamente se incluyeron cargas de tráfico de hasta 5 Mbps debido a que cuando los niveles del factor Tamaño de Flujo de Prueba crecieran, la cantidad de tráfico en los enlaces sería demasiada que las mediciones no serían apropiadas.

5.2.1.2 Diagrama del Experimento.

Para realizar este experimento se utilizó un modelo de conectividad aislado utilizando cuatro computadores, dos conmutadores (switches) y un enrutador interconectados como se muestra en la figura 29.

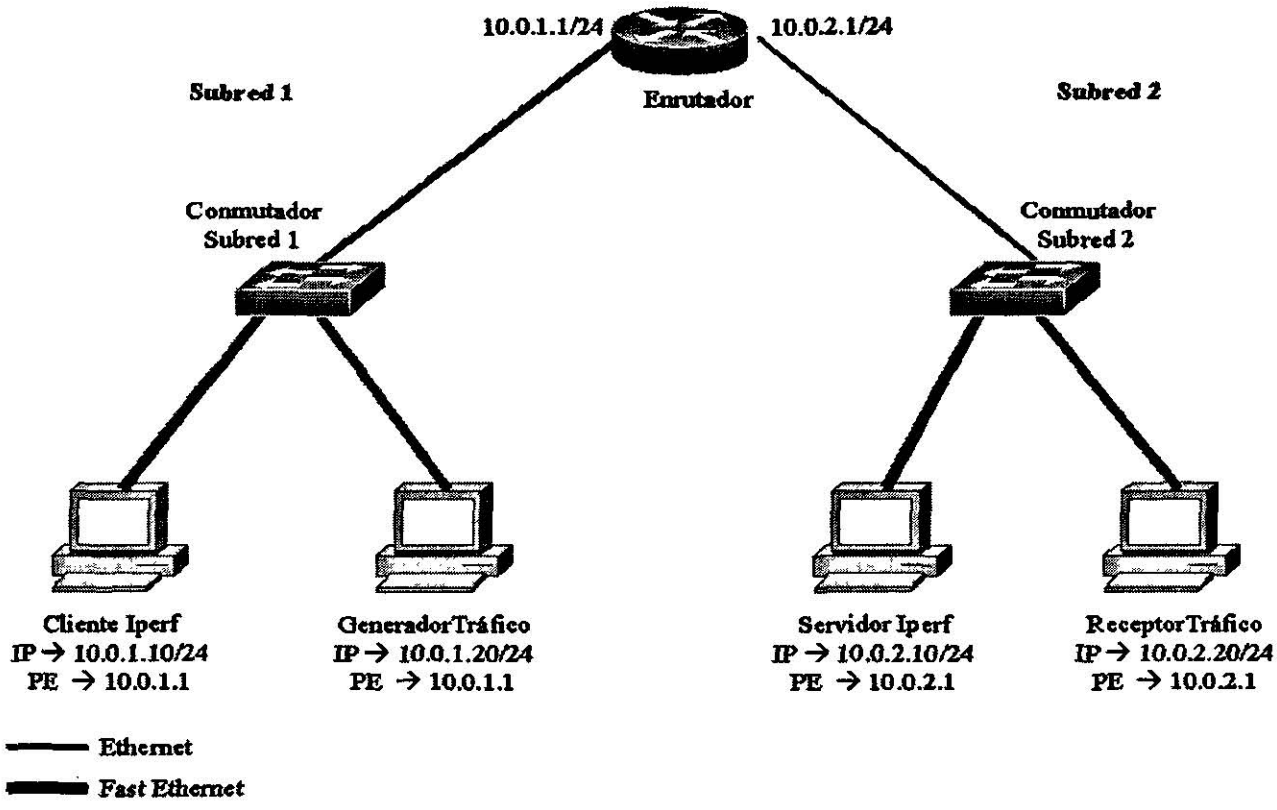


Figura 29. Esquema de conectividad del experimento B.

Al igual que en el experimento anterior se crearon dos subredes con el mismo tipo de conectividad y velocidad que el anterior. En este experimento el enrutador realizó las mismas funciones que el experimento anterior.

Los cuatro computadores del esquema cuentan con el sistema operativo Linux RedHat 8.0 (Kernel 2.4.18), dos en cada subred; los computadores de la subred 1 son el Cliente Iperf y el Generador de Tráfico, mientras que los de la subred 2 son el Servidor Iperf y el Receptor de Tráfico.

El Cliente Iperf es el encargado de generar y enviar los datos de prueba hacia el Servidor Iperf, en el cual se registran los resultados de las mediciones de los parámetros. Los computadores Generador y Receptor de Tráfico

realizaron las mismas funcionalidades que en el experimento anterior, solo que variando los niveles del factor C de acuerdo a la tabla XVII.

En este experimento, el conmutador además de ofrecer la conectividad a los computadores como en el experimento anterior, también jugó un papel importante, ya que se realizaron dos bloques de pruebas: priorizando los flujos de prueba en el conmutador, y otra sin priorizar los flujos de prueba.

Esta prioritización fue adicional a la prioritización en el enrutador, y se hizo debido a que la cantidad de tráfico enviada tanto por el Cliente Iperf como por el Generador de Tráfico en algunos casos era demasiada que el conmutador tiraba los paquetes debido a congestiones provocadas por el método de acceso al medio en Ethernet. Por eso, para entender el funcionamiento tanto cuando se priorizaran los flujos del Cliente Iperf, como sin prioritización de los flujos, se repitieron las mismas pruebas para los dos escenarios.

5.2.1.3 Procedimiento.

Este experimento se dividió en dos bloques. El primero consistió en la realización de 90 pruebas, abarcando todas las combinaciones de los niveles de los tres factores, sin realizar ninguna configuración en los conmutadores, solamente en los computadores (Factor B y C) y en el enrutador (Factor A). Es decir, estas pruebas se realizaron sin priorizar los flujos de prueba en el conmutador.

El segundo bloque de este experimento consistió en la realización de las mismas 90 pruebas, solamente que aquí sí se priorizaron los flujos de prueba en el conmutador de la subred 1. Con esto logramos que los flujos de prueba fueran tratados de manera preferencial en el conmutador de salida a diferencia del flujo de carga de tráfico.

La prioritización del conmutador de la subred 1 fue una prioritización estricta, es decir, que se asignó a la cola de mayor prioridad del conmutador los flujos de tráfico provenientes del computador Cliente Iperf, mientras que el tráfico del computador Generador de Tráfico fue asignado a la cola de menor prioridad del conmutador.

Para ambos apartados de este experimento se tomaron 30 replicas de los resultados, además de que las pruebas fueron realizadas en orden aleatorio.

Para realizar las mediciones de los parámetros de este experimento se utilizó la herramienta IPerf versión 1.6 para Linux del Laboratorio Nacional para la Investigación de Redes Aplicadas de la Fundación Nacional de Ciencia de los Estados Unidos [44].

En el computador Servidor Iperf se ejecutó la herramienta en modo servidor, mientras que el computador Cliente Iperf fue el encargado de iniciar las pruebas. La recolección de datos se realizó de los resultados mostrados por la herramienta en el computador Servidor Iperf.

En las pruebas en que era necesario generar tráfico, se utilizó al igual que en el experimento anterior la herramienta Traffic Generator versión 2.0. El uso de esta herramienta en este experimento fue de la misma manera que se utilizó en el experimento A.

Al igual que en el experimento anterior también se midieron los paquetes perdidos por el Generador de Tráfico, para lo cual se utilizó la herramienta IPTraf de la misma manera que en el experimento anterior.

Las pruebas se realizaron primeramente sin priorización en el conmutador de la subred 1, para después realizar el mismo conjunto de pruebas priorizando los flujos de paquetes de prueba en el conmutador. Para cada una de las pruebas se guardaron los resultados de los parámetros medidos, así como los paquetes perdidos por el generador de tráfico para su posterior análisis.

Las configuraciones de los equipos de comunicaciones (tanto del enrutador como del conmutador), así como los scripts de las herramientas utilizadas en este experimento pueden ser consultadas a detalle en el Apéndice A.

5.2.1.4 Factores No Controlados.

De la misma manera que en el experimento anterior, en este experimento tampoco se controlaron las colisiones en los conmutadores ocasionadas por el método de acceso al medio de Ethernet.

Aunque en uno de los apartados de las pruebas se priorizaron los flujos de paquetes de prueba en el conmutador de la subred 1, esto no garantiza que no haya colisiones a nivel de capa 2, además al conmutador de la subred 2 no se le realizó ninguna configuración adicional a la de fábrica, por lo que la posibilidad de que haya habido paquetes descartados ocasionados por los congestionamientos a nivel de capa 2 existió en ambos bloques de este experimento. Este fenómeno no fue posible detectarlo en este experimento ya que las mediciones realizadas fueron de extremo a extremo.

5.2.2 Resultados del Experimento B.

Los resultados de este experimento los mostraremos en base a tablas de resultados y figuras para una mejor comprensión. Se mostrarán los resultados tanto para las pruebas realizadas sin priorizar el conmutador, como para las pruebas realizadas con prioritización en el conmutador.

En el Apéndice C se muestran las tablas de resultados incluyendo la media muestral y los intervalos de confianza de los datos obtenidos en el experimento.

5.2.2.1 Resultados Con 0 Mbps de Carga de Tráfico en la Red.

Como podemos ver en la tabla XVIII, cuando había una carga de tráfico en la red de 0% y sin priorizar el conmutador hubo un mayor caudal eficaz utilizando el mecanismo de encolamiento PQ, así mismo, utilizando FIFO se obtuvo un mayor caudal eficaz que utilizando WFQ. Para flujos de datos

grandes se obtuvo menor variación en el retardo (jitter) utilizando PQ que FIFO y PQ; cuando se utilizó WFQ la variación en el retardo fue mayor que con FIFO. Con respecto a los paquetes perdidos, hubo menor pérdida con PQ en comparación con los demás métodos de encolamiento.

TABLA XVIII.

CAUDAL EFICAZ, VARIACIÓN DEL RETARDO Y PORCENTAJE DE PAQUETES PERDIDOS PROMEDIO, PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑO DE FLUJO DE PAQUETES DEL EXPERIMENTO B CON CARGA DE TRÁFICO DE 0 MBPS Y SIN PRIORITIZACIÓN EN EL CONMUTADOR.

		Método de encolamiento								
		FIFO			PQ			WFQ		
		Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps	Jitter (ms)	% Paquetes Perdidos
Datos enviados Mbps	1	0.996	2.241	0	1.000	0.542	0	1.000	0.696	0
	2	2.000	0.427	0	2.000	0.514	0	2.965	0.404	1.163
	3	3.000	0.426	0	3.000	0.426	0	2.918	0.412	2.672
	4	3.824	0.369	4.390	3.900	0.422	2.508	3.799	0.402	5.078
	5	4.646	0.347	7.087	4.773	0.436	4.535	4.663	0.330	6.743
	6	5.505	0.330	8.249	5.623	0.425	6.295	5.522	0.317	7.956
	7	6.346	0.454	9.341	6.440	0.530	7.415	6.349	0.472	9.284
	8	7.113	0.457	11.078	7.194	0.347	9.895	6.928	0.147	13.265
	9	7.972	0.324	11.441	8.133	0.210	9.644	7.931	0.336	11.910
	10	8.226	0.407	17.479	8.704	0.204	12.911	8.270	0.493	16.962

En la tabla XIX se muestran los resultados con carga de tráfico de 0% y priorizando el conmutador de salida. Podemos ver que el caudal eficaz es mayor utilizando PQ, sin embargo la diferencia entre los diferentes mecanismos

de encolamiento no es tan significativa como cuando no se priorizó el conmutador. Al igual que los resultados sin priorizar el conmutador, observamos que la variación en el retraso es menor utilizando el mecanismo de encolamiento PQ, y lo mismo sucede con la tasa de paquetes perdidos.

TABLA XIX.

CAUDAL EFICAZ, VARIACIÓN DEL RETARDO Y PORCENTAJE DE PAQUETES PERDIDOS PROMEDIO, PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑO DE FLUJO DE PAQUETES DEL EXPERIMENTO B CON CARGA DE TRÁFICO DE 0 MBPS PRIORIZANDO EL CONMUTADOR.

		Método de encolamiento								
		FIFO			PQ			WFQ		
		Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps	Jitter (ms)	% Paquetes Perdidos
Datos enviados Mbps	1	0.998	0.097	0	1.000	0.445	0	1.000	0.447	0
	2	2.000	0.515	0	2.000	0.528	0	2.000	0.537	0
	3	3.000	0.545	0	2.997	0.525	0	3.000	0.553	0
	4	4.000	0.524	0	3.993	0.550	0.159	3.993	0.549	0.159
	5	4.894	0.559	2.114	4.895	0.564	2.099	4.882	0.560	2.237
	6	5.791	0.699	3.464	5.794	0.700	3.412	5.794	0.700	3.412
	7	6.610	0.532	5.547	6.602	0.539	5.656	6.613	0.541	5.502
	8	7.408	0.398	7.387	7.404	0.406	7.408	7.397	0.408	7.512
	9	8.372	0.958	6.857	8.448	0.298	6.163	8.438	0.306	6.268
	10	8.911	0.917	10.835	8.977	0.938	10.221	8.990	0.916	10.099

Al observar las figuras 30, 31, 32, 33, 34 y 35, observamos que el caudal eficaz obtenido cuando se priorizó el conmutador fue mayor que el que se obtuvo sin priorizar el conmutador, independientemente del mecanismo de

encolamiento utilizado en cada caso, además el crecimiento de caudal eficaz fue mas estable al priorizar el conmutador. Podemos observar que la variación en el retardo tiene un comportamiento mas estable cuando no se priorizó el conmutador bajo estas condiciones de carga de tráfico en la red (0%). La tasa de paquetes perdidos fue mayor cuando no se priorizó el conmutador, como era de esperarse.

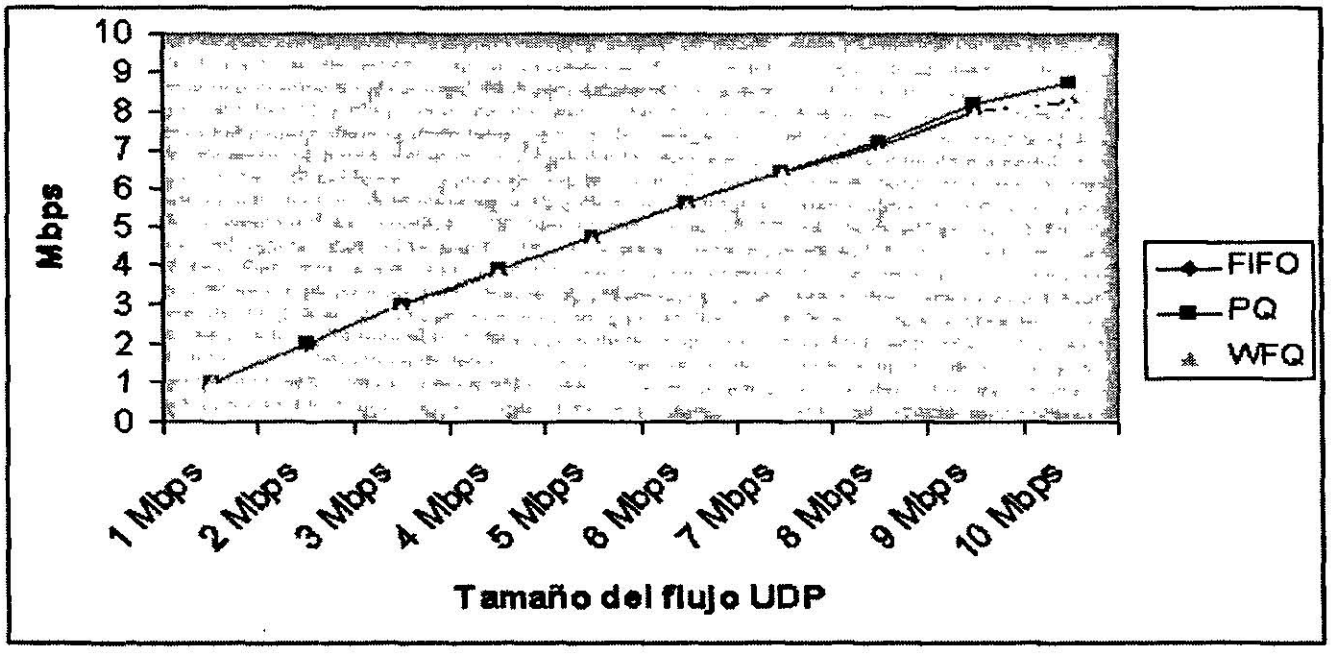


Figura 30. Comparativo del caudal eficaz entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 0 Mbps y sin prioritización en el conmutador.

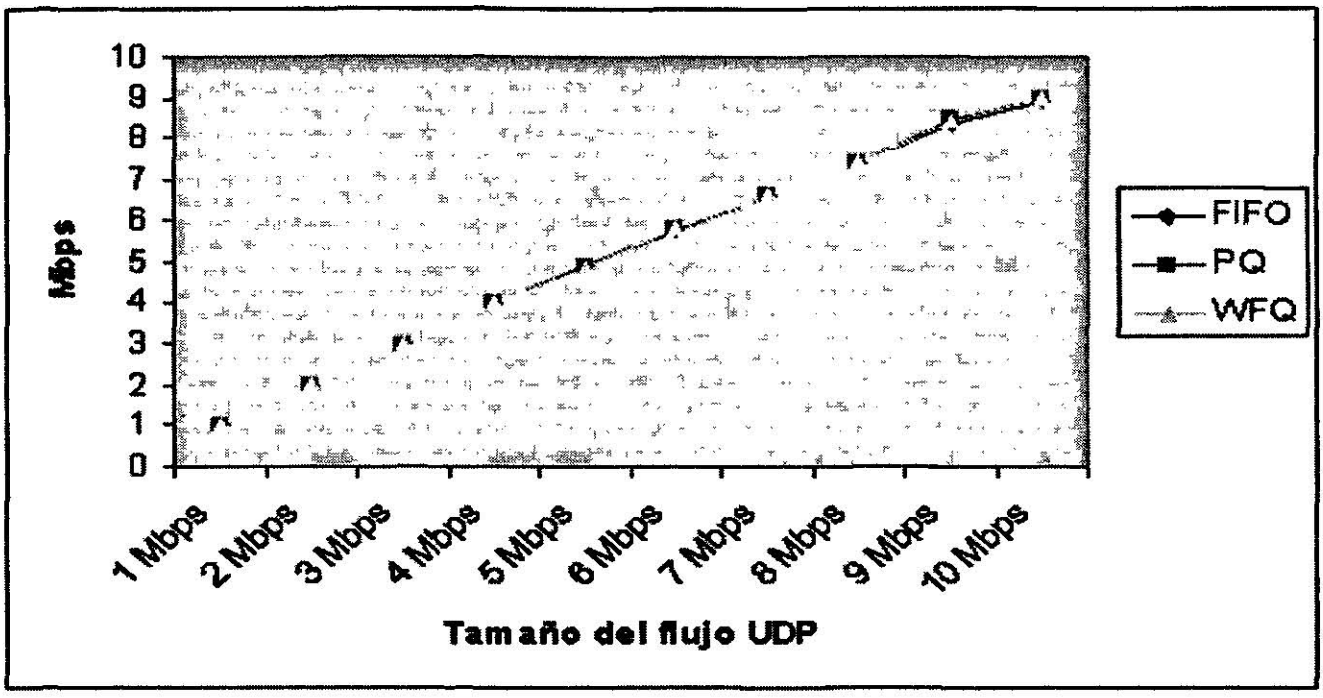


Figura 31. Comparativo del caudal eficaz entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 0 Mbps priorizando el conmutador.

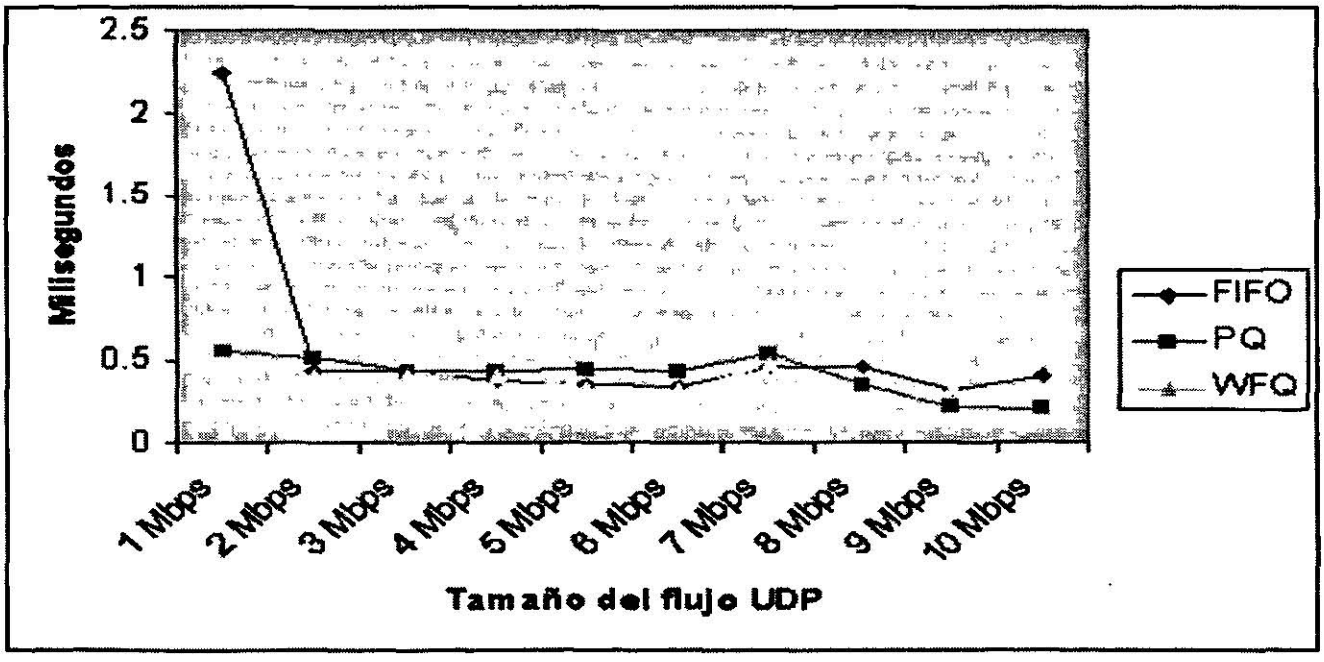


Figura 32. Comparativo de la variación del retardo entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 0 Mbps y sin prioritización en el conmutador.

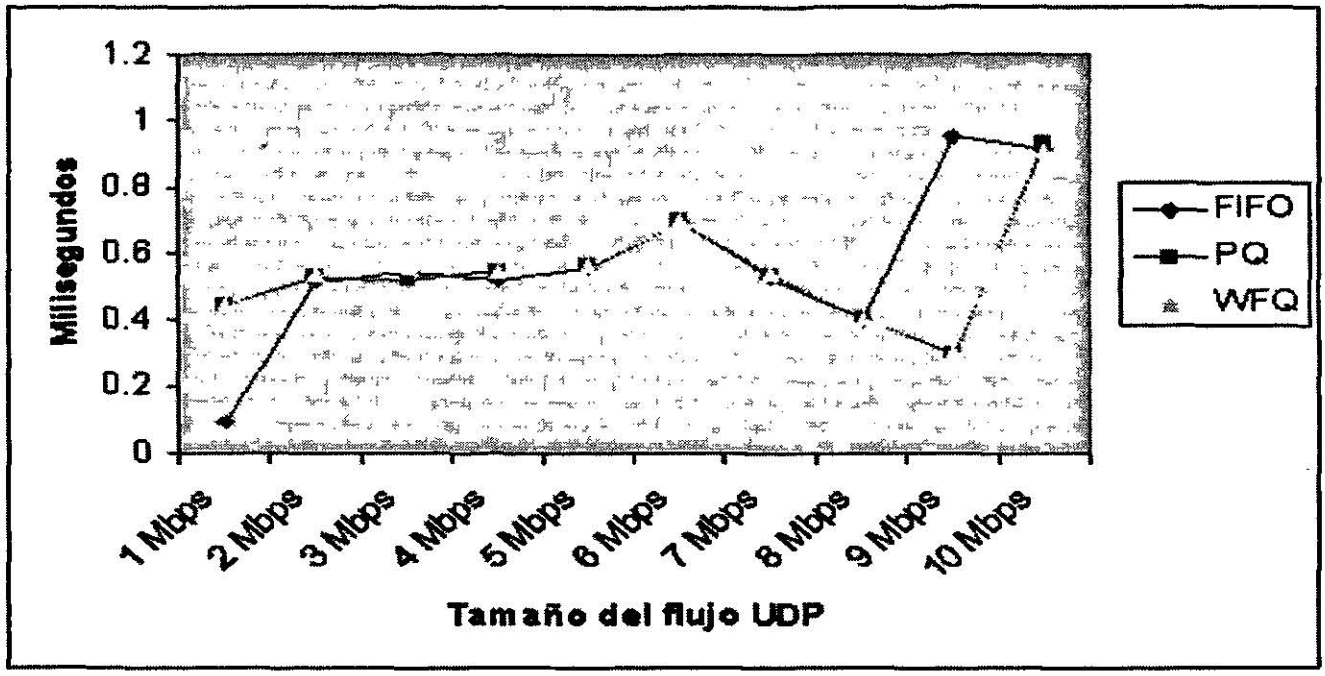


Figura 33. Comparativo de la variación del retardo entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 0 Mbps priorizando el conmutador.

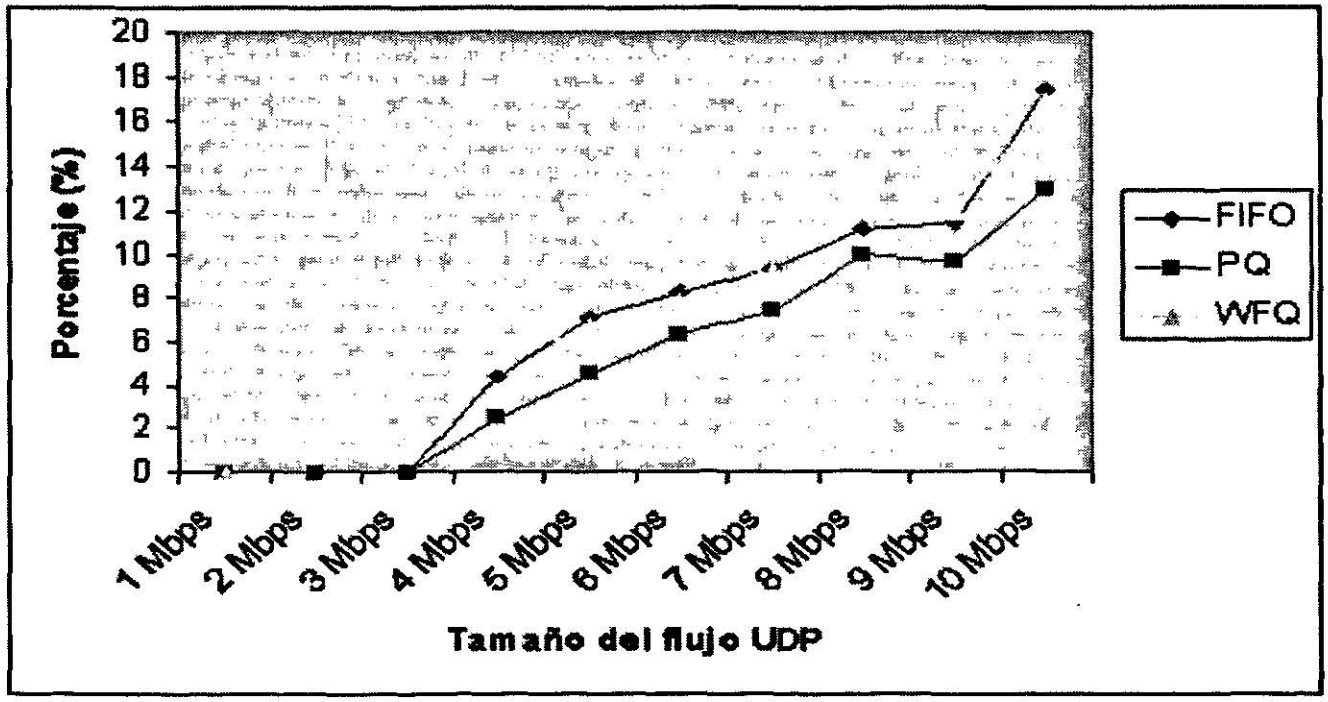


Figura 34. Comparativo del porcentaje de paquetes perdidos entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 0 Mbps y sin prioritización en el conmutador.

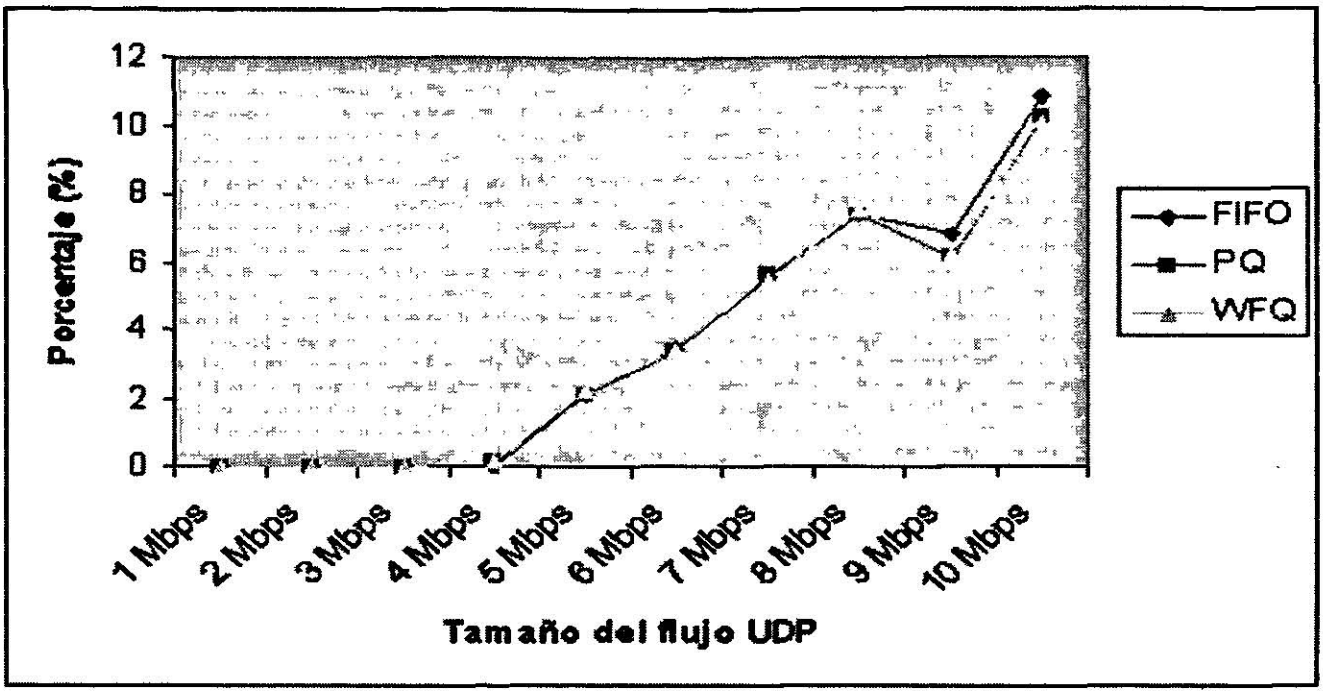


Figura 35. Comparativo del porcentaje de paquetes perdidos entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 0 Mbps priorizando el conmutador.

5.2.2.2 Resultados con 2.5 Mbps de carga de tráfico en la red.

La tabla XX muestra los resultados obtenidos con una carga de tráfico en la red de 2.5 Mbps (25%) y sin priorizar el conmutador. Vemos como el máximo caudal eficaz es de menos de 7 Mbps, y aunque en las pruebas se enviaban flujos mas grandes, debido a la carga de tráfico en la red solo se pudo obtener este valor. La variación en el retraso se comportó de manera estable para los tres mecanismos de encolamiento, salvo para flujos grandes donde utilizando FIFO hubo menor variación en el retraso. Al utilizar el mecanismo de encolamiento PQ obtuvimos una menor tasa de pérdida de paquetes en comparación de los otros dos mecanismos de encolamiento.

Aunque hubo pérdida de paquetes en los flujos de prueba, en el generador de tráfico no se registraron pérdidas de paquetes

TABLA XX.

CAUDAL EFICAZ, VARIACIÓN DEL RETARDO Y PORCENTAJE DE PAQUETES PERDIDOS PROMEDIO, PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑO DE FLUJO DE PAQUETES DEL EXPERIMENTO B CON CARGA DE TRÁFICO DE 2.5 MBPS Y SIN PRIORITIZACIÓN EN EL CONMUTADOR.

		Método de encolamiento								
		FIFO			PQ			WFQ		
		Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps	Jitter (ms)	% Paquetes Perdidos
Datos enviados Mbps	1	1.000	0.648	0	1.000	0.371	0	1.000	0.660	0
	2	2.000	0.626	0	2.000	0.546	0	2.000	0.581	0
	3	2.990	1.495	0.359	2.968	0.514	1.084	2.969	0.485	1.045
	4	3.834	0.546	3.247	3.821	0.513	4.481	3.827	0.528	4.331
	5	4.774	0.633	4.531	4.644	0.555	7.108	4.619	0.563	6.880
	6	5.554	0.545	7.303	5.440	0.544	9.323	5.464	0.551	8.919
	7	6.593	0.505	5.798	6.324	0.623	9.522	6.337	0.609	9.388
	8	6.606	1.598	17.307	6.442	0.816	19.476	6.408	1.008	20.009
	9	6.621	0.855	26.454	6.457	1.744	28.289	6.448	1.045	28.272
	10	6.607	0.784	33.929	6.411	1.751	35.863	6.479	1.666	35.365

Los resultados de las pruebas con prioritización en el conmutador se muestran en la tabla XXI. En estos resultados podemos observar que el caudal eficaz obtenido fue significativamente mayor a el que se logró sin priorizar el conmutador, además de comportarse de manera estable para los tres

mecanismos de encolamiento. La variación en el retraso se mantuvo estable, salvo para el flujo de 10 Mbps donde fue menor utilizando el mecanismo PQ.

TABLA XXI.

CAUDAL EFICAZ, VARIACIÓN DEL RETARDO Y PORCENTAJE DE PAQUETES PERDIDOS PROMEDIO, PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑO DE FLUJO DE PAQUETES DEL EXPERIMENTO B CON CARGA DE TRÁFICO DE 2.5 MBPS PRIORIZANDO EL CONMUTADOR.

		Método de encolamiento								
		FIFO			PQ			WFQ		
		Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps	Jitter (ms)	% Paquetes Perdidos
Datos enviados Mbps	1	1.000	0.580	0	1.000	0.600	0	1.000	0.574	0
	2	2.000	0.670	0	2.000	0.631	0	1.998	0.622	0
	3	2.988	0.782	0.150	2.991	1.254	0.102	3.000	0.689	0
	4	3.991	0.743	0.226	3.985	0.792	0.188	3.984	0.719	0.384
	5	4.870	0.699	2.589	4.870	0.685	2.597	4.857	0.680	2.856
	6	5.785	0.623	3.577	5.769	0.553	3.831	5.736	0.574	4.223
	7	6.609	0.569	5.503	6.642	0.573	5.093	6.680	0.547	4.548
	8	7.371	0.451	7.814	7.411	0.479	7.312	7.395	0.473	7.525
	9	8.366	0.415	6.927	8.192	0.402	8.881	8.343	0.473	7.335
	10	8.951	1.679	10.350	8.911	0.848	10.864	8.994	0.943	10.015

Al realizar las pruebas con prioritización en el conmutador, el porcentaje de paquetes perdidos por el generador de tráfico creció enormemente para flujos de más de 8 Mbps, tal como se muestra en la tabla XXII.

TABLA XXII.

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE FLUJO DE PAQUETE DEL EXPERIMENTO B CON CARGA DE TRÁFICO DE 2.5 MBPS PRIORIZANDO EL CONMUTADOR.

		Método de encolamiento		
		FIFO	PQ	WFQ
		% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico
Datos enviados Mbps	1	0	0	0
	2	0	0	0
	3	0	0	0
	4	0	0	0
	5	0	0	0
	6	0	0	0
	7	0	0	0
	8	27.111	27.215	27.215
	9	57.246	56.221	57.559
	10	58.481	58.883	58.954

Al observar las figuras 36 a la 43 vemos que para flujos de mas de 7 Mbps el priorizar los flujos de prueba en el conmutador se ve reflejado significativamente en el caudal eficaz alcanzado, independientemente del mecanismo de encolamiento utilizado para cada caso. Vemos también que la variación en el retraso es menor y más estable al priorizar el conmutador. La tasa de paquetes perdidos es considerablemente mayor cuando no se prioriza el conmutador que cuando sí se priorizan los flujos.

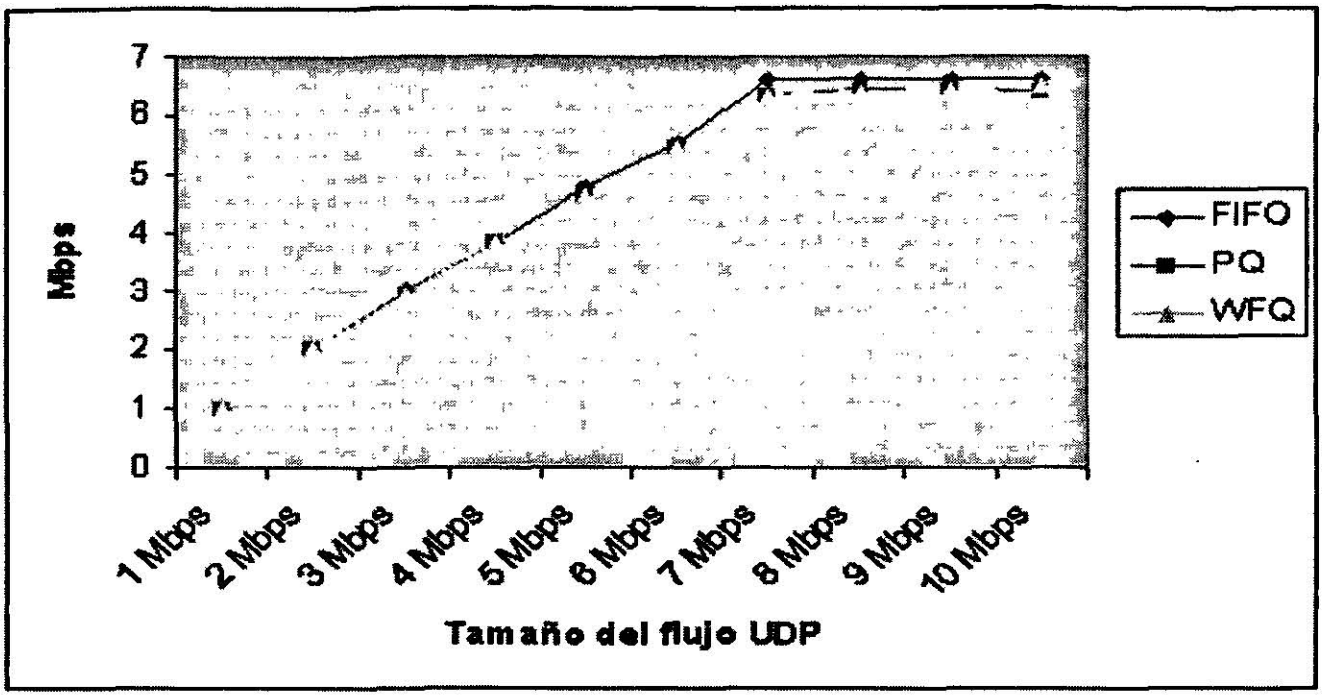


Figura 36. Comparativo del caudal eficaz entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 2.5 Mbps y sin prioritización en el conmutador.

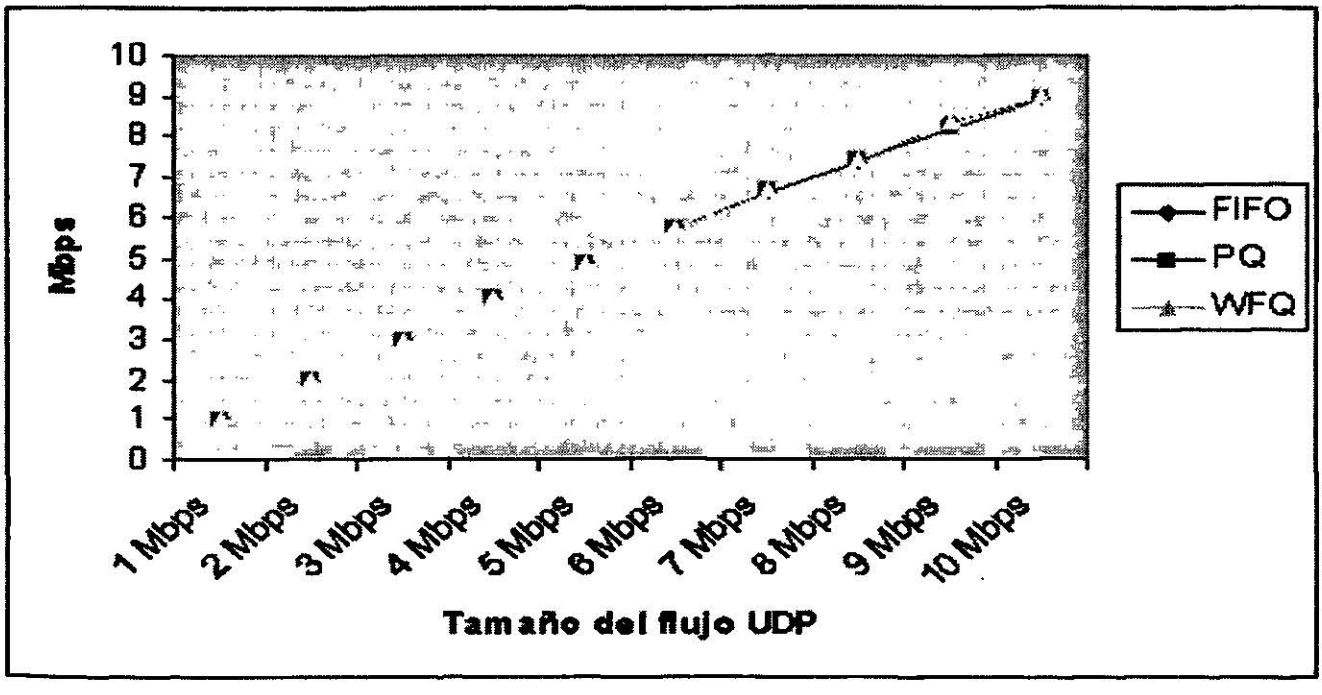


Figura 37. Comparativo del caudal eficaz entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 2.5 Mbps priorizando el conmutador.

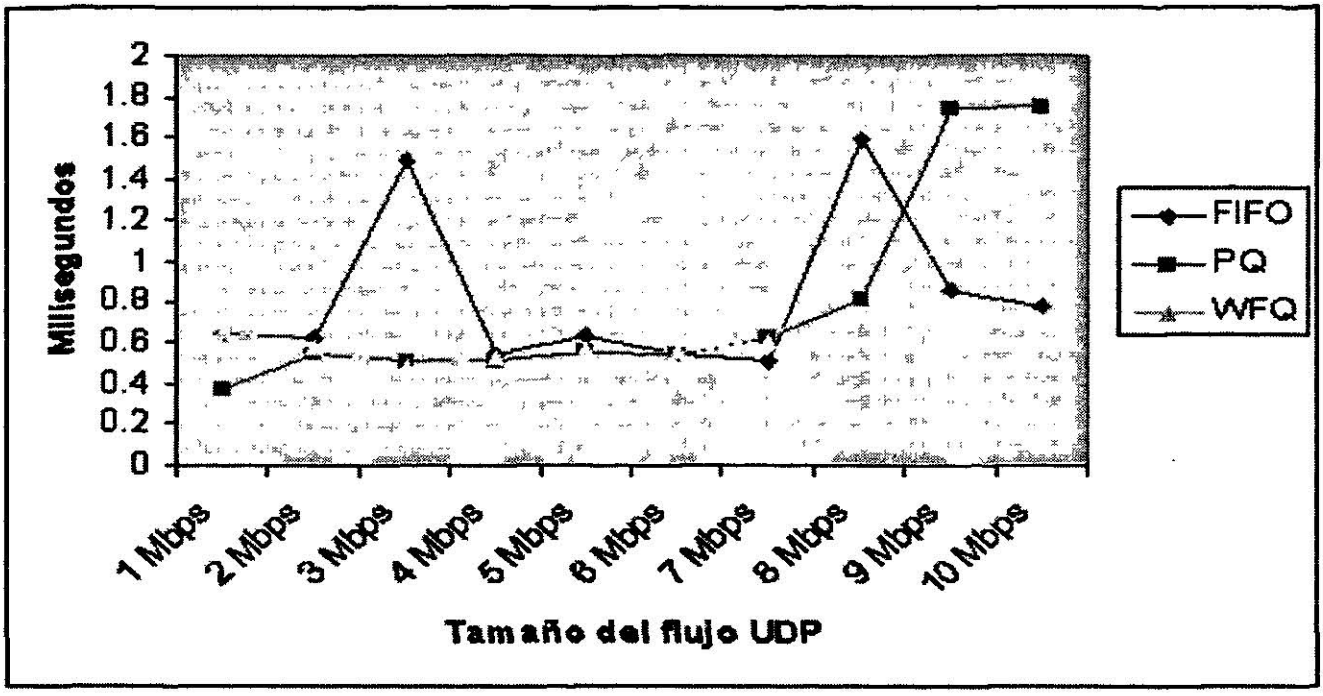


Figura 38. Comparativo de la variación del retardo entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 2.5 Mbps y sin prioritización en el conmutador.

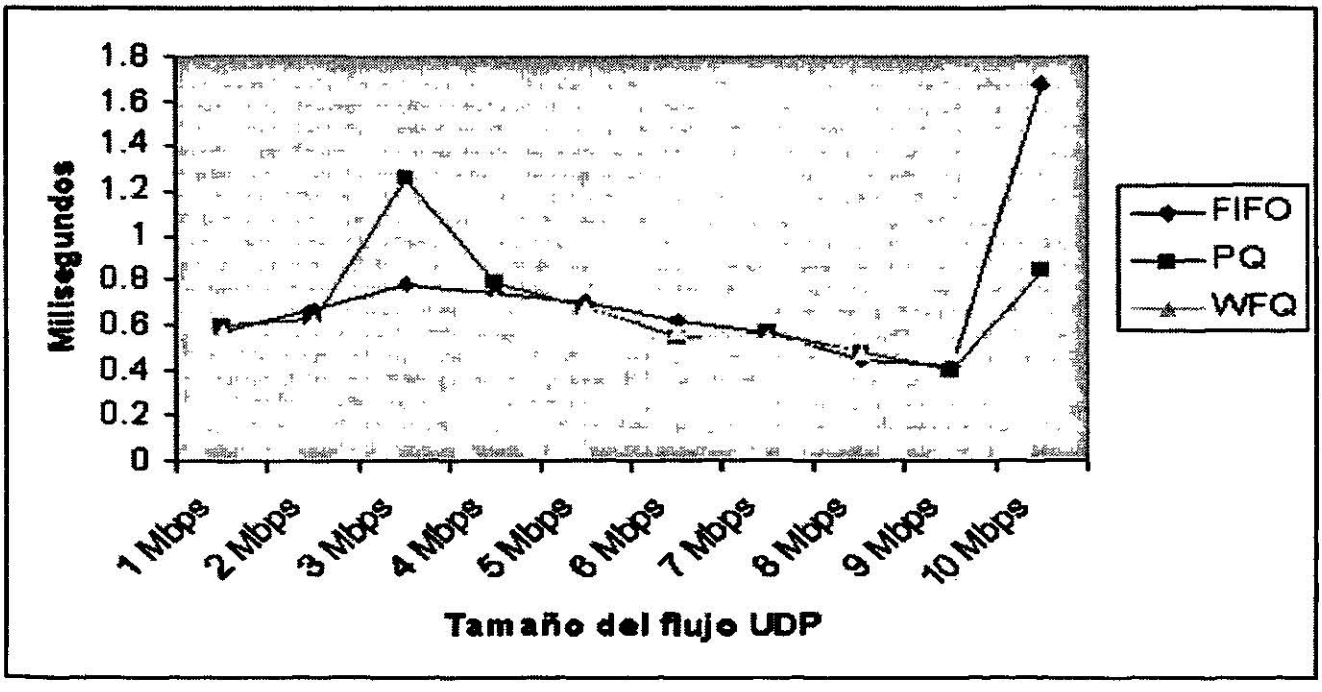


Figura 39. Comparativo de la variación del retardo entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 2.5 Mbps priorizando el conmutador.

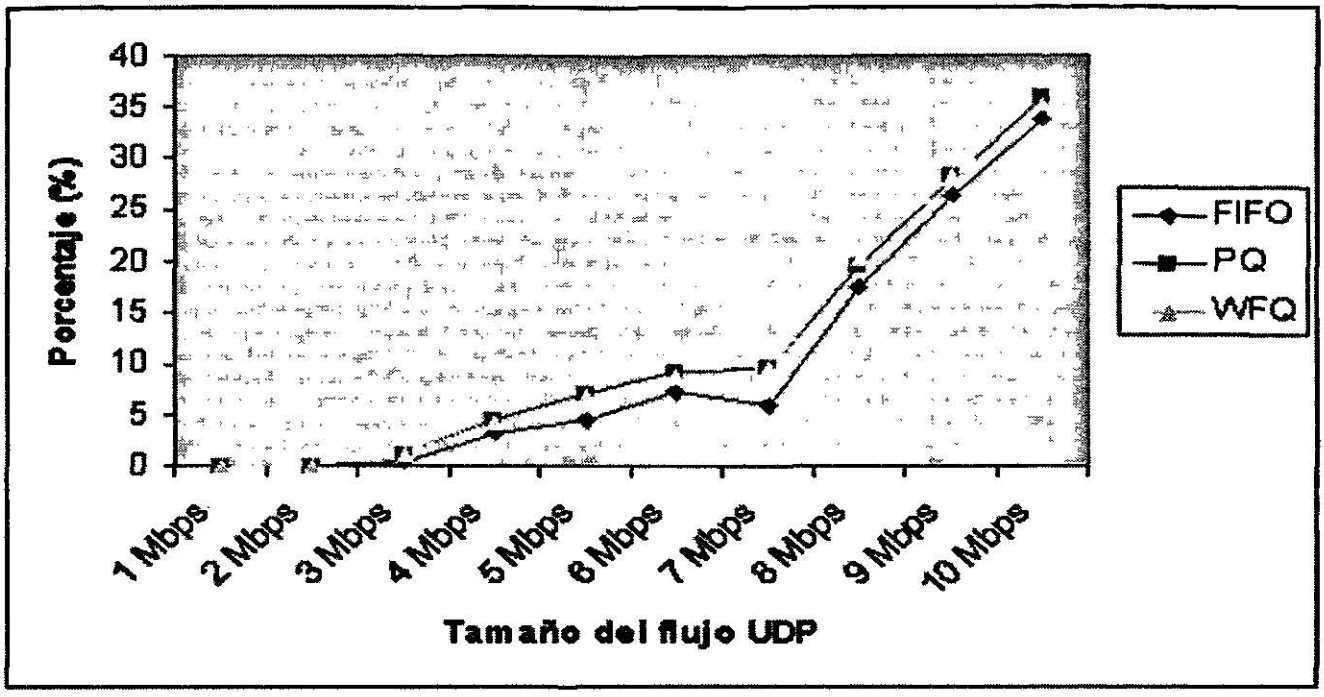


Figura 40. Comparativo del porcentaje de paquetes perdidos entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 2.5 Mbps y sin prioritización en el conmutador.

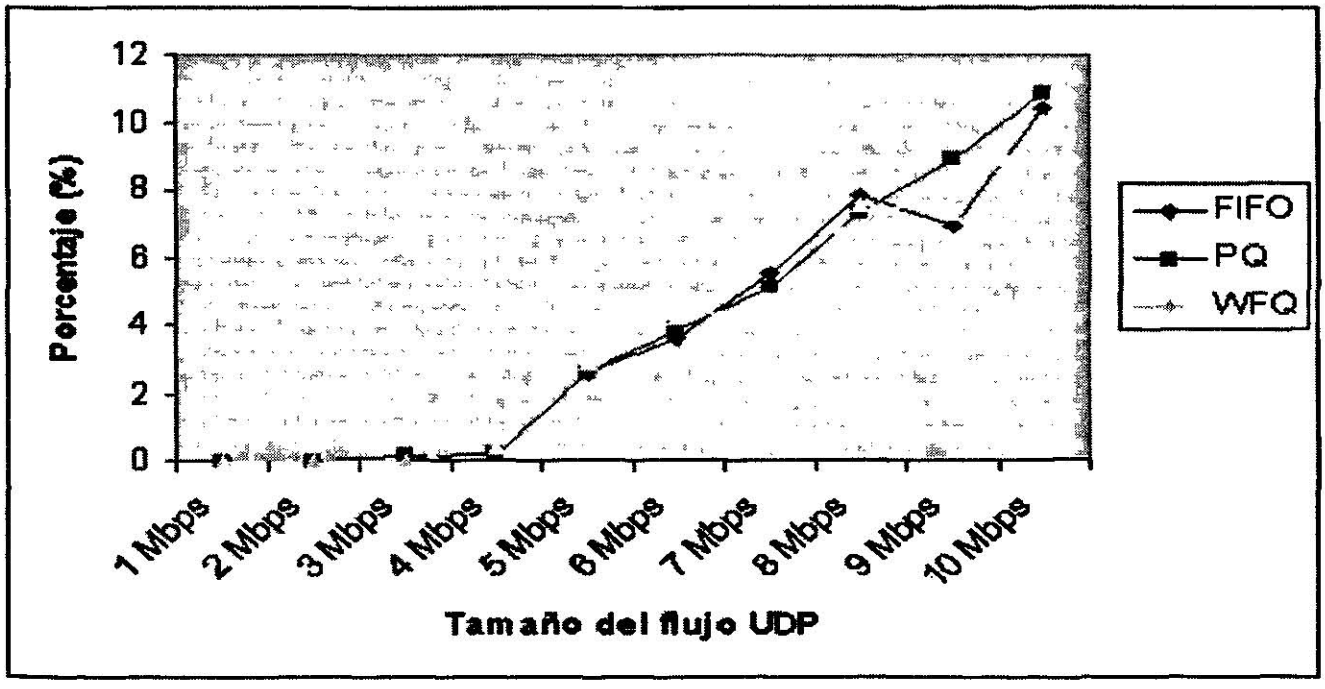


Figura 41. Comparativo del porcentaje de paquetes perdidos entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 2.5 Mbps priorizando el conmutador.

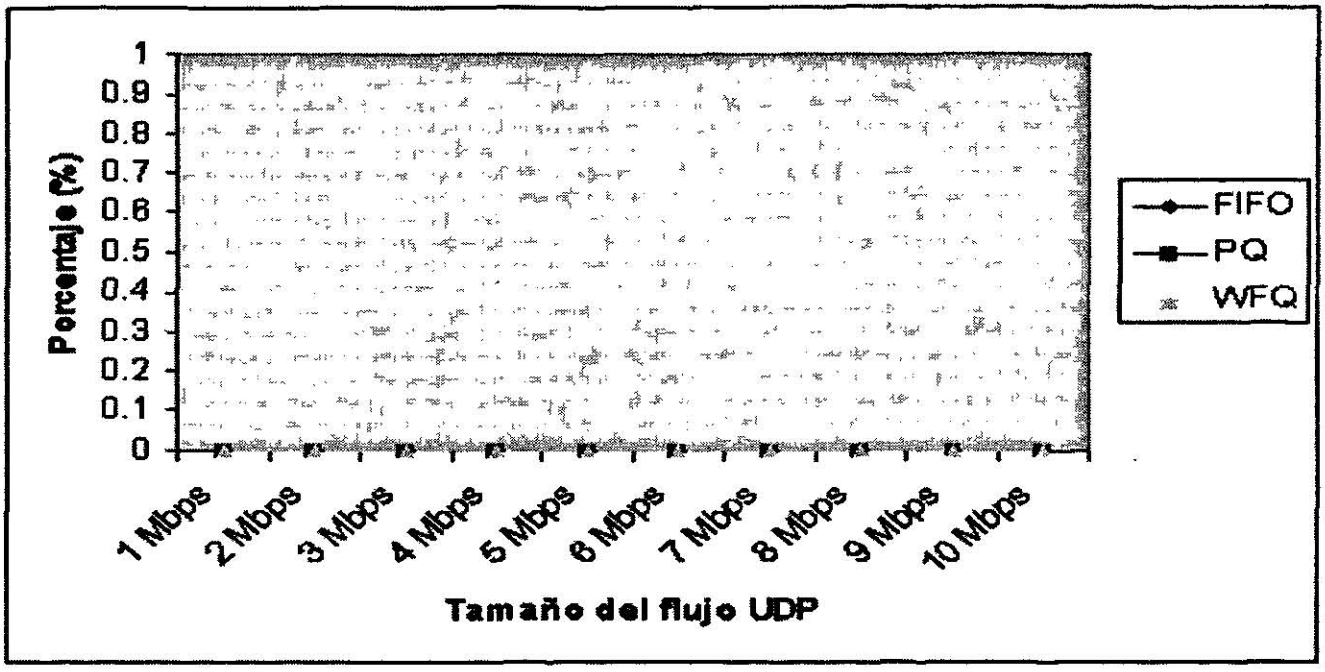


Figura 42. Comparativo del porcentaje de paquetes perdidos por el generador de tráfico entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 2.5 Mbps y sin prioritización en el conmutador.

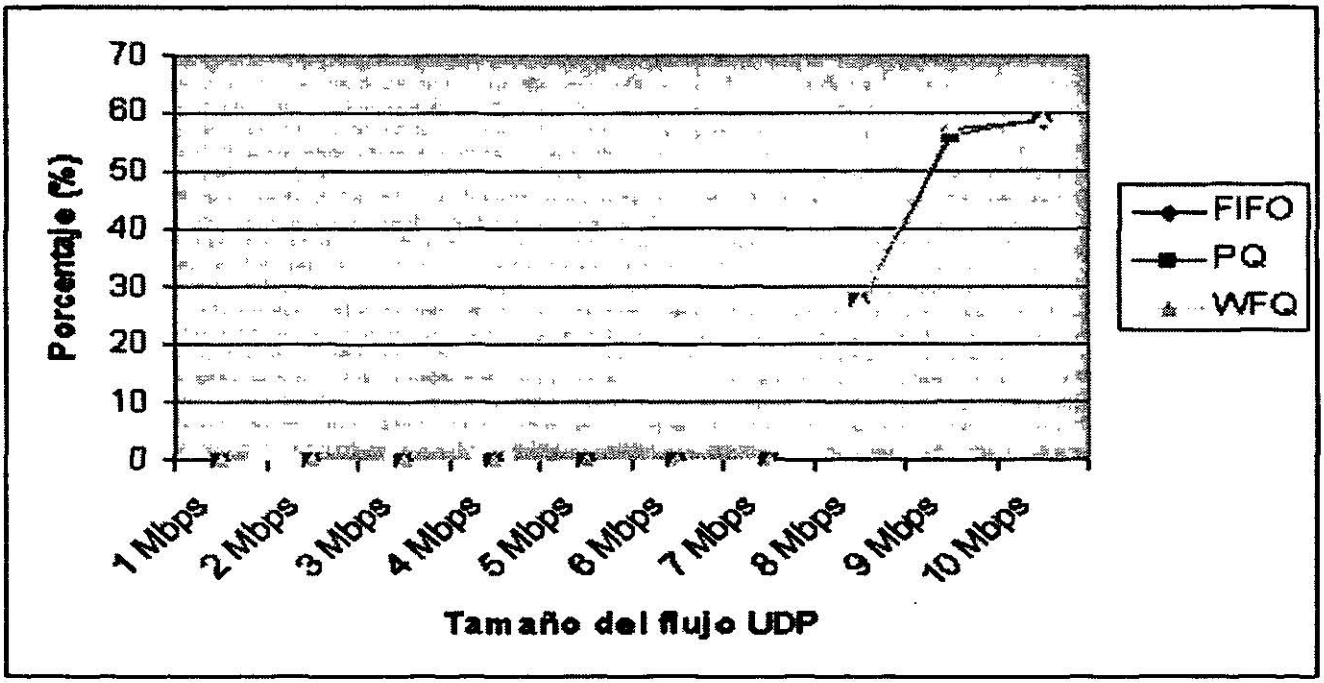


Figura 43. Comparativo del porcentaje de paquetes perdidos por el generador de tráfico entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 2.5 Mbps priorizando el conmutador.

5.2.2.3 Resultados con 5 Mbps de carga de tráfico en la red.

La tabla XXIII muestra los resultados para las pruebas con carga de tráfico de 5 Mbps (50%) y sin priorizar el conmutador. Observamos que el caudal eficaz obtenido fue de 4.545 Mbps máximo, aunque se enviaron flujos de prueba de hasta 10 Mbps; después de los flujos de prueba de 5 Mbps el caudal eficaz logrado se mantuvo estable en los 4.5 Mbps aproximadamente. La variación en el retraso fue alta y dispersa, donde para flujos grandes observamos que con el mecanismo de encolamiento PQ se obtuvo una menor variación en el retraso.

La tasa de pérdida de paquetes por los flujos de prueba de Iperf comenzó a crecer a partir del flujo de 3 Mbps para alcanzar una muy alta tasa de pérdida de paquetes para los flujos de mas de 6 Mbps. Utilizando el mecanismo de encolamiento PQ se obtuvieron tasas de pérdida de paquetes ligeramente menores en comparación con los otros mecanismos de encolamiento.

TABLA XXIII.

CAUDAL EFICAZ, VARIACIÓN DEL RETARDO Y PORCENTAJE DE PAQUETES PERDIDOS PROMEDIO, PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑO DE FLUJO DE PAQUETES DEL EXPERIMENTO B CON CARGA DE TRÁFICO DE 5 MBPS Y SIN PRIORITIZACIÓN EN EL CONMUTADOR.

		Método de encolamiento								
		FIFO			PQ			WFQ		
		Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps	Jitter (ms)	% Paquetes Perdidos
Datos enviados Mbps	1	1.000	1.621	0	0.997	2.086	0	1.000	2.879	0
	2	2.000	0.854	0	2.000	2.167	0	2.000	0.860	0
	3	2.962	0.855	1.155	2.965	0.927	1.084	2.848	0.596	4.927
	4	3.786	0.767	5.346	3.783	0.828	5.934	3.775	0.783	5.196
	5	4.494	1.576	9.938	4.461	1.557	10.533	4.475	1.625	10.446
	6	4.506	1.754	24.804	4.494	1.657	25.007	4.530	1.638	24.477
	7	4.446	1.361	36.510	4.443	2.045	36.644	4.493	1.850	35.743
	8	4.538	1.192	43.256	4.517	1.776	43.491	4.570	1.889	42.748
	9	4.530	1.880	49.548	4.554	1.312	49.425	4.520	2.044	49.778
	10	4.527	1.760	54.701	4.507	1.542	54.508	4.545	1.832	54.475

Como podemos observar en la tabla XXIV, la tasa de paquetes perdidos por el generador de tráfico fue muy baja, la cual no superó el 3% de paquetes perdidos aún para flujos de paquetes de prueba de 10 Mbps.

TABLA XXIV.

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE FLUJO DE PAQUETE DEL EXPERIMENTO B CON CARGA DE TRÁFICO DE 5 MBPS Y SIN PRIORITIZACIÓN EN ELCONMUTADOR.

		Método de encolamiento		
		FIFO	PQ	WFQ
		% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico
Datos enviados Mbps	1	0	0	0
	2	0.028	0	0
	3	0	0	0
	4	0	0	0
	5	1.212	1.177	1.159
	6	2.222	2.144	2.123
	7	2.429	2.404	2.496
	8	2.731	2.706	2.660
	9	2.859	2.848	2.845
	10	2.919	2.905	2.909

Los resultados de las pruebas con carga de tráfico de 5 Mbps y priorizando los flujos de prueba en el conmutador se muestran en la tabla XXV. Contrario a los resultados anteriores, podemos observar que priorizando los flujos de prueba en el conmutador obtenemos un caudal eficaz mucho mayor, al obtener hasta 8.999 Mbps de caudal eficaz para flujos de 10 Mbps; podemos observar que al utilizar el mecanismo de encolamiento PQ se obtuvieron mejores resultados de caudal eficaz.

La variación en el retraso se mantuvo estable, salvo cuando se utilizo el mecanismo PQ con flujos de 2 Mbps donde la variación en el retraso fue muy

alta. Podemos observar que al utilizar el mecanismo de encolamiento WFQ en la mayoría de los flujos de prueba se obtuvieron variaciones en el retraso mas bajas.

La tasa de paquetes perdidos de los flujos de prueba comenzó a generarse a partir de los flujos de 4 Mbps en adelante, y la tasa de perdida máxima alcanzada fue de menos de 11% aún para flujos de prueba de 10 Mbps. En la mayoría de los tamaños de flujos de prueba se obtuvieron tasas de pérdida mas bajas al utilizar el mecanismo de encolamiento PQ.

TABLA XXV.

CAUDAL EFICAZ, VARIACIÓN DEL RETARDO Y PORCENTAJE DE PAQUETES PERDIDOS PROMEDIO, PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑO DE FLUJO DE PAQUETES DEL EXPERIMENTO B CON CARGA DE TRÁFICO DE 5 MBPS PRIORIZANDO EL CONMUTADOR.

		Método de encolamiento								
		FIFO			PQ			WFQ		
		Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps)	Jitter (ms)	% Paquetes Perdidos	Caudal Eficaz (Mbps	Jitter (ms)	% Paquetes Perdidos
Datos enviados Mbps	1	1.000	0.861	0	1.000	0.567	0	1.000	0.842	0
	2	2.000	0.911	0	2.000	2.730	0	2.000	0.869	0
	3	2.999	0.858	0.013	3.000	0.846	0	3.000	0.847	0
	4	3.989	0.677	0.150	3.983	0.660	0.406	3.980	1.002	0.239
	5	4.888	0.823	2.238	4.887	0.824	2.246	4.897	0.791	2.058
	6	5.769	0.772	3.806	5.769	0.776	3.817	5.779	0.763	3.646
	7	6.557	0.578	6.294	6.546	0.614	6.274	6.505	0.771	6.899
	8	7.400	0.463	7.457	7.458	0.732	6.576	7.397	0.461	7.515
	9	8.798	0.746	2.290	8.343	0.372	7.328	8.427	0.369	6.399
	10	8.918	0.948	10.769	8.999	0.800	10.023	8.968	0.746	10.327

Sin embargo, como se observa en la tabla XXVI, al priorizar los flujos de prueba en el conmutador, la tasa de paquetes perdidos por el generador de tráfico creció enormemente, hasta alcanzar una tasa de pérdida de casi el 70% para los flujos de 10 Mbps.

TABLA XXVI.

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y TAMAÑOS DE FLUJO DE PAQUETE DEL EXPERIMENTO B CON CARGA DE TRÁFICO DE 5 MBPS PRIORIZANDO EL CONMUTADOR.

		Método de encolamiento		
		FIFO	PQ	WFQ
		% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico	% Paquetes Perdidos por el Generador de Tráfico
Datos enviados Mbps	1	0	0	0
	2	0	0	0
	3	0	0	0
	4	0	0	0
	5	4.996	4.960	4.989
	6	20.737	20.865	20.865
	7	36.478	36.481	36.940
	8	52.345	51.576	52.425
	9	67.437	67.731	67.463
	10	69.557	48.763	52.602

Al observar las figuras 44 a la 52 notamos que al priorizar los flujos de prueba en el conmutador el caudal eficaz obtenido es significativamente mayor para flujos de prueba grandes que cuando no se prioriza el conmutador. La variación en el retraso es menor y más estable al priorizar los flujos en el

conmutador, y la tasa de pérdida de paquetes es considerablemente mayor cuando no se realiza la prioritización en el conmutador. Observamos también que al priorizar los flujos en el conmutador, la tasa de paquetes perdidos por el generador de tráfico es mucho mayor que cuando no se realiza la prioritización.

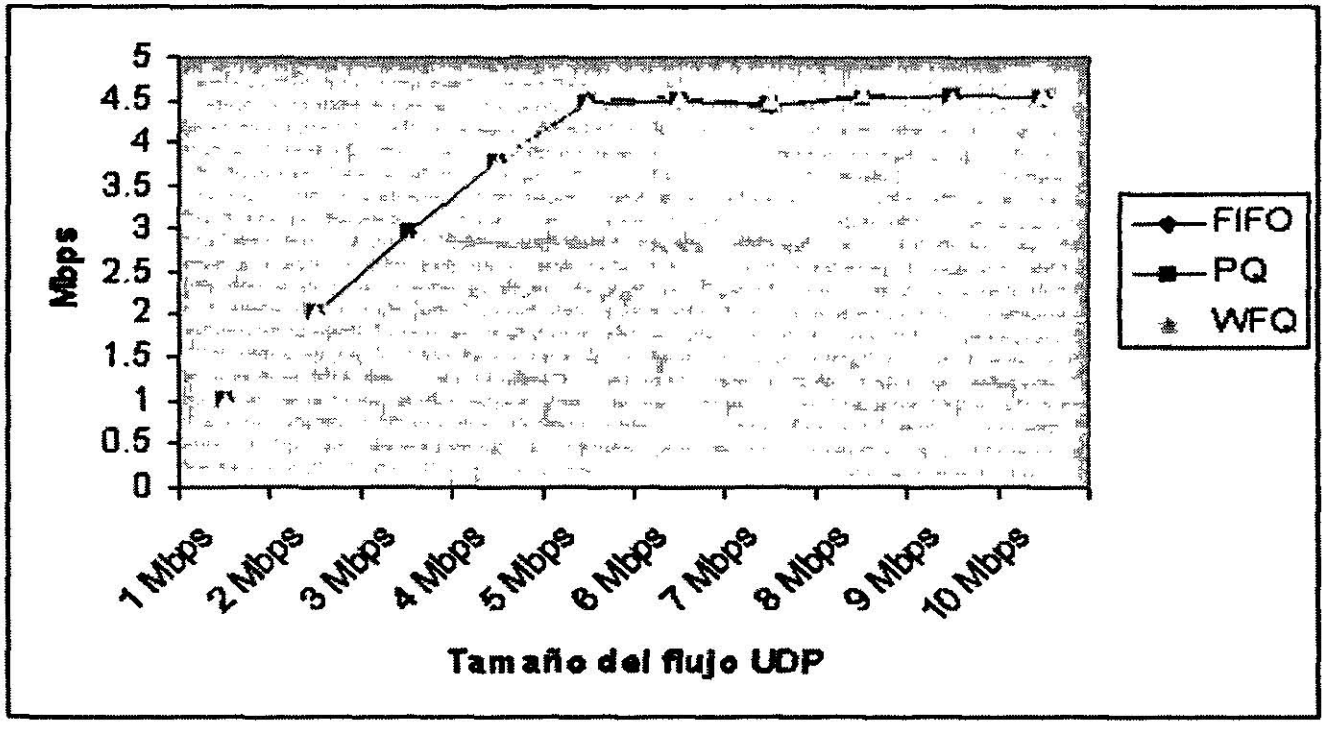


Figura 44. Comparativo del caudal eficaz entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 5 Mbps y sin prioritización en el conmutador.

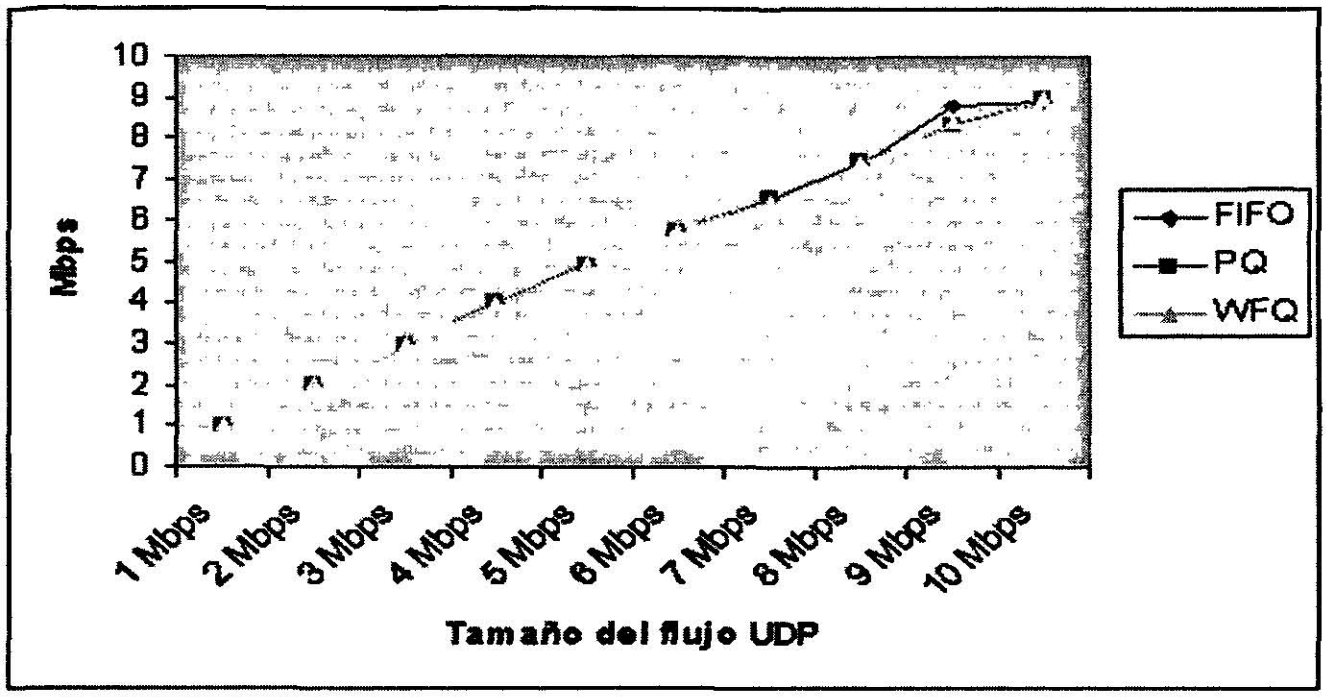


Figura 45. Comparativo del caudal eficaz entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 5 Mbps priorizando el conmutador.

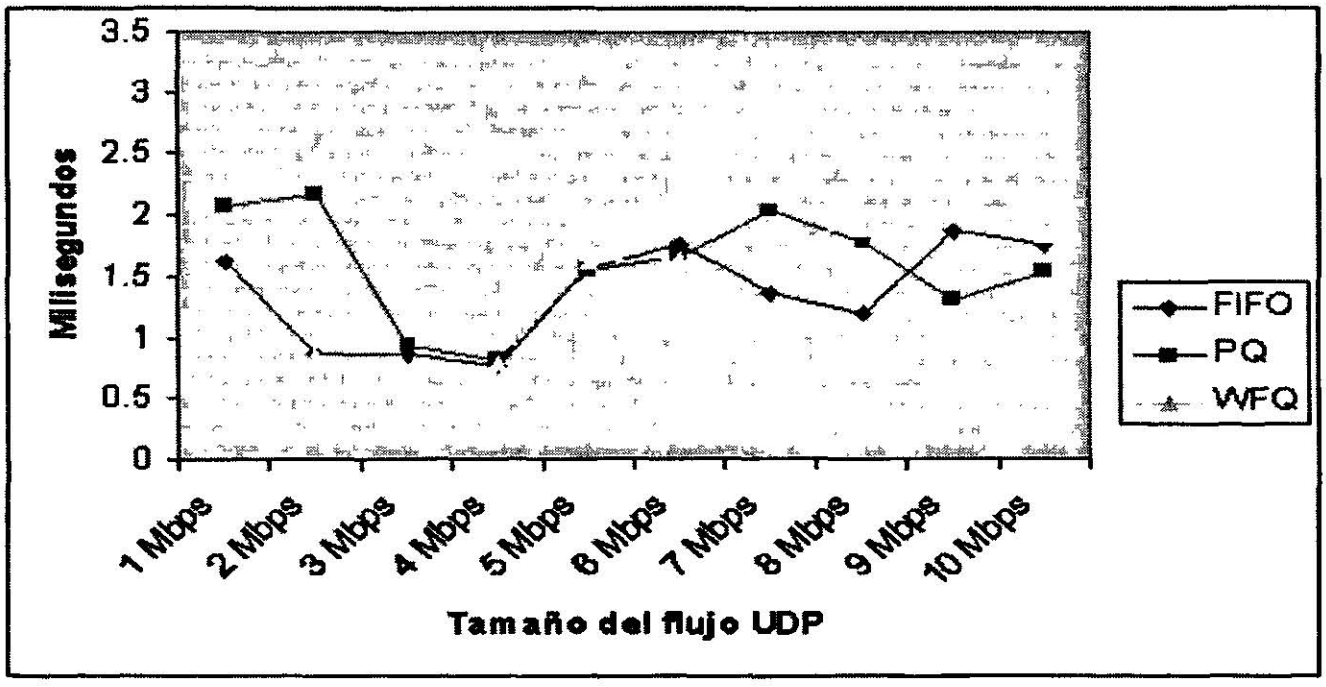


Figura 46. Comparativo de la variación del retardo entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 5 Mbps y sin prioritización en el conmutador.

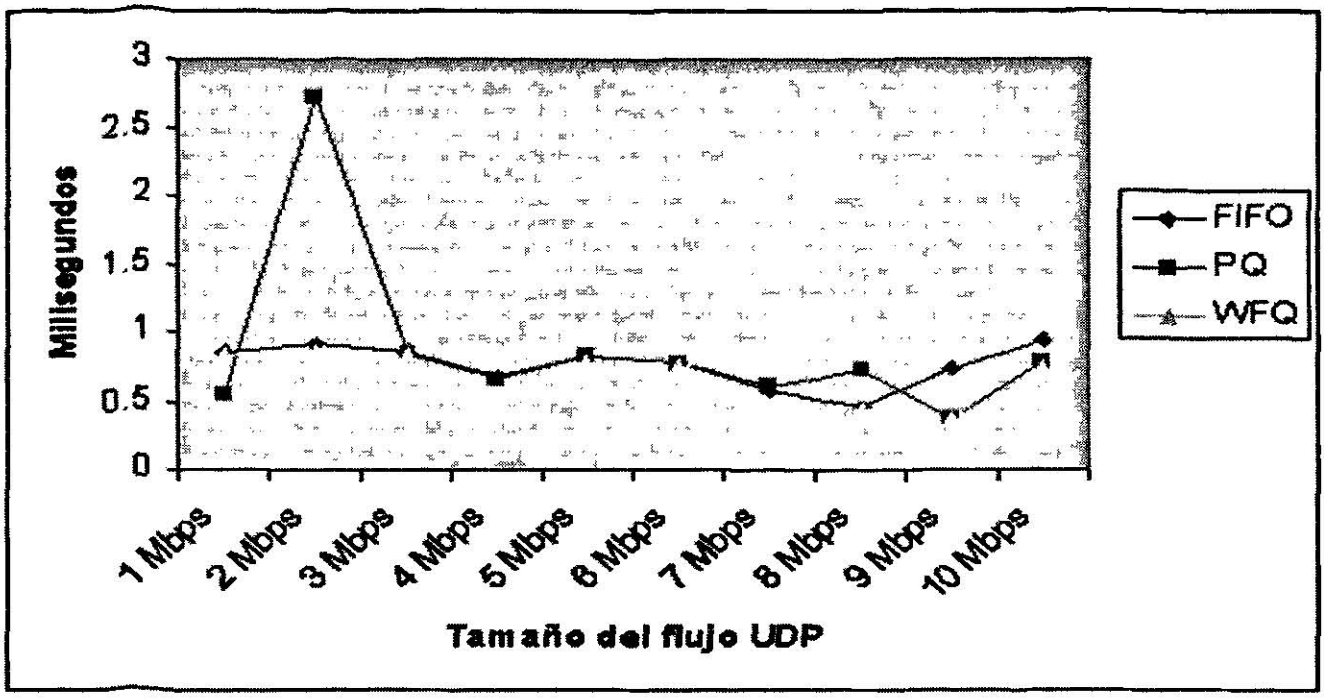


Figura 47. Comparativo de la variación del retardo entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 5 Mbps priorizando el conmutador.

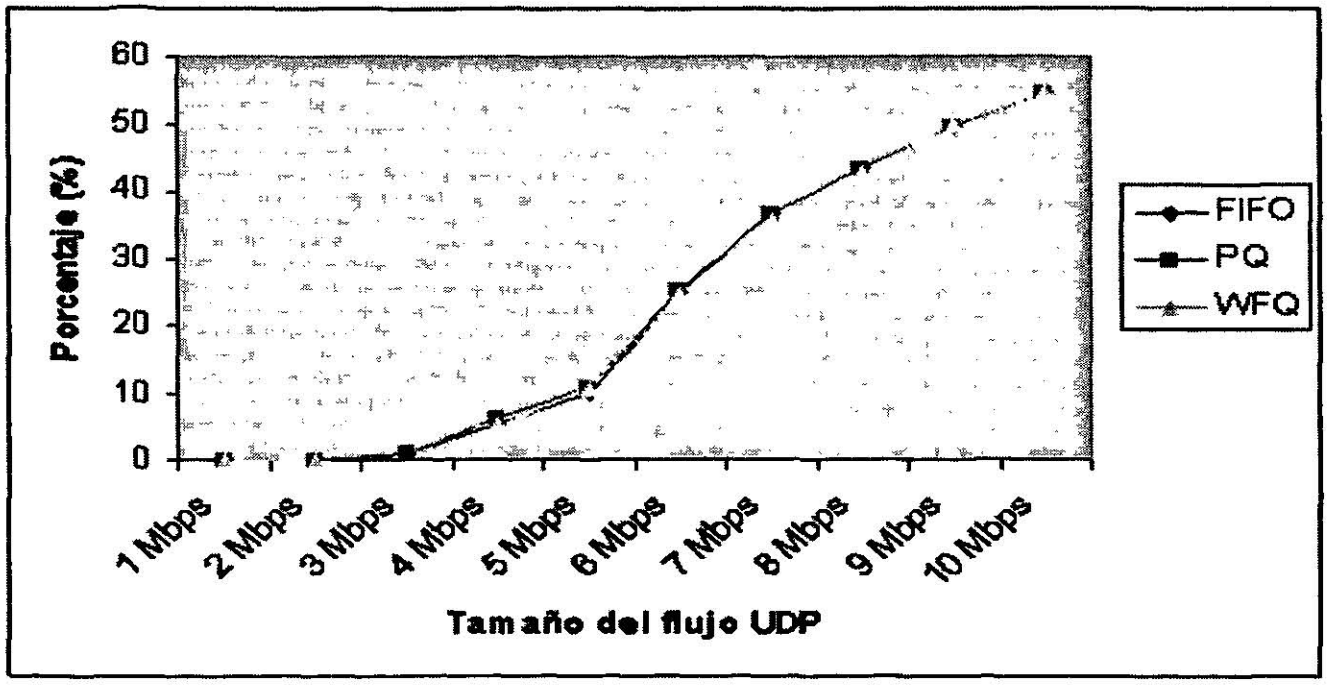


Figura 48. Comparativo del porcentaje de paquetes perdidos entre los distintos mecanismos de encolamiento del *Experimento B con una carga de tráfico de* 5 Mbps y sin priorización en el conmutador.

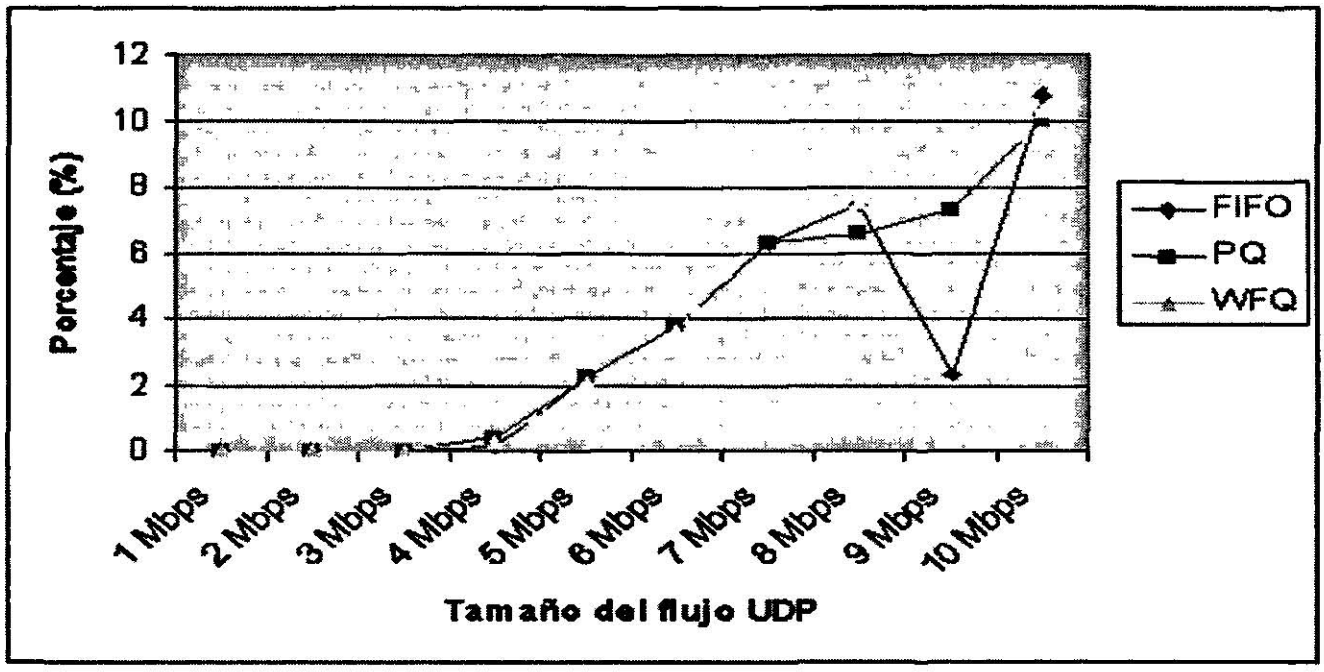


Figura 49. Comparativo del porcentaje de paquetes perdidos entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 5 Mbps priorizando el conmutador.

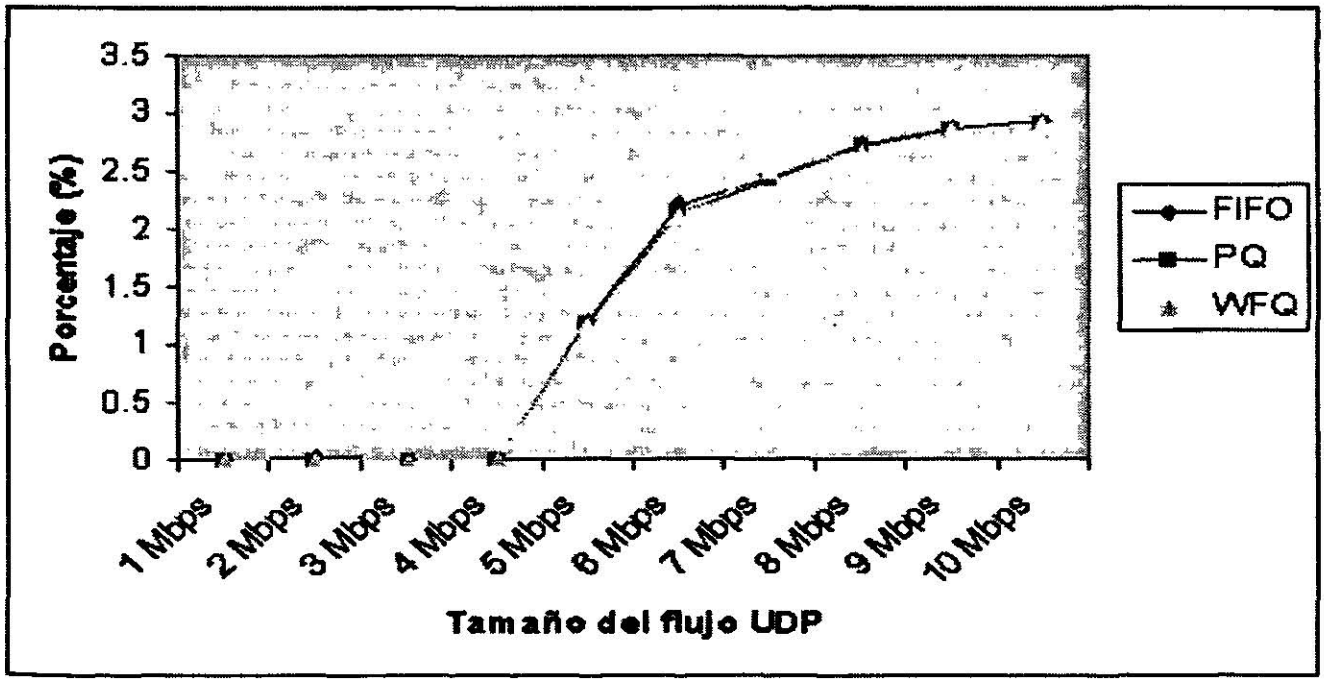


Figura 50. Comparativo del porcentaje de paquetes perdidos por el generador de tráfico entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 5 Mbps y sin prioritización en el conmutador.

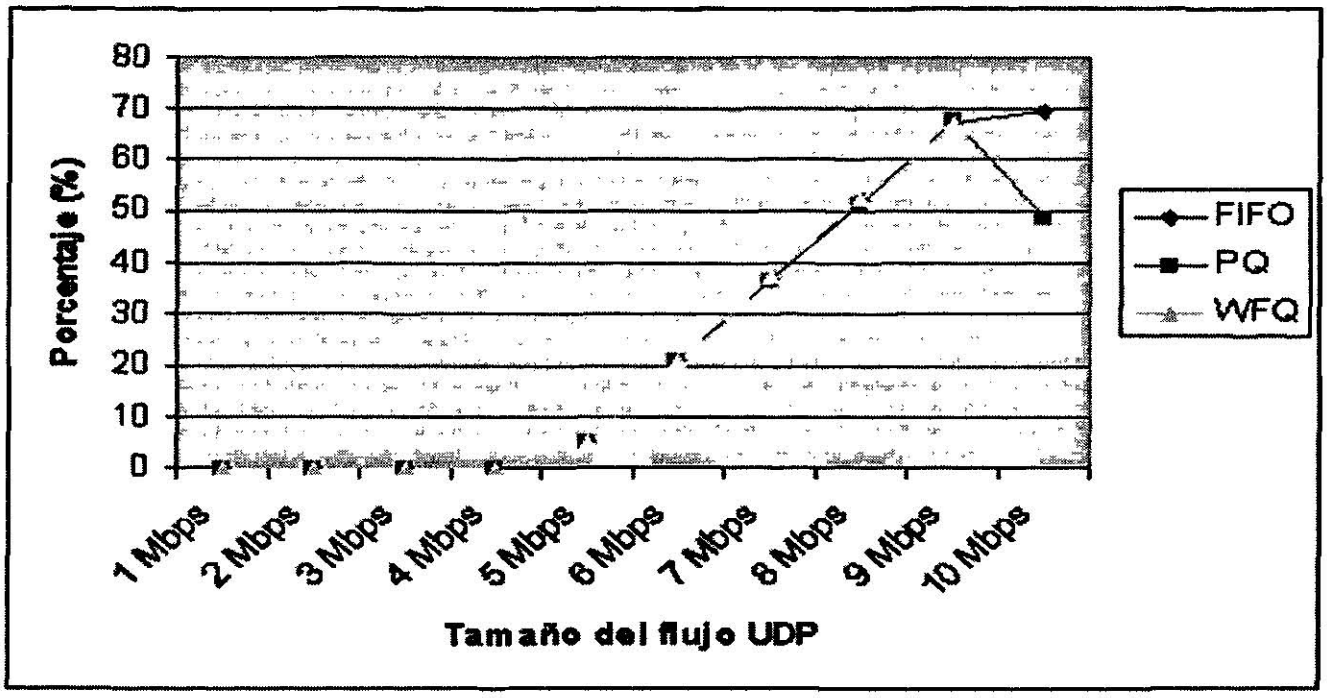


Figura 52. Comparativo del porcentaje de paquetes perdidos por el generador de tráfico entre los distintos mecanismos de encolamiento del Experimento B con una carga de tráfico de 5 Mbps priorizando el conmutador.

5.3 Experimento C: Medición de Tiempo y Tamaño de Colas de Espera.

En los dos experimentos anteriores se realizaron mediciones de extremo a extremo, es decir, solo se median los parámetros entre los computadores, sin importar que sucediera en cada uno de los saltos del trayecto de los paquetes.

Sin embargo, también es necesario observar lo que sucede en cada uno de los tramos o saltos en toda la ruta de comunicaciones para saber con mayor precisión en que parte de nuestra red se encuentra un eventual problema.

Este experimento tiene como objetivo observar el comportamiento en las colas de los equipos de comunicaciones en cada uno de los "saltos" por todo el trayecto que atraviesan los paquetes en la comunicación entre dos computadores.

5.3.1 Descripción del Experimento C.

Este experimento consistió en medir el comportamiento de las colas en cada uno de los saltos para apreciar lo que sucede en cada parte del trayecto por el cual atraviesan los paquetes.

Se midió el tiempo que los paquetes tardaban en cola así como el tamaño aproximado de la cola de espera. Para este experimento se identificaron dos factores que afectaron las mediciones, para los cuales se realizó una prueba a cada una de las combinaciones de estos factores.

5.3.1.1 Factores que Intervinieron en el Experimento.

Los factores que intervinieron en este experimento fueron: Método de encolamiento y la carga de tráfico. Cada uno con distintos niveles los cuales se resumen en la tabla XXVII.

TABLA XXVII

FACTORES ESTUDIADOS EN EL EXPERIMENTO C

Factores	
Factor A Método de encolamiento	Factor B Carga de tráfico
Nivel 1.- Primero en Entrar Primero en Salir (FIFO) Nivel 2.- Encolamiento Priorizado (PQ) Nivel 3.- Encolamiento Justo Ponderado (WFQ)	Nivel 1.- 7.5 Mbps (75%) Nivel 2.- 10 Mbps (100%) Nivel 3.- 12.5 Mbps (125%)

El factor método de encolamiento se refiere al algoritmo de encolamiento que se utilizó en el enrutador. Los tres algoritmos de encolamiento utilizados ya fueron descritos en la sección 4.2.

El factor carga de tráfico se refiere a la cantidad de tráfico existente en el modelo de pruebas debido a tráfico proveniente de otras entidades distintas a nuestras entidades de prueba. En este factor los niveles de carga de tráfico van del 75% al 125% (12.5Mbps) del enlace de comunicaciones entre el enrutador y los conmutadores. Solamente se consideraron estas cargas de tráfico ya que con cargas de tráfico menores no se formaban colas de espera en el enrutador, y debido a que para realizar las mediciones propias de este experimento era

necesario que se formaran colas de espera en el enrutador se utilizaron las cargas de tráfico descritas en la tabla XXVII.

5.3.1.2 Diagrama del Experimento.

Para realizar este experimento se utilizó un modelo de conectividad aislado utilizando cuatro computadores, un conmutador (switch), un conmutador-enrutador (switch-router) y un enrutador interconectados como se muestra en la figura 52.

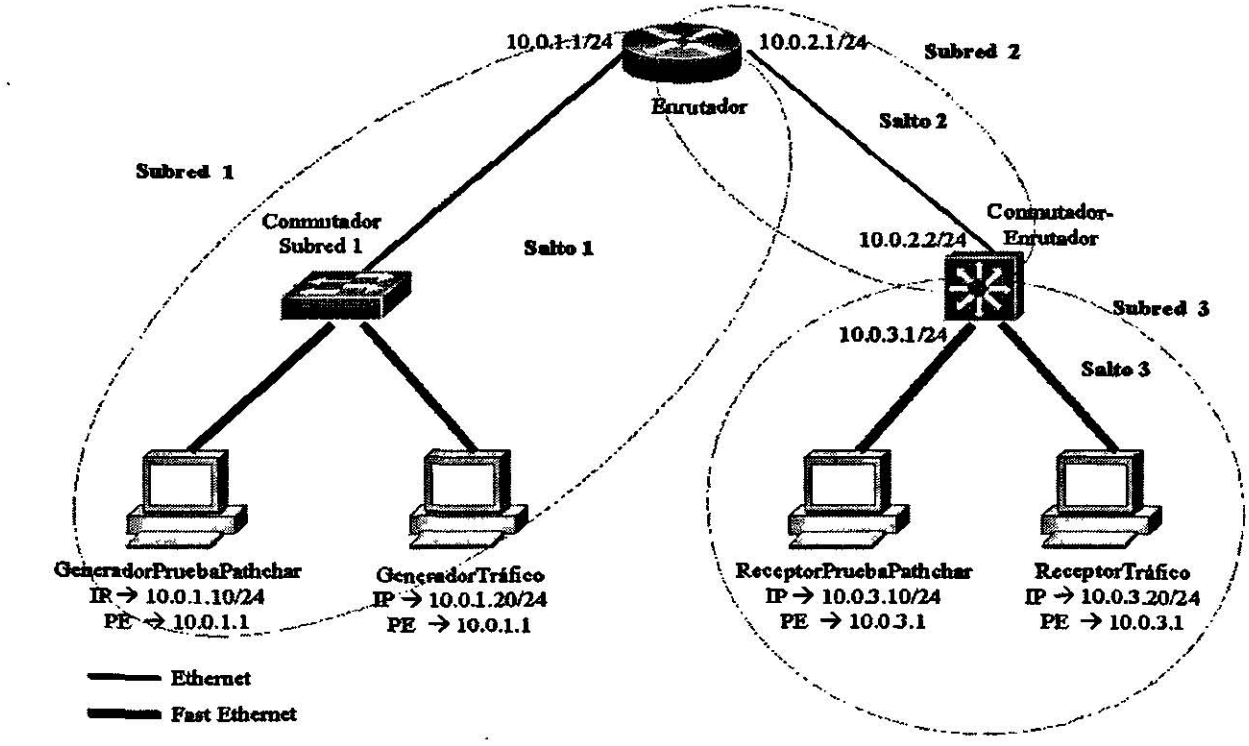


Figura 52. Esquema de conectividad del experimento C.

El esquema de este experimento varía un poco de los anteriores, debido a que fue necesario que estuvieran claramente definidos los "saltos" que tenían que dar los paquetes.

Se utilizaron cuatro computadores con sistema operativo Linux RedHat 8.0 (Kernel 2.4.18), el Generador de Prueba Pathchar el cual fue el encargado de enviar y medir los datos de la prueba hacia el computador Receptor de Prueba Pathchar. Los computadores Generador de Tráfico y Receptor de Tráfico realizaron las mismas funcionalidades que en los experimentos anteriores.

En los esquemas anteriores, el conmutador de la subred 2 había estado realizando solamente funciones de conmutación en capa 2 (Capa de Enlace del Modelo OSI), sin embargo para poder observar el comportamiento de las colas en los diferentes saltos fue necesario utilizar este conmutador con funcionalidad de capa 3 (Capa de Red), ya que las mediciones se basan en protocolos de la familia IP (específicamente en ICMP). A este dispositivo le llamamos Conmutador-Enrutador, ya que realiza ambas funciones. Este modelo está formado por tres subredes, cada una con un esquema de direccionamiento propio.

El primer salto, correspondiente al enlace entre los computadores Generador de Prueba Pathchar y Generador de Tráfico hacía el enrutador, nos permitió observar el comportamiento de las colas en la interfaz de entrada del enrutador, involucrando de manera implícita al conmutador de la subred 1.

El segundo salto, correspondiente al enlace entre el enrutador y el conmutador-enrutador nos permitió observar el comportamiento de las colas en la interfaz de salida del enrutador (donde se aplicaron los mecanismos de encolamiento del Factor A) y la interfaz de entrada del conmutador-enrutador.

El último salto, correspondiente a la conectividad entre el conmutador-enrutador y el computador, nos permitió observar el comportamiento de las colas de la interfaz de salida del conmutador-enrutador y el computador.

5.3.1.3 Procedimiento.

Al igual que el experimento anterior, este experimento se dividió en dos bloques. El primero consistió en la realización de 9 pruebas que abarcaron la combinación de todos los niveles de los dos factores involucrados en este experimento sin realizar ninguna configuración adicional en el conmutador de la subred 1; las configuraciones de este bloque se realizaron en el enrutador (Factor A) y en los computadores (Factor B).

El segundo bloque consistió en la realización de las mismas 9 pruebas, pero ahora se priorizaron los flujos de prueba tanto en el conmutador de la subred 1 como en el enrutador. Al igual que en el experimento anterior, la *prioritización en el conmutador* fue una prioritización estricta.

Las mediciones del tiempo en colas y el tamaño de la cola de espera fue realizada utilizando la herramienta Pathchar desarrollada por Van Jacobson [45]; es muy recomendable leer el Apéndice D donde se explica a detalle como funciona esta herramienta y como se aplica a nuestro experimento.

Para este experimento solamente se realizó una replica de cada prueba, esto debido a el tiempo que tarda en realizarse cada prueba es muy grande (45 minutos aproximadamente por prueba), por lo que los resultados de este experimento son meramente ilustrativos.

El computador Generador de Prueba Pathchar fue el encargado de realizar la prueba hacia el computador Receptor de Prueba Pathchar; al igual que en los demás experimentos se generó la carga de tráfico utilizando la herramienta Traffic Generator versión 2, y se midieron los paquetes perdidos por el generador de tráfico con la herramienta IPTraf.

Las pruebas se realizaron primeramente sin prioritización en el conmutador de salida, para después realizar el mismo conjunto de pruebas priorizando los flujos de paquetes de prueba en el conmutador. Para cada una de las pruebas se guardaron los resultados de los parámetros medidos, así como los paquetes perdidos por el generador de tráfico para su posterior análisis.

Las configuraciones de los equipos de comunicaciones, así como los scripts de las herramientas utilizadas en este experimento pueden ser consultadas a detalle en el Apéndice A.

5.3.1.4 Factores No Controlados.

Al igual que en los experimentos anteriores, las colisiones provocadas por el método de acceso al medio en Ethernet no estuvo controlado, por lo que

también en este experimento existió la probabilidad de que hubiera paquetes perdidos ocasionados por este fenómeno.

5.3.2 Resultados del Experimento C.

Los resultados de este experimento los dividiremos en dos secciones, una para cada bloque de pruebas que se realizaron: Sin prioritización en el conmutador y con prioritización en el conmutador. Como ya se había mencionado, los resultados aquí mostrados son únicamente con fines demostrativos, ya que solamente se realizó una réplica de cada prueba, por lo que los comentarios que acompañan a los resultados son solamente ilustrativos.

5.3.2.1 Sin Prioritización en el Conmutador.

En la tabla XXVIII se pueden observar los resultados obtenidos para estas pruebas sin priorizar los flujos de prueba en el conmutador. Como era de esperarse, el tiempo y tamaño de las colas con carga de tráfico de 7.5 Mbps fue cero para los tres saltos para los tres mecanismos de encolamiento utilizados.

TABLA XXVIII.

TIEMPO Y TAMAÑO EN COLA DE LOS PAQUETES EN CADA SALTO DEL ESQUEMA DEL EXPERIMENTO "C" PARA LAS DIFERENTESCOMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y CARGA DE TRÁFICO SIN PRIORIZACION EN EL CONMUTADOR.

				Método de encolamiento					
				FIFO		PQ		WFQ	
				Tiempo (ms)	Tamaño (KBytes)	Tiempo (ms)	Tamaño (KBytes)	Tiempo (ms)	Tamaño (KBytes)
Carga	7.5 Mbps	Salto	1	0	0	0	0	0	0
			2	0	0	0	0	0	0
			3	0	0	0	0	0	0
	10 Mbps	Salto	1	52.5	1100	52.5	363	52.8	454
			2	45.7	43.7	41.4	38.4	40.9	36.9
			3	47.6	52.1	50.2	59	49.6	54.8
	12.5 Mbps	SaltO	1	53	1200	53.3	379	53.2	373
			2	51.3	48.4	26.7	24.8	38	34.8
			3	44.8	49	39.6	46.6	41.3	46.7

Vemos que los tiempos en cola cuando hay una carga de tráfico de 10 Mbps (100%) son similares en el salto 1, sin embargo para el salto 2 si se nota una diferencia de tiempos cuando se utilizaron los mecanismos de encolamiento PQ y WFQ con respecto a FIFO. Al observar los resultados obtenidos del tamaño en cola vemos que en el primer salto es sumamente notoria la mejoría utilizando PQ con respecto a los otros dos mecanismos de encolamiento, donde con FIFO se obtiene un tamaño en cola sumamente

elevado; en los saltos 2 y 3 las diferencias de los resultados entre los diferentes mecanismos de encolamiento son mínimas.

Cuando hay una carga de tráfico de 12.5 Mbps observamos que los tiempos de respuesta obtenidos en el primer y tercer salto son similares entre los diferentes mecanismos de encolamiento; en el segundo salto la diferencia de los tiempos en cola entre los mecanismos de encolamiento se hace notoria, donde utilizando el mecanismo PQ se obtienen los menores tiempos.

La tabla XXIX muestra los resultados para la tasa de paquetes perdidos por el generador de tráfico.

TABLA XXIX.

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y CARGA DE TRÁFICO DEL EXPERIMENTO "C" SIN PRIORITIZACIÓN EN EL CONMUTADOR.

		Método de encolamiento		
		FIFO	PQ	WFQ
		Porcentaje Paquetes Perdidos	Porcentaje Paquetes Perdidos	Porcentaje Paquetes Perdidos
Carga	7.5 Mbps	1.105	1.302	1.282
	10 Mbps	12.572	8.774	9.782
	12.5 Mbps	30.656	38.348	33.273

5.3.2.2 Con Prioritización en el Conmutador

Los resultados de las pruebas realizadas priorizando los flujos de prueba en el conmutador se muestran en la tabla XXX. Al haber una carga de tráfico de 7.5 Mbps los resultados del tiempo en cola y el tamaño en cola fueron cero.

TABLA XXX.

TIEMPO Y TAMAÑO EN COLA DE LOS PAQUETES EN CADA SALTO DEL ESQUEMA DEL EXPERIMENTO "C" PARA LAS DIFERENTESCOMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y CARGA DE TRÁFICO PRIORIZANDO EL CONMUTADOR

				Método de encolamiento					
				FIFO		PQ		WFQ	
				Tiempo (ms)	Tamaño (KBytes)	Tiempo (ms)	Tamaño (KBytes)	Tiempo (ms)	Tamaño (KBytes)
Carga	7.5 Mbps	Salto	1	0	0	0	0	0	0
			2	0	0	0	0	0	0
			3	0	0	0	0	0	0
	10 Mbps	Salto	1	3.45	49.2	7.35	50.8	6.95	8.67
			2	6.76	4.93	6.91	6.09	7.64	19.1
			3	7.76	9.28	7.98	9.37	7.94	8.74
	12.5 Mbps	Salto	1	6.99	50.2	6.26	30.1	6.5	25.9
			2	7.7	7.16	7.63	6.92	7.51	7.97
			3	7.72	8.86	7.8	9.03	7.78	8.47

Para cargas de tráfico de 10 Mbps observamos que los tiempos en cola en el salto 1 se redujeron considerablemente, comportándose de manera similar para los diferentes mecanismos de encolamiento durante los tres saltos, salvo para el primer salto donde utilizando FIFO se obtuvo un menor tiempo en cola. Los resultados para el tamaño de la cola durante los tres saltos muestran un comportamiento estable cuando se utilizaron los mecanismos de encolamiento FIFO y PQ, mientras que utilizando el mecanismo WFQ el comportamiento fue un tanto inestable.

Los resultados obtenidos para cargas de tráfico de 12.5 Mbps muestran un comportamiento estable para los tiempos en cola en los tres saltos. El comportamiento del tamaño de la cola muestra una ligera ventaja en el primer salto al utilizar el mecanismo de encolamiento WFQ; para el salto dos y tres el comportamiento fue similar para los diferentes mecanismos de encolamiento.

La tabla XXXI muestra una los resultados obtenidos para la tasa de pérdida de paquetes por el generador de tráfico al realizar estas pruebas, donde se ve que al utilizar el mecanismo de encolamiento PQ la tasa de paquetes perdidos se incrementó considerablemente con respecto a los otros mecanismos de encolamiento.

TABLA XXXI.

PORCENTAJE DE PAQUETES PERDIDOS POR EL GENERADOR DE TRÁFICO PARA LAS DIFERENTES COMBINACIONES DE MECANISMOS DE ENCOLAMIENTO Y CARGA DE TRÁFICO DEL EXPERIMENTO "C" PRIORIZANDO EL CONMUTADOR.

		Método de encolamiento		
		FIFO	PQ	WFQ
		Porcentaje Paquetes Perdidos	Porcentaje Paquetes Perdidos	Porcentaje Paquetes Perdidos
Carga	7.5 Mbps	0.929	1.199	1.232
	10 Mbps	9.834	9.766	13.582
	12.5 Mbps	33.003	32.776	33.022

5.3.2.3 Comparando los resultados

En esta sección realizaremos un comparativo de los resultados obtenidos tanto sin priorizar los flujos de prueba en el conmutador como priorizandolos. Para esto nos basaremos en las gráficas de los resultados mostrados anteriormente. Para carga de tráfico de 7.5 Mbps no realizaremos ninguna comparación, ya que en ambas pruebas los resultados tanto del tiempo y tamaño en cola fueron cero.

En las figuras 53 y 54 podemos observar que cuando hay una carga de tráfico en la red de 10 Mbps entre el enlace que une los conmutadores y el enrutador, el tiempo en cola de los paquetes es significativamente menor cuando se priorizan los flujos de prueba en el conmutador de salida, esto ocurre para los tres saltos que da el paquete en todo el trayecto de comunicación.

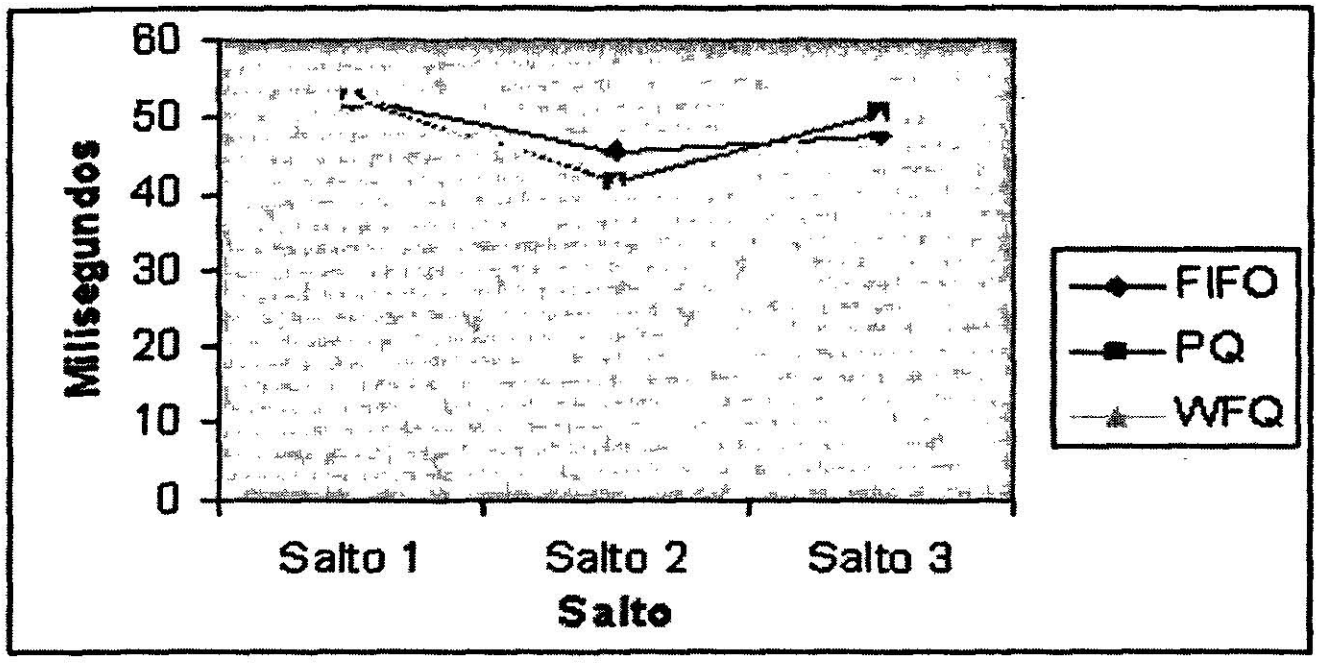


Figura 53. Comparativo del tiempo en cola por salto entre los distintos mecanismos de encolamiento con una carga de tráfico de 10 Mbps y sin prioritización en el conmutador.

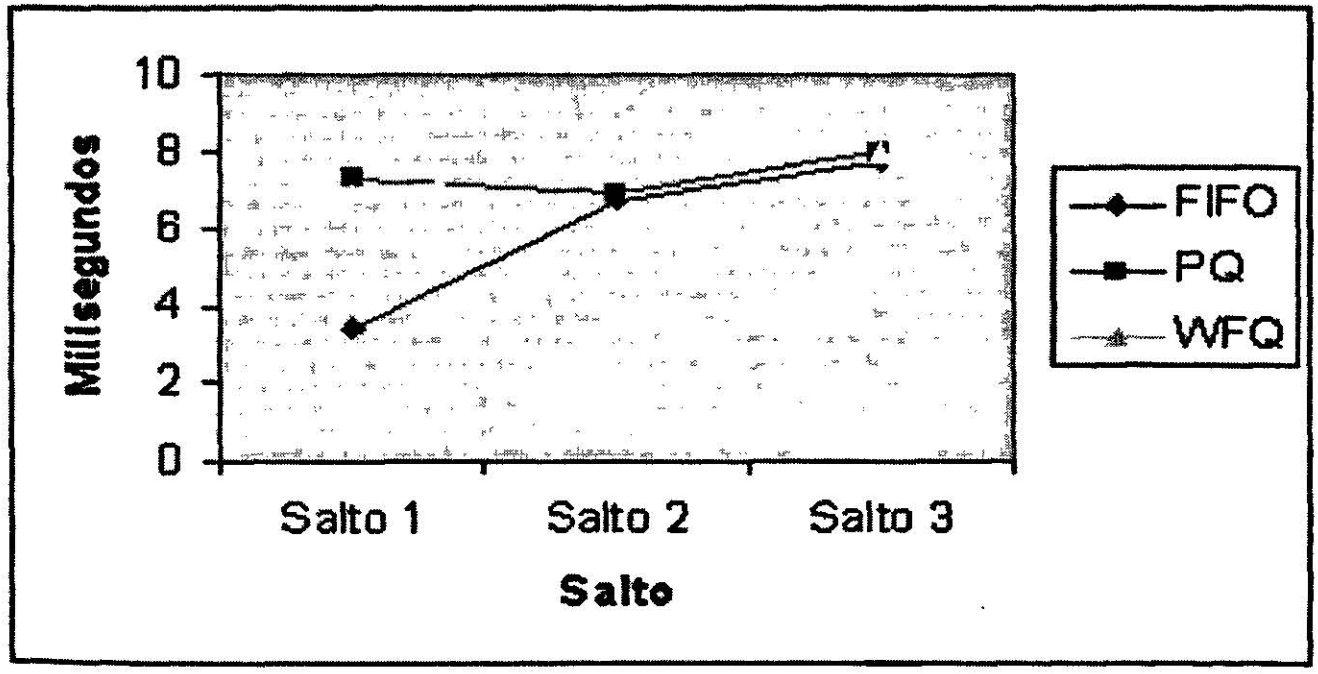


Figura 54. Comparativo del tiempo en cola por salto entre los distintos mecanismos de encolamiento con una carga de tráfico de 10 Mbps priorizando el conmutador.

El tamaño en cola cuando hay una carga de 10 Mbps, es mucho mas bajo en el salto 1 cuando se priorizan los flujos en el conmutador de salida; en los resultados del salto 1 es donde afecta la variación de configuraciones del conmutador de salida. En los saltos dos y tres la disminución del tamaño de cola al priorizar los flujos son también muy significativos, como se muestra en las figuras 55 y 56.

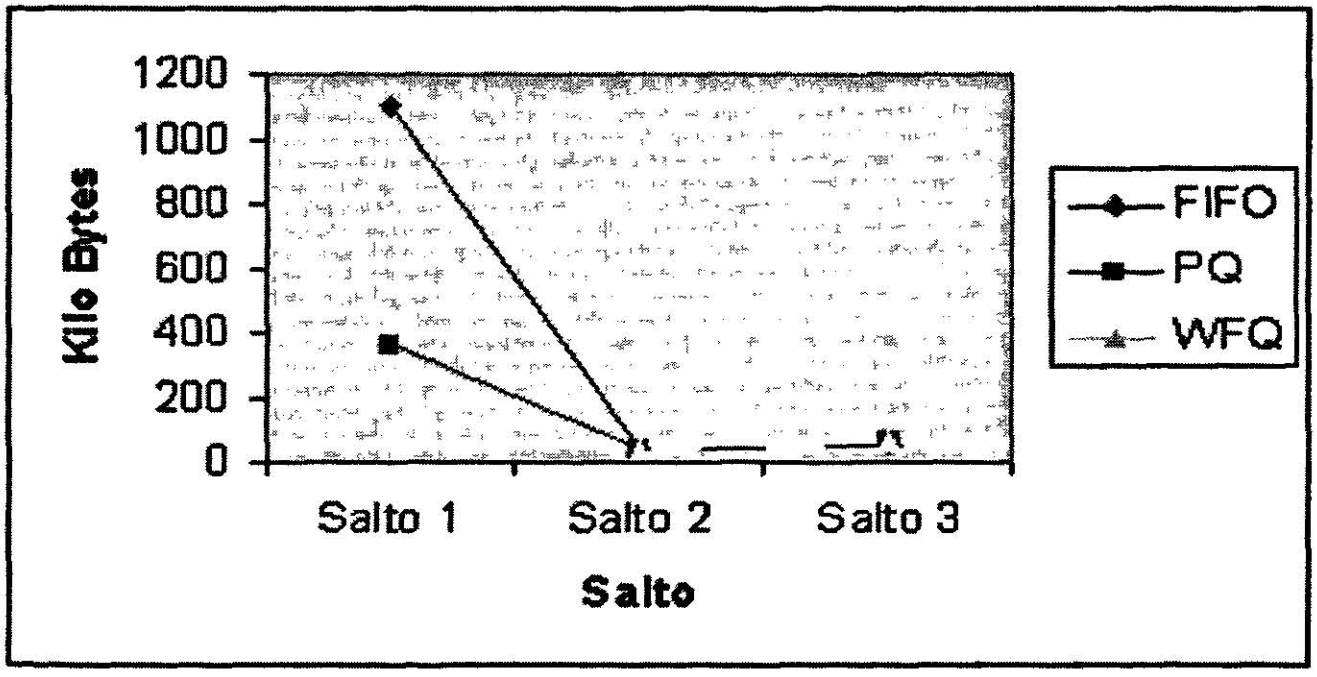


Figura 55. Comparativo del tamaño de la cola por salto entre los distintos mecanismos de encolamiento con una carga de tráfico de 10 Mbps y sin prioritización en el conmutador.

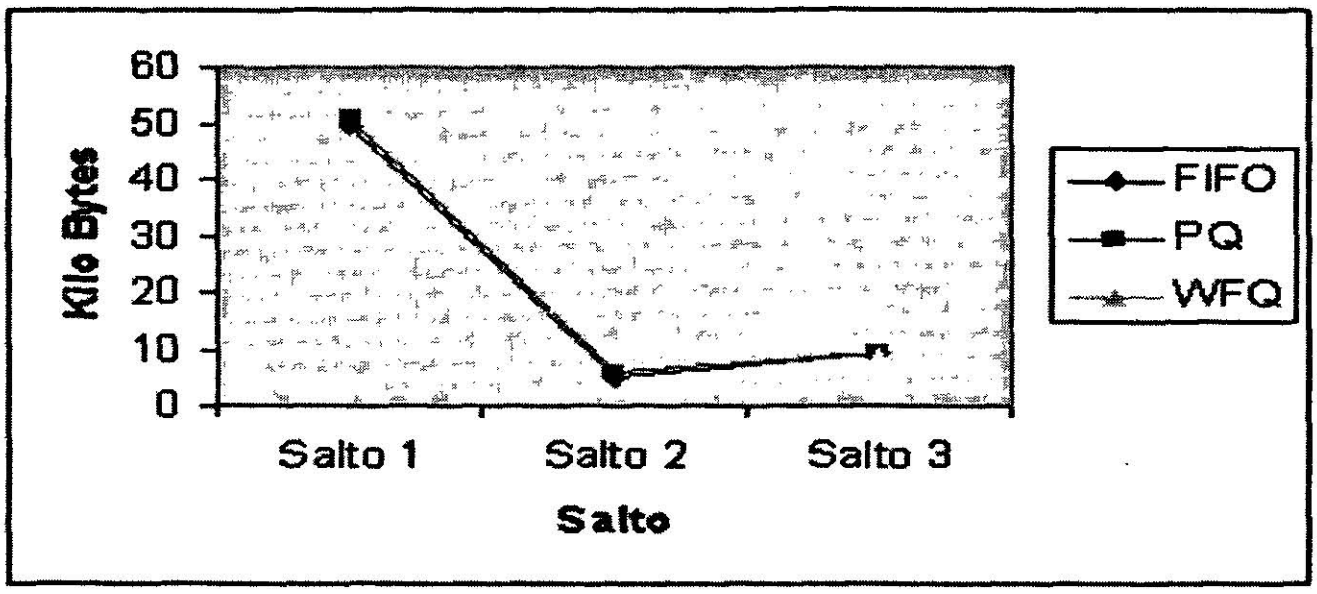


Figura 56. Comparativo del tamaño de la cola por salto entre los distintos mecanismos de encolamiento con una carga de tráfico de 10 Mbps priorizando el conmutador.

Como se muestra en las figuras 57 y 58, las tasas de paquetes perdidos para ambas pruebas no tienen una variación muy significativa.

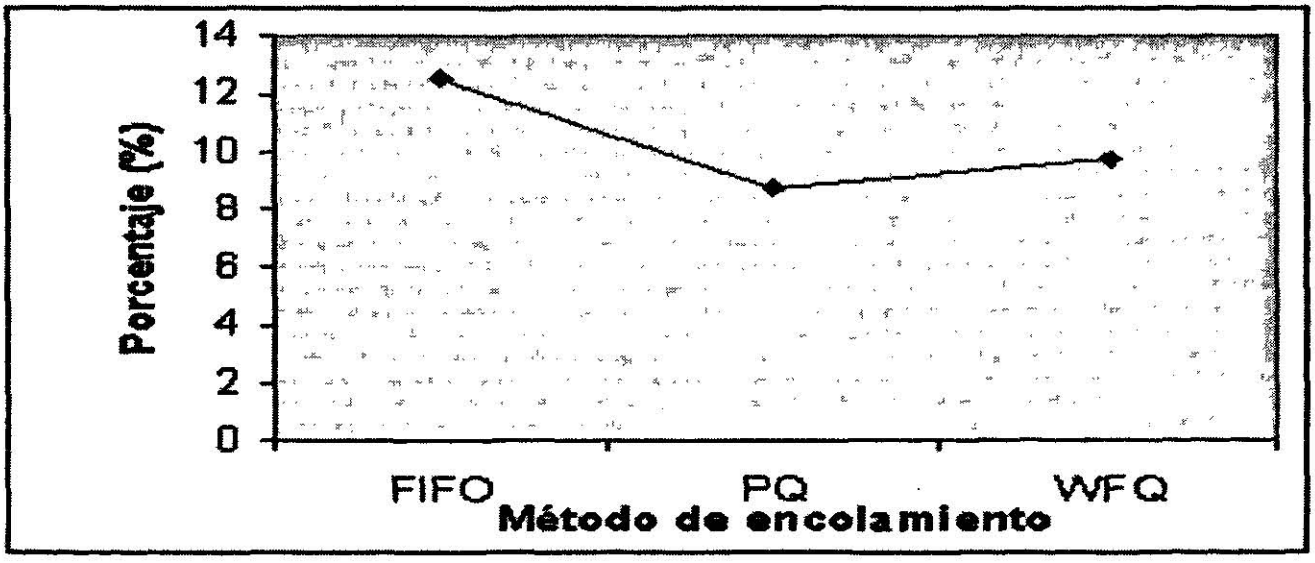


Figura 57. Comparativo del porcentaje de paquetes perdidos por el generador de tráfico entre los distintos mecanismos de encolamiento del Experimento C con una carga de tráfico de 10 Mbps y sin prioritización en el conmutador.

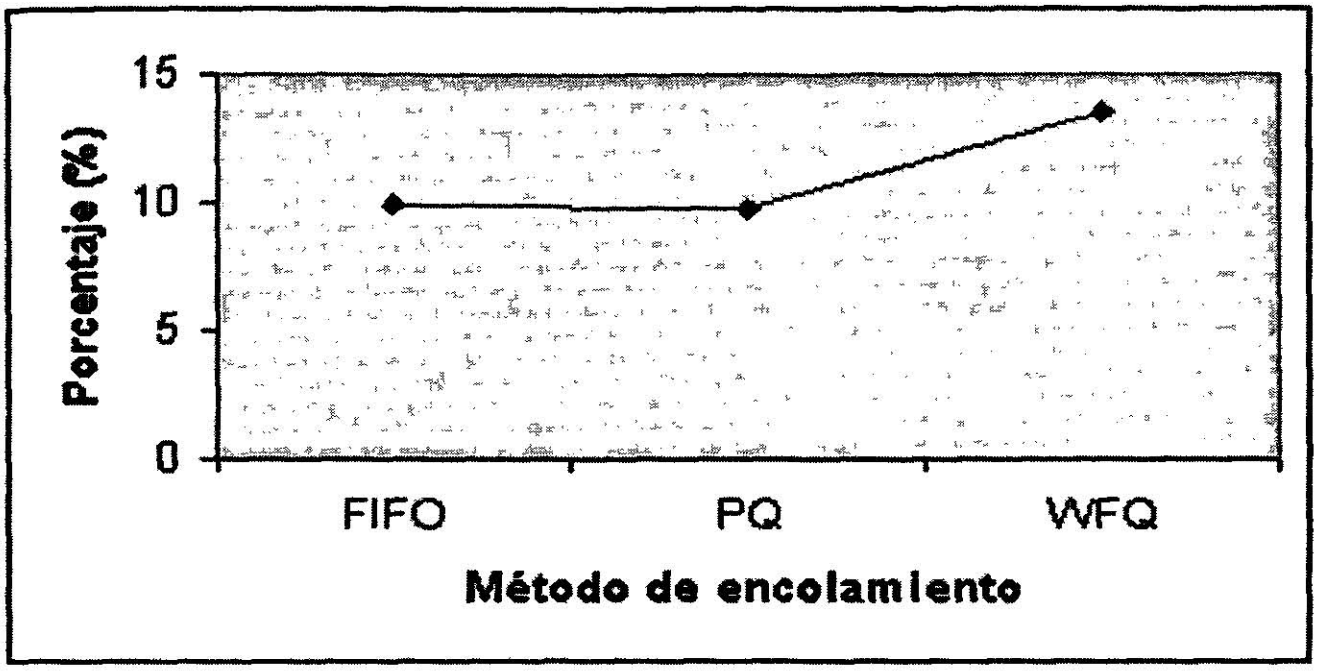


Figura 58. Comparativo del porcentaje de paquetes perdidos por el generador de tráfico entre los distintos mecanismos de encolamiento del Experimento C con una carga de tráfico de 10 Mbps priorizando el conmutador.

El comparativo de resultados con carga de tráfico en la red de 12.5 Mbps es muy similar al de 10 Mbps, donde se notan mejoras sustanciales al priorizar los flujos de prueba en el conmutador de salida. Las figuras 59 a la 62 muestran las gráficas de resultados.

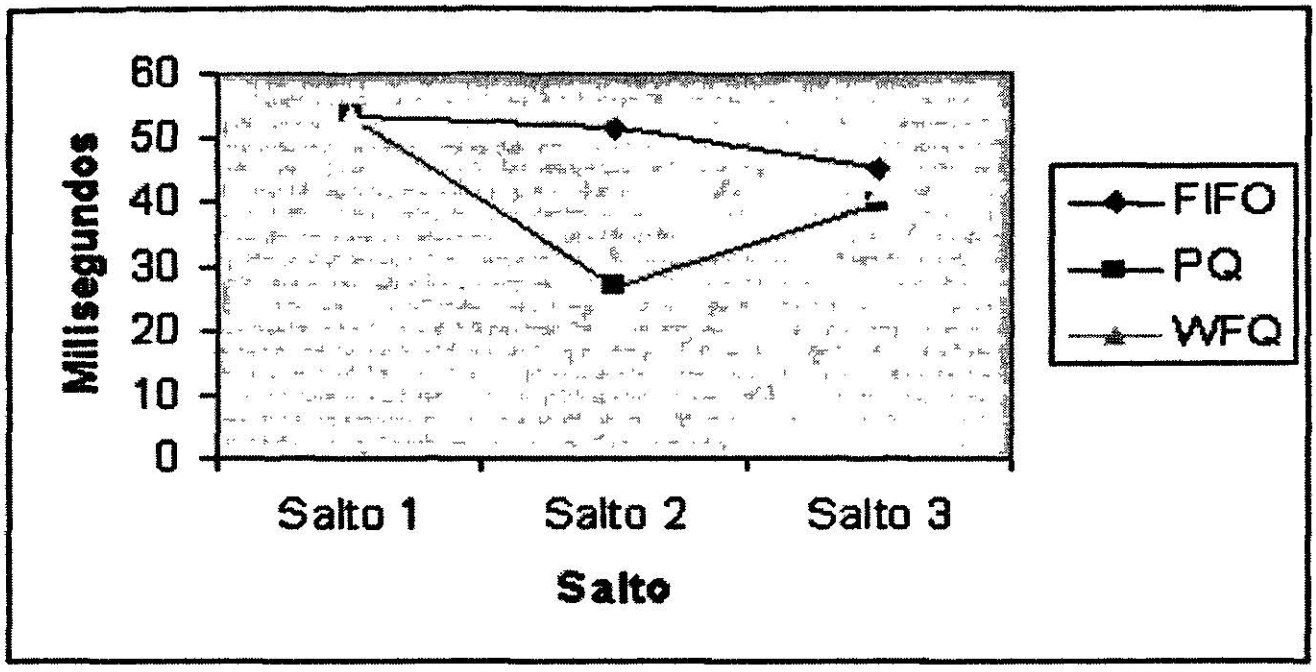


Figura 59. Comparativo del tiempo en cola por salto entre los distintos mecanismos de encolamiento con una carga de tráfico de 12.5 Mbps y sin prioritización en el conmutador.

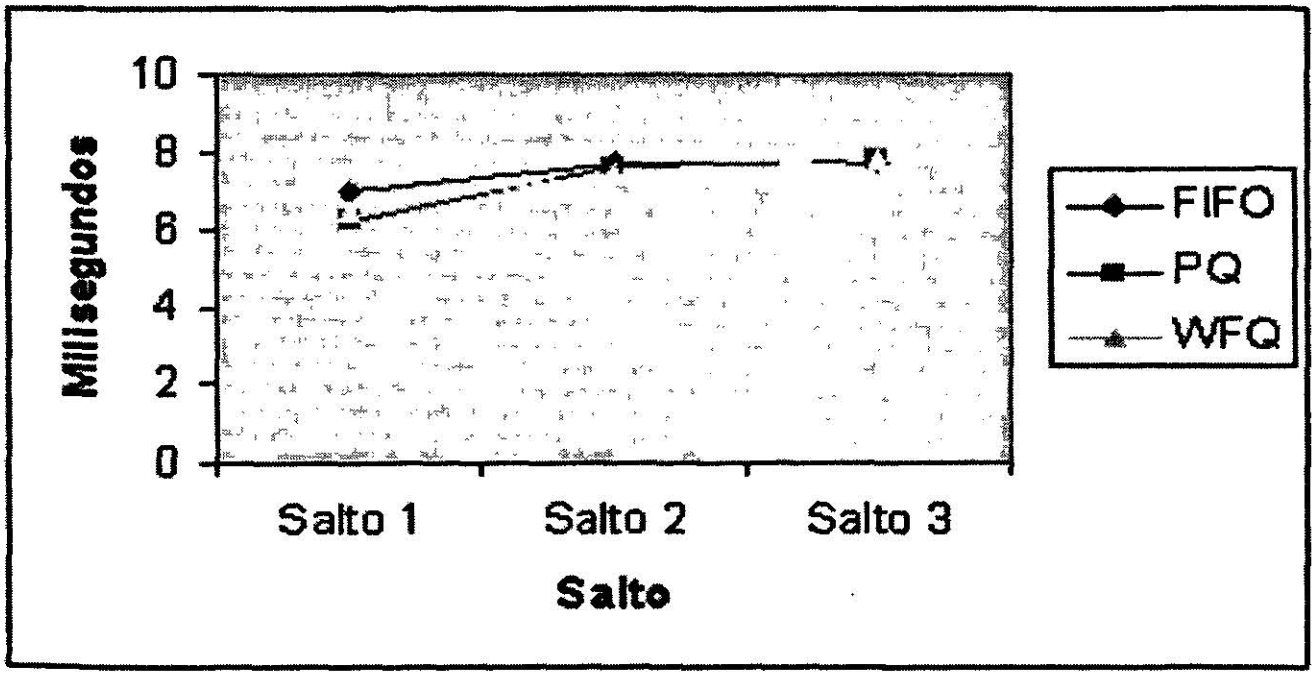


Figura 60. Comparativo del tiempo en cola por salto entre los distintos mecanismos de encolamiento con una carga de tráfico de 12.5 Mbps priorizando el conmutador.

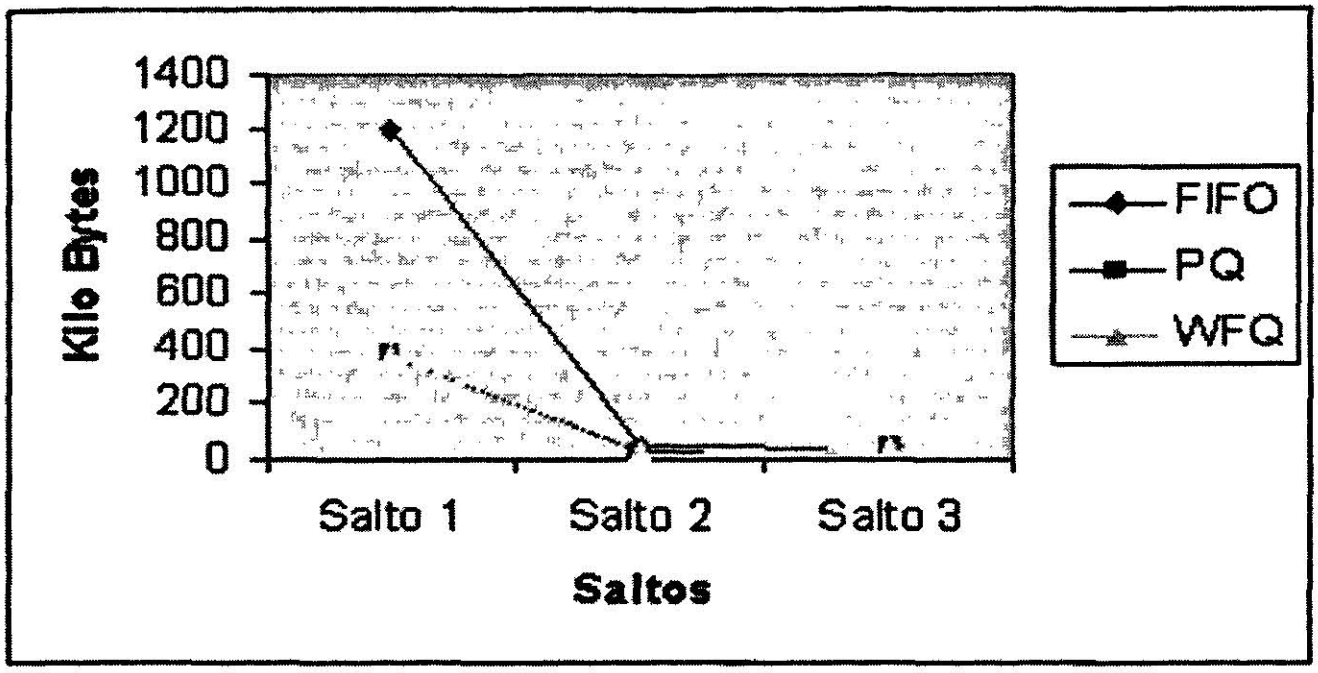


Figura 61. Comparativo del tamaño de la cola por salto entre los distintos mecanismos de encolamiento con una carga de tráfico de 12.5 Mbps y sin prioritización en el conmutador.

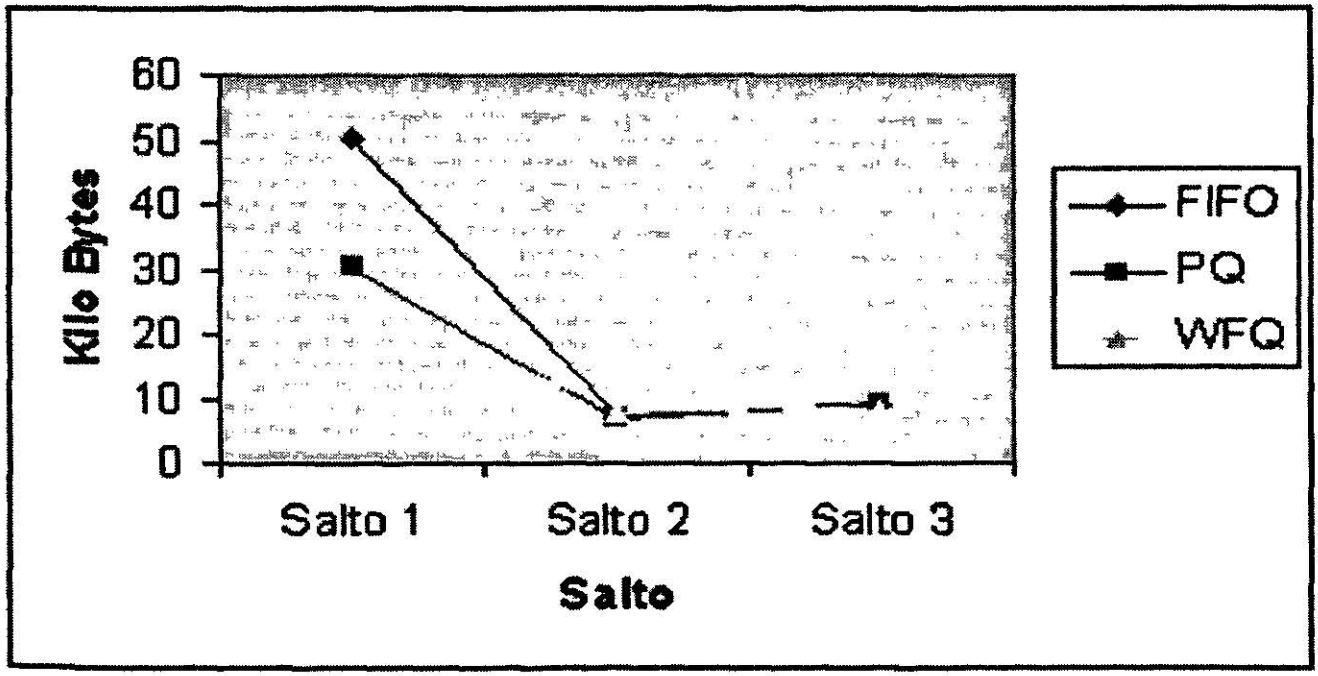


Figura 62. Comparativo del tamaño de la cola por salto entre los distintos mecanismos de encolamiento con una carga de tráfico de 12.5 Mbps priorizando el conmutador.

Las figuras 63 y 64, muestran la tasa de pérdida de paquetes por el generador de tráfico con una carga de 12.5 Mbps, la cual sí se vio afectada significativamente al priorizar los flujos de prueba en el conmutador.

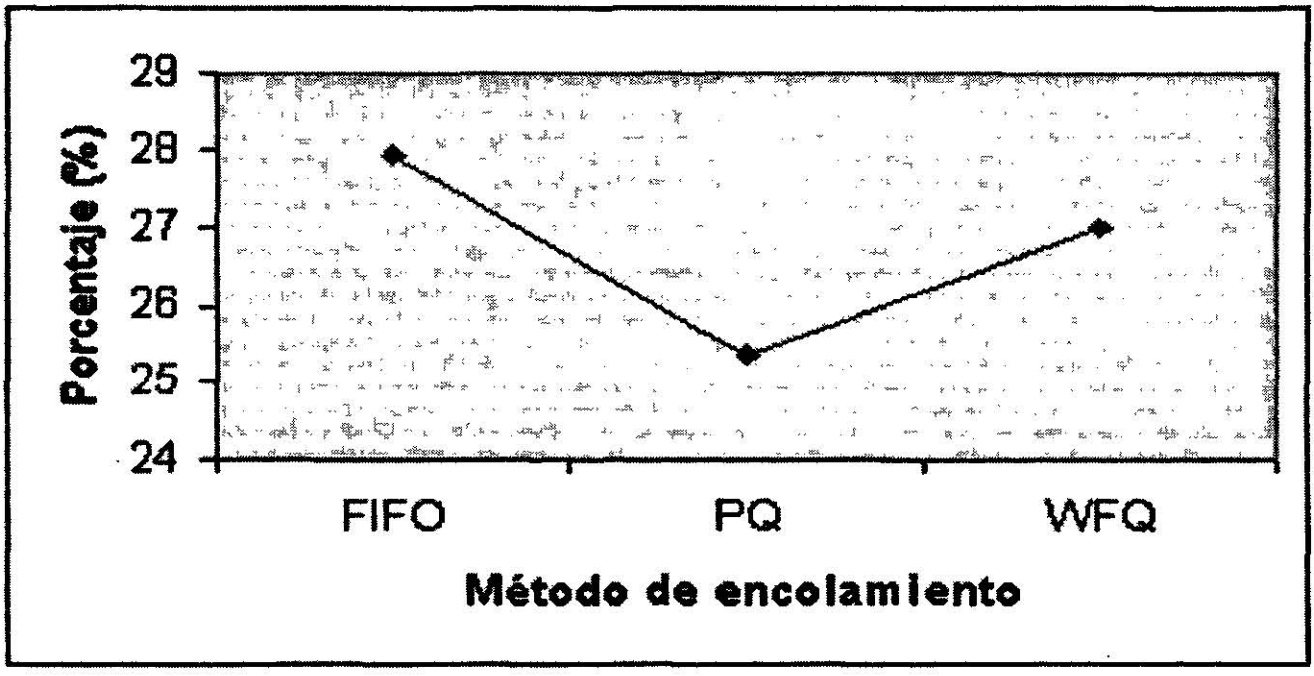


Figura 63. Comparativo del porcentaje de paquetes perdidos por el generador de tráfico entre los distintos mecanismos de encolamiento del Experimento C con una carga de tráfico de 12.5 Mbps y sin prioritización en el conmutador.

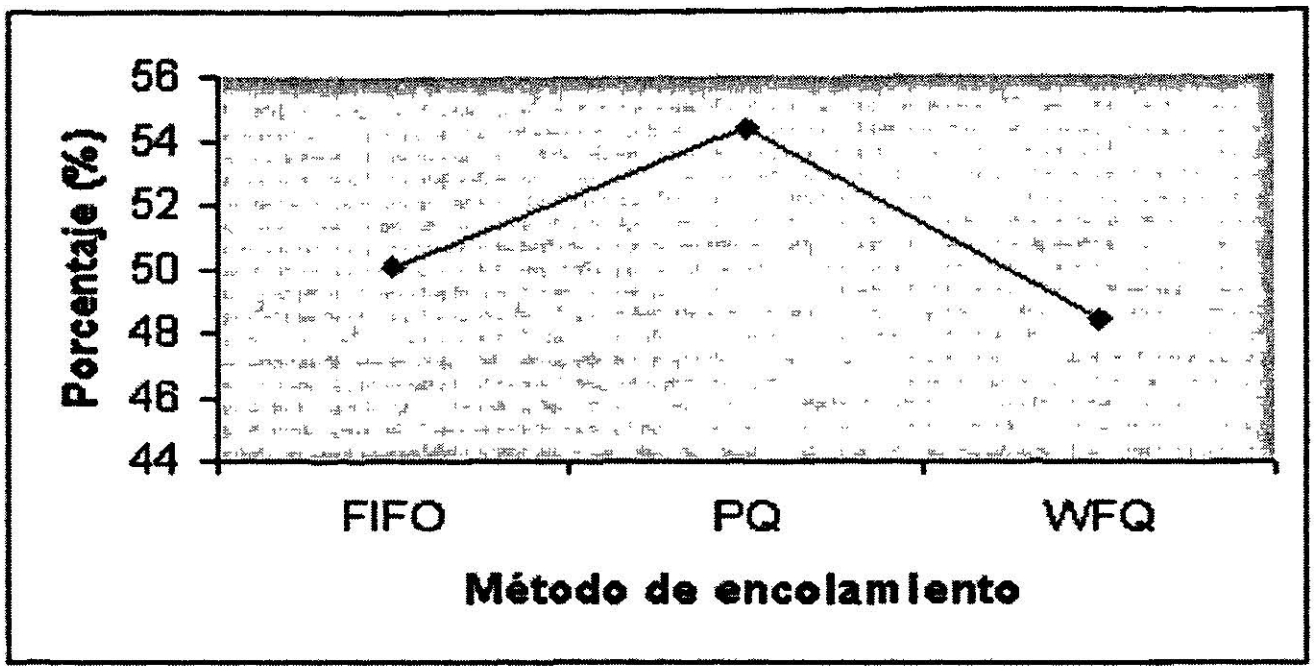


Figura 64. Comparativo del porcentaje de paquetes perdidos por el generador de tráfico entre los distintos mecanismos de encolamiento del Experimento C con una carga de tráfico de 12.5 Mbps priorizando el conmutador.

5.4 Conclusiones

En este capítulo se mostraron los experimentos realizados para la medición de parámetros de Calidad de Servicio; se mostró la descripción, los factores que intervinieron en el experimento, el diagrama de conectividad del experimento, el procedimiento y los factores no controlados para cada uno de los tres experimentos. Así mismo también se mostraron los resultados de cada uno de los experimentos, haciéndose comentarios acerca de estos resultados.

En el capítulo siguiente se mostrará una discusión de los resultados mostrados en este capítulo, así como las conclusiones generales del trabajo.

CAPITULO 6

CONCLUSIONES

En este capítulo veremos una discusión de los resultados mostrados en el capítulo anterior, posteriormente se muestran las conclusiones generales obtenidas en la realización de este trabajo, para finalmente ver los trabajos futuros que permitan enriquecer más los esfuerzos y trabajos sobre este tema.

6.1 Discusión de resultados

Como se vio en el capítulo anterior, se realizaron tres experimentos para medir parámetros que nos ayudan a determinar la Calidad del Servicio ofrecida. En el primer experimento se demostró a través de un análisis de varianza que al utilizar diferentes mecanismos de encolamiento en los equipos de comunicaciones el tiempo que tardan en transmitirse los paquetes entre el emisor y el receptor sí se ve afectado.

Los resultados de este experimento son útiles, ya que hoy en día muchas aplicaciones que usamos comúnmente son sensibles al tiempo, tales como las videoconferencias, la voz sobre redes IP, etcétera; estos servicios requieren que el tiempo de transferencia sea mínimo y además que se minimice la variación en el retardo, por lo que tener una medida cuantitativa del

comportamiento de los tiempos de respuesta sobre diferentes escenarios puede servirnos para tomar mejores decisiones en la implementación de los servicios mencionados.

Sin embargo, algunas otras aplicaciones no son sensibles al retardo, pero si les afecta el caudal eficaz o la tasa de paquetes perdidos, como por ejemplo la transferencia de archivos, la mensajería instantánea, etcétera, donde tener parámetros de medición bien definidos y datos cuantitativos de estos parámetros nos pueden ser de mucha utilidad en la implementación de estos servicios. En el experimento B se mostró una forma de medir y cuantificar parámetros del caudal eficaz, tasa de pérdida de paquetes y variación en el retardo sobre diferentes escenarios y utilizando diferentes algoritmos de encolamiento.

El realizar mediciones extremo a extremo entre el emisor y el receptor no nos permiten medir lo que sucede en cada uno de los elementos de la ruta de comunicaciones, lo cual para fines prácticos pudiera ser irrelevante, sin embargo cuando tenemos problemas de rendimiento en nuestra red es de suma importancia verificar el comportamiento de nuestros flujos de paquetes en cada uno de los puntos del trayecto. Los resultados del experimento C muestran el comportamiento en las colas de los equipos de comunicaciones para cada salto que dieron los paquetes de los flujos de prueba, en donde pudimos observar que al utilizar distintos mecanismos de encolamiento el tiempo y tamaño de la cola se ve afectado, llegando a ser en algunos casos muy significativo.

En los experimentos B y C se realizaron dos bloques de pruebas en cada experimento, ya que al momento de realizar las pruebas se obtenían resultados un tanto fuera de los esperados, y se comprobó que este comportamiento era debido a que en ciertas pruebas el conmutador de salida influía enormemente en los resultados obtenidos, por ello se decidió realizar las mismas pruebas e iteraciones en dos escenarios, uno sin priorizar los flujos de prueba en el conmutador de salida, y el otro priorizado los flujos de prueba en el conmutador.

Los resultados comprueban que al priorizar los flujos de prueba en el conmutador se obtienen mejores resultados en los parámetros medidos para estos flujos en comparación de los resultados obtenidos sin realizar la prioritización de los flujos en el conmutador.

6.2 Conclusiones

Este trabajo abordó el concepto de Calidad de Servicio, sus ventajas, escenarios de implementación, las diferentes tecnologías para su implementación; también se habló sobre la importancia de las colas de espera en los equipos de comunicaciones y los diferentes mecanismos de encolamiento que pueden ser utilizados, mostrando las ventajas y desventajas de cada uno de ellos. Se mostró la importancia de medir y cuantificar las implementaciones de Calidad de Servicio a través de una sección experimental. Al inicio de este trabajo se realizó una breve historia de las redes de comunicaciones, así como de los modelos de comunicaciones con el objetivo

de realizar una introducción y lograr una mejor comprensión de los temas tratados posteriormente. Se obtuvieron varias conclusiones tanto en la parte teórica como en la experimental, las cuales se muestran a continuación.

La Calidad de Servicio es útil solamente sobre ciertos escenarios, es decir, la implementación de la Calidad de Servicio no nos va a resolver todos nuestros problemas, ya que como se mencionaba en el capítulo 3 no es una varita mágica; como se mostró en los distintos escenarios explicados en la sección 3.2 en algunos casos el aplicar Calidad de Servicio no tiene sentido, y esto se demostró en la parte experimental, al ver como bajo ciertos escenarios el aplicar Calidad de Servicio no se obtenían mejoras sustanciales en los parámetros medidos.

Existen diferentes técnicas y tecnologías para la implementación de la Calidad de Servicio, cada una con sus ventajas y desventajas como se vieron en el capítulo 3; las técnicas existentes abarcan desde la implementación de Calidad de Servicio en las capas inferiores del modelo de comunicación como es la capa de enlace, hasta implementaciones en las capas de red, transporte y aplicación. El factor de decisión para seleccionar una determinada tecnología de implementación de Calidad de Servicio dependen principalmente de los requerimientos tanto de los usuarios como de las aplicaciones y servicios que se ejecutan en la red, además de la infraestructura con la que se cuenta para llevar a cabo la implementación.

En los equipos de comunicaciones se forman colas de espera cuando la entrada de datos sobrepasa la capacidad de atención del dispositivo. Estas colas de espera juegan un papel fundamental, ya que entre mayor tiempo pasen los paquetes en la cola de espera, mayor será el tiempo total de comunicación. Existen diferentes mecanismos de encolamiento basados en algoritmos que van desde los mas simples hasta los sumamente complejos. Cada mecanismo de encolamiento tiene sus ventajas y desventajas, así como escenarios dónde es más recomendable aplicar ese mecanismo en particular, como se vio en el capítulo 4. La elección del mecanismo de encolamiento a utilizar depende de lo que se quiera lograr, ya que mientras mecanismos como el Encolamiento Priorizado brinda un método para asegurar que los flujos o paquetes priorizados se atiendan primero en la cola, esto hace que los flujos que no son priorizados se vean perjudicados.

Como se mostró en la figura 27 del capítulo 4, en un modelo de comunicaciones extremo a extremo, el tiempo total que tardan los datos en ir de una aplicación del usuario emisor a la aplicación del usuario destino será la suma de tiempos que tardan los datos en atravesar toda la ruta de comunicaciones, por lo que entre mas equipos de comunicaciones haya en esa ruta, el tiempo total de transmisión será mayor. Por este motivo, el aplicar distintos mecanismos de encolamiento en los equipos de comunicaciones puede ayudar a disminuir el tiempo que tardan los paquetes en las colas de espera, obteniendo como consecuencia una disminución en el tiempo total extremo a extremo.

Como se vio en el capítulo 4, si el tiempo total o retardo disminuye, la variación en el retardo también disminuye, logrando beneficios para aplicaciones isócronas o en tiempo real como son la videoconferencia o Voz sobre IP. Debido a la ecuación para calcular el caudal eficaz, éste es inversamente proporcional a el retardo, por lo que sí el retardo disminuye el caudal eficaz se incrementa; así mismo, se reduce la probabilidad de paquetes perdidos debido a tiempos de espera agotados, por lo que la tasa de paquetes perdidos disminuirá, beneficiando enormemente a aplicaciones tales como las transferencias de archivos, servicios www, acceso remoto a bases de datos, etcétera.

Al realizar la parte experimental de este trabajo se obtuvieron varias conclusiones, algunas de ellas refuerzan las afirmaciones teóricas, como lo descrito en los párrafos anteriores sobre la relación del retardo con los demás parámetros medidos.

Otro punto importante es que la implementación de Calidad de Servicio no sólo debe realizarse en el núcleo de nuestra red (también llamado "core" o "backbone") sino que debe implementarse en todos los equipos de comunicaciones en los que se tenga el control. Esto es importante, ya que como se observó en el modelo experimental de este trabajo, existen enormes diferencias en los resultados obtenidos cuando se aplica la Calidad de Servicio a todos los equipos de comunicaciones en comparación a cuando solo se aplica en los enrutadores principales.

Como se vio en la parte experimental, es de suma importancia tener implementados mecanismos que nos permitan medir y recibir retroalimentación del comportamiento de los esquemas implementados, para poder cuantificar las mediciones de los parámetros y obtener datos que nos permitan verificar si realmente las implementaciones de Calidad de Servicio están funcionando en base a lo esperado. Si no realizamos mediciones, nunca podremos determinar de manera cuantitativa el comportamiento de nuestra red de comunicaciones.

Podemos concluir finalmente, que la implementación de la Calidad de Servicio requiere primeramente un arduo análisis sobre los requerimientos de los usuarios, además de las aplicaciones y servicios que se ejecutan en la red, para después de tener toda la información de los requerimientos desarrollar una estrategia de solución basada en las diferentes técnicas y tecnologías de implementación de Calidad de Servicio, aunado a una buena elección de los mecanismos de encolamiento a utilizar que satisfagan los requerimientos de uso de la red, para finalmente, una vez que la solución de Calidad de Servicio haya sido implementada, trabajar en el desarrollo e implementación de un esquema de medición y cuantificación de los parámetros que pueden determinarnos si la estrategia de Calidad de Servicio implementada está satisfaciendo los requerimientos.

6.3 Trabajos futuros

Al trabajar en la parte experimental de este trabajo noté la necesidad de poder medir la caracterización de las rutas de comunicaciones, lo cual nos permitiría observar y medir el comportamiento real de cada uno de los tramos del trayecto en la ruta de comunicaciones.

Para esto es necesario trabajar en la creación de un esquema que nos permita obtener estas mediciones de manera correcta, además de crear una herramienta de software la cual nos ayude en la realización de los cálculos en las mediciones de los parámetros para así obtener una caracterización completa del trayecto por el cual atraviesan nuestros paquetes.

BIBLIOGRAFIA

1. Comer, E. Douglas. *Redes Globales de Información con Internet y TCP/IP Principios Básicos, Protocolos y Arquitectura, Tercera Edición.* Ed. Pearson, 1996.

2. Stallings, William. *Comunicaciones y Redes de Computadores, Sexta Edición.* Ed. Prentice Hall, 2000.

3. Stallings, William. *Local and Metropolitan Area Networks, Sixth Edition.* Ed. Prentice Hall. 2000.

4. Walpole Ronald, Myers Raymond. *Probabilidad y Estadística, Cuarta Edición.* Ed. Mc Graw Hill, 2001.

REFERENCIAS

1. Zakon, R. Hobbes' *Internet Timeline*, RFC 2235 Noviembre 1997.

2. Resolución 269 del 107th Congreso de los Estados Unidos. http://www.popular-science.net/history/meucci_congress_resolution.html

3. Stallings, William. *Local and Metropolitan Area Networks Sixth Edition*, Prentice Hall 1999.

4. Stallings, William. *The Origins of OSI*, http://williamstallings.com/Extras/OSI.html

5. Diccionario de la Real Academia Española 2001, http://www.rea.es.

6. Crawley, E. et.al. *A Framework for QoS-based Routing in the Internet*, RFC 2386, Agosto 1998.

7. Bakin, Deborah S. *Application QoS Networking Infrastructure for High Performance, High Reliability Web Operations*, http://www.ewh.ieee.org/r2/baltimore/Chapter/Comm/dlt_talk/index.htm

8. QoS Forum. *QoS Protocols and Architectures*, http://www.qosforum.com, Julio 1999.

9. Blake, S. et.al. *Architecture for Differentiated Services*, RFC 2475, Diciembre 1998.

10. Postel, Jon. et.al. *Internet Protocol*, RFC 791, Septiembre1981.

11. Baker, *Requerimients for IP version 4 Routers*, RFC 1812, Junio 1995.

12. Deering, S. et.al. *Internet Protocol Version 6 (IPv6) Specification*, RFC 2460, Diciembre 1998.

13. Ek, Niclas. *IEEE 802.1p,q – QoS on the MAC level*,
Helsinki University of Technology. Abril 1999,
http://www.tml.hut.fi/Opinnot/Tik110.551/1999/papers/08IEEE802.1QosIn
MAC/qos.html

14. *Layer 2 Traffic Prioritization*,
Intel Corporation Press, 1999.

15. Almquist, P. *Type of* Service,
RFC 1349, Julio 1992.

16. Reynolds, J. y Postel, J. *Assigned Numbers*,
RFC 1060, Marzo 1990.

17. IETF Differentiated Services Work Group *Description of the Working Group*,
Septiembre 2002, http://www.ietf.org/html.charters/diffserv-charter.html

18. Blacke, S. et.al. *An Architecture for Differentiated Services*,
RFC 2475 , Diciembre 1998.

19. Nichols, K. et.al. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 headers*,
RFC 2474, Diciembre 1998.

20. *DiffServ – The Scalable End-to-End QoS Model*,
Cisco Systems White Papers, http://www.cisco.com

21. Jacobson, B. et.al. *An Expedited Forwarding PHB*,
RFC 2598, Junio 1999.

22. Davie, B. et.al. *An Expedited Forwarding PHB (Per Hop Behavior)*,
RFC 3246, Marzo 2002.

23. Heinanen, J. et.al. *Assured Forwarding PHB Group*,
RFC 2597, Junio 1999.

24. Grossman, D. *New Terminology and Clarifications for DiffServ*,
RFC 3260, Abril 2002.

25. Shenker, S. et.al. *Specification of Guaranteed Quality of Service*
RFC 2212, Septiembre 1997.

26. Wroclawski, J. *Specification of the Controlled-Load Network Element Service*,
RFC 2211, Septiembre 1997.

27. Rosen, E. et.al. *Multiprotocol Label Switching Architecture,*
RFC 3031, Enero 2001.

28. ATM Forum,
http://www.atmforum.com/

29. Campbell, Andrew T. *A Quality of Service Architecture,*
Tesis Doctoral, Universidad de Lancaster, Enero 1996.

30. Almes, G. et.al. *A One Way Delay Metric for IPPM,*
RFC 2679 Septiembre 1999.

31. Almes, G. et.al. *A Round Trip Delay Metric for IPPM,*
RFC 2681 Septiembre 1999.

32. Partridge, C. *Isochronous Applications Do Not Requiere Jitter-Conttroled Networks,*
RFC 1257 Septiembre 1991.

33. Schulzrinne, H. et.al. *RTP: A Transport Protocol for Real-Time Applications,*
RFC 1889 Enero 1996.

34. Almes, G. et.al. *A One-way Packet Loss Metric for IPPM,*
RFC 1889 Septiembre 1999.

35. *QoS Networking,*
Cisco Press, Abril 2000,
http://www.cisco.com/cpress/cc/td/cpress/fund/ith2nd/it2446.htm

36. Semeria, Chuck. *Supporting Differentiated Service Classes: Queue Sheduling Disciplines,*
Juniper Networks White Paper, Diciembre 2000.

37. *Weighted Fair Queuing Quick Reference,*
Scalable Networks White Paper, 2000,
http://www.scalable-networks.com

38. *Custom Queuing and Priority Output Queuing,*
Cisco White Papers 1999,
http://www.cisco.com/warp/public/cc/cisco/mkt/iworks/tech/cq_wp.htm

39. Bradner, S. McQuaid, J. *Benchmarking Methodology for Network Interconnect Devices,*
RFC 2544 Marzo 1999

40 Walpole. Myers. *Probabilidad y Estadística Cuarta Edición,* Mc Graw Hill, 1992.

41. Kessler, G. Shepard, S. *A primer On Internet an TCP/IP Tools,* RFC 1739 Diciembre 1994

42. Traffic Generator versión 2 http://www.postel.org/tg/tg.html

43. IPTraf versión 2.7.0 IP Network Monitoring Software http://iptraf.seul.org/

44. IPerf versión 1.6 http://dast.nlanr.net/Projects/Iperf/

45. Pathchar Van Jacobson ftp://ftp.ee.lbl.gov/pathchar

46. Nitzan, Becca y Nemeth Evy. *Pathchar notes,* http://www.caida.org/tools/utilities/others/pathchar/pathcharnotes.html

47. Downey, Allen. *Using pathchar to estimate Internet link characteristics,* Colby Collage, 1999.

48. Jacobson, Van. *Pathchar – a tool to infer characteristics of Internet paths,* Lawrence Berkeley Nacional Laboratory, Abril 1997.

49. Postel, J. *Internet Control Message Protocol,* RFC 792, Septiembre 1981.

APENDICES

APENDICE A

CONFIGURACIONES DE HARDWARE Y SOFTWARE PARA LOS EXPERIMENTOS

APENDICE A

CONFIGURACIONES PARA EL EXPERIMENTO A

Configuraciones de los equipos de comunicaciones y de los computadores utilizados en el experimento A.

Enrutador

Marca: Cabletron Smart Switch Router 2000
Sistema Operativo: 3.0.0.0

Velocidad de los puertos:

- Puerto 1.- 10Mbps/Full Duplex
- Puerto 2.- 10Mbps/Full Duplex

Configuración básica:

```
1 : port set et.1.1 speed 10mbps duplex full
2 : port set et.1.2 speed 10mbps duplex full
    !
3 : interface create ip subred1 address-netmask 10.0.1.1/24 port et.1.1 up
4 : interface create ip subred2 address-netmask 10.0.2.1/24 port et.1.2 up
    !
5 : ip add route 10.0.1.0/24 gateway 10.0.2.1
6 : ip add route 10.0.2.0/24 gateway 10.0.1.1
```

Configuración con encolamiento priorizado:

```
1 : port set et.1.1 speed 10mbps duplex full
2 : port set et.1.2 speed 10mbps duplex full
    !
3 : interface create ip subred1 address-netmask 10.0.1.1/24 port et.1.1 up
4 : interface create ip subred2 address-netmask 10.0.2.1/24 port et.1.2 up
    !
 5 : qos set ip PING_A high 10.0.1.10/32 10.0.2.10/32 any any any subred1,subred2 any any
any any
 6 : qos set ip PING_B high 10.0.2.10/32 10.0.1.10/32 any any any subred1,subred2 any any
any any
 7 : qos set ip TG_A low 10.0.1.20/32 10.0.2.20/32 any any any subred1,subred2 any any any
any
 8 : qos set ip TG_B low 10.0.2.20/32 10.0.1.20/32 any any any subred1,subred2 any any any
any
    !
9 : ip add route 10.0.1.0/24 gateway 10.0.2.1
```

10 : ip add route 10.0.2.0/24 gateway 10.0.1.1

Configuración con encolamiento justo ponderado

1 : port set et.1.1 speed 10mbps duplex full
2 : port set et.1.2 speed 10mbps duplex full
!
3 : interface create ip subred1 address-netmask 10.0.1.1/24 port et.1.1 up
4 : interface create ip subred2 address-netmask 10.0.2.1/24 port et.1.2 up
!
5 : qos set ip PING_A high 10.0.1.10/32 10.0.2.10/32 any any any subred1,subred2 any any any any
6 : qos set ip PING_B high 10.0.2.10/32 10.0.1.10/32 any any any subred1,subred2 any any any any
7 : qos set ip TG_A low 10.0.1.20/32 10.0.2.20/32 any any any subred1,subred2 any any any any
8 : qos set ip TG_B low 10.0.2.20/32 10.0.1.20/32 any any any subred1,subred2 any any any any
9 : qos set queueing-policy weighted-fair port all-ports
10 : qos set weighted-fair control 10 high 25 medium 65 low 0
!
11 : ip add route 10.0.1.0/24 gateway 10.0.2.1
12 : ip add route 10.0.2.0/24 gateway 10.0.1.1

Conmutador Subred 1

Marca: Enterasys Vertical Horizon VH-2402-L3
Sistema Operativo: v1.0

Velocidad de los puertos:

- Puerto 1.- 10Mbps/Full Duplex
- Puerto 2.- 100Mbps/Full Duplex
- Puerto 3.- 100Mbps/Full Duplex

Configuración básica:

Funcionamiento de conmutador capa 2

Conmutador Subred 2

Marca: 3COM CoreBuilder 3500
Sistema Operativo: v2.1.0

Velocidad de los puertos:

- Puerto 1.- 10Mbps/Full Duplex
- Puerto 2.- 100Mbps/Full Duplex
- Puerto 3.- 100Mbps/Full Duplex

Configuración básica:

Funcionamiento de conmutador capa 2

Host – Generador Pruebas Ping

Procesador: Intel Pentium IV 1.6 GHz
Memoria: 512 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: 3COM 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.1.10
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.1.1

Software de prueba:

- Ping de Linux

Scripts:

```
Ping 10.0.2.10 –s 64 –t 30
```

Host – Generador Tráfico

Procesador: Intel Pentium Celeron 466 MHz
Memoria: 128 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: Intel PRO/100+ 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.1.20
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.1.1

Software de prueba:

- Traffic Generator 2.0
- IPTraf 1.7.0

Scripts:

Traffic Generator 2.0.-
Para 2.5 Mbps

```
on 0:10
udp 10.0.2.20.2345
at 5 setup
at 7 arrival constant .0032 length constant 954
time 70
```

Para 5 Mpbs

```
on 0:10
udp 10.0.2.20.2345
at 5 setup
at 7 arrival constant .0016 length constant 954
time 70
```

Para 7.5 Mbps

```
on 0:10
udp 10.0.2.20.2345
at 5 setup
at 7 arrival constant .001 length constant 892
time 70
```

Para 10 Mbps

```
on 0:10
udp 10.0.2.20.2345
at 5 setup
at 7 arrival constant .0008 length constant 954
time 70
```

IPTraf 1.7.0.-

En modo "Estadísticas detalladas de la interfaz"

Host – Receptor Pruebas Ping

Procesador: Intel Pentium Celeron 466 MHz
Memoria: 128 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: Intel PRO/100+ 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.2.10
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.2.1

Software de prueba:

- Ninguno

Scripts:

Ninguno

Host – Receptor Tráfico

Procesador: Intel Pentium Celeron 466 MHz
Memoria: 128 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: Intel PRO/100+ 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.2.20
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.2.1

Software de prueba:

- Traffic Generator 2.0
- IPTraf 1.7.0

Scripts:
Traffic Generator 2.0.-

```
on 0:10 udp 10.0.2.20.2345 server
at 1.1 wait
```

IPTraf 1.7.0.-

En modo "Estadísticas detalladas de la interfaz"

Hay algunas notas importantes con respecto a las configuraciones mostradas en este apéndice.

- En la configuración del encolamiento justo ponderado (WFQ) se asignó la siguiente ponderación:
  - 10% para tráfico de control de la red.
  - 25% para tráfico de alta prioridad; en nuestro caso, los flujos de paquetes de la prueba Ping.
  - 65% para el tráfico de mediana prioridad; en nuestro caso, los flujos de paquetes provenientes del generador de tráfico.
  - 0% para el tráfico de baja prioridad.

  Por simplicidad se asignó el 65% a la cola de mediana prioridad y 0% a la de baja prioridad, pero en una implementación real esta ponderación varía de acuerdo a las necesidades propias de la red, es decir, no existe ningún estándar para la asignación de la ponderación para este mecanismo de encolamiento.

- En los scripts del generador de tráfico, el tamaño de paquete que se especifica corresponde al valor deseado de paquete menos 46 bytes, que corresponden a las cabeceras de UDP (8 bytes) + IP (20 bytes) + Ethernet (18 bytes).

  Por ejemplo, cuando se generaron 2.5 Mbps, si se realiza el cálculo en base a los datos del script, tendríamos que el flujo de paquetes por segundo sería de: (954 bytes) * (8) * (1/.0032 seg) = 2.385 Mbps.

  Sin embargo, como ya se mencionó los datos de las cabeceras también cuentan, e incrementan el tamaño de los paquetes. Así, si le incrementamos los 46 bytes de las cabeceras, el flujo de paquetes por segundo sería de 2.5 Mbps.

APENDICE A

CONFIGURACIONES PARA EL EXPERIMENTO B

Configuraciones de los equipos de comunicaciones y de los computadores utilizados en el experimento B.

Enrutador

Marca: Cabletron Smart Switch Router 2000
Sistema Operativo: 3.0.0.0

Velocidad de los puertos:

- Puerto 1.- 10Mbps/Full Duplex
- Puerto 2.- 10Mbps/Full Duplex

Configuración básica:

```
1 : port set et.1.1 speed 10mbps duplex full
2 : port set et.1.2 speed 10mbps duplex full
    !
3 : interface create ip subred1 address-netmask 10.0.1.1/24 port et.1.1 up
4 : interface create ip subred2 address-netmask 10.0.2.1/24 port et.1.2 up
    !
5 : ip add route 10.0.1.0/24 gateway 10.0.2.1
6 : ip add route 10.0.2.0/24 gateway 10.0.1.1
```

Configuración con encolamiento priorizado:

```
1 : port set et.1.1 speed 10mbps duplex full
2 : port set et.1.2 speed 10mbps duplex full
    !
3 : interface create ip subred1 address-netmask 10.0.1.1/24 port et.1.1 up
4 : interface create ip subred2 address-netmask 10.0.2.1/24 port et.1.2 up
    !
 5 : qos set ip PING_A high 10.0.1.10/32 10.0.2.10/32 any any any subred1,subred2 any any
any any
 6 : qos set ip PING_B high 10.0.2.10/32 10.0.1.10/32 any any any subred1,subred2 any any
any any
 7 : qos set ip TG_A low 10.0.1.20/32 10.0.2.20/32 any any any subred1,subred2 any any any
any
 8 : qos set ip TG_B low 10.0.2.20/32 10.0.1.20/32 any any any subred1,subred2 any any any
any
    !
9 : ip add route 10.0.1.0/24 gateway 10.0.2.1
```

10 : ip add route 10.0.2.0/24 gateway 10.0.1.1

Configuración con encolamiento justo ponderado

1 : port set et.1.1 speed 10mbps duplex full
2 : port set et.1.2 speed 10mbps duplex full
!
3 : interface create ip subred1 address-netmask 10.0.1.1/24 port et.1.1 up
4 : interface create ip subred2 address-netmask 10.0.2.1/24 port et.1.2 up
!
5 : qos set ip PING_A high 10.0.1.10/32 10.0.2.10/32 any any any subred1,subred2 any any any any
6 : qos set ip PING_B high 10.0.2.10/32 10.0.1.10/32 any any any subred1,subred2 any any any any
7 : qos set ip TG_A low 10.0.1.20/32 10.0.2.20/32 any any any subred1,subred2 any any any any
8 : qos set ip TG_B low 10.0.2.20/32 10.0.1.20/32 any any any subred1,subred2 any any any any
9 : qos set queueing-policy weighted-fair port all-ports
10 : qos set weighted-fair control 10 high 25 medium 65 low 0
!
11 : ip add route 10.0.1.0/24 gateway 10.0.2.1
12 : ip add route 10.0.2.0/24 gateway 10.0.1.1

Conmutador Subred 1

Marca: Enterasys Vertical Horizon VH-2402-L3
Sistema Operativo: v1.0

Velocidad de los puertos:

- Puerto 1.- 10Mbps/Full Duplex
- Puerto 2.- 100Mbps/Full Duplex
- Puerto 3.- 100Mbps/Full Duplex

Configuración sin prioritización:

Funcionamiento de conmutador capa 2

Configuración con prioritización:

Funcionamiento de conmutador capa 2, con las siguientes políticas de encolamiento:

- Computador Cliente IPerf asignado a la cola de alta prioridad para los paquetes de entrada y salida del conmutador.
- Computador Generador de Tráfico asignado a la cola de baja prioridad para los paquetes de entrada y salida del conmutador.

Conmutador Subred 2

Marca: 3COM CoreBuilder 3500
Sistema Operativo: v2.1.0

Velocidad de los puertos:

- Puerto 1.- 10Mbps/Full Duplex
- Puerto 2.- 100Mbps/Full Duplex
- Puerto 3.- 100Mbps/Full Duplex

Configuración básica:

Funcionamiento de conmutador capa 2

Host – Cliente Iperf

Procesador: Intel Pentium IV 1.6 GHz
Memoria: 512 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: 3COM 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.1.10
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.1.1

Software de prueba:

- Iperf v.1.6.4

Scripts:

./iperf –c 10.0.2.10 –u –b (tamaño) -f b –t 30 –i 1

Host – Generador Tráfico

Procesador: Intel Pentium Celeron 466 MHz
Memoria: 128 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: Intel PRO/100+ 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.1.20
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.1.1

Software de prueba:

- Traffic Generator 2.0
- IPTraf 1.7.0

Scripts:

Traffic Generator 2.0.-
Para 2.5 Mbps

```
on 0:10
udp 10.0.2.20.2345
at 5 setup
at 7 arrival constant .0032 length constant 954
time 45
```

Para 5 Mpbs

```
on 0:10
udp 10.0.2.20.2345
at 5 setup
at 7 arrival constant .0016 length constant 954
time 45
```

IPTraf.-
En modo "Estadísticas detalladas de la interfaz"

Host – Servidor Iperf

Procesador: Intel Pentium Celeron 466MHz
Memoria: 128 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: Intel PRO/100+ 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.2.10
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.2.1

Software de prueba:

- Iperf v.1.6.4

Scripts:

./iperf –s –u –I 1 –f b

Host – Receptor Tráfico

Procesador: Intel Pentium Celeron 466 MHz
Memoria: 128 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: Intel PRO/100+ 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.2.20
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.2.1

Software de prueba:

- Traffic Generator 2.0
- IPTraf 1.7.0

Scripts:
Traffic Generator 2.0.-
on 0:10 udp 10.0.2.20.2345 server
at 1.1 wait

IPTraf 1.7.0.-
En modo "Estadísticas detalladas de la interfaz"

Para estas configuraciones aplican las mismas observaciones hechas para el experimento anterior, en cuanto a la ponderación del mecanismo de encolamiento justo ponderado y el tamaño del paquete enviado por el generador de tráfico.

APENDICE A

CONFIGURACIONES PARA EL EXPERIMENTO C

Configuraciones de los equipos de comunicaciones y de los computadores utilizados en el experimento C.

Enrutador

Marca: Cabletron Smart Switch Router 2000
Sistema Operativo: 3.0.0.0

Velocidad de los puertos:

- Puerto 1.- 10Mbps/Full Duplex
- Puerto 2.- 10Mbps/Full Duplex

Configuración básica:

```
1 : port set et.1.1 speed 10mbps duplex full
2 : port set et.1.2 speed 10mbps duplex full
    !
3 : interface create ip subred1 address-netmask 10.0.1.1/24 port et.1.1 up
4 : interface create ip subred2 address-netmask 10.0.2.1/24 port et.1.2 up
    !
5 : ip add route 10.0.2.0/24 gateway 10.0.2.2
6 : ip add route 10.0.3.0/24 gateway 10.0.2.2
7 : ip add route 10.0.1.0/24 gateway 10.0.1.1
```

Configuración con encolamiento priorizado:

```
1 : port set et.1.1 speed 10mbps duplex full
2 : port set et.1.2 speed 10mbps duplex full
    !
3 : interface create ip subred1 address-netmask 10.0.1.1/24 port et.1.1 up
4 : interface create ip subred2 address-netmask 10.0.2.1/24 port et.1.2 up
    !
5 : qos set ip PING_A high 10.0.1.10/32 10.0.3.10/32 any any any subred1,subred2 any any
any any
6 : qos set ip PING_B high 10.0.3.10/32 10.0.1.10/32 any any any subred1,subred2 any any
any any
7 : qos set ip TG_A medium 10.0.1.20/32 10.0.3.20/32 any any any subred1,subred2 any any
any any
8 : qos set ip TG_B medium 10.0.3.20/32 10.0.1.20/32 any any any subred1,subred2 any any
any any
    !
```

9 : ip add route 10.0.2.0/24 gateway 10.0.2.2
10 : ip add route 10.0.3.0/24 gateway 10.0.2.2
11 : ip add route 10.0.1.0/24 gateway 10.0.1.1

Configuración con encolamiento justo ponderado

1 : port set et.1.1 speed 10mbps duplex full
2 : port set et.1.2 speed 10mbps duplex full
!
3 : interface create ip subred1 address-netmask 10.0.1.1/24 port et.1.1 up
4 : interface create ip subred2 address-netmask 10.0.2.1/24 port et.1.2 up
!
5 : qos set ip PING_A high 10.0.1.10/32 10.0.3.10/32 any any any subred1,subred2 any any any any
6 : qos set ip PING_B high 10.0.3.10/32 10.0.1.10/32 any any any subred1,subred2 any any any any
7 : qos set ip TG_A medium 10.0.1.20/32 10.0.3.20/32 any any any subred1,subred2 any any any any
8 : qos set ip TG_B medium 10.0.3.20/32 10.0.1.20/32 any any any subred1,subred2 any any any any
9 : qos set queueing-policy weighted-fair port all-ports
10 : qos set weighted-fair control 10 high 40 medium 50 low 0
!
11 : ip add route 10.0.2.0/24 gateway 10.0.2.2
12 : ip add route 10.0.3.0/24 gateway 10.0.2.2
13 : ip add route 10.0.1.0/24 gateway 10.0.1.1

Conmutador Subred 1

Marca: Enterasys Vertical Horizon VH-2402-L3
Sistema Operativo: v1.0

Velocidad de los puertos:

- Puerto 1.- 10Mbps/Full Duplex
- Puerto 2.- 100Mbps/Full Duplex
- Puerto 3.- 100Mbps/Full Duplex

Configuración sin prioritización:

Funcionamiento de conmutador capa 2

Configuración con prioritización:

Funcionamiento de conmutador capa 2, con las siguientes políticas de encolamiento:

- Computador Generador Prueba Pathchar asignado a la cola de alta prioridad para los paquetes de entrada y salida del conmutador.
- Computador Generador de Tráfico asignado a la cola de baja prioridad para los paquetes de entrada y salida del conmutador.

Conmutador-Enrutador

Marca: 3COM CoreBuilder 3500
Sistema Operativo: v2.1.0

Velocidad de los puertos:

- Puerto 1.- 10Mbps/Full Duplex
- Puerto 2.- 100Mbps/Full Duplex
- Puerto 3.- 100Mbps/Full Duplex

Configuración:

VLAN's:

Index	VID	Name	Type	Ports	Origin
1	1	Default	Open	1-13	Static
2	2	ToSSR	Open	1,2	Static
3	3	ToPCs	Open	3-6	Static

Interfaces:

Index	Type	IP Address	Subset mask	State	VLAN Index
1	VLAN	10.0.2.2	255.255.255.0	Up	2
2	VLAN	10.0.3.1	255.255.255.0	Up	3

Rutas

Destination	Subset mask	Gateway	Status
10.0.1.0	255.255.255.0	10.0.2.1	Static
10.0.2.0	255.255.255.0	--	Direct
10.0.2.2	255.255.255.255	--	Local
10.0.3.0	255.255.255.0	--	Direct
10.0.3.1	255.255.255.255	--	Local

Host – Generador Prueba Pathchar

Procesador: Intel Pentium IV 1.6 GHz
Memoria: 512 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: 3COM 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.1.10
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.1.1

Software de prueba:

- Pathchar

Scripts:

```
./pathchar 10.0.3.10 –q 16 –m 1500
```

Host – Generador Tráfico

Procesador: Intel Pentium Celeron 466 MHz
Memoria: 128 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: Intel PRO/100+ 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.1.20
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.1.1

Software de prueba:

- Traffic Generator 2.0
- IPTraf 1.7.0

Scripts:

Traffic Generator 2.0.-
Para 7.5 Mbps

```
on 0:10
udp 10.0.3.20.2345
at 5 setup
at 7 arrival constant .001 length constant 892
```

Para 10 Mpbs

```
on 0:10
udp 10.0.3.20.2345
at 5 setup
at 7 arrival constant .0008 length constant 954
```

Para 12.5Mbps

```
on 0:10
udp 10.0.3.20.2345
at 5 setup
at 7 arrival constant .0008 length constant 1204
```

IPTraf.-

En modo "Estadísticas detalladas de la interfaz"

Host – Receptor Pruebas Pathchar

Procesador: Intel Pentium Celeron 466 MHz
Memoria: 128 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: Intel PRO/100+ 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.3.10
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.3.1

Software de prueba:

- Ninguno

Scripts:

Ninguno

Host – Receptor Tráfico

Procesador: Intel Pentium Celeron 466 MHz
Memoria: 128 MB
Sistema Operativo: RedHat 8.0 (Kernel 2.4.18)
Tarjeta de Red: Intel PRO/100+ 100Mbps Full-Duplex

Configuración de la red:
IP: 10.0.3.20
Mascara de subred: 255.255.255.0
Puerta de enlace: 10.0.3.1

Software de prueba:

- Traffic Generator 2.0
- IPTraf 1.7.0

Scripts:
Traffic Generator 2.0.-

```
on 0:10 udp 10.0.3.20.2345 server
at 1.1 wait
```

IPTraf 1.7.0.-

En modo "Estadísticas detalladas de la interfaz"

APENDICE B

PRUEBAS DE HIPOTESIS Y ANALISIS DE VARIANZA DEL EXPERIMENTO A

APENDICE B

PRUEBAS DE HIPOTESIS Y ANALISIS DE VARIANZA DEL EXPERIMENTO A

El objetivo de este apéndice es mostrar las pruebas de hipótesis para cada uno de los factores y la interacción entre factores, para finalmente presentar la tabla del análisis de varianza de los datos recabados para el experimento A. También se indicará si la hipótesis nula se acepta o se rechaza basándose en el planteamiento propio de cada prueba.

Para la realización del análisis de varianza los paquetes perdidos en el ping fueron descartados, debido a que estos se pueden considerar como paquetes con tiempo de respuesta infinito, lo cual afectaría enormemente el resultado del análisis de varianza. El no tomar en cuenta estos paquetes no afecta, ya que solamente hubo 11 paquetes perdidos en las 3150 réplicas, lo cual no es significativo, además de que no fueron tomados en cuenta para los cálculos.

Todas las pruebas de hipótesis se realizaron con un nivel de significancia de 0.05 ($\alpha = 0.05$), es decir, se tiene un 95% de certeza que los resultados mostrados sean correctos.

Factor A.

No existe diferencia entre los tiempos de respuesta cuando se utilizan diferentes mecanismos de encolamiento.

H'_0: $\alpha_1 = \alpha_2 = \ldots = \alpha_a = 0$
H'_1: al menos una de las α_i's no es igual a cero.

Se rechaza la hipótesis nula a un nivel de significancia de 0.05 cuando:

$$f_1 > f_{0.05}\,[a\text{-}1,\ abc(n\text{-}1)]$$

Sustituyendo los valores encontramos:

$$4.93 > 3.00$$

Por lo tanto:

Se rechaza la hipótesis nula y se concluye que al utilizar diferentes mecanismos de encolamiento el tiempo de respuesta varía.

Factor B.

No existe diferencia entre los tiempos de respuesta cuando se varía el tamaño del paquete del ping.

H''_0: $\beta_1 = \beta_2 = \ldots = \beta_b = 0$
H''_1: al menos una de las β_j's no es igual a cero.

Se rechaza la hipótesis nula a un nivel de significancia de 0.05 cuando:

$$f_2 > f_{0.05}\,[b-1,\ abc(n-1)]$$

Sustituyendo los valores encontramos:

$$23.21 > 2.10$$

Por lo tanto:

Se rechaza la hipótesis nula y se concluye que al utilizar diferentes tamaños de paquete en el ping, el tiempo de respuesta varía.

Factor C.

No existe diferencia entre los tiempos de respuesta cuando se varía la carga de tráfico en el sistema de pruebas.

H'''_0: $\gamma_1 = \gamma_2 = \ldots = \gamma_b = 0$
H'''_1: al menos una de las γ_k's no es igual a cero.

Se rechaza la hipótesis nula a un nivel de significancia de 0.05 cuando:

$$f_3 > f_{0.05}\,[c-1,\ abc(n-1)]$$

Sustituyendo los valores encontramos:

$$437.68 > 2.37$$

Por lo tanto:

Se rechaza la hipótesis nula y se concluye que al utilizar cargas de tráfico diferentes en el sistema, el tiempo de respuesta varía.

Factor AB.

No existe interrelación entre los diferentes mecanismos de encolamiento y los diferentes tamaños de paquete del ping.

H^{IV}_0: $(\alpha\beta)_{11} = (\alpha\beta)_{12} = \ldots = (\alpha\beta)_{ab} = 0$
H^{IV}_1: al menos una de las $(\alpha\beta)_{ij}$'s no es igual a cero.

Se rechaza la hipótesis nula a un nivel de significancia de 0.05 cuando:

$$f_4 > f_\alpha [(a-1)(b-1), abc(n-1)]$$

Sustituyendo los valores encontramos:

$$1.59 > 1.75$$

Por lo tanto:

Este resultado implica aceptar la hipótesis nula, sin embargo, basándose en la tabla XXXII, utilizando el valor de Prob>F que indica la probabilidad que se acepte la hipótesis nula cuando esta es falsa, vemos que el valor es 0.0877, necesitando ser éste de por lo menos 0.1 para poder ser totalmente aceptada. Por lo tanto, podemos decir que estadísticamente no hay información suficiente para aceptar la hipótesis nula, ni tampoco para rechazarla.

Factor AC.

No existe interrelación entre los diferentes mecanismos de encolamiento y las diferentes cargas de tráfico del sistema de pruebas.

H^{V}_0: $(\alpha\gamma)_{11} = (\alpha\gamma)_{12} = \ldots = (\alpha\gamma)_{ac} = 0$
H^{V}_1: al menos una de las $(\alpha\gamma)_{ik}$'s no es igual a cero.

Se rechaza la hipótesis nula a un nivel de significancia de 0.05 cuando:

$$f_5 > f_{0.05} [(a-1)(c-1), abc(n-1)]$$

Sustituyendo los valores encontramos:

$$3.66 > 1.94$$

Por lo tanto:

Se rechaza la hipótesis nula y se concluye que al utilizar mecanismos de encolamiento distintos y cargas de tráfico diferentes en el sistema, el tiempo de respuesta varía.

Factor BC.

No existe interrelación entre los diferentes tamaños de paquete del ping y las diferentes cargas de tráfico del sistema de pruebas.

$H^{(6)}_0$: $(\beta\gamma)_{11} = (\beta\gamma)_{12} = \ldots = (\beta\gamma)_{bc} = 0$
$H^{(6)}_1$: al menos una de las $(\beta\gamma)_{jk}$'s no es igual a cero.

Se rechaza la hipótesis nula a un nivel de significancia de 0.05 cuando:

$$f_6 > f_{0.05}\,[(b-1)(c-1),\ abc(n-1)]$$

Sustituyendo los valores encontramos:

$$2.28 > 1.52$$

Por lo tanto:

Se rechaza la hipótesis nula y se concluye que al utilizar tamaños de paquete del ping distintos y diferentes cargas de tráfico en el sistema, el tiempo de respuesta varía.

Factor ABC.

No existe interrelación entre los diferentes mecanismos de encolamiento, los diferentes tamaños de paquete del ping y las diferentes cargas de tráfico del sistema de pruebas.

$H^{(7)}_0$: $(\alpha\beta\gamma)_{111} = (\alpha\beta\gamma)_{112} = \ldots = (\alpha\beta\gamma)_{abc} = 0$
$H^{(7)}_1$: al menos una de las $(\alpha\beta\gamma)_{ijk}$'s no es igual a cero.

Se rechaza la hipótesis nula a un nivel de significancia de 0.05 cuando:

$$f_7 > f_{0.05}\,[(a-1)(b-1)(c-1),\ abc(n-1)]$$

Sustituyendo los valores encontramos:

$$1.97 > 1.355$$

Por lo tanto:

Se rechaza la hipótesis nula y se concluye que al utilizar diferentes mecanismos de encolamiento, tamaños de paquete del ping distintos y diferentes cargas de tráfico en el sistema, el tiempo de respuesta varía.

TABLA XXXII

ANÁLISIS DE VARIANZA PARA EL EXPERIMENTO A

Fuente de variación	Suma de cuadrados	Grados de libertad	Cuadrado medio	*f* calculada	Prob>F
Efectos Principales					
A	1081.67	2	540.84	4.93	0.0073
B	15272.99	6	2545.50	23.21	<0.0001
C	192000	4	48011.80	437.68	<0.0001
Interacción de dos factores					
AB	2091.04	12	174.25	1.59	0.0877
AC	3112.59	8	401.57	3.66	0.0003
BC	6014.15	24	250.59	2.28	0.0004
Interacción de tres factores					
ABC	10381.91	48	216.29	1.97	<0.0001
Error	332800	3034*	109.70		
Total	562600	3138*			

* Para obtener estos grados de libertad, se restaron los 11 paquetes perdidos al número total de réplicas.

APENDICE C

TABLAS DE RESULTADOS DEL EXPERIMENTO B INCLUYENDO LOS INTERVALOS DE CONFIANZA

APENDICE C

TABLAS DE RESULTADOS DEL EXPERIMENTO B INCLUYENDO LOS INTERVALOS DE CONFIANZA

Las tablas de este Apéndice muestran la media muestral y el intervalo de confianza obtenido de las 30 réplicas realizadas en cada prueba. El intervalo de confianza mostrado tiene un 95% de confiabilidad.

TABLA XXXIII

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO FIFO CON CARGA DE TRAFICO DE 0 MBPS SIN PRIORIZAR EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	0.996	0.010	2.241	0.847	0.000	0.000
	2	2.000	0.000	0.427	0.069	0.000	0.000
	3	3.000	0.000	0.426	0.069	0.000	0.000
	4	3.824	0.021	0.369	0.059	4.390	0.536
	5	4.646	0.013	0.347	0.068	7.087	0.259
	6	5.505	0.045	0.330	0.048	8.249	0.741
	7	6.346	0.065	0.454	0.040	9.341	0.907
	8	7.113	0.117	0.457	0.073	11.078	1.417
	9	7.972	0.113	0.324	0.080	11.445	1.219
	10	8.226	0.460	0.407	0.143	17.479	2.970

TABLA XXXIV

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO PRIORIZADO CON CARGA DE TRAFICO DE 0 MBPS SIN PRIORIZAR EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.542	0.043	0.000	0.000
	2	2.000	0.000	0.514	0.052	0.000	0.000
	3	3.000	0.000	0.426	0.069	0.000	0.000
	4	3.900	0.013	0.422	0.425	2.508	0.332
	5	4.773	0.035	0.436	0.130	4.535	0.706
	6	5.623	0.053	0.425	0.072	6.295	0.571
	7	6.440	0.114	0.530	0.054	7.415	0.998
	8	7.194	0.098	0.347	0.061	9.895	1.020
	9	8.133	0.046	0.210	0.068	9.644	0.419
	10	8.704	0.051	0.204	0.050	12.911	0.573

TABLA XXXV

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO JUSTO PONDERADO CON CARGA DE TRAFICO DE 0 MBPS SIN PRIORIZAR EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.696	0.043	0.000	0.000
	2	2.965	0.005	0.404	0.057	1.163	0.182
	3	2.918	0.098	0.412	0.063	2.672	3.146
	4	3.799	0.038	0.402	0.057	5.078	1.089
	5	4.663	0.024	0.330	0.066	6.743	0.466
	6	5.522	0.060	0.317	0.054	7.956	0.987
	7	6.349	0.072	0.472	0.041	9.284	1.017
	8	6.928	0.097	0.147	0.013	13.265	1.172
	9	7.931	0.078	0.336	0.059	11.910	0.823
	10	8.270	0.451	0.493	0.257	16.962	2.939

TABLA XXXVI

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO FIFO CON CARGA DE TRAFICO DE 2.5 MBPS SIN PRIORIZAR EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.648	0.042	0.000	0.000
	2	2.000	0.000	0.626	0.060	0.000	0.000
	3	2.990	0.022	1.495	0.577	0.359	0.547
	4	3.834	0.113	0.546	0.051	3.247	1.687
	5	4.774	0.036	0.633	0.026	4.531	0.449
	6	5.554	0.080	0.545	0.102	7.303	0.995
	7	6.593	0.039	0.505	0.014	5.798	0.607
	8	6.606	0.062	1.598	0.037	17.307	0.631
	9	6.621	0.060	0.855	0.087	26.454	0.688
	10	6.607	0.039	0.784	0.036	33.929	0.392

TABLA XXXVII

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO PRIORIZADO CON CARGA DE TRAFICO DE 2.5 MBPS SIN PRIORIZAR EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.007	0.371	0.037	0.000	0.000
	2	2.000	0.000	0.546	0.047	0.000	0.000
	3	2.968	0.003	0.514	0.058	1.084	0.120
	4	3.821	0.022	0.513	0.055	4.481	0.494
	5	4.644	0.028	0.555	0.040	7.108	0.543
	6	5.440	0.060	0.544	0.028	9.323	0.961
	7	6.324	0.102	0.623	0.071	9.522	1.088
	8	6.442	0.059	0.816	0.043	19.476	0.738
	9	6.457	0.082	1.744	0.063	28.289	0.913
	10	6.411	0.064	1.751	0.077	35.863	0.664

TABLA XXXVIII

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO JUSTO PONDERADO CON CARGA DE TRAFICO DE 2.5 MBPS SIN PRIORIZAR EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.660	0.051	0.000	0.000
	2	2.000	0.000	0.581	0.059	0.000	0.000
	3	2.969	0.005	0.485	0.058	1.045	0.153
	4	3.827	0.022	0.528	0.057	4.331	0.553
	5	4.619	0.085	0.563	0.059	6.880	0.643
	6	5.464	0.067	0.551	0.032	8.919	1.070
	7	6.337	0.097	0.609	0.055	9.388	1.285
	8	6.408	0.137	1.008	0.074	20.009	1.924
	9	6.448	0.074	1.045	0.331	28.272	0.737
	10	6.479	0.073	1.666	0.050	35.365	0.836

TABLA XXXIX

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO FIFO CON CARGA DE TRAFICO DE 5 MBPS SIN PRIORIZAR EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	1.621	0.060	0.000	0.000
	2	2.000	0.001	0.854	0.052	0.000	0.000
	3	2.962	0.033	0.855	0.069	1.155	0.155
	4	3.786	0.020	0.767	0.041	5.346	0.564
	5	4.494	0.038	1.576	0.151	9.938	0.749
	6	4.506	0.055	1.754	0.172	24.804	0.833
	7	4.446	0.151	1.361	0.487	36.510	2.147
	8	4.538	0.040	1.192	0.065	43.249	0.512
	9	4.530	0.051	1.880	0.106	49.548	0.468
	10	4.527	0.034	1.760	0.142	54.701	0.449

TABLA XL

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO PRIORIZADO CON CARGA DE TRAFICO DE 5 MBPS SIN PRIORIZAR EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	0.997	0.009	2.086	0.106	0.000	0.000
	2	2.000	0.002	2.167	0.057	0.000	0.000
	3	2.965	0.005	0.927	0.061	1.084	0.174
	4	3.783	0.034	0.828	0.043	5.934	1.477
	5	4.461	0.048	1.557	0.145	10.533	0.812
	6	4.494	0.044	1.657	0.173	25.007	0.749
	7	4.443	0.139	2.045	0.454	36.644	2.386
	8	4.517	0.043	1.776	0.117	43.491	0.544
	9	4.554	0.047	1.312	0.084	49.425	0.530
	10	4.507	0.119	1.542	0.468	54.508	0.805

TABLA XLI

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO JUSTO PONDERADO CON CARGA DE TRAFICO DE 5 MBPS SIN PRIORIZAR EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.002	2.879	0.110	0.000	0.000
	2	2.000	0.001	0.860	0.056	0.000	0.000
	3	2.848	0.022	0.596	0.048	4.927	0.671
	4	3.775	0.044	0.783	0.059	5.196	0.669
	5	4.475	0.034	1.625	0.176	10.446	0.670
	6	4.530	0.045	1.638	0.104	24.477	0.692
	7	4.493	0.049	1.850	0.113	35.743	0.813
	8	4.570	0.026	1.889	0.167	42.748	0.301
	9	4.520	0.040	2.044	0.083	49.778	0.510
	10	4.545	0.047	1.832	0.101	54.475	0.582

TABLA XLII

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO FIFO CON CARGA DE TRAFICO DE 0 MBPS PRIORIZANDO EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	0.998	0.006	0.097	0.054	0.000	0.000
	2	2.000	0.003	0.515	0.036	0.000	0.000
	3	3.000	0.010	0.545	0.037	0.000	0.000
	4	4.000	0.017	0.524	0.034	0.000	0.000
	5	4.894	0.023	0.559	0.035	2.114	0.449
	6	5.791	0.040	0.699	0.034	3.464	0.635
	7	6.610	0.063	0.532	0.029	5.547	0.812
	8	7.408	0.082	0.398	0.039	7.387	0.763
	9	8.372	0.061	0.958	0.094	6.857	0.544
	10	8.911	0.071	0.917	0.117	10.835	0.632

TABLA XLIII

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO PRIORIZADO CON CARGA DE TRAFICO DE 0 MBPS PRIORIZANDO EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.445	0.038	0.000	0.000
	2	2.000	0.005	0.528	0.034	0.000	0.000
	3	2.997	0.048	0.525	0.047	0.000	0.000
	4	3.993	0.013	0.550	0.033	0.159	0.311
	5	4.895	0.022	0.564	0.043	2.099	0.437
	6	5.794	0.039	0.700	0.030	3.412	0.629
	7	6.602	0.074	0.539	0.032	5.656	0.967
	8	7.404	0.086	0.406	0.039	7.408	0.979
	9	8.448	0.083	0.298	0.049	6.163	0.878
	10	8.977	0.063	0.938	0.162	10.221	0.631

TABLA XLIV

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO JUSTO PONDERADO CON CARGA DE TRAFICO DE 0 MBPS PRIORIZANDO EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.447	0.038	0.000	0.000
	2	2.000	0.004	0.537	0.035	0.000	0.000
	3	3.000	0.010	0.553	0.030	0.000	0.000
	4	3.993	0.013	0.549	0.035	0.159	0.311
	5	4.882	0.058	0.560	0.043	2.237	0.606
	6	5.794	0.039	0.700	0.030	3.412	0.629
	7	6.613	0.064	0.541	0.034	5.502	0.827
	8	7.397	0.054	0.408	0.040	7.512	0.560
	9	8.438	0.078	0.306	0.051	6.268	0.818
	10	8.990	0.061	0.916	0.117	10.099	0.732

TABLA XLV

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO FIFO CON CARGA DE TRAFICO DE 2.5 MBPS PRIORIZANDO EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.580	0.038	0.000	0.000
	2	2.000	0.004	0.670	0.048	0.000	0.000
	3	2.988	0.019	0.782	0.080	0.150	0.295
	4	3.991	0.017	0.743	0.064	0.226	0.317
	5	4.870	0.025	0.699	0.041	2.589	0.462
	6	5.785	0.039	0.623	0.037	3.577	0.621
	7	6.609	0.053	0.569	0.040	5.503	0.613
	8	7.371	0.136	0.451	0.039	7.814	1.634
	9	8.366	0.108	0.415	0.054	6.927	0.822
	10	8.951	0.101	1.679	0.147	10.350	0.994

TABLA XLVI

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO PRIORIZADO CON CARGA DE TRAFICO DE 2.5 MBPS PRIORIZANDO EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.600	0.052	0.000	0.000
	2	2.000	0.003	0.631	0.040	0.000	0.000
	3	2.991	0.042	1.254	0.641	0.102	0.177
	4	3.985	0.019	0.792	0.050	0.188	0.314
	5	4.870	0.026	0.685	0.071	2.597	0.440
	6	5.769	0.020	0.553	0.043	3.831	0.528
	7	6.642	0.052	0.573	0.040	5.093	0.663
	8	7.411	0.080	0.479	0.061	7.312	0.931
	9	8.192	0.405	0.402	0.080	8.881	3.062
	10	8.911	0.079	0.848	0.108	10.864	0.744

TABLA XLVII

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO JUSTO PONDERADO CON CARGA DE TRAFICO DE 2.5 MBPS PRIORIZANDO EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.574	0.037	0.000	0.000
	2	1.998	0.020	0.622	0.032	0.000	0.000
	3	3.000	0.010	0.689	0.061	0.000	0.000
	4	3.984	0.017	0.719	0.032	0.384	0.431
	5	4.857	0.023	0.680	0.031	2.856	0.405
	6	5.736	0.083	0.574	0.044	4.223	0.889
	7	6.680	0.048	0.547	0.057	4.548	0.644
	8	7.395	0.080	0.473	0.041	7.525	0.924
	9	8.343	0.104	0.473	0.214	7.335	1.138
	10	8.994	0.062	0.943	0.133	10.015	0.713

TABLA XLVIII

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO FIFO CON CARGA DE TRAFICO DE 5 MBPS PRIORIZANDO EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.861	0.042	0.000	0.000
	2	2.000	0.005	0.911	0.029	0.000	0.000
	3	2.999	0.012	0.858	0.024	0.013	0.025
	4	3.989	0.047	0.677	0.059	0.150	0.294
	5	4.888	0.022	0.823	0.025	2.238	0.426
	6	5.769	0.046	0.772	0.014	3.806	0.751
	7	6.557	0.058	0.578	0.026	6.294	0.629
	8	7.400	0.082	0.463	0.040	7.457	1.095
	9	8.798	0.026	0.746	0.034	2.290	0.293
	10	8.918	0.044	0.948	0.132	10.769	0.482

TABLA XLIX

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO PRIORIZADO CON CARGA DE TRAFICO DE 5 MBPS PRIORIZANDO EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.006	0.567	0.046	0.000	0.000
	2	2.000	0.005	2.730	0.151	0.000	0.000
	3	3.000	0.009	0.846	0.035	0.000	0.000
	4	3.983	0.019	0.660	0.041	0.406	0.461
	5	4.887	0.027	0.824	0.032	2.246	0.525
	6	5.769	0.039	0.776	0.009	3.817	0.611
	7	6.546	0.072	0.614	0.051	6.274	0.903
	8	7.458	0.092	0.732	0.048	6.576	0.999
	9	8.343	0.034	0.372	0.062	7.328	0.322
	10	8.999	0.078	0.800	0.092	10.023	0.690

TABLA L

MEDIA MUESTRAL E INTERVALOS DE CONFIANZA PARA LOS RESULTADOS OBTENIDOS EN EL EXPERIMENTO B UTILIZANDO ENCOLAMIENTO JUSTO PONDERADO CON CARGA DE TRAFICO DE 5 MBPS PRIORIZANDO EL CONMUTADOR

		Parámetro					
		Caudal Eficaz (Mbps)		Jitter (ms)		% Paquetes Perdidos	
		Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)	Media Muestral	Intervalo de Confianza (95%)
Datos Enviados Mbps	1	1.000	0.000	0.842	0.042	0.000	0.000
	2	2.000	0.003	0.869	0.039	0.000	0.000
	3	3.000	0.010	0.847	0.029	0.000	0.000
	4	3.980	0.028	1.002	0.065	0.239	0.468
	5	4.897	0.025	0.791	0.029	2.058	0.481
	6	5.779	0.041	0.763	0.013	3.646	0.657
	7	6.505	0.075	0.771	0.037	6.899	0.646
	8	7.397	0.063	0.461	0.035	7.515	0.717
	9	8.427	0.074	0.369	0.038	6.399	0.685
	10	8.968	0.067	0.746	0.106	10.327	0.676

APENDICE D

FUNCIONAMIENTO DE LA HERRAMIENTA PATHCHAR Y FORMA DE UTILIZACION EN EL EXPERIMENTO C

APENDICE D

FUNCIONAMIENTO DE LA HERRAMIENTA PATHCHAR Y FORMA DE UTILIZACION EN EL EXPERIMENTO C

La herramienta Pathchar desarrollada por Van Jacobson es una herramienta que permite al usuario encontrar el ancho de banda, retraso, tiempo y tamaño promedio de colas, así como tasa de pérdida de paquetes para cada salto en la ruta entre un emisor y un destino en una red de paquetes. Esta herramienta se basa en el protocolo ICMP para la obtención de los tiempos.

La distribución de esta herramienta es solamente en versión Alpha, nunca fue terminada en una versión final, por lo que hay la posibilidad de que tenga algunos errores en las mediciones o resultados; aunque solo es una posibilidad, ya que la documentación existente sobre la herramienta no indica ningún error detectado [46].

Aunque esta herramienta muestra como salida otras mediciones tales como el caudal eficaz en cada uno de los puntos, nosotros solamente la utilizamos para medir las características de las colas de espera; la decisión de no utilizar la herramienta para medir el caudal eficaz y otros parámetros es debido a que esta herramienta obtiene mediciones erróneas al utilizar enlaces de alta velocidad [47].

Veamos ahora como funciona la herramienta. Los equipos de comunicaciones tienen enlaces, un motor de reenvío y colas, como se muestra en la figura 65.

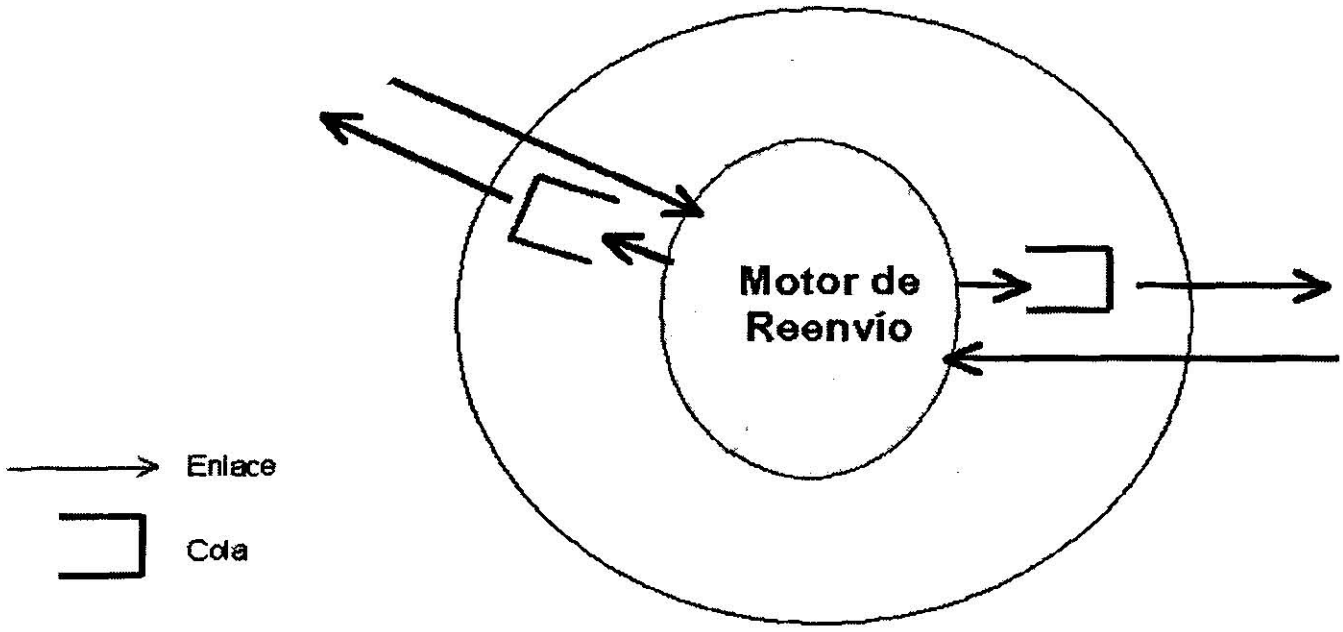


Figura 65. Elementos conceptuales de los equipos de comunicaciones.

Para obtener la información de las colas, pathchar se basa en la premisa de que las colas solamente pueden ser agregadas a un tiempo determinístico, por lo que si se toman una gran cantidad de muestras, en tiempos aleatorios bien espaciados, el conjunto mínimo de tiempos puede aproximarse a un modelo de reenvío adecuado [48].

De manera conceptual, las mediciones obtenidas entre cada nodo se basan en el modelo de la figura 66.

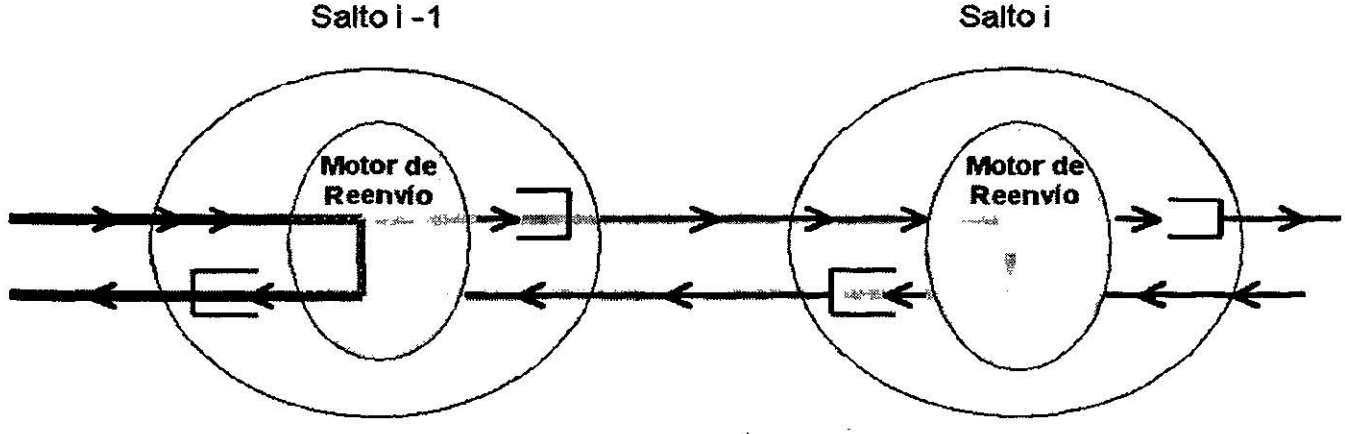


Figura 66. Mediciones obtenidas en general por cada nodo usando la herramienta Pathchar.

La sombra oscura indica la ruta de medición para el salto i – 1, mientras que el sombreado claro indica las mediciones para el salto i.

En nuestro experimento, las mediciones para cada salto están representadas como se muestra en la figura 67.

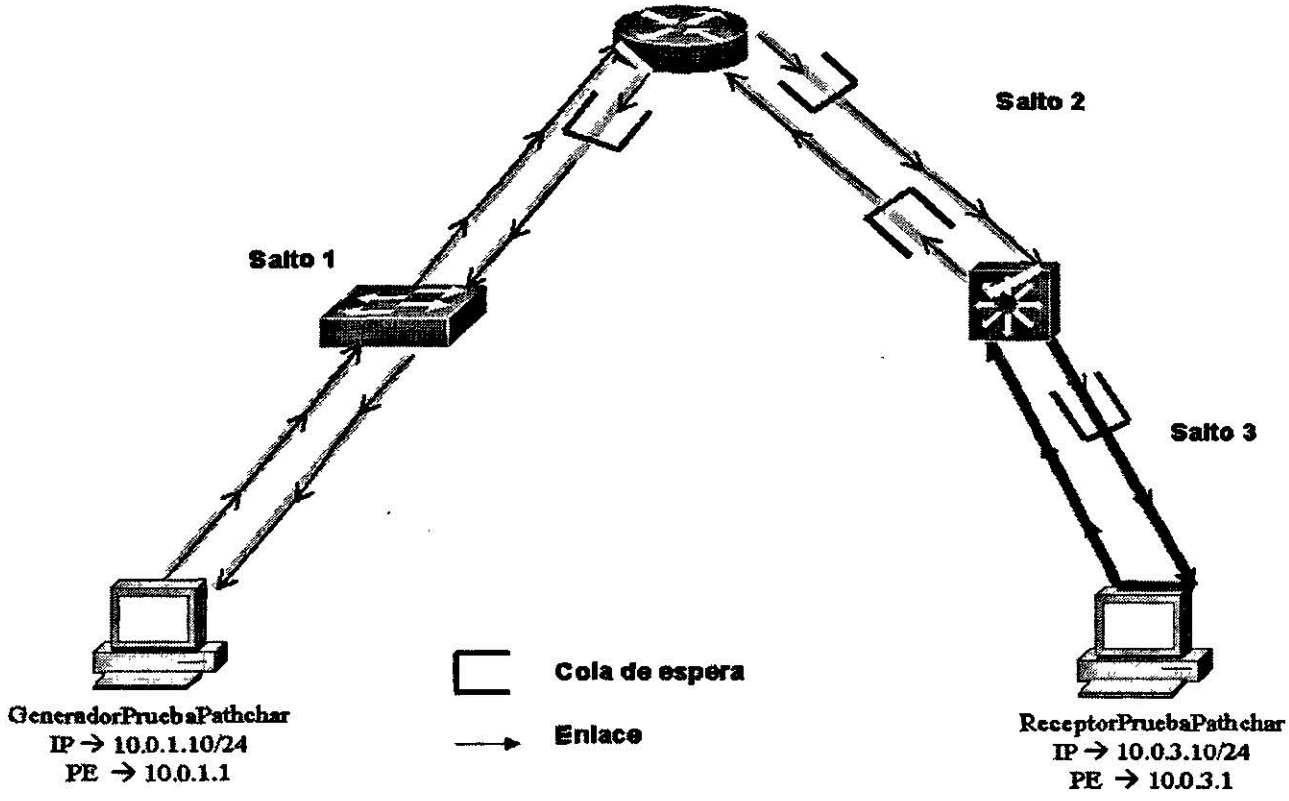


Figura 67. Colas de espera de donde se obtuvieron las mediciones por salto del Experimento C.

La figura 67 muestra el mismo esquema de conectividad de la figura 52, solamente que aquí se identifican los saltos que se tomaron para llevar a cabo las mediciones.

En el primer salto, la cola de espera a la cual se realizaron las mediciones, es la correspondiente a la interfaz de salida del puerto 1 del enrutador, y se forma la cola cuando el enrutador contesta la petición de mensajes de Solicitud de Eco de ICMP con mensajes de Respuesta de Eco [49]; aquí esta cola de espera no tienen ninguna configuración, y en los resultados de este salto están implícitos los tiempos que toman los paquetes en atravesar en conmutador de la Subred 1.

En el segundo salto, las mediciones muestran las características de las colas que se forman en la interfaz de salida del enrutador y en la interfaz de entrada del conmutador- enrutador. En la cola que se forma en la interfaz de salida del enrutador fue donde se aplicaron los distintos mecanismos de encolamiento del Factor A del Experimento C.

El tercer salto corresponde a la conectividad entre en conmutador-enrutador y el computador Receptor de Prueba Pathchar; los resultados mostrados de este salto, corresponden a la cola de espera que se forma en la salida del conmutador-enrutador.

Para este experimento, se configuró la herramienta Pathchar para que realizara 16 pruebas para cada salto, utilizando tamaños de paquetes que iban desde los 64 bytes hasta los 1500 con incrementos de 32 bytes.

APENDICE E

GLOSARIO

APENDICE E

GLOSARIO

Acondicionador de tráfico

Entidad la cual realiza funciones de acondicionamiento de tráfico, el cual puede contener Medidores, Marcadores, Desechador o Tirador, Configurador o Modelador. Los Acondicionadores de Tráfico son típicamente implementados solamente en los nodos frontera.

Anfitrión (Host)

Cualquier computadora de usuario final que se conecta a una red.

Arquitectura de comunicaciones

Una estructura consistente compuesta por un conjunto de módulos que realizarán las funciones para llevar a cabo la comunicación

ATM

Modo de Transferencia Asíncrono (Asynchronous Transfer Mode), tecnología de red orientada a conexión la cual utiliza pequeñas celdas de tamaño fijo para la transmisión de los datos.

BA

Grupo de Comportamiento (Behavior Aggregate), es una colección de paquetes con el mismo valor en el punto de código, los cuales atraviesan un enlace en determinada dirección.

Backbone

Red de columna vertebral, es cualquier red que forme la interconexión central entre redes.

Calidad de Servicio (QoS)

Es la capacidad de un elemento de la red (aplicación, anfitrión o enrutador) de proveer un nivel de aseguramiento para que ese tráfico y los requerimientos de servicio puedan ser satisfechos.

Campo TOS

Campo dentro de la cabecera del protocolo IPv4 utilizado para señalizar una solicitud para un nivel de Calidad de Servicio específico.

Carga ofrecida a la red

Es el total de paquetes ofrecidos a la red incluyendo paquetes de control, colisiones, etcétera.

Caudal eficaz (throughput)

Es la tasa total de datos transmitidos entre nodos; expresa una fracción de la capacidad y puede ser interpretado como la utilización.

Clasificador

Entidad la cual selecciona paquetes basándose en el contenido de las cabeceras de los paquetes de acuerdo a reglas definidas.

CLNP

Protocolo de Red Sin Conexión (Connectionless Network Protocol), protocolo de la capa de red en la arquitectura OSI.

Comportamiento de reenvío ToS

Indica como debe ser tratado el paquete cuando este es reenviado, basándose en el valor del campo TOS.

Comportamiento por salto (PHB, Per Hop Behaviors)

Es el comportamiento de reenvío externamente observable aplicado en un nodo que soporte Servicios Diferenciados a un grupo de comportamiento (BA).

Comportamientos por grupo

Conjunto de uno o más Comportamientos por Salto que solo pueden ser especificados de manera significativa e implementados simultáneamente, debido a restricciones comunes aplicados a todos los PHB en conjunto, tales como servicios en colas o políticas de administración de colas.

Comunicación peer-to-peer

Es un esquema mediante el cual cada parte involucrada tiene las mismas capacidades, por lo que cualquiera puede iniciar el proceso de comunicación.

Configurador o Modelador

Dispositivo que realiza el proceso de retrasar paquetes dentro de un flujo para causar que estos paquetes se acoplen a algún perfil de tráfico definido.

Conmutador

Es un dispositivo que filtra y reenvía paquetes entre segmentos de redes de área local. Operan en la capa de enlace de datos, aunque algunos conmutadores operan también en la capa de red.

Datagrama

Unidad básica de información que pasa a través de una red TCP/IP. Contiene (entre otras cosas), las direcciones fuente y destino.

Descriptor de Flujo (Flow-descriptor)

En Servicios Integrados, es una solicitud de reserva emitida por un sistema final destino, que consta de una especificación de solicitud y una especificación de filtro.

Desechador o Tirador

Dispositivo que realiza el proceso de descartar paquetes basándose en reglas o políticas determinadas.

DNS

Sistema de Nombre de Dominio (Domain Name System), es un sistema de base de datos distribuida en línea, la cual es utilizada para transformar nombres de máquina en direcciones IP para que puedan ser más legibles.

Dominio de Servicios Diferenciados

Conjunto contiguo de nodos los cuales operan con un conjunto común de servicios proporcionando políticas y definiciones de Comportamientos Por Salto.

DSA

Distributed System Architecture (Arquitectura de Sistemas Distribuidos), nombre original que se le dio al modelo OSI.

DSCP

Punto de Código de Servicios Diferenciados (Differentiated Services Code Point),

Enrutamiento

Determinación del camino o ruta que las unidades de datos atravesarán desde la fuente al destino.

Especificación de Filtro (Filter Spec, *Filter Specification*)

En Servicios Integrados, define un conjunto de paquetes para los que se solicita una reserva de recursos.

Especificación de *Solicitud* (RSpec, *Request Specification*)

Indica el tipo de Servicios Integrados requeridos (puede ser Carga Controlada o Garantizado).

Especificación de *Tráfico* (TSpec, *Traffic Specification*)

En Servicios Integrados, comunica información acerca de la tasa de datos y rango de tamaño de paquetes de un flujo de datos.

Ethernet

Tecnología para redes de área local, la cual proporciona un sistema de entrega de mejor esfuerzo basado en la tecnología CSMA/CD.

FIFO

Primero en Entrar Primero en Salir (First In First Out), algoritmo básico de colas, se basa en la premisa el Primero en Entrar es el Primero en Salir.

Firewall (cortafuegos)

Es un sistema diseñado para prevenir accesos no autorizados desde o hacia redes privadas. Todos los mensajes que entran o dejan la red privada pasan a través del firewall, el cual examina los paquetes y aplica criterios de seguridad con cada uno de ellos.

Flujo

Conjunto de paquetes agrupados por características comunes (protocolo de transporte, dirección origen, dirección destino, etc.) que fluyen unidireccionalmente entre emisor y receptor.

FQ

Encolamiento Justo (Fair Queuing), mecanismo de encolamiento en el cual los paquetes primeramente son clasificados en flujos por el sistema y después asignados a la cola correspondiente dedicada específicamente para ese flujo. Después, las colas son servidas un paquete por turno en un orden round-robin.

Frame Relay

Tecnología de conmutación de paquetes basada en el uso de tramas de la capa de enlace de longitud variable.

IEEE 802.1p

Técnica de señalización del Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronic Engineering, IEEE), la cual permite priorizar tráfico en la capa de enlace de datos (capa 2 del modelo OSI) utilizando la tecnología Ethernet.

IP

Protocolo Internet (Internet Protocol), es el protocolo estándar que define a los datagramas IP como la unidad de intercambio de información. Proporciona las bases para el servicio de entrega de paquetes sin conexión y con el mejor esfuerzo. Trabaja en la capa de red.

Jitter

Variación en el retraso ó jitter, es la variación en los tiempos entre llegadas de los datos a un destino final.

Latencia

Ver Retraso.

Marcador

Dispositivo que realiza el proceso de establecer el Punto de Código en un paquete basándose en reglas definidas.

Medidor

Dispositivo que realiza el proceso de medir propiedades temporales sobre los flujos de tráfico seleccionados por el Clasificador. Estas mediciones pueden ser utilizadas por el Marcador, el Configurador o el Desechador, o bien, ser utilizado solo con fines de contabilidad y medición.

Memoria temporal ó buffer

Área de almacenamiento temporal utilizada por los equipos de comunicaciones para almacenar los datos entrantes en espera de ser atendidos.

OSI

Interconexión de Sistemas Abiertos (Open Systems Interconnection), es un conjunto de estándares y protocolos de la ISO para la interconexión de sistemas.

PDU

Unidad de Datos de Protocolo (Protocol Data Unit), en un modelo de comunicaciones, es la unión de los datos de la cabecera del protocolo de la capa actual con los datos de la capa superior.

Pérdida de paquetes

Son paquetes que no llegaron a su destino. Esto puede ser debido a diferentes causas, tales como: la suma de comprobación no es correcta, fallas en el medio físico de transmisión, o bien, por congestiones en los equipos de comunicaciones.

PQ

Encolamiento Priorizado (Priority Queuing), mecanismo de encolamiento el cual permite clasificar los paquetes para después ser asignados a colas con diferentes prioridades. Ya que estén en cada una de las colas los paquetes, primero se transmiten las colas de alta prioridad, una vez que esta se encuentre vacía, se transmite la cola de menor prioridad siguiente, y así sucesivamente.

Precedencia IP

Provee la habilidad de clasificar los paquetes de la red en la capa de Internet de la arquitectura TCP/IP. La precedencia es un esquema para la asignación de recursos en la red basados en la importancia de los diferentes flujos de datos

Protocolo

Conjunto de reglas que gobiernan el intercambio de datos entre dos entidades

Puerto

Abstracción que los protocolos de transporte TCP/IP utilizan para distinguir diferentes aplicaciones destinos en un ambiente multitarea.

Punto de código (CodePoint)

Valor específico de la sección DSCP del campo DS.

Puntos de Código de Selección de Clase (Class-Selector CodePoint)

Cualquiera de los ocho puntos de código en el rango 'xxx000' (donde 'x' puede ser 0 o 1).

Región de Servicios Diferenciados

Conjunto contiguo de dominios de servicios diferenciados, los cuales pueden proporcionar servicios diferenciados sobre rutas que cruzan estos dominios.

Retraso

Expresión que determina el tiempo que toma un paquete de datos en ir de un punto de origen a un punto destino. También conocido como latencia.

Round-robin

Es un método para seleccionar igualmente a todos los elementos de un grupo en un orden racional, usualmente empezando del primero hasta al último, para después comenzar de nuevo la lista, y así sucesivamente.

RSVP

Estándar de la IETF el cual permite a un dispositivo final y a la red negociar características específicas de QoS. RSVP es el componente principal de la arquitectura de Servicios Integrados.

Enrutador

Dispositivo que conecta dos o más redes y envía paquetes de una red a otra. Un enrutador utiliza las direcciones destino de un datagrama para decidir el próximo salto al que enviará el datagrama.

SAP

Punto de Acceso al Servicio (Service Access Point), identificador único que asigna los protocolos OSI a cada aplicación para que los protocolos puedan identificar el destino de la información en un ambiente multitarea.

Sello de tiempo (time stamp)

La marca de tiempo, es un registro del tiempo actual en el que sucedió un evento. Es utilizado principalmente para sincronización de procesos.

Servicios Diferenciados (DiffServ ó DS)

Arquitectura para proveer Calidad de Servicio en las redes, el cual consta de un estándar de la IETF para un pequeño, pero bien definido conjunto de bloques de construcción por paquete mediante los cuales se pueden construir una variedad de servicios.

Servicios Integrados (IntServ)

Arquitectura de Calidad de Servicio la cual consta de un conjunto de estándares de la IETF que abarcan cómo los servicios de aplicación definen sus requerimientos de Calidad de Servicio, cómo esta información está disponible para los equipos de comunicaciones, y métodos para probar y validar que la Calidad de Servicio acordada sea mantenida.

SLA

Acuerdo de Nivel de Servicio (Service Level Agreements), es un contrato entre el consumidor y el proveedor de servicios de red que especifica, en términos medibles (caudal eficaz, tasa de pérdida de paquetes, retraso, etcétera), que servicios se deben proveer.

SNA

Arquitectura de Sistemas de Red (Systems Network Architecture), arquitectura propietaria de IBM para la interconexión de sistemas.

SONET

Red Óptica Síncrona (Synchronous Optical Network), estándar norteamericano de transporte para redes ópticas con velocidades de 51.84 Mega bits por segundo (OC-1) a 40 Giga bits por segundo (OC-768).

Tasa de paquetes perdidos

Es la fracción de todos los paquetes que no llegaron a su destino.

TCP

Protocolo de Control de Transmisión (Transmisión Control Protocol), es el protocolo de nivel de transporte TCP/IP estándar que proporciona el servicio de flujos de datos confiable.

Telnet

Protocolo estándar de la arquitectura TCP/IP para el servicio de terminal remota.

Token Ring

Tecnología de red de área local en la cual las computadoras son conectadas en un anillo o topología estrella, y un dígito binario o token es utilizado para decidir que máquina puede enviar mensajes.

ToS

Tipo de Servicio (Type of Service), arquitectura que provee una especificación de parámetros abstractos de la Calidad de Servicio deseada. Estos parámetros son usados para guiar la selección de los parámetros de servicio cuando se transmite un datagrama sobre una red en particular.

Tráfico Isócrono

Tráfico que requiere una coordinación de tiempo para que la información se despliegue correctamente. Ejemplos de este tipo de tráfico es la voz y video en tiempo real. También llamado Tráfico Síncrono.

Trama

Grupo de bits que incluye datos, además de las direcciones e información de control del protocolo. Unidad básica del protocolo de la capa de enlace.

UDP

Protocolo de Datagrama de Usuario (User Datagram Protocol), protocolo estándar TCP/IP que permite a un programa de aplicación de una máquina enviar un datagrama hacia un programa de aplicación de otra máquina. Este protocolo provee un servicio de datagramas poco confiable no orientado a conexión.

WFQ

Encolamiento Justo Ponderado (Weighted Fair Queuing), mecanismo de encolamiento el cual aplica prioridades o ponderaciones para identificar el tráfico y clasificarlo para determinar cuanto ancho de banda le corresponde. Típicamente cada prioridad tiene su propia cola; las ponderaciones son asignadas a cada cola para determinar la asignación del ancho de banda.

APENDICE F

LISTA DE ACRONIMOS

APENDICE F

LISTA DE ACRONIMOS

Acrónimo	Término en inglés	Término en español
AF PHB	Assured Forwarding Per Hop Behavior	Comportamiento por Salto de Reenvío Asegurado
ANSI	American Nacional Standards Institute	Instituto Nacional de Estándares Americanos
ARPA	Advanced Research Project Agency	Agencia de Proyectos de Investigación Avanzada
ARPANET	Advanced Research Projects Agency Network	Red de la Agencia de Proyectos de Investigación Avanzada
ATM	Asynchronous Transfer Mode	Modo de Transferencia Asíncrono
BA	Behavior Agrégate	Grupo de Comportamiento
CBQ	Class-based Queuing	Encolamiento Basado en Clases
CLNP	Connectionless Network Protocol	Protocolo de Red Sin Conexión
CQ	Custom Queuing	Encolamiento Customizado
CSMA/CD	Carrier Sense Multiple Access/Collision Detect	Acceso múltiple sensible a portadora con detección de colisión.
CSNET	Computer Science Network	Red de las Ciencias de la Computación
DARPA	Defense Advanced Research Projects Agency	Agencia de Proyectos de Desarrollo para la Defensa
DiffServ	Differentiated Service	Servicios Diferenciados
DNS	Domain Name System	Sistema de Nombre de Dominio
DoD	Departament of Defense	Departamento de Defensa
DS	Differentiated Service	Servicios Diferenciados
DSA	Distributed System Architecture	Arquitectura de Sistemas Distribuidos
DSCP	Differentiated Services Code Point	Punto de Código de Servicios Diferenciados
DWRR	Deficit Weighted Round Robin	Encolamiento Round Robin con Déficit Ponderado

Acrónimo	Término en inglés	Término en español
EF PHB	Expedited Forwarding Per Hop Behavior	Comportamiento por Salto de Reenvío Acelerado
FIFO	First In First Out	Primero en Entrar Primero en Salir
FQ	Fair Queuing	Encolamiento Justo
IEEE	Institute of Electrical and Electronics Engineers	Instituto de Ingenieros Eléctricos y Electrónicos
IETF	Internet Engineering Task Force	Grupo de Trabajo de Ingeniería de Internet
IntServ	Integrated Services	Servicios Integrados
IP	Internet Protocol	Protocolo Internet
ISO	International Organization Standardization	Organización Internacional de Estandarización
JANET	Joint Acedemic Network	Red de Unión Academica
JUNET	Japan Unix Network	Red Unix de Japón
MAC	Medium Access Control	Control de Acceso al Medio
MBZ	Must Be Zero	Debe ser cero
MILNET	Militar Network	Red Militar
MINET	Movement Information Net	Red de Movimiento de Información
MPLS	MultiProtocol Label Switching	Conmutación de Etiquetas MultiProtocolo
NCP	Network Control Protocol	Protocolo de Control de Red
NSF	Nacional Science Foundation	Fundación Nacional para las Ciencias
NSFNET	Nacional Science Foundation Network	Red de la Fundación Nacional para las Ciencias
OSI	Open Systems Interconnection	Interconexión de Sistemas Abiertos
PDU	Protocol Data Unit	Unidad de Datos de Protocolo
PHB	Per Hop Behaviors	Comportamiento por Salto
PING	Packet InterNet Groper	Buscador entre redes de paquetes
PQ	Priority Queuing	Encolamiento Priorizado
PRNET	Packet Radio Network	Red de Paquetes por Radio
QoS	Quality of Service	Calidad de Servicio
RESV	Reservation Request	Solicitud de Reservación
RFC	Request For Comments	Abierto al Debate
RSpec	Request Specification	Especificación de Solicitud
RSVP	Resource ReSerVation Protocol	Protocolo de Reservación de Recursos
SAP	Service Access Point	Punto de Acceso al Servicio

Acrónimo	Término en inglés	Término en español
SLA	Service Level Agreements	Acuerdo de Nivel de Servicio
SNA	Systems Network Architecture	Arquitectura de Sistemas de Red
SONET	Synchronous Optical Network	Red Óptica Síncrona
TCB	Traffic Conditioner Block	Bloque de Acondicionamiento de Tráfico
TCI	Tag Control Info	Etiqueta de Control de Información
TCP	Transmission Control Protocol	Protocolo de Control de Transmisión
ToS	Type of Service	Tipo de Servicio
TP	Transport Protocol	Protocolo de Transporte
TSpec	Traffic Specification	Especificación de Tráfico
UDP	User Datagram Protocol	Protocolo de Datagrama de Usuario
VLAN	Virtual Local Area Network	Red de Área Local Virtual
VoIP	Voice over Internet Protocol	Voz sobre el Protocolo Internet
WFQ	Weighted Fair Queuing	Encolamiento Justo Ponderado
WRR	Weighted Round Robin	Encolamiento Round Robin Ponderado
WWW	World Wide Web	Red Mundial

RESUMEN AUTOBIOGRAFICO

Juan Antonio Castilleja García

Candidato para el Grado de

Licenciado en Ciencias Computacionales

Tesis: CALIDAD DE SERVICIO EN REDES DE COMPUTADORES

Campo de Estudio: *Telecomunicaciones*

Biografía:

Datos Personales: Nacido en Monterrey, Nuevo León el 16 de Febrero de 1980, hijo de José Ángel Castilleja Herrera y María Margarita García Nieto.

Educación: Egresado de la Universidad Autónoma de Nuevo León, de la Licenciatura en Ciencia Computacionales de la Facultad de Ciencias Físico Matemáticas en Agosto de 2002.

Experiencia Profesional: Integrante del Laboratorio de Interoperabilidad de la Dirección de Sistemas e Informática de la UANL desde Abril del 2000 participando en el proyecto de Internet 2 en México.