

ANÁLISIS DE TIPOS DE GESTORES DE COLAS EN NS-2

ANALYSIS OF TYPES OF QUEUE MANAGERS IN NS-2

ABSTRACT

Network packet switch must have an order in the processing of requests arriving at a node so that the network is effective and efficient, that is, that packet loss due to changes in bandwidth and buffer capacities of the routers in charge of communication, is the smallest. For this there are different types of queue which define the characteristics of incoming and outgoing packets on a particular node. In this paper the differences between these types of managers by implementing a network topology data in NS2 software will be studied in order to determine what the proper protocol to avoid massive loss of messages in that configuration.

Keywords:

Buffer, switching, DropTail cola, cola CBQ, cola RED, cola SFQ

Rosa María Mejía Montañez
jiam@correo.udistrital.edu.co

Juan Camilo Bernal Soto
jucbernals@correo.udistrital.edu.co

Danilo Alfonso López
sarmiento
Universidad Distrital Francisco
José Caldas
dalopez@udistrital.edu.co

Tipo de Artículo: Investigación

Fecha de recepción
Agosto 22 2013
Fecha de Aceptación
Octubre 15 de 2013

RESUMEN

En redes de conmutación de paquetes, es necesario tener un orden en el procesamiento de solicitudes, que lleguen a un nodo con el fin de que la red sea eficaz y eficiente, es decir, que la pérdida de paquetes debida a cambios de ancho de banda y a las capacidades de buffer de los enrutadores encargados de la comunicación, sea la menor posible. Para esto existen diferentes tipos de gestores de colas los cuales definen las características de entrada y salida de paquetes en un determinado nodo. En este documento se estudiarán las diferencias entre estos tipos de gestores mediante la implementación de una topología de red de datos en el software NS2, con el fin de determinar cuál sería el protocolo adecuado para evitar la pérdida masiva de mensajes en dicha configuración.

Palabras clave

Android, IOS, seguridad, dispositivos móviles.

1. INTRODUCCIÓN

El control de la congestión en redes IP ha sido un problema recurrente en los últimos años debido al crecimiento en la transmisión de datos con nuevas aplicaciones como video conferencias y telefonía IP. Es por esto que se deben desarrollar nuevos protocolos para manejar eficientemente la gestión de los mensajes en las colas, lo cual hace referencia al proceso de atención y servicio del mensaje o paquetes de información que se emplean en el nodo que determina su almacenamiento en una memoria durante su espera de atención [1].

Algunos de los algoritmos encargados del manejo de la congestión y las colas presentes en un enrutador, pueden ser muy efectivos y a la vez sencillos, como por ejemplo Droptail; el cual tiene un bajo nivel de procesamiento y otros más complejos como CBQ, SFQ o RED satisfacen la necesidad de calidad de servicio en la red de acuerdo a la demanda en las comunicaciones.

A continuación se dará una breve definición de algunos tipos de gestores de colas y sus principales características, seguido se propondrá una topología de una red de datos con un único enlace sobre el cual debe circular el tráfico de diferentes nodos y será simulada en el software NS2 para diferentes parámetros

de entrada (ancho de banda, tamaño de paquete, velocidad de envío) para luego analizar las diferencias entre cada gestor de colas definido previamente.

2. MARCO TEÓRICO

La formación de colas es un concepto que surge de manera muy natural al estudiar redes de conmutación de paquetes; los paquetes que llegan por un punto de entrada o por un nodo intermedio en la trayectoria al destino, se almacenan temporalmente, se procesan para determinar la conexión apropiada de enlace de transmisión de salida al siguiente nodo en la trayectoria [2].

Dado que puede existir diferencia entre la tasa a la que llegan paquetes a un nodo, y el régimen con el que éste es capaz de atenderlos, se crearon algoritmos encargados de ordenar los paquetes en cola de acuerdo a sus características de modo que el envío satisficiera las necesidades del usuario ya sea ancho de banda, tiempo de respuesta, variación del tiempo de respuesta, probabilidad de pérdida de paquetes, etc. Estos algoritmos pueden ser clasificados teniendo en cuenta sus requerimientos de almacenamiento de información para llevar a cabo su

trabajo [3] como se muestra en la figura 1.

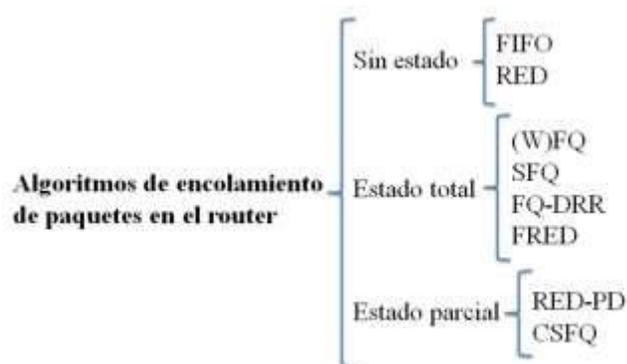


Fig. 1. Clasificación de los algoritmos de encolamiento[3].

Droptail:

Este algoritmo de administración de colas usado por los enrutadores de internet para decidir cuándo desechar paquetes, el tráfico en este algoritmo no es diferenciado, es decir funciona como una memoria tipo FIFO (First Input First Output), luego cada paquete es tratado exactamente igual.

El modo de operación está en revisar el nivel del buffer. Una vez este buffer está lleno se rechazan todos los paquetes entrantes sin discriminar su tipo.

Esta pérdida de datagramas hace que el transmisor entre en un estado de "inicio lento", el cual reduce la cantidad de tráfico que introduce el transmisor hasta que este comience a recibir *acknowledgements* (ACK) [4] incrementando su ventana de congestión.

A medida que la cola se llena desde el punto de inicio hasta el punto final de encolamiento, se da una pérdida de paquetes en un patrón lineal suave, llamado una interpolación gráfica, como se muestra en la Figura 2. No hay pérdida de paquetes hasta que el nivel de llenado de colas alcanza el punto de inicio de 30 por ciento, luego de esto los paquetes comienzan a caer. A medida que la cola se llena, el porcentaje de paquetes rechazados aumenta en forma lineal. Cuando la cola se llena hasta el punto final de 50 por ciento, la tasa de pérdida de paquetes se incrementa hasta el 80 por ciento. Cuando el nivel de encolamiento supera el punto final de 50 por ciento, se pierden todos los paquetes hasta este caiga por debajo de 50 por ciento.

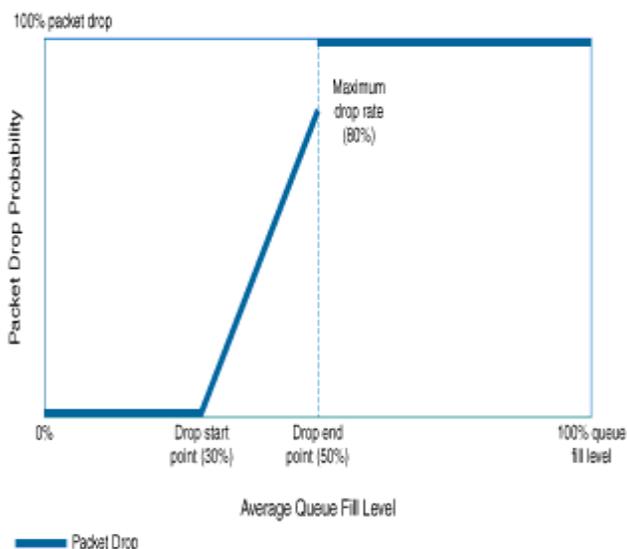


Fig. 2. Perfil de pérdida de paquetes para el encolamiento tipo Droptail.[6]

El problema más severo es la “sincronización global” el cual ocurre cuando datagramas de múltiples conexiones TCP son desechados.

Esto hace que múltiples conexiones generen retardos y reenvíen sus datos al mismo tiempo. Esto pasa porque en lugar de descartar muchos paquetes de una conexión se descarta un paquete de cada conexión.

CBQ:

CBQ (Class Based Queuing), es un algoritmo basado en el encolamiento justo, en este protocolo los paquetes son asignados a una clase (tiempo real, transferencia de archivos, etc.) de esta forma los paquetes son colocados en la cola de la clase correspondiente como se muestra en la figura 3.

La forma en la que se accede a los paquetes a enviar es por medio de round-robin (acceso por tiempo) así que se dará mayor tiempo para enviar un número mayor de paquetes de una cola con un índice de prioridad mayor.

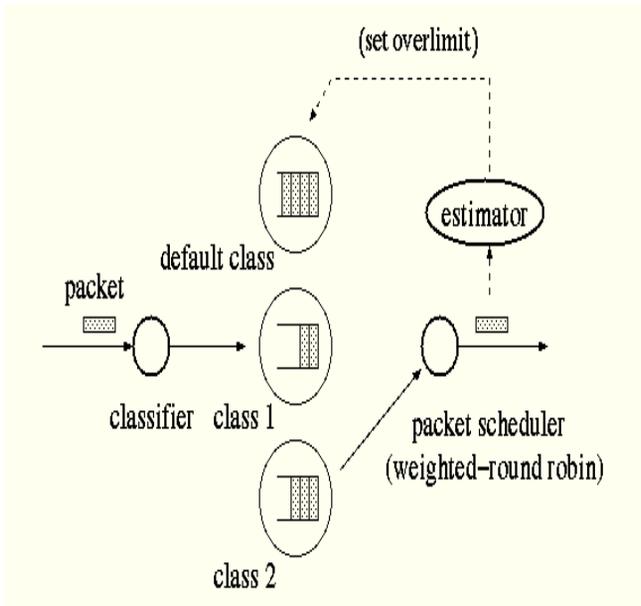


Fig. 3. Componentes básicos del gestor de cola tipo CBQ.

CBQ divide el ancho de banda de la conexión de red entre múltiples clases o colas. Una cola puede ser configurada para que tome prestado ancho de banda de otra cola si esta está siendo subutilizada.

Las colas CBQ son arregladas de una manera jerárquica. En la cima está la cola raíz la cual define el total del ancho de banda disponible. Las colas hijas son creadas debajo de la cola raíz. A cada una de ellas se le puede asignar una porción de ancho de banda de la cola raíz. Por otro lado CBQ otorga a cada cola un nivel de prioridad. A medida que este nivel es más alto, esa cola será preferida en el momento en el que haya congestión [5].

RED:

RED (random Early Detection) es un algoritmo para evitar la congestión en una red. Su trabajo se realiza por medio del continuo cálculo la de longitud promedio de la cola y comparándola con dos umbrales uno mínimo y uno máximo.

Si el promedio de la cola está por debajo del umbral mínimo entonces ningún paquete deberá ser arrojado pero si el promedio está sobre el umbral máximo entonces todos los nuevos paquetes que lleguen serán desechados. Si el promedio se encuentra entre los umbrales entonces los paquetes son arrojados basándose en un promedio calculado sobre el tamaño de la cola.

En otras palabras cuando el promedio de la longitud de la cola se aproxima al umbral máximo más y más paquetes son desechados. A la hora de desechar paquetes RED escoge aleatoriamente cuales conexiones desechar. Conexiones que utilicen una gran cantidad de ancho de banda tienen una mayor probabilidad de que sus paquetes sean arrojados. El tamaño medio de la cola a la llegada de paquetes k se calcula mediante un promedio móvil ponderado exponencial (EWMA) de las longitudes de cola actual y anteriores $Q_k, Q_{k-1}, Q_{k-2}, \dots, Q_{k-n}$ de acuerdo al ancho de banda w , como se observa en la ecuación 1[7]

$$AVG_k = (1 - w)AVG_{k-1} + w Q_k \quad (1)$$

RED es uno de los protocolos más utilizados debido a que evita la sincronización global. Otra gran ventaja de RED es que evita los derramamientos de tráfico ya que empieza a descartar paquetes aún antes de tener el buffer lleno. Sin embargo RED debería ser usado únicamente cuando el protocolo de transporte es capaz de responder a los indicadores de congestión de la red. En la mayoría de casos debería usarse en colas con tráfico TCP y no UDP. Adicionalmente, RED utiliza ECN (Explicit Congestion Notification) para notificar a dos Hosts que se están comunicando en la red de cualquier congestión sobre el camino de comunicaciones.

FQ:

Fair queuing es un algoritmo de planificación que permite que múltiples flujos de paquetes compartan las capacidades del canal. El espacio del buffer es dividido en muchas filas, cada una de las cuales es usada para mantener los paquetes de flujo. Imaginando un link con tasa de transmisión R a un momento N con flujo de datos activo, cada flujo es servido con una tasa promedio de R/N . en un corto periodo la tasa podría fluctuar alrededor de este valor.

Fair queuing puede implementar “max-min fairness” (maxima minima

justicia). Ya que su mayor prioridad es maximizar la mínima tasa de transmisión por cada flujo activo. La segunda prioridad es maximizar el segundo mínimo de tasa de transmisión y así sucesivamente, lo que se traduce en un protocolo de bajo rendimiento en redes inalámbricas. A diferencia de “round-robin scheduling” cuando se implementa “max-min fairness” el tamaño del paquete a ser transmitido influye en la decisión de cuanto ancho de banda otorgar a esa conexión y así asegurar que cada flujo de datos tiene igual oportunidad de transmitir igual cantidad de datos.

SFQ:

Stochastic Fairness Queueing (SFQ) es una implementación sencilla que es menos precisa que otras de su tipo pero que requiere muchos menos cálculos mientras es casi perfectamente justa.

Sin duda alguna a la hora de hablar de SFQ la palabra clave es conversación, la cual corresponde a una sesión TCP o a un “streaming” UDP. El tráfico es dividido en muchas filas tipo FIFO (first in First out), una para cada conversación para ser enviadas de una forma “round robin” (acceso al canal por medio de turnos) dando así a cada sesión la oportunidad de un turno para enviar datos (ver figura 4).

Esto hace que la distribución de capacidad del canal sea justa y no permite que ninguna conversación se pierda.

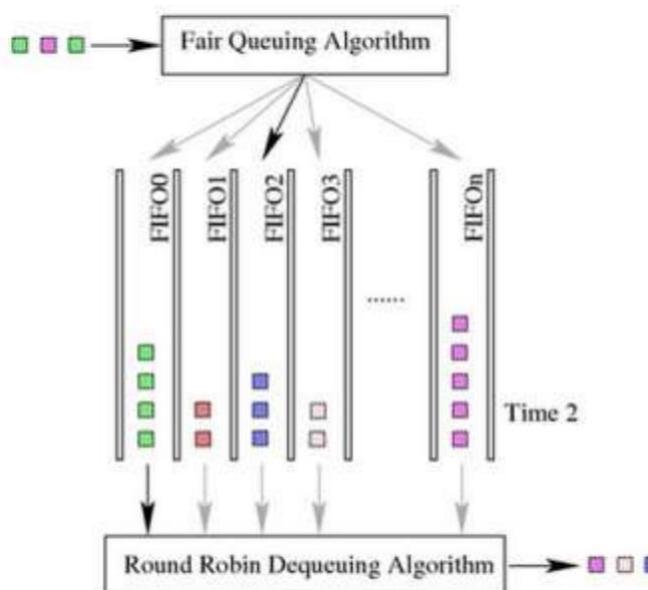


Fig. 4. Comportamiento de una red de datos con encolamiento SFQ

Debido a la naturaleza de su algoritmo, múltiples sesiones podrían acabar en el mismo "cubo", lo cual reduce la oportunidad de enviar un paquete, por lo tanto se reduce a la mitad la velocidad disponible. Para prevenir que sea notoria la fila donde se pondrá el paquete. SFQ cambia su algoritmo de "hashing" muy frecuentemente de acuerdo a como lo determine el administrador o como esté configurado por defecto. SFQ cumple su mayor efectividad cuando la interfaz de salida está muy llena. Y cumple su mayor efectividad al combinarse con otros protocolos.

DRR:

Déficit round robin (DRR) es una forma de agendar el canal de transmisión para un paquete basado en su longitud sin conocerla exactamente. La mayor ventaja del DRR es que puede manipular paquetes de diferentes tamaños sin conocer su tamaño exacto. DRR selecciona una longitud de paquetes que se puede enviar por servicio. Si el primer paquete de la fila a examinarse excede esa longitud se deja atrás hasta la próxima visita del generador de la agenda de transmisión.

DRR funciona dando líneas de "crédito" a cada fila. En una ronda de visitas DRR escoge un número llamado quantum, el cual es diferente para cada fila. Simultáneamente a quantum, DRR lleva un contador de déficit.

Al seleccionar una cola, DRR compara la longitud del primer paquete existente en ella con quantum, si este es menor se le da servicio a la cola por tantos paquetes como el quantum lo permita, una vez cese esta transmisión el valor de quantum y el contador de déficit son sumados y ese resultado se utilizará en la próxima visita que DRR haga a esa fila. De lo contrario si el primer paquete excede al quantum correspondiente para esa fila. La fila no es servida pero se realiza la misma operación de adición entre quantum y el contador de déficit para la próxima visita [8].

DRR es muy efectivo en redes cuyas aplicaciones arrojen paquetes de diferentes tamaños. Adicionalmente presenta un ahorro de cálculo al no tener en cuenta en su algoritmo filas vacías las cuales puede identificar mediante un auxiliar.

3. DESARROLLO DE LA TOPOLOGÍA

Mediante el software para análisis de redes telemáticas NS2 (Network Simulator), simularemos la estructura de red presentada en la figura 5 la cual consta de 8 nodos dónde los número 0, 1 y 2 son transmisores de tráfico constante CBR a distintas tasas y los nodos 5, 6 y 7 son los respectivos receptores.

Los distintos gestores de colas serán aplicados y capturados entre los enrutadores 3 y 4 cuyo enlace será de menor capacidad que los demás con el fin de generar una cola sobre el primer enrutador. Los retardos en general para esta topología serán de 10 milisegundos, sin embargo, para objeto del estudio, los retardos se variarán en el enlace intermedio de acuerdo al gestor de cola que se esté tratando para así tener diferentes respuestas.

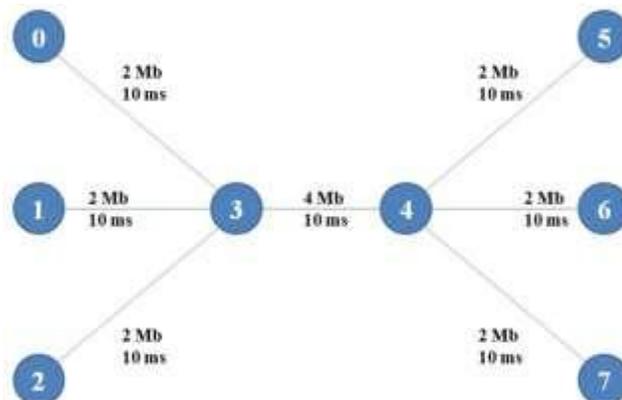


Fig. 5. Estructura de red para análisis de gestores de colas

Para cada gestor de colas implementado en la simulación se tienen algunos parámetros básicos; para DropTail, FQ, SFQ y DRR se tiene la siguiente configuración:

```

$simulacion duplex-link $nodo0 $nodo2
1Mb 10ms DropTail, donde, 1 Mb y 10
ms son el ancho de banda del enlace y
el tiempo de respuesta respectivamente.
  
```

Para RED se necesitan unos parámetros de entrada más específicos con el fin de tener una óptima utilización de ancho de banda, como sigue:

```

#variable asociada la cola para
modificar sus parámetros (RED)
set Cola [[ $simulacion link $nodo3
$nodo4] queue]
# Numero máximo de paquetes en la
cola en total
$Cola set limit_ 50
# Numero medio máximo de paquetes
encolados
$Cola set thresh_ 15
  
```

```
# Numero medio máximo de
paquetes nivel de umbral.
$Cola set maxthresh_ 40
# Establece la probabilidad de
desechar un paquete a 1/30
$Cola set linterm_ 30
# Los paquetes se marcaran para
indicar congestión.
$Cola set setbit_ true
```

Adicionalmente se hace uso de la herramienta XGRAPH de NS2 para generar junto con la simulación una gráfica que mostrará el envío satisfactorio de paquetes para los tres enlaces propuestos, a través del tiempo.

El lenguaje de programación usado en NS-2 (extensión .tcl) para la estructura descrita previamente fue tomado de [1].

4. SIMULACIÓN

La simulación se hace para un tiempo de 5 segundos, con envío constante de paquetes (CBR) desde los 0,1 hasta los 4 segundos.

Cada mensaje enviado varía de color de acuerdo al nodo generador de modo que se pueda diferenciar de que nodo hacen parte los paquetes en cola o descartados.

Para el tipo de gestión Droptail con 4 Mb de ancho de banda y 5ms de tiempo de respuesta en el enlace 3 – 4, se obtuvo la gráfica presentada en la figura 6.

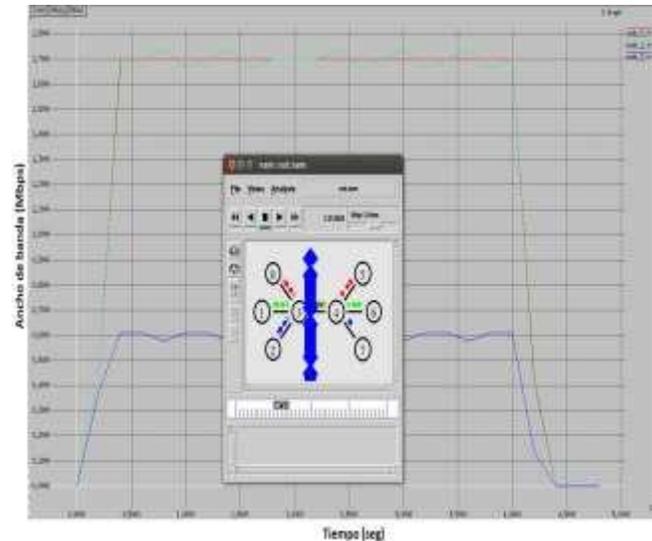


Fig. 6. Gráfica de análisis de envío de paquetes para gestión de colas tipo Droptail en la estructura de red mostrada.

Como se observa en la figura 6, se tiene un intervalo en los primeros segundos en el cual el envío de paquetes es satisfactorio para todos los nodos, este tiempo depende del límite de paquetes en cola establecidos en la simulación la cual en este caso es de 10 paquetes, luego se forma un encolamiento y dado que el tipo de gestor elegido es FIFO los paquetes con prioridad son los enviados por el nodo 0 (rojo) y los que más se descartan son los enviados por el nodo 2 (azul). Por lo tanto se tiene que el ancho de banda entre los nodos 3 y 4 se distribuye principalmente entre el envío de paquetes del nodo 0 y 1. Claramente carece de mecanismos para controlar la

congestión de paquetes en el nodo 3 y distribución equitativa de capacidad de su respectivo enlace, en consecuencia se tiene que notificar a los nodos emisores del estado de la red para que el encolamiento y a su vez descarte de paquetes disminuya, sin embargo esto puede conducir a una subutilización de la capacidad del enlace.

En la simulación de la figura 7. Se presenta el tipo de gestión de cola Droptail con una variación en el tamaño de los paquetes enviados por cada nodo (nodo0, nodo1, nodo2), los cuales corresponden a 650bytes, 850bytes y 1000bytes respectivamente, cada uno con tiempos de retardos en el envío de paquetes de 4ms.

Con relación a la simulación 1 se puede observar que la variación del tamaño de los paquetes no ocasiona cambios en el principio de funcionamiento del tipo de gestión de cola Droptail, el cual da prioridad al flujo 0 y 1, los cuales hacen uso de 3MB de ancho de banda y dejan un 1MB para el flujo 3 el cual presenta una tasa de pérdidas del 50% aproximadamente, en general una mayor utilización del ancho de banda del canal, genera mayor pérdida de datos, es decir que la cantidad de datos de la fuente o nodo que llegan a su destino se reducen drásticamente.

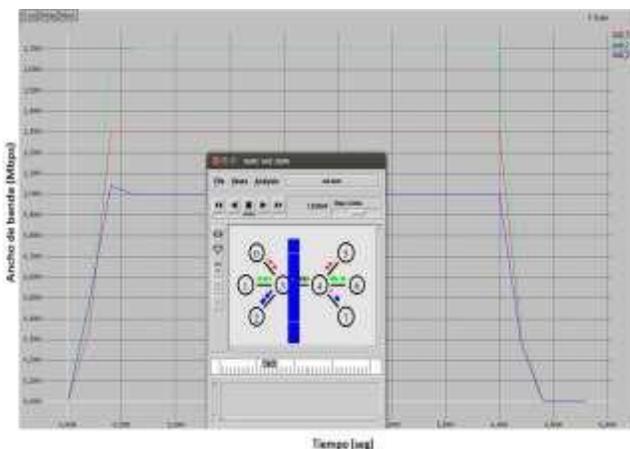


Fig. 7. Gráfica de análisis de envío de paquetes para gestión de colas tipo Droptail en la estructura de red mostrada con diferentes parámetros para cada nodo emisor.

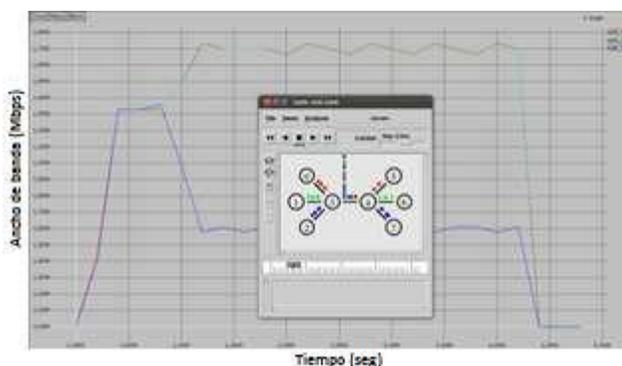


Fig. 8. Droptail para diferente límite de paquetes en cola.

En el tipo de gestión de cola Droptail no se descartan paquetes hasta que la cola este completamente llena, sin embargo la cantidad de paquetes en cola es una característica que puede ser modificada, en el caso de la figura 8 el límite se estableció en 100 paquetes, a diferencia de la simulación 1 en la cual el límite eran 10 paquetes, de allí que la pérdida de datos sea mucho más lenta ya que la cola no se llena tan rápido cuando el

límite establecido son 100 paquetes y no se descartaran estos hasta llenar el buffer.

Algoritmos como FQ intentan dar garantías en cuanto a distribución adecuada del enlace sin embargo presentan algunos problemas de acuerdo a la velocidad de procesamiento del router o memoria disponible en el mismo.

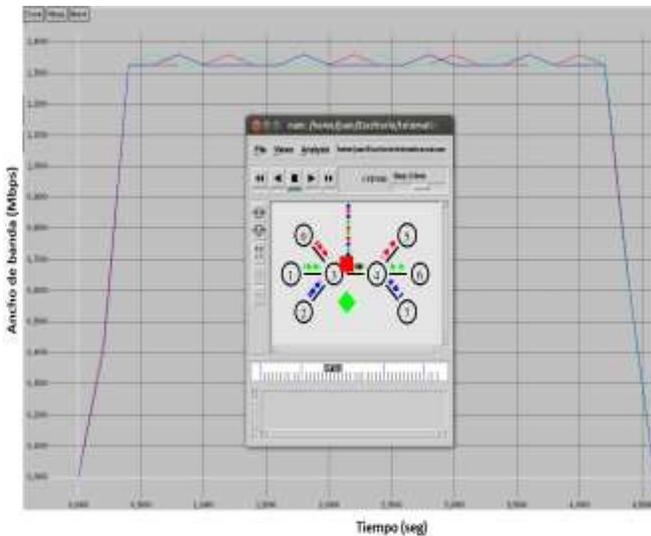


Fig. 9 Gráfica de análisis de envío de paquetes para gestión de colas tipo FQ en la estructura de red mostrada.

En la figura 9 se observa el alto nivel de justicia que presenta el algoritmo FQ dado que reserva la misma cantidad de ancho de banda a cada flujo, luego los paquetes descartados por el nodo 3 son equitativos para cada nodo emisor y su tasa de pérdida no es muy alta.

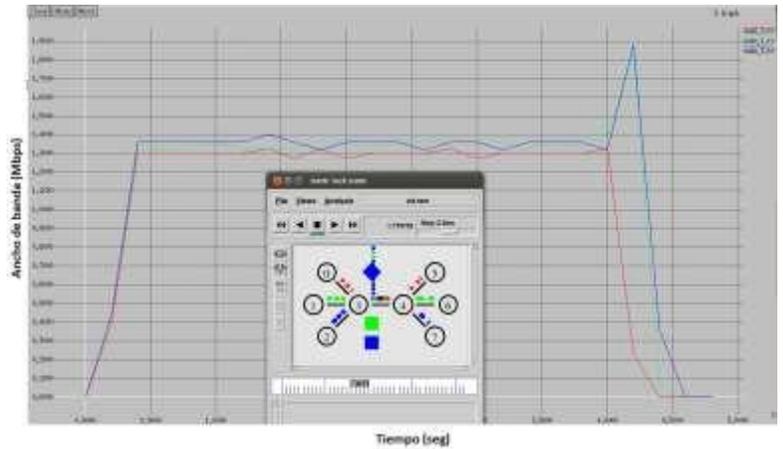


Fig. 10. FQ para diferentes tamaños de paquetes

Un caso particular en FQ se presenta cuando los tamaños de los paquetes son diferentes en cada flujo, la distribución del ancho de banda de salida es el mismo para los 3 flujos independientemente si un nodo envía paquetes de mayor o menor tamaño con respecto a los demás, este fenómeno se puede observar en la anterior simulación en la cual cada flujo maneja 3 tamaños de paquetes diferentes (650bytes, 850bytes, 1000bytes), en este caso la tasa de pérdidas de cada flujo varía con respecto al tamaño de los paquetes (ver figura 10).

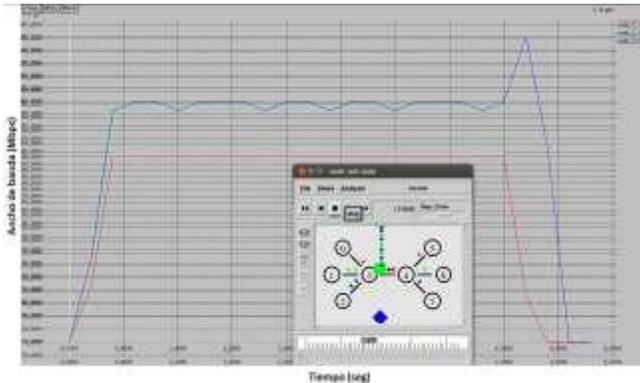


Fig. 11. Gestor de cola FQ para diferentes tiempos de retardo.

Otro parámetro que es posible variar es el tiempo de envío de paquetes, en este caso se varió el tiempo del flujo 0 (rojo) de 4ms a 10ms, esto ocasiona que la cantidad de paquetes que llegan al destino con respecto a los demás flujos disminuya, como se observa en la figura 11.

Una mejora para el algoritmo FQ es el SFQ el cual está más enfocado a la implementación de la estructura de red. En la figura 12 se observa que el comportamiento de la red es similar al del representado con FQ sin embargo la pérdida de paquetes es un poco más alta dado que el número de colas es finito.

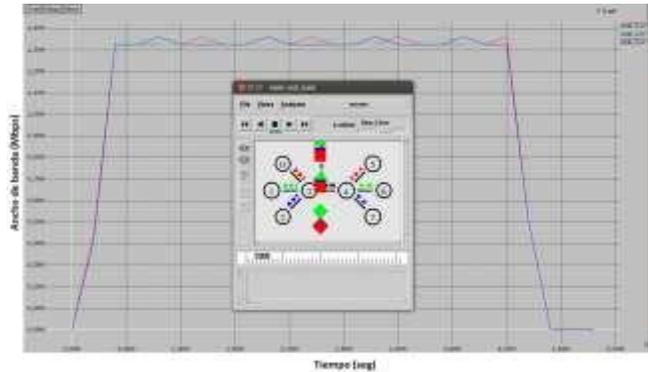


Fig. 12. Gráfica de análisis de envío de paquetes para gestión de colas tipo SFQ en la estructura de red mostrada.

El tipo de gestión de cola SFQ realiza una distribución equitativa entre todas las conexiones abiertas es decir implementa un proceso de atención igualitaria por flujo dependiendo del tamaño de los paquetes, así se representa en la figura 13 en la cual cada flujo presenta diferentes tamaños de sus paquetes 650bytes, 850bytes y 1000bytes (rojo, verde, azul) respectivamente, la atención prestada a cada flujo es igual, es decir que la tasa de pérdidas en cada flujo es la misma, gracias a la asignación de ancho de banda en el nodo 3 dependiendo del tamaño del paquete.

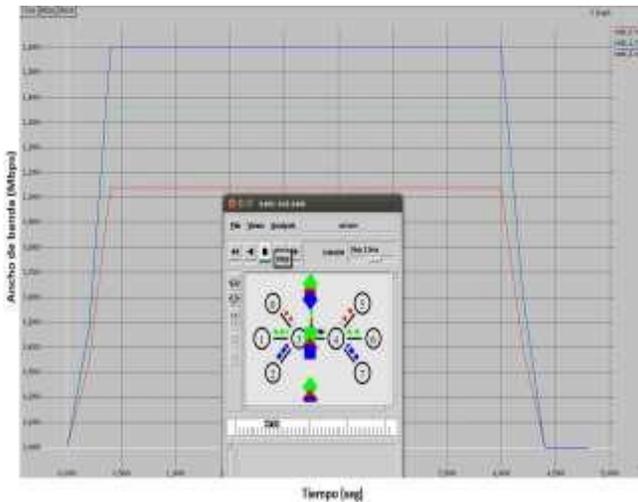


Fig. 13. Gestor de cola SFQ para diferentes tamaños de paquetes.

Otro algoritmo de gestión de colas es el DRR el cual distribuye equitativamente el ancho de banda del enlace en cuestión diferenciando el tamaño del paquete enviado.

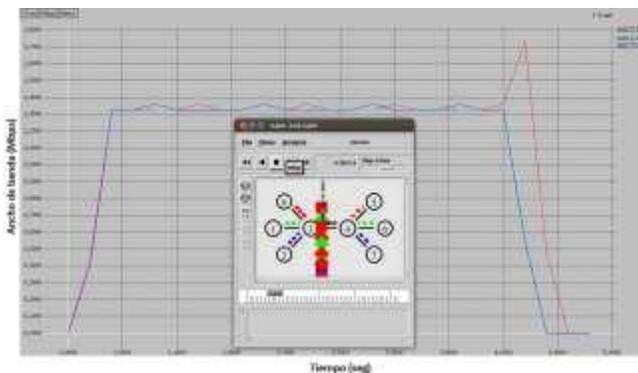


Fig. 14. Gráfica de análisis de envío de paquetes para gestión de colas tipo DRR en la estructura de red mostrada.

La gráfica 14 muestra el comportamiento de la red mostrada en la misma, donde el envío de paquetes de los tres diferentes nodos es equitativo al igual que su pérdida. En contraste se tiene la figura 15

donde el tamaño de paquetes varió (650, 850, 1000 para los nodos 0, 1 y 2 respectivamente) se puede observar que la pérdida de paquetes es más alta para el nodo que envía paquetes de mayor tamaño ya que son los que consumen mayor ancho de banda.

Por último, el algoritmo que necesita mayor cantidad de parámetros de entrada para garantizar su óptimo funcionamiento es el mostrado en la figura 16, RED.

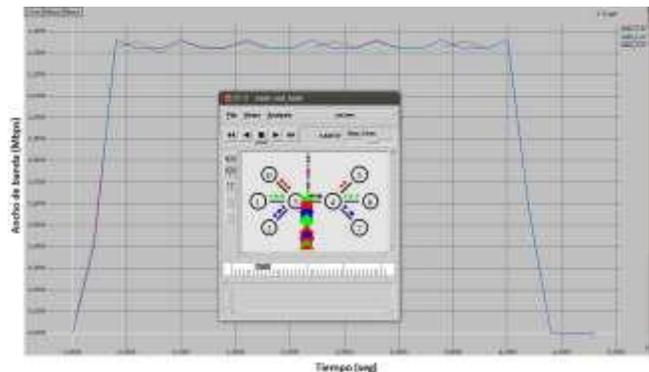


Fig. 15. Gestor de colas DRR para diferentes tamaños de paquetes.

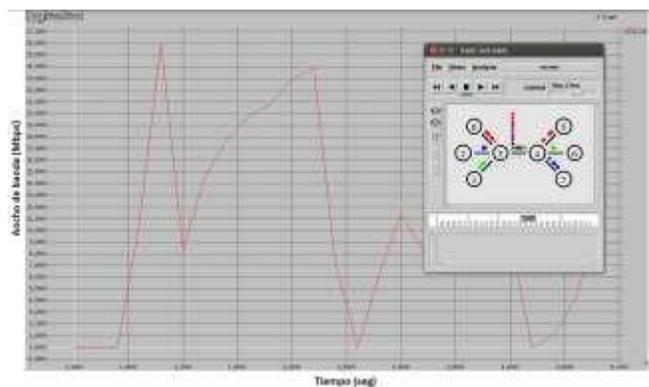


Fig. 16. Gestor de colas RED para la topología expuesta.

En el tipo de gestión de cola RED lo que se busca es evitar que la cola llegue hasta su límite máximo, como se evidencia en la figura 16 donde el número o cantidad de paquetes en cola, es decir que este tipo de gestión no permite que llegue a su máximo valor y empieza a votar paquetes de forma anticipada de acuerdo a un pre monitorización.

5. ANÁLISIS DE RESULTADOS

Tabla 1. Tabla de resultados gestores de cola con paquetes de igual tamaño

		ancho de banda emitido	ancho de banda recibido	tasa de perdida
DROPTAIL	flujo1	1.7MB	1.7MB	0%
	flujo2	1.7MB	1.7MB	0%
	flujo3	1.7MB	600KB	64.7%
FQ	flujo1	1.7MB	1.33MB	21.56%
	flujo2	1.7MB	1.33MB	21.56%
	flujo3	1.7MB	1.33MB	21.56%
SFQ	flujo1	1.7MB	1.33MB	21.56%
	flujo2	1.7MB	1.33MB	21.56%
	flujo3	1.7MB	1.33MB	21.56%
DRR	flujo1	1.7MB	1.33MB	21.56%
	flujo2	1.7MB	1.33MB	21.56%
	flujo3	1.7MB	1.33MB	21.56%

En la tabla se presentan algunos resultados relevantes de las simulaciones realizadas por medio del software ns2, entre estos se encuentra el ancho de banda

requerido por cada nodo emisor, el ancho de banda asignado a cada flujo en el nodo 3 y por último la tasa de perdida presentada en cada flujo, la cual nos permite hacer un análisis detallado del funcionamiento y eficiencia de cada uno de los gestores de colas.

- En el tipo de gestor Droptail se dio prioridad a los dos primeros flujos asignándoles el ancho de banda requerido por cada uno, mientras que el flujo 3 presento una pérdida considerable del 64.7%.
- FQ presento una atención igualitaria a cada flujo con un ancho de banda de 1.33MB lo cual genero una tasa de perdidas iguales de 21.56%.

Tabla 2. Tabla de resultados gestores de cola con paquetes de tamaño variables

		ancho de banda emitido	ancho de banda recibido	tasa de perdida
DROPTAIL	flujo1	1.3MB	1.3MB	0%
	flujo2	1.7MB	1.7MB	0%
	flujo3	2MB	1MB	50%
FQ	flujo1	1.3MB	1.3MB	0%
	flujo2	1.7MB	1.35MB	20.58%
	flujo3	2MB	1.35MB	32.5%
SFQ	flujo1	1.3MB	1.04MB	20%
	flujo2	1.7MB	1.36MB	20%
	flujo3	2MB	1.6MB	20%
DRR	flujo1	1.3MB	1.3MB	0%
	flujo2	1.7MB	1.35MB	20.58%
	flujo3	2MB	1.35MB	32.5%

En la tabla 2 se presentan los resultados de las simulaciones de cada uno de los tipos de gestores de cola con paquetes de diferentes tamaños en cada nodo (nodo0, nodo1, nodo2) de 650bytes, 850bytes y 1000bytes respectivamente.

- Droptail presenta el mismo funcionamiento independiente del tamaño de los paquetes, la administración del buffer de la cola funciona como una memoria tipo FIFO dando prioridad a los flujos 1 y 2.
- FQ asigna el mismo ancho de banda a cada flujo por lo que la tasa de pérdidas dependerá del tamaño de los paquetes enviados, entre más grande sean los paquetes mayor será el ancho de banda requerido y por ende se tendrá mayor pérdida de paquetes como es el caso del flujo 3.
- SFQ presenta un tasa de perdidas igual de 20% para los 3 flujos, para ello la distribución del ancho de banda lo asigna dependiendo del tamaño de los paquetes enviados por cada nodo emisor.
- DRR asigna el mismo ancho de banda a cada flujo por lo que la tasa de pérdidas dependerá del tamaño de los paquetes enviados, entre más grande sean los paquetes mayor será el

ancho de banda requerido y por ende se tendrá mayor pérdida de paquetes como es el caso del flujo 3 que presenta una tasa de pérdidas del 32.5%.

6. CONCLUSIONES

- Se logró implementar una topología en NS2 y analizar la función de diversos gestores de cola en situaciones de cuellos de botella donde los anchos de banda de los nodos son diferentes.
- El tipo de encolamiento FQ tiene la gran ventaja (para esa razón fue diseñada) de prestar un servicio igualitario para cada usuario. Dividiendo la cola en diferentes flujos y recibiendo los equitativamente.
- Al momento de implementar una determinada topología de red es de gran importancia tener en cuenta los parámetros de anchos de banda, tasas de transmisión y tamaños de los paquetes enviados, para determinar qué tipo de gestor de cola podría ser el más conveniente en la red en cuestión y así minimizar las perdidas y los retardos en la red.

7. REFERENCIAS

- [1] Danilo López, “Tipos de gestores de colas en NS-2”, Anotaciones entregadas, pp. 1-3, 2011
- [2] Luis Iturri, “Calidad de servicio en la red ATM-MPLS de PEMEX”, Instituto Politécnico Nacional, pp 8. 2009
- [3] Eduardo J. Ortega, “Algoritmos de encolamiento en routers para una distribución justa de la capacidad de un enlace compartido en una red TCP/IP”, DIS, Universidad Nacional de Colombia, pp. 5, 2008
- [4] S. M. Mun , J. Y. Son W. R. Jo and W. B. Lee “An Implementation of AIS-Based AD hoc Routing Protocol for Maritime Data Communications Networks”, 2012
- [5] L.Boukhalifa, P. Minet, S. Midonnet, and L. George “Comparative Evaluation o CBQ and PriQueue un a MANET” 2005
- [6] Technical Documentation in Juniper networks, “Understanding CoS Tail-Drop Profiles”, pp. 1, 2012
- [7] Walter Cerroni, “Active Queue Management”, University of Bologna – Italy, pp. 8, 2005
- [8] M. Shreedhar, G. Varghese, “Efficient fair queuing using deficit round robin”, Proc. Of ACM SIGCOMM '95, pp. 231-242, 1995.