

CALIDAD DE SERVICIO EN REDES IPv4 Y SU SIMULACIÓN EN NS-2

Jaime Andrés Vallejo Avellaneda

Estudiante de Ingeniería electrónica de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia.
javallejoa@correo.udistrital.edu.co

Jordi Orlando Rico Rodríguez

Estudiante de Ingeniería electrónica de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia.
joricor@correo.udistrital.edu.co

Tipo: Artículo reporte de caso

Fecha de Recepción: Octubre 14 de 2011

Fecha de Aceptación: Diciembre 18 de 2011

QUALITY OF SERVICE ON IPv4 NETWORKS AND ITS CORRESPONDING NS2-BASED SIMULATIONS

ABSTRACT

This paper presets some of the characteristics that are present when attempting to guarantee Quality of Service (QoS) (e.g. bandwidth, packet loss) over a network topology that involves three types of traffic, namely CBR, Pareto y Exponential. Five configurations are implemented using NS2, the first corresponds to the case where no QoS is required and the channel capacity is large enough to cope with the three types of traffic, while in the four remaining configurations channel capacity is reduced in order to study QoS performance under different conditions.

In order to observe traffic behavior, a network without QoS is simulated including two types of queuing policies. Subsequently, a different configuration that includes QoS is deployed by using models like IntServ and DiffServ.

Results are plotted and compared to determine the most convenient traffic behavior between IP networks.

Keywords: Quality of Service, Differentiated Services, Integrated Services

RESUMEN

Se evalúan algunas características presentadas a la hora de garantizar calidad de servicio como son el ancho de banda y la pérdida de paquetes, sobre una topología de red que involucra tres tipos de tráfico (CBR, Pareto y Exponencial). Se implementan cinco configuraciones diferentes en el simulador de redes NS2, donde el primero corresponde a un caso sin QoS, con una capacidad de canal que es suficientemente grande para transmitir los tres tipos de tráfico, mientras que en las siguientes cuatro configuraciones la capacidad se reduce a un menor tamaño y se estudian diferentes casos de calidad de servicio.

Para observar el comportamiento de los tráficos se implementa inicialmente una red sin QoS con dos tipos de encolamiento y posteriormente se realiza una configuración que implementa QoS mediante los modelos IntServ y DiffServ.

Los resultados obtenidos de estas simulaciones son graficados y comparados con el fin de determinar el comportamiento más adecuado en redes IP.

Palabras clave: Calidad de servicio, Servicios Diferenciados, Servicios Integrados

1. INTRODUCCIÓN

Con el transcurso de los años han surgido nuevas tecnologías en Internet, las cuales requieren cierto grado de calidad del servicio para poder implementarse y utilizarse, ya que de lo contrario sería algo tedioso su uso, podrían presentar fallos y no prosperarían. Es por ello que se da la necesidad de implementar nuevos modelos referentes a QoS, los cuales garanticen que la información enviada llegue a su destino de forma correcta y de esta manera poder usar aplicaciones como lo son la voz sobre IP, el video a distancia, la realización de conferencias, entre otras.

Para medir esta variable de red tan importante se deben tener en cuenta cuatro factores importantes los cuales son el ancho de banda (AB), buscando que sea lo más grande posible sin representar costos excesivos, la pérdida de paquetes, el retardo end-to-end y el jitter, que por el contrario se desea que sean mínimos.

Para ello se han implementado dos modelos de arquitecturas como extensión al protocolo IP; estas extensiones reciben el nombre de Servicios Diferenciados (DiffServ) y de Servicios Integrados (IntServ).

En el caso de IntServ, se reservan recursos en todos los nodos de la red por los que se enviarán los datos. Por su parte, Diffserv, clasifica la información en grupos a los cuales se les asignará un valor de prioridad diferente; esto para otorgar una mayor seguridad de recepción a los paquetes con mayor prioridad.

Para observar el comportamiento de estos dos modelos, se recurre al uso de la herramienta Network Simulator, la cual ofrece la oportunidad de variar parámetros y poder observar mediante gráficas, el comportamiento de la red para cada caso.

Este artículo inicia con una breve definición de los principales temas sobre los que se soporta la investigación. Seguidamente se presenta la topología que se usará para poner a prueba estos modelos, las simulaciones realizadas, y

finalmente se presentan los resultados obtenidos.

2. INTERNET PROTOCOL (IP)

IP es un protocolo no orientado a la conexión, utilizado tanto por el emisor como por el receptor para el intercambio de datos por medio de una red no fiable (llamada de mejor esfuerzo - best effort) de la mejor forma posible pero sin garantías, debido a que no posee ningún mecanismo que determine si un paquete ha llegado a no a su destino [1].

Es por esto que la fiabilidad requerida debe ser proporcionada por la capa de transporte.

Este protocolo utiliza generalmente Drop Tail como algoritmo gestor de colas para decidir cuándo descartar paquetes, en donde, en contraste con algoritmos más complejos como RED y WRED[2], el tráfico no es diferenciado, es decir, que cada paquete es tratado de la misma forma que los demás, por lo cual, en el momento que el canal se llena a su máxima capacidad, los nuevos paquetes son eliminados hasta que haya más espacio en cola para aceptar el tráfico entrante.

No obstante, la pérdida de los datagramas causa que el emisor entre en un modo lento, reduciendo así el rendimiento en esa sesión, hasta que este recibe nuevamente reconocimientos de sus datagramas y aumente su ventana de congestión. Esto se debe a que en lugar de descartar muchos segmentos de una conexión, el router tiende a descartar un segmento de cada conexión [3].

3. CALIDAD DE SERVICIO

Hace referencia a la capacidad de una red, de reservar recursos para un tráfico con el fin de proporcionar un servicio [4].

En la actualidad existen dos tipos de tecnologías de QoS asociadas hasta cierto punto directamente con IP. La primera de ellas denominada servicios diferenciados, se caracteriza por marcar los paquetes de acuerdo a su prioridad,

mientras que la segunda tecnología, denominada servicios integrados, se basa en reserva y asignación de recursos dependiendo del tipo de tráfico [4].

3.1. Diffserv

Tiene como finalidad dar una mayor confiabilidad de envío de paquetes, clasificándolos por la información que manejan y dándole a estos grupos un valor de prioridad (clasificación por clases o agregados). Con esto se busca, por una parte dar una mayor seguridad de recepción de los paquetes con mayor prioridad y por la otra, evitar los excesivos procesamientos dados en los routers troncal, trasladándolos a los extremos de la red [5].

Para el proceso de acondicionamiento, inicialmente se ubican los paquetes a transmitir ubicados en los nodos de acceso. Para cada uno se configura un PHB (Per-hop-behaviour), que definirá las políticas de envío [5], para lo cual se marca cada uno de los tráficos con un valor determinado y se asigna a este un número de cola física y virtual.

Posteriormente a cada pareja cola física-cola virtual, se les establece unos valores de configuración de acuerdo a la prioridad que se desee. Esto es así, debido a que el manejo de esta variable en DiffServ se realiza mediante RED (descarte aleatorio temprano) [6].

Mientras que Drop-Tail descarta todo el tráfico excesivo implicando una retransmisión, RED descarta paquetes aleatoriamente hasta llegar a un umbral establecido. Con esto se logran reducir las congestiones y se transmite de una forma más rápida [7].

Esta configuración consta de 3 parámetros:

- El primer parámetro corresponde a la pareja de cola física y virtual. El valor inicial indicará el número de cola física y el siguiente, el de cola virtual.
- El segundo parámetro define el rango de ocupación del gestor de paquetes sobre el que se aplicarán los valores de prioridad.

Este se define con un valor mínimo, y un valor máximo de paquetes y se representan como Min y Max [8].

- El tercer parámetro hace referencia a la probabilidad que se tiene de descartar un mensaje y estará dado en porcentaje. La probabilidad de descarte irá aumentando en el canal hasta llegar como máximo a este valor.

Finalmente, una vez encontrado y clasificado un flujo se verifica si cumple con los requisitos establecidos en los SLS (Service level specifications) y si es así, es enviado por la red; de lo contrario se descarta o se marca como no confirmado.

3.2. Intserv

Este modelo de arquitectura tiene como fin principal el gestionar los servicios necesarios para garantizar QoS [9], para lo cual se hace una reserva de recursos y mecanismos de control de admisión [10].

Este modelo surge con el fin de implementar nuevas aplicaciones en tiempo real como lo son las conferencias multimedia, el video a distancia y la realidad virtual.

Para que exista IntServ es necesario que los host soliciten una QoS específica a la red para una aplicación o un flujo de datos en particular, adicionalmente los routers deben enviar estas solicitudes a todos los nodos de la red a lo largo de la trayectoria [11]. Para ello se utiliza el protocolo de reserva de recursos (RSVP).

RSVP controla el envío de paquetes para garantizar la calidad del servicio, desde el emisor hacia el receptor, cruzando por todos sus nodos intermedios, trabajando sobre IP y reservando recursos en cada nodo de la ruta.

Para ello una sección de RSVP consta de la dirección de destino, un identificador del protocolo IP y del puerto de destino [13].

Como cada emisor desea transmitir intenta hacer reservas en cada nodo intermedio, pero

esta solicitud sólo será atendida si en cada dispositivo es el solicitante con más prioridad.

4. TOPOLOGÍA A SIMULAR

La topología de la red a simular se visualiza en la Fig. 1.

Esta red se compone de tres fuentes de tráfico con sus respectivos equipos destino donde el núcleo de la misma se compone de un dispositivo de capa 3 central con dos enlaces que lo conectan a un par de DTEs en los extremos.

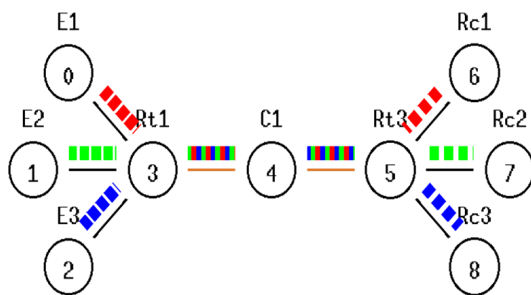


Fig. 1. Topología simulada

Las fuentes están conectadas a los routers mediante enlaces de 2Mbps con un retardo de 10 ms, mientras que en enlace troncal se integra a la estructura mediante canales de 4Mbps con delay de 10 ms y transferencia de tráfico full-duplex.

Cada una de las fuentes corresponde a un flujo distinto, los cuales iniciarán y terminarán en tiempos diferentes con el objetivo de observar el comportamiento del canal al adicionar los tráfico y la forma de funcionar al llegar a la saturación del mismo.

Al flujo de datos simulados tiene las siguientes características:

- El primero se identifica con color rojo y corresponde a un tráfico CBR (tasa constante de bits). Puede utilizarse en servicios como telefonía y videoconferencias [14]. Sin embargo depende de la sincronización de tiempo entre la fuente y el destino y utiliza gran parte del ancho de banda del canal, generando congestiones. Para el caso de la

simulación éste, empezará a ser transmitido a partir de 0.5 s finalizando a los 4.5 s y manejará paquetes de 500 bytes con un uso de 1.6 Kbps. Adicionalmente será el tráfico de mayor prioridad en la implementación de los dos modelos de calidad de servicio.

- El segundo se representa con color verde y es un flujo tipo Pareto [15]. Se caracteriza por largas ráfagas causadas por manejos de grandes paquetes [16]. Se manejarán igualmente mensajes de 500 bytes con un uso de ancho de banda de 1.6Kbps y serán enviados en el intervalo de 1 s a 4 s. Además tendrá una prioridad intermedia al implementar el modelo DiffServ.
- El último tipo de tráfico es el de color azul y tiene un comportamiento tipo exponencial. Su nombre se debe a que los intervalos de llegada de paquetes tienen una distribución de esta forma [8]. El tamaño de los mensajes y requerimientos de capacidad es el mismo que los casos anteriores y su tiempo de simulación esta entre los 1.5 a los 3.5seg.

Cada uno de estos tráfico tiene varios parámetros de configuración, los cuales se presentan en la Sección 5 correspondiente a Simulaciones.

Para analizar y estudiar IP se realizaron cuatro simulaciones:

- En la primera se presenta un escenario sin QoS, en donde no se generan pérdidas por saturación en el canal. Para ello se tiene en cuenta la capacidad requerida por las tres fuentes, la cual suma 4.8Mbps. Como el ancho de banda de la red troncal es de 4Mbps, se hizo necesario aumentarlo. Para el caso de estudio se aumentó a 6Mbps.
- La segunda simulación corresponde a los requerimientos establecidos al comienzo de esta sección, es decir, una red sin QoS, con un requerimiento para la red troncal de 4Mbps y manteniendo los otros parámetros iguales. En este caso se puede ver que el ancho de banda solicitado por los tráfico es mayor al disponible, con lo cual se ocasionarán pérdidas durante la transmisión. En este escenario se implementa encola-

miento Drop-Tail.

- La tercera evaluación, implementa QoS utilizando el modelo DiffServ. Aquí se le dará mayor prioridad a CBR, asignándole una probabilidad de pérdida de paquetes del 1% como máximo. Pareto tendrá una prioridad intermedia donde la probabilidad de pérdida aumentará hasta llegar a un 40%. Finalmente la fuente exponencial tendrá la prioridad más baja, asignándosele una probabilidad de pérdida de paquetes máxima de 90%.
- Por último, en la cuarta simulación se implementará el modelo IntServ, en donde se dará mayor prioridad CBR, garantizando que siempre tendrá el ancho de banda que requiere (1.6 Mbps), mientras que los 2.4 Mbps restantes serán compartidos entre Pareto y Exponencial

5. SIMULACIONES

Para crear la topología requerida y simular su comportamiento se utiliza el programa Network Simulator 2. Los resultados serán analizados mediante gráficas obtenidas con el complemento Xgraph y tablas con los datos finales. Para esto se debe crear inicialmente un archivo .TCL donde se programarán todos los aspectos de la red, como son sus nodos, las conexiones y los tipos de tráfico que se utilizarán.

A continuación se presentan algunos de los aspectos más relevantes de la configuración de cada simulación.

5.1. IP sin QoS – Garantía de ancho de banda

Inicialmente se crean todos los nodos requeridos y se define si servirán como fuentes y receptores de tráfico o como routers. Seguidamente se generan las conexiones como se aprecian a continuación:

```
$ns duplex-link $n0 $r3 2Mb 10ms DropTail
$ns duplex-link $c4 $r3 6Mb 10ms SFQ
$ns duplex-link $r5 $c4 6Mb 10ms SFQ
$ns duplex-link $n6 $r5 2Mb 10ms DropTail
```

Cada uno de los enlaces se configura como bidireccional asignando un tiempo de retardo y un ancho de banda.

Se puede apreciar que la capacidad de canal de la red troncal es suficiente para que los tres tráficos pasen por el sin tener ningún tipo de pérdidas. Por esta razón no importará el gestor de cola utilizado en este caso, aspecto que si afectará en el siguiente escenario [17].

El siguiente paso consiste en crear las fuentes de transmisión sobre agentes de tráfico UDP y adjuntarlas a los nodos correspondientes de la siguiente forma:

```
set udp1 [new Agent/UDP]
$ns attach-agent $n0 $udp1
```

Por último se definen los tráficos mencionados en la sección anterior:

Para CBR debe configurarse el tamaño de los paquetes que será de 500 bytes y el intervalo entre cada envío o tasa de transmisión de 1,6Mbps. Este tráfico luego se adjunta al agente UDP creado [18].

```
set cbr [new Application/Traffic/CBR]
$cbr set packetSize_ 500
$cbr set interval_ 0.0025
$cbr attach-agent $udp1
```

En el caso de Pareto se configura de igual forma el tamaño de los paquetes, la velocidad de envío y adicionalmente se incluye el tiempo de ráfaga (burst time) y el tiempo muerto durante el cual no habrá transmisión de la fuente (idle time). La longitud de los paquetes y la velocidad de transmisión se mantienen igual al tráfico CBR mientras que el tiempo de ráfaga se establece en 1 ms y el tiempo muerto en 0.2 ms. Con esto se tendrán tiras cortas sin esperar mucho tiempo entre ellas.

```
set pareto [new Application/Traffic/Pareto]
$pareto set packetSize_ 500
$pareto set burst_time_ 1ms
$pareto set idle_time_ 0.2ms
$pareto set rate_ 1600k
$pareto attach-agent $udp2
```

El Exponencial se configura de la misma forma que el tráfico Pareto cambiando únicamente los valores y el nombre de la aplicación. En este caso tanto el tiempo de ráfaga como el tiempo muerto se aumentarán, pasando a ser 1 ms y 2 ms respectivamente. La consecuencia de esto será un flujo donde en ocasiones no se utilizará el enlace y se verá gráficamente representado como varios saltos.

```
set expon [new Application/Traffic/Exponential]
$expon set packetSize_ 500
$expon set burst_time_ 2ms
$expon set idle_time_ 1ms
$expon set rate_ 1600k
$expon attach-agent $udp3
```

Una vez generada la topología y definidos los tráfico se establecen los tiempos de simulación y se procede a compilar el programa creado.

```
$ns at 0.5 "$cbr start"
$ns at 1.0 "$pareto start"
$ns at 1.5 "$expon start"
$ns at 3.5 "$expon stop"
$ns at 4.0 "$pareto stop"
$ns at 4.5 "$cbr stop"
$ns at 5.0 "finish"
$ns run
```

Los resultados y comparación de este y los demás modelos se apreciarán en la siguiente sección

5.2. IP sin QoS – Encolamiento Drop-Tail

En este caso, se mantiene la configuración definida anteriormente, cambiando únicamente los requerimientos de la red troncal, pasando a ser 4Mbps con el fin de ocasionar saturación del canal:

```
$ns duplex-link $c4 $r3 4Mb 10ms Drop Tail
$ns duplex-link $r5 $c4 4Mb 10ms Drop Tail
```

Cabe destacar que aquí se eliminarán los datos excedentes provenientes de la última fuente en transmitir hasta el momento en que se descongestione el canal de transmisión.

5.3. QoS Diffserv

NS2 implementa servicios diferenciados mediante la configuración de dos nodos de red: El nodo frontera y el nodo troncal o de núcleo [5]. Por lo tanto la topología se definirá de la misma forma que sin QoS modificando únicamente las conexiones entre los routers de la siguiente forma:

```
$ns simplex-link $r3 $c4 4Mb 10ms dsRED/edge
$ns simplex-link $c4 $r3 4Mb 10ms dsRED/core

$ns simplex-link $c4 $r5 4Mb 10ms dsRED/core
$ns simplex-link $r5 $c4 4Mb 10ms dsRED/edge
```

Los canales de comunicación bidireccionales configurados anteriormente se convierten en dos enlaces unidireccionales donde uno irá desde la frontera hasta el núcleo y el otro estará en sentido contrario.

El manejo de colas se hará mediante RED como se especificó en la Sección 3 configurando para esto el módulo dsRED y especificando el sentido de la conexión mediante “edge” y “core”.

A continuación, a cada una de las uniones se le asigna una variable:

```
set qE1C [[ $ns link $r3 $c4 ] queue]
set qCE1 [[ $ns link $c4 $r3 ] queue]
set qCE2 [[ $ns link $c4 $r5 ] queue]
set qE2C [[ $ns link $r5 $c4 ] queue]
```

Finalmente se diferencian los servicios y se establecen las políticas para DiffServ. Estas consisten en un identificador el cual se tomará como 10, 20 y 30 para los 3 gestores virtuales y dos valores “cir” y “cbs”, donde el primero determinará la velocidad media de transmisión y el segundo el volumen de tráfico alcanzable enviando a la velocidad determinada. Esto se agregará a una tabla de políticas requerida para este modelo:

```
$qE1C addPolicyEntry [$n0 id] [$n6 id] TokenBucket 10 $cir0 $cbs0
```

Seguidamente, se asigna una ruta cada identificador y esto se almacena en una segunda tabla

llamada `PolicerEntry`:

```
$qE1C addPolicerEntry TokenBucket 10 11
$qE1C addPolicerEntry TokenBucket 20 21
$qE1C addPolicerEntry TokenBucket 30 31
```

Una vez hecho esto se asigna la pareja de cola física-virtual al identificador como se había mencionado en la Sección 3:

```
$qE1C addPHBEntry 10 0 0
$qE1C addPHBEntry 20 0 1
$qE1C addPHBEntry 30 0 2
```

Y finalmente se establecen los parámetros de configuración:

```
$qE1C configQ 0 0 1 20 0.01
$qE1C configQ 0 1 1 20 0.4
$qE1C configQ 0 2 1 20 0.9
```

Esto mismo se realiza para cada una de las variables de unión

5.4. QoS Intserv

Aquí, debe reservarse y garantizarse un ancho de banda para un tráfico o tráficos determinados. En este caso se desea priorizar el flujo CBR como se menciona. Para esto se reservan los 1.6 Mbps que requiere con el fin de no tener pérdidas, mientras que los otros flujos se repartirán la capacidad restante del medio de transmisión.

Inicialmente se configura la conexión y se determina el porcentaje de rango de frecuencia reservable para la aplicación. Además se configurarán los parámetros de retardo de acceso y el ancho de banda asignado a los paquetes de mejor esfuerzo, que para este caso en particular se establecen en cero:

```
proc create_link {src_node dst_node
rate} {
global ns
set delay 10ms
set reservable 0.9
set rsvp_rate 0
set bo_queue_size 5000
$ns duplex-rsvp-link $src_node $dst_
node $rate $delay $reservable $rsvp_
```

```
rate $bo_queue_size Param Null; }
```

A continuación se utiliza el proceso entre los nodos requeridos y se asigna una tasa de transmisión a cada enlace de la siguiente forma:

Conexión fuente a router:

```
create_link $n0 $r3 2Mb
```

Conexión Red troncal:

```
create_link $r3 $c4 4Mb
```

Y se adjunta un agente RSVP a cada uno de los nodos:

```
set rsvp0 [$n0 add-rsvp-agent]
```

Finalmente se configura la sesión RSVP para el enlace que se desea priorizar. Para esto se crea una nueva sesión a la cual se le asignará un identificador y en donde se debe especificar el agente de origen, el nodo de destino y un indicador de flujo.

Ya que el tráfico prioritario será CBR, el agente de origen será `rsvp0` y el nodo de destino será `n6`:

```
set rsvp_session0 [$rsvp0 session $n6
$flow_id0]
```

Para dar inicio a la sesión RSVP, se debe especificar el agente de envío y la reserva que se hará. En el agente de envío se definirá la velocidad de transmisión, establecida en 1600 kbps así como el bucket size que representará el número de muestras que podrá acumular el flujo:

```
$ns at 0.01 "$rsvp0 sender $rsvp_ses-
sion0 +16000000 5000"
```

Por su parte la reserva se hará sobre el agente RSVP que servirá de receptor y aquí es donde se definirá la velocidad requerida por el tráfico prioritario, que en este caso será de 1.6Mbps, de igual forma se define el bucket size y finalmente se define el nodo origen:

```
$ns at 0.1 "$rsvp6 reserve $rsvp_ses-
sion0 FF +1600000 5000 $n0"
```

6. ANÁLISIS DE RESULTADOS.

6.1. IP sin QoS

Garantía de ancho de banda. Tras compilar el programa creado se obtienen una ventana de simulación correspondiente al Network Animator (NAM), donde se verá de forma dinámica el comportamiento de los paquetes a través de los canales y la pérdida de estos al exceder la capacidad del ancho de banda.

En este escenario el ancho de banda utilizado por cada tráfico corresponde al máximo teórico determinado y por consiguiente las pérdidas son nulas.

En la fig. 2 se aprecia el uso del canal para este caso. El tráfico CBR al ser continuo se representa como un uso constante del ancho de banda requerido, mientras que los otros dos, al tener tiempos muertos, presentan saltos durante los cuales su uso desciende. Por esto se ven varios picos durante su transmisión.

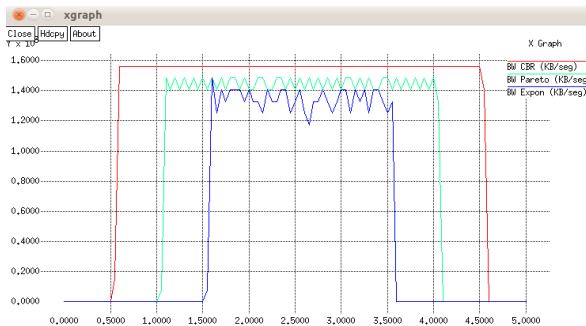


Fig. 2. Ancho de banda utilizado (escenario sin QoS)

Por otra parte, a través de la fig. 3 se puede comprobar que no existen pérdidas para este caso.

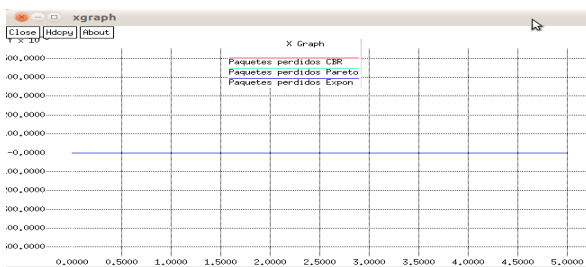


Fig. 3. Cantidad de pérdida de paquetes (escenario sin QoS)

```
Número de paquetes enviados = 3396
Número de paquetes enviados E1 = 1600
Número de paquetes enviados E2 = 1111
Número de paquetes enviados E3 = 685
-----
Número de paquetes recibidos = 3396
Número de paquetes recibidos R1 = 1600
Número de paquetes recibidos R2 = 1111
Número de paquetes recibidos R3 = 685
```

Fig. 4. Verificación paquetes enviados y recibidos (escenario sin QoS)

6.2. IP sin QoS - Drop Tail

La utilización del ancho de banda seguirá siendo la misma para cada tráfico. Sin embargo, como la distribución no será justa, cada uno utiliza el AB que encuentre disponible. Solo cuando las tres fuentes transmitan al mismo tiempo, la capacidad será de 1.33 Mbps.

Por otra parte, al utilizarse Drop-Tail para el manejo de colas, el tráfico excesivo se acumula y se descarta. Al descongestionarse el router se da paso a una retransmisión, donde los paquetes descartados causan una “ráfaga retardada” que se puede apreciar en la Fig. 5 como una utilización cercana a 2Mbps durante un corto tiempo.



Fig. 5. Ancho de banda utilizado (sin QoS - Drop Tail).

En este caso no se clasifica la información excedente, sino que se descarta en el momento en que se sature el canal. Aquí se pierden menos paquetes de CBR y de Pareto, mientras que en el tráfico Exponencial aumentan las pérdidas (Fig. 6).

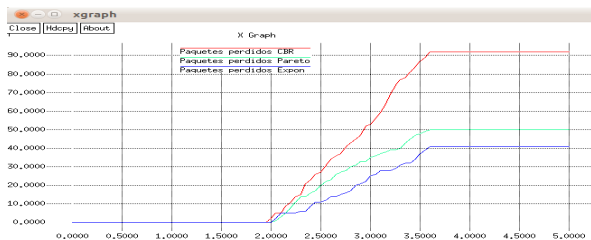


Fig. 6. Pérdida de paquetes (sin QoS - Drop Tail)

En total se transmitieron 3396 paquetes de los cuales 3217 fueron recibidos (Fig. 7). El tráfico CBR perdió 92 paquetes, Pareto perdió 50 y el tráfico exponencial aumentó sus pérdidas a 37 paquetes.

```

Número de paquetes enviados = 3396
Número de paquetes enviados E1 = 1600
Número de paquetes enviados E2 = 1111
Número de paquetes enviados E3 = 685
-----
Número de paquetes recibidos = 3217
Número de paquetes recibidos R1 = 1508
Número de paquetes recibidos R2 = 1061
Número de paquetes recibidos R3 = 648

```

Fig. 7. Cantidad de paquetes enviados y recibidos (Sin QoS – Drop Tail)

En cuanto al retardo de envío extremo a extremo (End-to-end Delay) presentado en la Fig. 8, se puede ver que es muy similar para los tres tráficos.

En el intervalo de 0 s hasta 0.5 s el retardo es inexistente ya que no hay tráfico.

A partir de 0.5 s, cuando CBR empieza a transmitir el retardo es constante e igual a 46 ms y se mantiene hasta 1 s cuando empieza a transmitir el flujo Pareto.

En este momento el retardo no aumenta significativamente pues no se ha ocupado la capacidad en su totalidad, por lo que se obtiene un valor cercano a 46 ms con algunos picos hasta 47 ms. Este retardo se da para las dos primeras fuentes.

Desde los 1.5 s cuando inicia la transmisión del tráfico exponencial, el retardo aumenta para los tres flujos aunque de forma distinta para cada uno, estableciéndose en un valor promedio de 68 ms. Sin embargo como se puede ver en la Fig. 8, el tráfico exponencial presenta una disminución de retardo en el intervalo 2.5 s a 3 s, debido a una utilización mayor de AB.

En el momento en que la última fuente deja de transmitir, el retardo de las demás vuelve a disminuir y llega al valor que se obtuvo al inicio de 46 s.

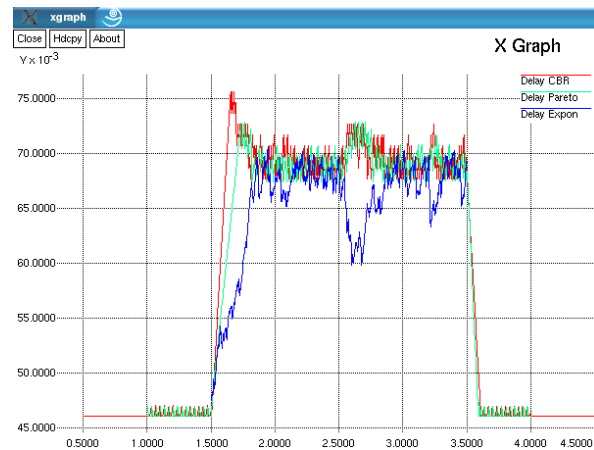


Fig. 8. Comportamiento del retardo.

El jitter por su parte hace referencia a la desviación respecto al valor de retardo inmediatamente anterior y se requiere que sea un valor pequeño ya que sirve como indicador de sincronismo en la transmisión de paquetes. Si este se hiciera muy grande indicaría que se están presentando saltos no deseados durante la transmisión por posibles fallas de sincronización.

Los valores negativos de jitter indican que el retardo actual fue menor que el precedente, es decir que la transmisión fue más rápida, mientras que un valor positivo indica que la transmisión tardó más que la anterior.

De las Fig. 9 a la 11 se observa el jitter para cada uno de las fuentes.

El primero de ellos corresponde a CBR (Fig. 9). Se puede ver que este inicia a partir de 0.5 s que es el momento en el cual este tráfico empieza a enviarse. Desde este instante hasta 1 s se mantiene en 0 ms pues ningún otro flujo está haciendo uso de la red.

En el momento en que empieza a enviar la segunda fuente, este valor de jitter aumenta, variando en un rango de 0 ms hasta aproximadamente 6 ms.

Al ser transmitido el último tráfico desde 1.5 ms, aumenta de nuevo y se mantiene en un valor promedio cercano a 0 ms, es decir, que el retardo para este tendrá mucha variación. Si se

vuelve a observar la Fig. 8 se puede ver que este análisis es correcto ya que el retardo para CBR se mantiene en un promedio de 70 ms.

Al dejar de transmitirse los tráficos Pareto y exponencial, el jitter disminuye, volviendo al valor inicial.

El siguiente jitter corresponde al dado en el caso Pareto (Fig. 10). Este inicia a partir de 1 s pero no lo hace desde cero, puesto que ya se estaba transmitiendo el tráfico anterior. Su valor es similar al del CBR en este mismo rango.

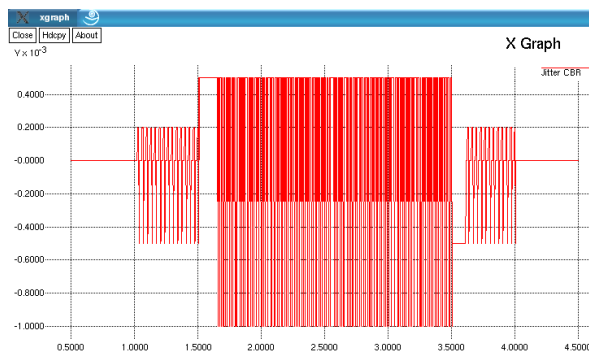


Fig. 9. Comportamiento del Jitter (caso CBR)

En el momento que se inicia la tercera transmisión aumenta y se mantiene igualmente en un valor promedio cercano a cero. Es decir que el retardo se mantendrá casi constante. Al igual que en el caso anterior se puede comprobar el resultado remitiéndose a la Fig. 8.

La Fig. 11 corresponde al jitter obtenido para el tráfico exponencial. Siguiendo con los intervalos de envío, este inicia en 1.5 s y lo hace en un valor cercano a cero, el cual aumenta rápidamente debido a la presencia de las dos fuentes de tráfico anteriores.

Este no tiene un valor promedio de cero sino que es muy variable, lo cual se ve representado en variaciones abruptas del retardo, particularmente entre 2.5 s y 3 s se obtiene una ráfaga negativa que corresponde a la disminución en

el retardo visto en la Fig. 8.

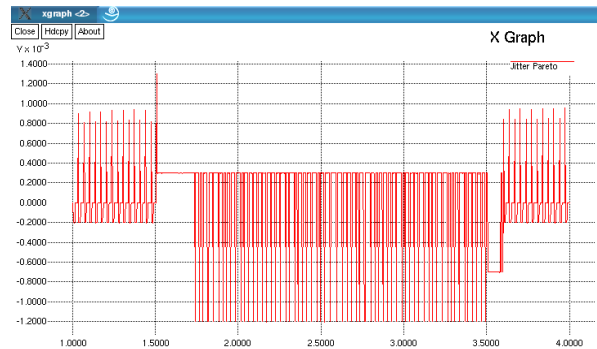


Fig. 10. Comportamiento del Jitter (caso Pareto)

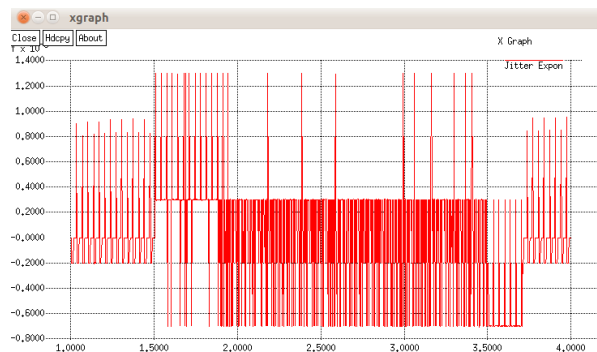


Fig. 11. Comportamiento del Jitter (caso Exponencial)

6.3. QoS Modelo Diffserv

El ancho de banda utilizado por cada usuario es el mismo que el de la red sin calidad de servicios. Sin embargo, en la sección anterior se le asignó un valor de prioridad (Probabilidad de pérdida de paquetes) a cada uno, el cual indica qué cantidad de paquetes deben descartarse al momento de saturarse el canal.

En el caso de la topología tratada, se le dio mayor prioridad a CBR y menor al Exponencial, viéndose representado como un mayor uso de ancho de banda por parte del primero, y grandes caídas para el segundo. Si se promedia el uso de ancho de banda del tráfico CBR se obtiene una utilización de los 1.6Mbps requeridos (Fig. 12)

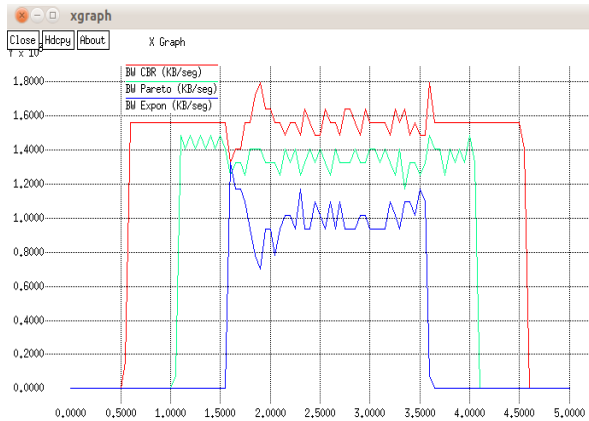


Fig. 12. AB en DiffServ

El mismo análisis aplica para las pérdidas de datos, ya que las obtenidas para CBR son nulas a diferencia de las otros dos (Fig. 13).

En total se transmitieron 3402 paquetes de los cuales 3161 fueron recibidos. Las pérdidas totales fueron mayores que las del caso anterior, así que debe estudiarse si este resultado es aceptable para la aplicación requerida. Esto se aprecia en la Fig. 14.

El CBR no tuvo pérdidas, mientras que Pareto perdió 64 paquetes y el tráfico exponencial aumentó sus pérdidas a 177.

En los resultados obtenidos en la gráfica 14, "ldrops" hace referencia a la pérdida causada por el desbordamiento del enlace entre la fuente y el router, mientras que "edrops" es el número de paquetes descartados por la RED.

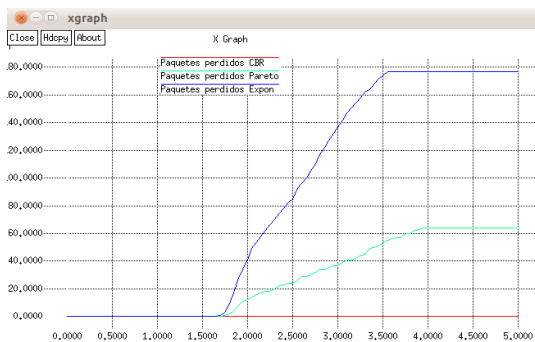


Fig. 13. Cantidad de paquetes perdidos en DiffServ

CP	TotPkts	TxPkts	ldrops	edrops
All	3402	3161	0	241
10	1600	1600	0	0
20	1111	1047	0	64
30	691	514	0	177

Fig. 14. Cantidad de paquetes enviados y recibidos - Diffserv

Un detalle importante en este modelo es que la prioridad dada a cada tráfico debe mantenerse aunque el prioritario haya dejado de transmitir. Esto quiere decir que si únicamente se encuentran los tráficos de prioridad media y baja, el primero tendrá garantizada la probabilidad de pérdida establecida.

El retardo en este caso fue variable e igual para los tres casos.

Al igual que en el caso sin QoS, CBR presenta un retardo de 0ms ya que no se transmiten más tráficos. Y posteriormente aumenta al entrar a transmitir la fuente Pareto.

El cambio se presenta cuando la fuente Exponencial empieza a generar información ya que el canal se congestiona y todos los paquetes sufren un retardo mientras se define el canal con mayor prioridad. Posteriormente este retardo disminuye al llegar todos los paquetes a su destino y en este momento busca estabilizarse en un valor cercano a 50 ms.

El modelo DiffServ garantiza prioridad de envío para un tipo de datos. Sin embargo, ya que no se está reservando ancho de banda para este, los otros tráficos ocupan también el canal presentándose este retardo (Fig. 15).

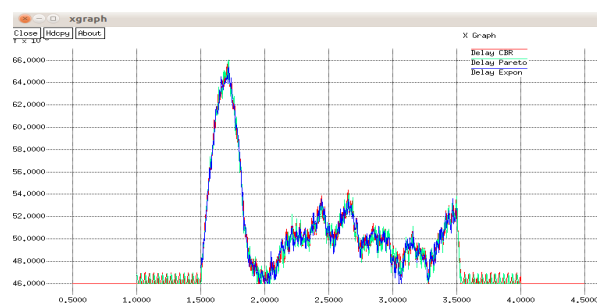


Fig. 15. Retardo Vs Tiempo de envío en el caso DiffServ

Viendo lo obtenido en cuanto a retardo, se puede deducir que el jitter será similar en todos los tráficos, cambiando tan solo en los tiempos de inicio. Este jitter tendrá en cada uno de ellos un valor máximo de 0.5 ms y se mantendrá en constante cambio de signo indicando una variación continua en el retardo. Las Fig. 16, 17 y 18 presentan los valores de jitter obtenidos

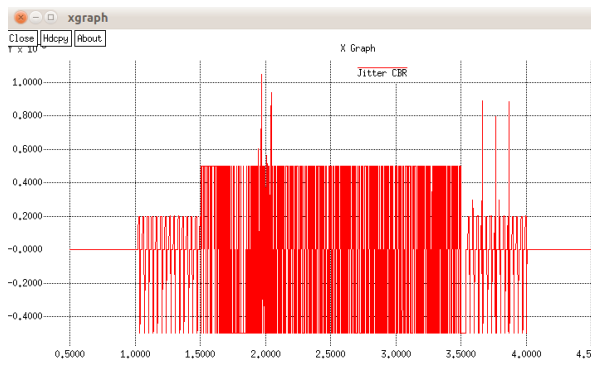


Fig. 16. Jitter cuando la cola es tipo CBR

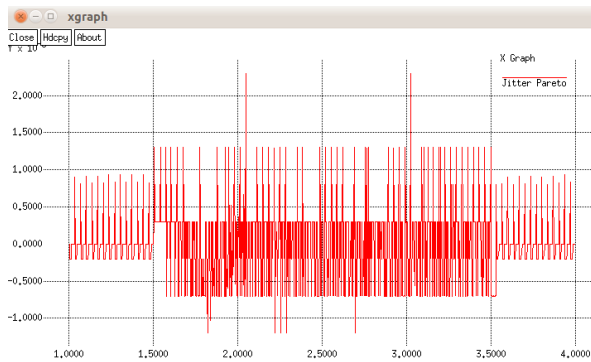


Fig. 17. Jitter cuando la cola es tipo Pareto

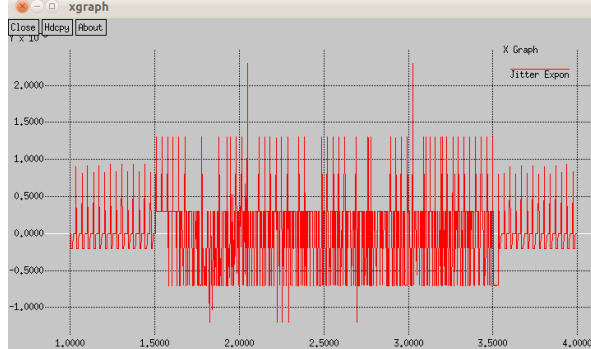


Fig. 18. Jitter cuando la cola es tipo Pareto exponencial

6.4. Intserv

La capacidad del canal utilizado por cada tráfi-

co es el mismo que el empleado en la red sin calidad de servicios y con implementación de Servicios Diferenciados. Con la única diferencia de la prioridad asignada a cada secuencia de datos dependiendo de su tipo, puesto que para esta topología se garantizó el ancho de banda que requería el tráfico CBR y a los otros dos tipos de flujos se les ha dejado libres para repartirse el resto del canal (Fig. 19).

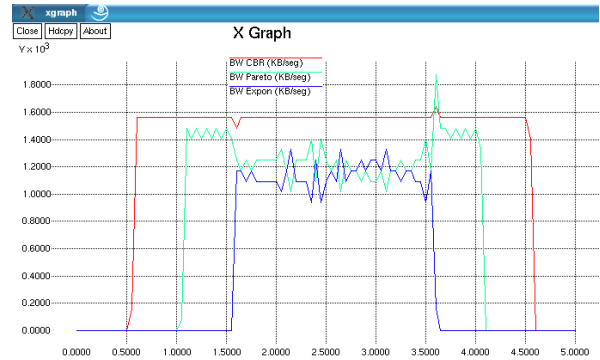


Fig. 19. Ancho de banda utilizado (Kbps) en IntServ

Así mismo se puede emplear el análisis mencionado con respecto al AB, ya que se observa que las pérdidas de paquetes para el tráfico CBR es nulo a diferencia de los otros 2. Particularmente se evidencia una mayor pérdida de datos para el caso Pareto; esto se debe a que este tipo de tráfico envía muchos más paquetes que el tráfico exponencial, por ende la probabilidad de perderlos es mayor (Fig. 20).

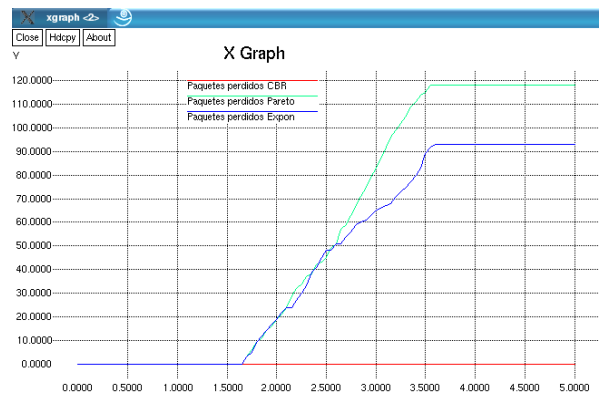


Fig. 20. Comportamiento de los flujos de pérdidas

En total se transmitieron 3401 paquetes de los cuales 3179 fueron recibidos. CBR no perdió

ningún dato debido a que se le garantizó el AB que requería, Pareto perdió 118 y el tráfico exponencial 104 bytes. (Fig. 21).

Número de paquetes enviados	= 3401
Número de paquetes enviados E1	= 1600
Número de paquetes enviados E2	= 1111
Número de paquetes enviados E3	= 690
Número de paquetes recibidos	= 3179
Número de paquetes recibidos R1	= 1600
Número de paquetes recibidos R2	= 993
Número de paquetes recibidos R3	= 586

Fig. 21. Cantidad de paquetes enviados y recibidos para IntServ

La Fig. 22 muestra el comportamiento del retardo con fairserv. Inicialmente se observa que este es igual a 46 ms para el caso Pareto cuando solo esta fuente transmite. Una vez comienza a funcionar los demás emisores este retardo se ve afectado de forma mínima ya que solo sube hasta 48 ms. No obstante para los otros dos casos esta variable se eleva hasta los 62 ms una vez el canal entra en congestión.

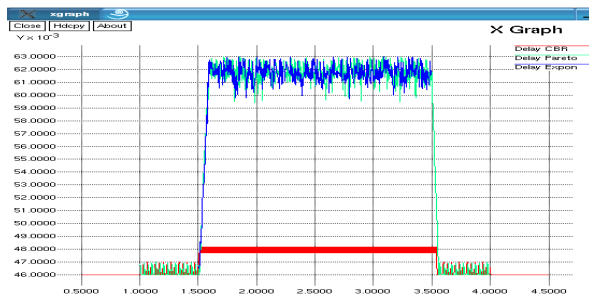


Fig. 22. Comportamiento del retardo con IntServ

Por su parte, el jitter para el tráfico prioritario (CBR) inicia de nuevo en 0.5 s y aumenta al transmitir los otros tráficos. Sin embargo, en este caso tiene un valor máximo muy pequeño de 500 us con un promedio de 0 us. Esto ocurre debido a que al reservar y garantizar un ancho de banda para un determinado tráfico, este se transmite como si no existieran fuentes adicionales, por lo que el retardo se mantendrá siempre igual (Fig. 23).

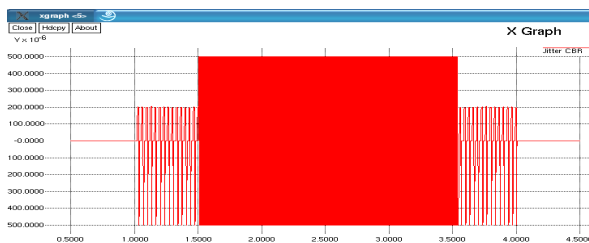


Fig. 23. Jitter con cola de gestión tipo CBR

El jitter para los dos casos restantes será muy similar ya que estos transmiten a la misma velocidad, causando que la capacidad sobrante del canal se divida en partes iguales.

Por esto, tanto el retardo como el jitter se diferencian únicamente por el tiempo muerto de cada uno y por el tiempo de inicio de transmisión.

El valor máximo para los dos tráficos es de 1 ms con un promedio cercano a 0 ms, manteniéndose así constante el retardo (Fig. 24 y 25).

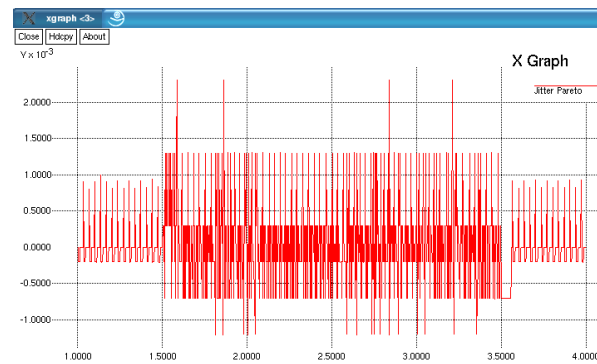


Fig. 24. Jitter con cola de gestión tipo Pareto

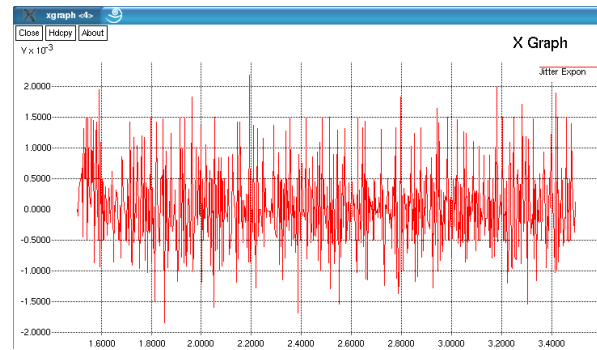


Fig. 25. Jitter con cola de gestión tipo exponencial

A continuación se presentan los resultados obtenidos a nivel de pérdidas para cada caso estudiado a nivel IP (tabla 1). Aquí se encuentra que para el caso de una red sin calidad de servicio, el tráfico que envió más datos es quien terminó perdiendo la mayor cantidad de paquetes, sin embargo al garantizar un ancho de banda donde puedan ser transmitidos todos los tráficos no se dieron pérdidas.

Al implementar calidad de servicio se garantizó

que todos los paquetes de un tráfico prioritario llegaran a su destino cuando el canal fue saturado, haciéndolo conveniente para aquellas aplicaciones en donde es de vital importancia que los datos lleguen correctamente en el menor tiempo posible, como lo es el caso de las videoconferencias.

Es importante recalcar que al incluir QoS se debe tener en cuenta si se desea aplicar prioridad a más de un tráfico, ya que si este es el caso, es preferible la implementación de DiffServ, con lo cual se le asigna una prioridad un poco más baja al flujo que menos necesite.

Tabla 1. Comparación de pérdidas obtenidas

Pérdidas de paquetes					
	Ideal	Sin QoS SFQ	Sin QoS Drop Tail	DiffServ	IntServ
CBR	0	126	92	0	0
Pareto	0	67	50	64	118
Exponencial	0	11	37	177	104

En la tabla 2 se puede apreciar que la red sin calidad de servicio presentó los mayores retardos de transmisión tanto máximos como en su valor promedio, mientras que al implementar QoS este disminuyó.

Para el flujo prioritario el modelo IntServ tuvo el mejor tiempo promedio de transmisión aunque no muy distante del modelo DiffServ, sin embargo, en los demás tráficos el modelo DiffServ obtuvo los mejores valores promedio de retardo.

Tabla 2. Comparación de retardos

End to end Delay (ms)						
	Sin QoS		DiffServ		IntServ	
	Max	Prom	Max	Prom	Max	Prom
CBR	75	70	65	50	48	48
Pareto	72	68	66	51	63	62
Exponencial	70	69	66	50	63	61.8

En cuanto al jitter, la tabla 3 permite ver que tanto la red sin calidad servicio como la misma con la implementación del modelo Diffserv tu-

vieron promedio similares cercanos a 0.1 ms. Este valor aunque es despreciable frente al valor de retardo obtenido, es mayor al encontrado con la implementación de IntServ, donde el valor para el tráfico prioritario fue muy cercano a cero.

Tabla 3. Comparación de valores de Jitter

	Jitter (ms)					
	Sin QoS		DiffServ		IntServ	
	Max	Prom	Max	Prom	Max	Prom
CBR	1	0.1	1	0	1.5	0.001
Pareto	2.3	0.13	2	0.1	2.3	0.02
Exponencial	2.3	0.14	1.98	0.11	2.2	0.02

7. CONCLUSIONES

Al momento de implementar una topología deben tenerse en cuenta diversos aspectos, muchos de los cuales fueron mencionados durante la evaluación de cada modelo. Aspectos como el retardo, la latencia o jitter, las pérdidas, entre otros.

Sin embargo existe un aspecto no mencionado pero fundamental a tener en cuenta y es la aplicación que esta red tendrá. Deben conocerse muy bien los tráficos que se trabajarán con el fin de establecer si se requiere una red con pérdidas equitativas para cada fuente de transmisión, o si solo algunas de las fuentes son importantes.

Surge entonces la necesidad de seleccionar alguno de los dos modelos. El modelo IntServ aunque garantiza el ancho de banda para el tipo de tráfico seleccionado, deja los tráficos restantes sin QoS, causando que luchen entre ellos para poder enviar sus paquetes, a diferencia del modelo DiffServ, el cual asigna un tipo de prioridad a cada tipo de tráfico otorgando así una calidad de servicio diferente a cada uno, por ello, con DiffServ se pierde mucha información de un tipo de datos si estos tienen asignado el valor más bajo de prioridad y si el canal se encuentra saturado por mucho tiempo.

Finalmente, la evaluación realizada permite determinar que los modelos de calidad de servi-

cio representaron una mejora frente a una red sin uso de estos, ya que se evitó en ambos casos la ráfaga retardada causada por el manejo de

colas establecido, evitando así esta retransmisión de datos.

Referencias Bibliográficas

- [1] Wikipedia, [En línea], consultado en Agosto 13 de 2011, disponible en: http://es.wikipedia.org/wiki/Internet_Protocol.
- [2] Cisco, [En línea], consultado en Junio 1 de 2011, disponible en: <http://search.unifiedcommunications.techtarget.com/tip/Ciscos-WRED>.
- [3] Cisco, [En línea], consultado en Julio 10 de 2011, disponible en: http://en.wikipedia.org/wiki/Tail_drop.
- [4] J. Escribano, C. García, C. Seldas, J. Moreno, J. Ignacio; Diffserv como solución a la provisión de QoS en Internet. Universidad Carlos III, Madrid, 2010.
- [5] R. Romeral, S. Miraut; Soporte a la calidad de servicio (QoS) utilizando Internet de servicios diferenciados (DiffServ). Madrid, 2010.
- [6] S. Floyd, V. Jacobson; Random early detection (RED) gateways for congestion avoidance.
- [7] B. Braden, D. Clark., J. Crowcroft., B. Davie, S. Deering, D. Estrin; Recommendations on queue management and congestion avoidance in the Internet, 1998.
- [8] R. Salazar, J. Carlos, A. Urrea; Sistema operativo linux y control de tráfico en redes de computadores. Universidad de Antioquia, Departamento de electrónica. Medellin, 2004.
- [9] Wikipedia, [En línea], consultado en Septiembre 9 de 2011, disponible en: http://es.wikipedia.org/wiki/Servicios_integrados.
- [10] Cisco, [En línea], consultado en Julio 7 de 2011, disponible en: http://www.cisco.com/en/US/products/ps6611/products_ios_protocol_group_home.html.
- [11] A. Barenco, C. Jacy; Modelo IntServ/ protocolo RSVP, Barcelona, 2000.
- [12] Cisco, [En línea], consultado en Marzo 15 de 2011, disponible en: http://www.cisco.com/en/US/products/ps6611/products_ios_protocol_group_home.html.
- [13] M. Saldaña, M. Aguilar; Calidad de servicios en redes IP, Mexico, 2009.
- [14] Tráfico constante de bits, [En línea], consultado en Junio 20 de 2011, disponible en: http://en.wikipedia.org/wiki/Traffic_contract#Constant_Bit_Rate_.28CBR.29.
- [15] Wikipedia, [En línea], consultado en Julio 17 de 2011, disponible en: http://es.wikipedia.org/wiki/Principio_de_Pareto.
- [16] D. Rincón; Introducción a los modelos de tráfico para redes de banda ancha.. Departamento de matemática aplicada y telemática (UPC), Barcelona, 2008.
- [17] E. Altman, T. Jiménez; NS Simulator for begginers. Universidad de Los Andes, Venezuela and ESSI, Diciembre 4, 2003.
- [18] A. Padilla, J. Becerra, L. Yasmin; Manual de prácticas con NS2. Grupo de investigación en Telecomunicaciones, Facultad de Ingeniería Electrónica, Universidad Pontificia Bolivariana. Bucaramanga, 2008.