

**“ANALISIS DE ALGORITMOS DE ENCOLAMIENTO EMPLEADOS  
EN REDES LAN y WAN”**

**ARQUIMEDES LAFONT CABALLERO  
URIEL PAJARO LOPEZ  
JUAN CARLOS CUELLO ROBLES**

**UNIVERSIDAD TECNOLOGICA DE BOLIVAR  
FACULTAD DE INGENIERIA  
DIRECCION DE PROGRAMA DE INGENIERIA DE SISTEMAS  
CARTAGENA DE INDIAS D.T Y C.  
2009**

Cartagena de Indias D. T. y C., 24 de Abril de 2009

Señores  
Comité Curricular de Evaluación de Proyectos  
Programa de Ingeniería de Sistemas  
La Ciudad

Respetados señores:

Con toda atención nos dirigimos a ustedes, con el fin de presentar a su consideración, estudio y aprobación, el trabajo de grado titulado "Análisis de algoritmos de encolamiento empleados en redes LAN y WAN", como requisito final para obtener el título de ingeniería de sistemas.

Atentamente

---

Juan Cuello  
C.C. 3.839.730 de Corozal

---

Uriel Pájaro L.  
C.C. 73.206.904 de Cartagena

---

Arquímedes Lafont C.  
C.C. 1.047.388.171 de Cartagena

---

Isaac Zúñiga S.  
C.C. 73.116.898 de Cartagena

Cartagena de Indias D. T. y C., 24 de Abril de 2009

Señores  
Comité Curricular de Evaluación de Proyectos  
Programa de Ingeniería de Sistemas  
La Ciudad

Respetados señores:

Tengo el agrado de presentar a su consideración, estudio y aprobación, el trabajo de grado en el cual me desempeño como director titulado "Análisis de algoritmos de encolamiento empleados en redes LAN y WAN", desarrollado por los estudiantes Arquímedes Lafont Caballero, Juan Carlos Cuello Robles y Uriel Pájaro López, como requisito final para obtener el título de ingeniería de sistemas.

Atentamente

---

Juan Cuello  
C.C. 3.839.730 de Corozal

---

Uriel Pájaro L.  
C.C. 73.206.904 de Cartagena

---

Arquímedes Lafont C.  
C.C. 1.047.388.171 de Cartagena

---

Isaac Zúñiga S.  
C.C. 73.116.898 de Cartagena

Cartagena de Indias, D.T.H y C Enero de 2009

NOTA DE ACEPTACION

---

---

---

---

---

PRESIDENTE DEL JURADO

---

JURADO

---

JURADO

Cartagena de Indias D.T y C. Mayo de 2009

## AUTORIZACION

Yo ARQUIMEDES LAFONT CABALLERO, identificado con la cedula de ciudadanía numero 1.047.388.171 de la ciudad de Cartagena, autorizo a la Universidad Tecnológica de Bolívar para hacer uso de mi trabajo de grado y publicarlo en el catálogo online de la biblioteca.

---

Arquímedes Lafont Caballero  
C.C 1.047.388.171 de Cartagena

Cartagena de Indias D.T y C. Mayo de 2009

## AUTORIZACION

Yo URIEL PAJARO LOPEZ, identificado con la cedula de ciudadanía numero 73.206.904 de la ciudad de Cartagena, autorizo a la Universidad Tecnológica de Bolívar para hacer uso de mi trabajo de grado y publicarlo en el catalogo online de la biblioteca.

---

Uriel Pájaro López  
C.C 73.206.904 de Cartagena

Cartagena de Indias D.T y C. Mayo de 2009

## AUTORIZACION

Yo JUAN CARLOS CUELLO ROBLES, identificado con la cedula de ciudadanía numero 3.839.730 de la ciudad de Corozal, autorizo a la Universidad Tecnológica de Bolívar para hacer uso de mi trabajo de grado y publicarlo en el catalogo online de la biblioteca.

---

Juan Carlos Cuello Robles  
C.C 3.839.730 de Corozal

# TABLA DE CONTENIDO

**INTRODUCCION**

**OBJETIVOS**

**JUSTIFICACION**

**ESTADO DEL ARTE**

**CAPITULO 1. ANÁLISIS DE CONGESTIÓN EN LAS REDES ACTUALES 24**

1.1 Causas y consecuencias.....25

1.2 Calidad de servicio (QoS).....28

1.2.1 Métodos de calidad de servicio.....28

1.2.1.1 Manejo de congestión.....29

1.2.1.2 Evasión de congestión.....29

1.2.2 Métodos de encolado por defecto en routers Cisco.....31

**CAPITULO 2. DEFINICION Y CONFIGURACION DE LOS ALGORITMOS DE ENCOLADO MÁS UTILIZADOS AL APLICAR QoS 32**

2.1 Definición de Priority Queueing (PQ).....33

2.1.1 Configurando PQ.....33

2.1.1.1 Definiendo la lista de prioridad.....34



2.1.1.2 Asignando la lista de prioridad a una interfaz.....	38
<b>2.2 Definición de Custom Queueing (CQ).....</b>	<b>39</b>
2.2.1 Configurando CQ.....	39
2.2.1.1 Definiendo la lista de colas personalizadas.....	40
2.2.1.2 Especificando el tamaño máximo de las colas personalizadas.....	41
2.2.1.3 Asignando paquetes a las colas personalizadas.....	42
2.2.1.4 Monitoreando las listas de colas personalizadas.....	43
<b>2.3 Definición de Weighted Fair Queueing (WFQ).....</b>	<b>44</b>
2.3.1 Configurando WFQ.....	45
<b>CAPITULO 3. ANALISIS DE CONGESTION EN UNA RED LAN Y WAN</b>	<b>47</b>
<b>3.1 Tipos de tráfico en la red.....</b>	<b>48</b>
3.1.1 Usuario LAN sin acceso a internet.....	48
3.1.2 Usuario LAN con acceso a internet.....	49
<b>3.2 Descripción de protocolos y servicios más comunes en el tráfico de una red...50</b>	
3.2.1 SMTP (Simple Mail Transfer Protocol).....	50
3.2.2 POP (Post Office Protocol).....	51

3.2.3 FTP (File Transfer Protocol).....	51
3.2.4 HTTP (Hipertext Transfer Protocol).....	52
3.2.5 SQL (Structured Query Language).....	53
3.2.6 IM (Instant Messaging).....	53
3.2.7 Telnet.....	54
<b>3.3</b> Monitoreo en una topología de red WAN.....	<b>55</b>
3.3.1 Protocolo Netflow.....	56
3.3.2 Análisis de resultados en la WAN sin conexión a internet.....	59
3.3.2.1 Tráfico sin priorización.....	59
3.3.2.2 SQL implementando PQ.....	62
3.3.2.3 FTP implementando PQ.....	68
3.3.2.4 SMTP implementando PQ.....	72
3.3.2.5 IM implementando PQ.....	77
3.3.2.6 UDP (Unreal) implementando PQ.....	78
3.3.2.7 SQL implementando CQ.....	83
3.3.2.8 FTP implementando CQ.....	88
3.3.2.9 SMTP implementando CQ.....	91

3.3.2.10 UDP (Unreal) implementando CQ.....	96
3.3.2.11 Implementación de WFQ para todos los protocolos.....	101
3.3.3 Análisis de resultados en la WAN con conexión a internet.....	107
3.3.3.1 Tráfico sin priorización.....	108
3.3.3.2 HTTP implementando PQ.....	110
3.3.3.3 POP implementando PQ.....	115
3.3.3.4 HTTP implementando CQ.....	120
3.3.3.5 POP implementando CQ.....	125
3.3.3.6 Implementación de WFQ para todos los protocolos.....	130
3.3.3.7 Cuadro comparativo general de algoritmos de encolamiento.....	135
3.3.3.8 Cuadro comparativo de algoritmos de encolamiento por protocolos.....	136
CONCLUSIONES.....	137
RECOMENDACIONES.....	141
ANEXOS.....	143
BIBLIOGRAFIA.....	180

## INTRODUCCION

A medida que las redes (en especial el internet) han ido evolucionando, los usuarios también han ido cambiando. No importa si sean empresas o personas particulares, el tratamiento que se le dé al manejo de su información es cada vez más importante. Por lo tanto, cada usuario será cada vez más exigente de acuerdo a la evolución de sus necesidades.

El volumen de la información que transita por las redes hoy en día es muy alto pero la capacidad de las redes no es infinita, por lo tanto ésta se tiene que ver como un recurso que debe ser utilizado de la mejor manera y además de eso garantizar la entrega de estos datos. Por eso es necesario hablar de calidad de servicio (QoS). Para obtener esta calidad de servicio y evitar que la congestión haga posible la pérdida o el retraso de datos, hacen su aparición los algoritmos de encolamiento, los cuales actúan al momento de la congestión.

En este trabajo de grado haremos una descripción de los algoritmos de encolamiento más utilizados en las redes empresariales o comunes, de tipo LAN y WAN, y su aplicación para priorizar determinados flujos de datos considerados de mayor importancia de acuerdo a las necesidades del usuario.

Veremos cómo su implementación en routers Cisco serie 2800 afecta el desempeño de la red en momentos de gran congestión, y determinaremos si son adecuados para tratar distintos tipos de tráfico existentes de acuerdo a su comportamiento durante el monitoreo que se efectúe.

## RESUMEN

En nuestro primer capítulo hacemos referencia a la calidad de servicio (QoS) como una alternativa para la optimización del ancho de banda en las redes LAN y WAN, definiendo brevemente las formas en que trata o maneja los recursos de red y los métodos empleados para dicha tarea.

En el segundo capítulo se definen más profundamente cada uno de los algoritmos de encolamiento empleados como métodos de manejo de congestión en la red, sus comandos y parámetros de configuración en routers Cisco. Haremos énfasis en 3 de ellos, los cuales consideramos de mayor importancia, factible implementación y utilizaremos en nuestras prácticas de laboratorio.

El tercer capítulo de este trabajo determinará nuestros objetivos de investigación en su totalidad. Realizaremos una clasificación por tipos de usuario de acuerdo a sus necesidades, que nos servirá de guía para emular las condiciones de una red empresarial, institucional o de hogar. Además describiremos cada uno de los protocolos de uso más frecuente en una red utilizados por la capa de aplicación, los tipos de datos que por ella circulan y su naturaleza de sensibilidad a retardos. Esto nos servirá para posteriormente crear el ambiente de cada tipo de red, simular casos de alta congestión, emplear los algoritmos de encolado seleccionados para priorizar cada tipo de tráfico por separado y mirar el comportamiento que la red presenta.

## **OBJETIVO GENERAL**

Comparar los distintos algoritmos de encolamiento de paquetes con el propósito de evaluar y mejorar la calidad de servicios en redes.

## **OBJETIVOS ESPECIFICOS**

- Estudiar y analizar el desempeño de los algoritmos de encolamiento que existen actualmente para el manejo de la congestión en las redes.
- Realizar prácticas básicas de laboratorio donde se configuren los distintos tipos de encolado.
- Monitorear el tráfico de la LAN y WAN con el propósito de confrontar las técnicas de encolamiento frente a situaciones de congestión en la red.
- Determinar condiciones de tráfico en la red en las que los algoritmos estudiados presentan dificultades y en las que podrían mostrar un mejor desempeño.
- Determinar las condiciones adecuadas para seleccionar el algoritmo que se acople más eficientemente a los requerimientos de cada administrador de red en la LAN y WAN.

## JUSTIFICACION

Para hacer viable el uso de las aplicaciones que consumen grandes anchos de banda y manejar los tráficos multimedia actuales, no es suficiente con añadir mayor capacidad a las redes implantando Gigabit Ethernet o fibra óptica. Es necesario además utilizar métodos que permitan controlar los tráficos dando prioridad al más crítico, dependiendo de las actividades realizadas en la red. Aquí una de las técnicas de calidad de servicio (QoS), como lo es el manejo de tráfico a través de algoritmos de encolamiento, es fundamental.

Cuando el tráfico de negocio crítico en una empresa, como videoconferencias o telefonía LAN, debe competir con el resto de datos que circulan por la red corporativa, lo más probable es que se presenten **cuellos de botella** que degraden la calidad de las aplicaciones más sensibles a los retardos. Aunque la mayoría de las veces el tiempo de respuesta de la red sea el suficiente, puede haber ocasiones en que se produzcan congestiones. Además una red tradicional no es capaz de diferenciar los distintos tipos de tráficos y, en consecuencia, a todos les da el mismo trato.

Las técnicas de calidad de servicio (QoS) permiten a los administradores de TI<sup>1</sup> tomar el control del tráfico de datos para asegurar el rendimiento de la red de un modo eficiente. En condiciones normales no es necesaria, pero hay eventos que afectan el rendimiento de las redes incluso cuando están bien diseñadas. Aunque la red se sobrecargue, QoS asegura que el tráfico crítico no sea ni perdido ni retardado. En otras palabras, añade fiabilidad y disponibilidad.

En general, el motivo por el cual iniciamos este proceso de investigación es enfocarnos hacia las tendencias del tráfico en la WAN. A medida que transcurre el tiempo, surgen diferentes tipos de contenidos en la red y con ellos las necesidades se hacen mayores y diversas. Para garantizar calidad de servicio, es fundamental

---

<sup>1</sup> TI: Tecnologías de la Información. Se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como medio de sistema informático.

analizar si los algoritmos de encolamiento y los dispositivos empleados actualmente se adaptan a estas nuevas necesidades o deben ser rediseñados en un futuro.

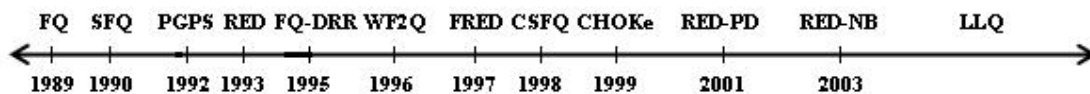
Por esto, es de vital importancia hacer un estudio detallado de las técnicas de “manejo de congestión” con dispositivos fabricados actualmente, a fin de observar el alcance o magnitud de la combinación de estas tecnologías.



## ESTADO DEL ARTE

En el pasado se han realizado diferentes aproximaciones al problema de encolamiento de los paquetes en el router dirigidos a distribuir de manera equitativa entre los diferentes flujos la capacidad de un enlace compartido.

En esta sección presentamos las principales investigaciones realizadas en el área en orden cronológico.



**FIFO** Es el tipo más simple de encolamiento, se basa en el siguiente concepto: el primer paquete en entrar a la interfaz, es el primero en salir. Es adecuado para interfaces de alta velocidad, sin embargo, no para bajas, ya que FIFO es capaz de manejar cantidades limitadas de ráfagas de datos. Si llegan más paquetes cuando la cola está llena, éstos son descartados. No tiene mecanismos de diferenciación de paquetes.

### 1989

Aunque originalmente propuesto por Nagle en 1987, Demers et al publicó en 1989 el diseño más popular del primer algoritmo de Fair Queuing (**FQ**).

FQ es un algoritmo de encolamiento de paquetes en el router que opera generando una cola para cada flujo que atraviesa el router; es decir, siempre que un paquete llega al router, este debe determinar a qué flujo pertenece con base en sus direcciones IP y puertos TCP / UDP origen y destino, para con base en ello

clasificarlo como perteneciente a determinado flujo y colocarlo en la cola correspondiente. Cada cola es luego atendida en una forma que se aproxima a un Round Robin bit a bit, asegurando, por tanto igual distribución de canal entre los diferentes flujos. Round Robin bit a bit significa que el router lee el primer bit del primer paquete de la primera cola y lo transmite; a continuación lee el primer bit del primer paquete de la segunda cola y lo transmite, y continúa sobre todas las colas generadas. Al llegar a la última, vuelve a la primera y transmite el segundo bit, dando inicio a otra nueva ronda. De esta manera, todas las colas transmiten la misma cantidad de información por unidad de tiempo.

## 1990

En 1990 McKenney desarrolló una versión de FQ que pretendía disminuir sus costos en términos de procesamiento en la etapa de clasificación de paquetes, cantidad de memoria requerida, así como en el cálculo del Round Robin bit a bit, denominada Stochastic Fair Queuing (**SFQ**).

Para esto, McKenney decidió que en vez de mapear cada flujo en una cola propia, resultaba más económico mapearlas a una de un conjunto fijo de colas disponibles en el router mediante una función de hashing, requiriendo menos memoria y menos velocidad en el manejo de la misma.<sup>2</sup>

## 1992

K. J. Parekh en su trabajo de doctorado desarrolló un modelo conocido como Packet by packet generalized processor sharing (**PGPS**), que resultó equivalente a FQ excepto porque no se asigna la misma prioridad (la misma participación sobre el

---

<sup>2</sup>Algoritmos de encolamiento en routers para distribución justa de la capacidad de un enlace compartido en una red TCP/IP, Eduardo J. Ortega, universidad nacional de Colombia

canal) a todos los flujos, sino que se permite que algunos tengan más prioridad que otros (por esto, a esta variedad de FQ se le denomina Weighted Fair Queuing, o WFQ).

El trabajo de Parekh despertó amplio interés para el desarrollo de redes que garantizaran ciertos servicios (las denominadas redes de servicios integrados). Sin embargo, presentaba los mismos problemas de WFQ; elevado costo computacional para su implementación.

### **1993**

En 1993 S. Floyd y V. Jacobson propusieron un algoritmo para evitar la congestión en redes denominado Random Early Detection (**RED**). Como tal, no está explícitamente diseñado para distribuir justamente el canal entre los diferentes flujos. Sin embargo, dado que varios algoritmos desarrollados posteriormente se basan en RED, lo expondremos aquí. Como ya se mencionó, el objetivo de RED es prevenir la congestión. La forma en que pretende realizarlo es monitoreando el tamaño de la cola de paquetes a ser transmitidos y descartando paquetes aleatoriamente (aún cuando la cola no esté llena) cuando detecta el comienzo de un periodo de congestión.

### **1995**

Shreedhar y Varghese, en un intento por disminuir la complejidad computacional de FQ, desarrollaron una variante de éste, denominada Fair Queuing with Deficit Round Robin (**FQ-DRR**). En realidad, FQ-DRR no es una variación de FQ directo, sino de SFQ. Shreedhar y Varghese realizaron una aproximación al problema en un punto medio entre FQ y SFQ: intentaron aproximar el comportamiento de round Robin bit a bit de una manera más eficiente que FQ, y más precisa que el Round Robin paquete

por paquete de SFQ. El resultado es lo que denominaron Deficit Round Robin (DRR), que explicaremos a continuación.

En cada ronda, DRR decide si transmitir el paquete de la cabeza con base en su tamaño; si transmitirlo significa otorgarle mayor participación a la cola en la distribución del enlace de la que debería, el paquete no es transmitido, pero el router guarda la memoria de este evento. De esta forma, en la siguiente ronda, el router sabe cuánto “le debe” a dicha cola y la compensa. De esta manera, en promedio cada cola es atendida equitativamente.

## **1996**

Bennet y Zhang, estudiaron la propuesta de WFQ (o PGPS) Parekh, encontrando que, aunque correcta, había sido malinterpretada posteriormente, aduciendo que se trataba de la mejor aproximación posible al modelo GPS. Bennett y Zhang determinan que WFQ no es, después de todo, una precisa aproximación al modelo GPS en lo referente a sus propiedades de justicia en la distribución del canal cuando este se comparte. En particular, encuentran que bajo condiciones de enlace compartido, no es cierto que el delay por flujo de WFQ esté acotado por máximo el tiempo de transmisión de un paquete de dicho flujo. Estas inexactitudes de WFQ para simular GPS, así como su alto costo computacional, motivaron el desarrollo de Worst case Fair Queuing (**WF2Q y WF2Q+**), dos algoritmos que presentan un comportamiento más cercano a GPS que WFQ, y uno de los cuales (WF2Q+) cuenta con menor complejidad. Los resultados de las simulaciones realizadas por Bennett y Zhang muestran que estos dos algoritmos superan en precisión a WFQ.

## **1997**

En 1997, Lin y Morris realizaron estudios sobre el desempeño de RED y probaron que estaba lejos de ser óptima, propusieron una versión modificada de RED,

denominada Flow Random Early Drop (**FRED**). FRED es RED con las siguientes modificaciones:

FRED no descarta los paquetes al estilo de RED cuando llegan si el número de paquetes en la cola de este flujo es menor que un parámetro dado *cmin*, esto previene que los flujos frágiles sufran ante la presencia de flujos robustos o no adaptivos en tiempos de congestión.

FRED impone una cota *cmax* al número de paquetes que cada flujo puede tener en la cola. Si un flujo intenta encolar más de *cmax* paquetes, son descartados. Adicionalmente, al flujo se le sanciona descartando todos sus paquetes entrantes hasta que el número de paquetes que tenga en la cola sea menor que el número medio de paquetes por flujo en la cola. Esto previene que los flujos no adaptivos tomen control de toda la cola y nieguen el servicio a los adaptivos.

## **1998**

Zhang et al. diseñaron el algoritmo Core-Stateless fair Queuing (**CSFQ**). CSFQ divide los routers de la red en dos categorías: **Routers de núcleo o Core routers y Routers de borde o Edge routers.**

CSFQ opera de la siguiente manera: cuando un paquete llega a uno de los routers de borde, éste determina a qué flujo pertenece y calcula su tasa de transferencia promedio, “marcando” los paquetes que reciben con esta información. Por otro lado, todos los routers (de borde y núcleo) calculan la tasa promedio total a la que reciben y aceptan paquetes (donde aceptarlos significa que no son descartados probabilísticamente, según explicaremos a continuación). Con esta información, así como con la capacidad total del canal, los routers estiman  $\alpha$ , la porción justa del enlace que debe tener cada flujo. Cada vez que un paquete es recibido por un router (de borde o de núcleo), es descartado con probabilidad proporcional al exceso de la tasa de llegada del mismo (almacenada en la etiqueta colocada por los routers de núcleo al paquete) sobre  $\alpha$ .

## 1999

En 1999 se desarrolló un nuevo algoritmo denominado CHOOSE and Keep for responsive flows, CHOOSE and Kill for unresponsive flows (**CHOKe**), que pretende brindar mejores niveles de justicia que RED sin requerir, al igual que RED, de almacenamiento de información por flujo en el router. El diseño de CHOKe se encuentra fuertemente basado en RED; se definen dos tamaños de colas  $s_{th}$  y  $s_{max}$ . Cuando un paquete llega se calcula el tamaño medio de la cola  $s_{avg}$  y se compara contra  $s_{th}$  y  $s_{max}$ . Si  $s_{avg} < s_{th}$ , el paquete es aceptado y se pone en la cola de salida, que es un FIFO tradicional. Si  $s_{avg} > s_{max}$ , el paquete es descartado.

## 2001

Floyd et al. desarrolló un algoritmo basado en RED, denominado Random Early Detection with Preferential Dropping (**RED-PD**), que aumenta la probabilidad de descarte de los paquetes de flujos responsables de una amplia fracción del tráfico. Para ello, un router RED-PD mantiene un historial de los flujos que recientemente han sufrido descarte de paquetes debido al tratamiento usual de RED. Al recibir un nuevo paquete, RED-PD determina si éste pertenece a un flujo que se encuentra en este historial. Si es así, entonces RED-PD calcula una probabilidad de descartarlo con base en la tasa de descartes actual de RED y el número de descartes promedio por flujo durante las últimas  $t$  unidades de tiempo. Si el flujo no es descartado, o si no se trataba de un flujo monitoreado (un flujo en el historial), entonces el paquete es entregado a RED. En ese sentido, el descarte preferencial (PD en el nombre RED-PD) actúa como un prefiltro para un RED tradicional, cuyos parámetros se calculan con base en la historia de descartes de RED.

## 2003

Wang trabajó en el mejoramiento de un algoritmo denominado Random Early Detection with No Bias (**RED-NB**), que supere a RED en términos de justicia. Aunque a la fecha su trabajo no ha sido presentado formalmente más allá de un reporte técnico, a primera vista los resultados son promisorios, especialmente si se tiene en cuenta que RED-PD brinda mejores resultados que la mayoría de algoritmos que no presentan administración por flujo completa (como la familia FQ), y RED-NB parece superarlo.

## 2003 – 2008

El Encolamiento de Baja Latencia (**LLQ**: Low-Latency Queueing) es una mezcla entre Priority Queueing y Class-Based Weighted-Fair Queueing (CBWFQ). Utilizado ampliamente por Cisco, este algoritmo permite el uso de una cola de prioridad estricta, la cual tiene preferencia absoluta sobre las demás colas. Se crean clases de prioridad, y el tráfico de todas ellas es dirigido hacia la única cola de prioridad estricta. Si la cola de prioridad no está encolando paquetes, se procede a atender las otras colas según su prioridad.

LLQ trabaja más que todo con datos sensibles a retardos, como por ejemplo voz. Tráficos como el video pueden introducir variaciones en el retardo que comprometen la transmisión correcta del tráfico de voz. Es actualmente el método de encolamiento recomendado para Voz sobre IP (VoIP) y Telefonía IP, que también trabajará apropiadamente con tráfico de videoconferencias.

Una de las características más importantes que tiene es que, cuando hay congestión, el tráfico que sobrepasa el ancho de banda asignado se descarta. Para solucionar esto se restringe el tráfico de prioridad a su ancho de banda asignado, de manera que los paquetes que no tienen prioridad no se pierdan.

## CAPITULO 1. ANÁLISIS DE CONGESTIÓN EN LAS REDES ACTUALES

Debido a la creciente demanda de servicios de telecomunicaciones con características especiales, así como a la constante integración que ha tenido el mercado mundial a nivel de servicios, las redes de telecomunicaciones han abrigado la necesidad de interconectarse de formas diversas y sofisticadas.

Hoy en día hay una completa integración de diversos tipos de contenido y servicios que generan tráfico en el canal de la red, tales como datos, contenidos multimedia, servicios VoIP, tv digital, etc. Es deseable que no ocurra pérdida de información, que exista un gran ancho de banda disponible, y que los retrasos en los envíos de estos paquetes de datos sean mínimos.

Muchos de las redes empresariales de hoy día no tienen la certeza de si los flujos de información que circulan a través de sus canales de comunicación están siendo manejados de la mejor manera (optimización de recursos), o si cuando se está expuesto a la congestión las técnicas usadas para su manejo son las más adecuadas. Esto conlleva a generar mayores costos de inversión en nuevos equipos que logren satisfacer las necesidades de los usuarios que dan uso a estos servicios, o a los incrementos innecesarios en la capacidad de los canales de comunicación.

Es por ello que surge la necesidad de utilizar mecanismos de priorización para las comunicaciones que se generan en la WAN y LAN a fin de poder distribuir el tráfico de acuerdo a su importancia y utilidad. Por esto y otros factores, se han desarrollado varios algoritmos de administración de la cola del router a fin de garantizar la Calidad del Servicio (QoS) en la red.

Hay que avanzar en materia de manipulación y tratamiento de congestión, porque si garantizamos QoS en nuestras redes, estamos generando una mejor alternativa que de cierta forma se convierte en una vital herramienta para hacer un excelente uso de los recursos con los que contamos actualmente, disminuyendo costos, garantizando calidad y perdurabilidad en el tiempo, ofreciendo confiabilidad y ante todo certificando un excelente servicio en las redes de telecomunicaciones.



## 1.1 CAUSAS Y CONSECUENCIAS DE LA CONGESTION EN LAS REDES

### ► Causas:

- **Memoria insuficiente en los conmutadores:** Los paquetes se reciben demasiado deprisa para ser procesados (lo que produce que se llene la memoria de entrada). Además puede ser que en la memoria de salida haya demasiados paquetes esperando ser asentidos, entonces se llena memoria de salida.
- **Un solo usuario acapara la conexión:** Cuando tenemos más de una maquina conectada a la red a través de la misma salida suele pasar que si una de ellas empieza a consumir mucho ancho de banda, acapara la conexión haciendo que las demás maquinas no tengan casi ancho de banda y la latencia (tiempo de respuesta de la conexión) sea muy baja.
- **Los buffers del router/modem no dan abasto:** En teoría las conexiones a Internet de banda ancha (cable, adsl, etc) son fullduplex, por lo que se puede enviar y recibir a la vez sin problemas. Pero a menudo cuando se satura un canal (el de subida o el de bajada) el otro canal se resiente. Esto es debido a que los paquetes que no puede enviar el router o modem se guardan en unos buffers internos, los cuales suelen ser bastante limitados y se comportan de una manera que no podemos controlar.
- **Exceso de aplicaciones que operan sobre la red:** En muchos casos se corren aplicaciones que requieren recursos superiores a los que puede proporcionar la red y que no han sido adecuadamente previstos. Por ejemplo, es frecuente que en empresas que utilizan sistemas de información que realizan consultas frecuentes sobre bases de datos, no se haya previsto adecuadamente el modo en que opera la aplicación, y que las

consultas a la base de datos generen congestión sobre la red en los horarios pico.

Adicionalmente hay que tener presente que en la actualidad hay múltiples aplicaciones (desde los sistemas de VoIP hasta aplicaciones de audio y video sobre Internet) que generan tráfico de cadenas de paquetes sobre UDP que tienden a ocupar todo el ancho de banda disponible. Una combinación de estos elementos puede volver excesivamente lenta aún una red FastEthernet de 100Mbps.<sup>3</sup>

### ► Consecuencias:

- **Retardos:** Trabajar cerca de la capacidad de los enlaces es ideal desde el punto de vista de la productividad, pero no lo es respecto al retardo. Se experimentan grandes retardos en una cola según la tasa de llegadas de paquetes se acerca a la capacidad del enlace.
- **Pérdidas de datos:** Como los búferes no son de tamaño infinito el emisor debe realizar retransmisiones para compensar los paquetes perdidos debido al desbordamiento de los búferes.
- **Desperdicio de recursos tecnológicos:** Las retransmisiones innecesarias realizadas por el emisor en presencia de grandes retardos, que provocan que venzan los temporizadores de retransmisión antes de que lleguen los asentimientos, hacen que el ancho de banda de los enlaces se utilice para encaminar copias innecesarias de los paquetes.

Cuando un paquete es desechado a lo largo de un camino, la capacidad de almacenamiento, procesamiento y transmisión que fue utilizada en cada uno de los nodos y enlaces anteriores, para encaminar ese paquete hasta el punto en el que es desechado, está siendo desperdiciada.<sup>4</sup>

---

<sup>3</sup> [http://es.wikipedia.org/wiki/Congestión\\_de\\_red](http://es.wikipedia.org/wiki/Congestión_de_red)

<sup>4</sup> [http://www-gris.det.uvigo.es/~estela/SC0708/control\\_congestion.pdf](http://www-gris.det.uvigo.es/~estela/SC0708/control_congestion.pdf)

- **Baja productividad y calidad en los servicios:** Si existe congestión en una red empresarial, las operaciones podrían no realizarse o demorarse y los recursos humanos destinados a ellas no serían aprovechados al máximo. Por lo tanto, se incrementan los costos, los tiempos de respuesta y se afecta la competencia.

## **1.2 CALIDAD DE SERVICIO (QoS)**

Se entiende por “Calidad de Servicio”, a la capacidad de una red para sostener un comportamiento adecuado del tráfico que transita por ella, cumpliendo a su vez con los requerimientos de ciertos parámetros relevantes para el usuario final. Esto puede entenderse también, como el cumplimiento de un conjunto de requisitos estipulados en un contrato (SLA: Service Level Agreement) entre un ISP (Internet Service Provider, proveedor de servicios de Internet) y sus clientes.

Al contar con QoS, es posible asegurar una correcta entrega de la información necesaria o crítica, para ámbitos empresariales o institucionales, dando preferencia a aplicaciones de desempeño crítico, donde se comparten simultáneamente los recursos de red con otras aplicaciones no críticas. QoS hace la diferencia, al prometer un uso eficiente de los recursos ante la situación de congestión, seleccionando un tráfico específico de la red, priorizándolo según su importancia relativa, y utilizando métodos de control y evasión de congestión para darles un tratamiento preferencial. Implementando QoS en una red, hace al rendimiento de la red más predecible, y a la utilización de ancho de banda más eficiente.

### **1.2.1 MÉTODOS DE CALIDAD DE SERVICIO**

Existen varios niveles en los cuales se puede proveer de calidad de servicio en una red. Uno de ellos es el de contar con una estrategia de manipulación de los paquetes en caso de congestión (manejo de congestión), o el evitar que la red alcance este estado (evasión de congestión), descartando paquetes a medida que estos ingresan a la red.

### 1.2.1.1 MANEJO DE CONGESTION

El “manejo de congestión” es un término general usado para nombrar los distintos tipos de **estrategia de encolamiento** que se utilizan para manejar situaciones donde la demanda de ancho de banda solicitada por las aplicaciones excede el ancho de banda total de la red, controlando la inyección de tráfico a la red, para que ciertos flujos tengan prioridad sobre otros.

Para ello se han diseñado los **algoritmos de encolamiento**, los cuales son herramientas flexibles, sencillas y poderosas, con las que se pueden implementar políticas de routing en las redes que permiten a los administradores una mejor gestión de sus recursos informáticos para el manejo de tráfico y control de congestión, y así alcanzar QoS. Entre los principales algoritmos se encuentran FIFO, FQ, SFQ, PGPS, FQ-DRR, WF2Q, CSFQ, CHOKe y LLQ, los cuales detallamos anteriormente en el estado del arte.

### 1.2.1.2 EVASION DE CONGESTION

Las metodologías de “evasión de congestión” se basan en la manera que los protocolos operan, con el fin de no llegar a la congestión de la red. Generalmente tratan con situaciones de Sincronización Global, descartando paquetes de forma aleatoria. Lo que limita a estas técnicas es que sólo sirven para tráfico basado en TCP, ya que otros protocolos no utilizan el concepto de ventana deslizante. Las técnicas más utilizadas son:

**RED (*Random Early Detection*):** Es un algoritmo que se utiliza para evitar la congestión. Su trabajo es evitar la congestión en la red, asegurándose de que la cola no se llene. Para ello calcula constantemente la longitud media (el tamaño) de la cola y la compara con dos umbrales o límites, un umbral mínimo y otro máximo. Si el tamaño medio de la cola se encuentra por debajo del umbral mínimo, entonces no se bloqueará ningún paquete. Si el tamaño medio se encuentra por encima del

umbral máximo, entonces *todos* los paquetes nuevos que lleguen serán bloqueados. Si el tamaño medio se encuentra entre los valores de los dos umbrales, entonces se bloquearán los paquetes de acuerdo con un cálculo de probabilidad obtenido a raíz del tamaño medio de la cola. En otras palabras, según se va aproximando el tamaño medio de la cola al umbral máximo, se va bloqueando un número cada vez mayor de paquetes. Cuando bloquea los paquetes, RED escoge de qué conexiones bloqueará los paquetes de una forma aleatoria. Las conexiones que usen mayores cantidades de ancho de banda serán las que tengan una probabilidad más alta de que se bloqueen sus paquetes.

**WRED (*Weighted Random Early Detection*):** Generalmente descarta paquetes selectivamente basándose en una precedencia de IP. Los paquetes con una precedencia alta de IP tienen menor probabilidad de ser descartados que aquellos con una precedencia más baja.

### 1.2.2 MÉTODOS DE ENCOLADO POR DEFECTO EN ROUTERS CISCO

Además del proceso de enrutamiento de paquetes y todas las etapas que conlleva, la mayoría de las series de routers Cisco no distribuidos implementan en su IOS algoritmos por defecto para el manejo de la congestión en la red, dependiendo de la interfaz. Cuando no existe ninguna configuración de QoS aplicada, las IOS de Cisco utilizan Fair Queueing (FQ) y Weighted Fair Queueing (WFQ) en aquellas interfaces de baja velocidad, como las seriales que son capaces de transmitir unos pocos megabits por segundo. Y para aquellas interfaces de alta velocidad de transmisión, como las Fast Ethernet, se utiliza el algoritmo **FIFO** para el encolado de paquetes. Este último método consiste en el proceso simple de encolar y atender cada paquete conforme va llegando, sin alterar su orden.

Para verificar el mecanismo actual de manejo de congestión en cualquier router Cisco, se utiliza el comando **show queueing interface name**. Al aplicarlo antes de realizar las pruebas de laboratorio pertinentes para el presente trabajo sobre las interfaces de routers Cisco 2800, se obtuvieron los siguientes resultados:

```
router#show queueing interface serial 0/1/0
Interface Serial0/1/0 queueing strategy: fair
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 212296
  Queueing strategy: weighted fair
  Output queue: 0/20/64/212295 (size/max total/threshold/drops)
    Conversations 0/149/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 1158 kilobits/sec
```

```
router#show queueing interface fastethernet 0/1
Interface FastEthernet0/1 queueing strategy: none
```

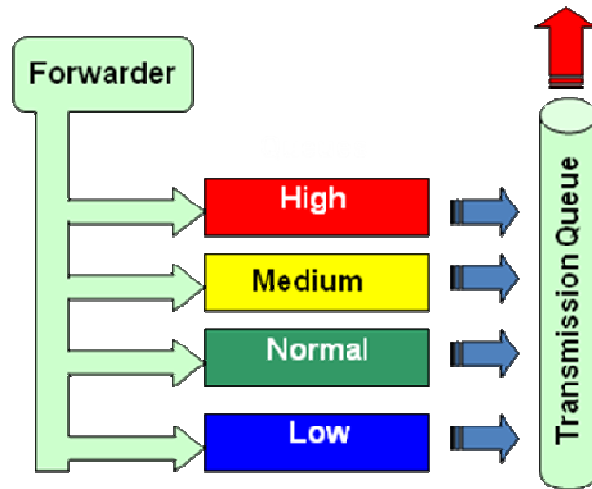
## **CAPITULO 2. DEFINICION Y CONFIGURACION DE LOS ALGORITMOS DE ENCOLADO MÁS UTILIZADOS AL APLICAR QoS**

Los algoritmos existentes han tenido diferentes niveles de éxito a diferentes costos. Aquellos que requieren menor poder de cómputo del router son también aquellos que brindan menores garantías en cuanto a una adecuada distribución del enlace. En el otro extremo, están aquellos que brindan garantías totales, pero requieren grandes cantidades de memoria y un procesador veloz para procesar los paquetes y decidir cómo colocarlos en la cola o entregarlos a la red. Si bien con el avance de la tecnología se ha hecho más económica la fabricación de dispositivos más rápidos, también se han incrementado las velocidades típicas de los enlaces de comunicación, haciendo que dicho procesamiento en el router introduzca un retraso considerable (frente al tiempo de propagación por los enlaces) en la transmisión de paquetes.

A continuación describiremos los algoritmos de encolado más utilizados en las redes actuales, dado el tipo de tráfico para el que fueron creados y deberían acoplarse más efectivamente.



## 2.1 DEFINICIÓN DE PRIORITY QUEUEING (PQ)



El algoritmo de encolamiento por prioridad (PQ) consiste en un conjunto de colas definidas por el usuario para el manejo del tráfico, clasificadas desde alta a baja prioridad. Cada paquete es asignado a una de estas colas, las cuales son servidas en estricto orden de prioridad. Las colas de mayor prioridad son siempre atendidas primero, luego la siguiente de menor prioridad y así sucesivamente. Si una cola de menor prioridad está siendo atendida, y un paquete ingresa a una cola de mayor prioridad, ésta es atendida inmediatamente. Este mecanismo se ajusta a condiciones donde existe un tráfico importante, pero puede causar la total falta de atención de colas de menor prioridad.

### 2.1.1 CONFIGURANDO PQ

Para configurar PQ, se desarrollan las tareas descritas en las siguientes secciones. Las tareas de las primeras 2 secciones son requeridas; la tarea de la sección faltante es opcional.

- Definir la lista de prioridad (Requerido)

- Asignar la lista de prioridad a una interfaz (Requerido)
- Monitorear las listas de encolado de prioridad (Opcional)

### **2.1.1.1 DEFINIENDO LA LISTA DE PRIORIDAD**

Una lista de prioridad contiene las definiciones de un conjunto de colas de prioridad. La lista de prioridad especifica en qué cola va a ser ubicado un paquete, y opcionalmente, la longitud máxima de las diferentes colas.

Para realizar un encolado usando una lista de prioridad, debe asignarse la lista a una interfaz. Una misma lista de prioridad puede ser aplicada a múltiples interfaces. Alternativamente, se pueden crear muchas políticas de prioridad diferentes para interfaces distintas.

Para definir una lista de prioridad, desarrolle las tareas descritas en las siguientes secciones. La tarea de la primera sección es requerida; la de la sección restante es opcional.

- Asignar paquetes a las colas de prioridad (Requerido)
- Especificar el tamaño máximo de las colas de prioridad (Opcional)

#### **Asignando paquetes a las colas de prioridad**

Se asignan paquetes a las colas de prioridad basándose en las siguientes características:

- Tipo de protocolo
- Interfaz por la cual los paquetes entran al router

Se pueden asignar múltiples reglas de asignación. Los comandos **priority-list** son leídos en orden de aparición hasta que el tipo de protocolo o de interfaz sea encontrado. Cuando se encuentra una coincidencia, el paquete es asignado a la cola apropiada y la búsqueda termina. Aquellos paquetes que no concuerden con ninguna de las demás reglas de asignación son asignados a la cola por defecto.

Para especificar en qué cola se va a ubicar un paquete, use los siguientes comandos en modo de configuración global:

	Command	Purpose
Step 1	Router(config)# <b>priority-list</b> list-number <b>protocol</b> protocol-name { <b>high</b>   <b>medium</b>   <b>normal</b>   <b>low</b> }; queue-keyword keyword-value	Establishes queueing priorities based on the protocol type.
Step 2	Router(config)# <b>priority-list</b> list-number <b>interface</b> interface-type interface-number { <b>high</b>   <b>medium</b>   <b>normal</b>   <b>low</b> };	Establishes queueing priorities for packets entering from a given interface.
Step 3	Router(config)# <b>priority-list</b> list-number <b>default</b> { <b>high</b>   <b>medium</b>   <b>normal</b>   <b>low</b> };	Assigns a priority queue for those packets that do not match any other rule in the priority list.

**Paso 1:** Establece prioridades de encolado basadas en el tipo de protocolo.

**Paso 2:** Establece prioridades de encolado para aquellos paquetes que entran por una interfaz dada.

**Paso 3:** Asigna una cola de prioridad a aquellos paquetes que no pertenezcan a ninguna otra regla en la lista de prioridad.

Todos los protocolos soportados por Cisco son permitidos. El argumento *queue-keyword* provee opciones adicionales incluyendo contador de bytes, asignaciones de servicios TCP y sus números de puerto, asignaciones de lista de acceso para XNS, AppleTalk, IP, IPX o VINES.

<b>queue-keyword</b>	<b>Descripción</b>
<b>fragments</b>	Asigna un nivel de prioridad definido a paquetes IP fragmentados (para uso con IP solamente). Más específicamente, este comando trabaja con paquetes IP cuyo campo offset es diferente de cero. El fragmento inicial de un paquete IP fragmentado tiene un fragmento offset de cero, así que dichos paquetes no son considerados por este comando.
<b>gt</b> <i>byte count</i>	Especifica un contador "mayor que". El nivel de prioridad asignado entra en acción cuando el tamaño de un paquete excede el valor especificado por el argumento <i>byte-count</i> . Este tamaño debe también incluir unos bytes adicionales debido a la encapsulación de la MAC sobre la interfaz de salida.
<b>list</b> <i>list-number</i>	Asigna prioridades de tráfico de acuerdo a una lista especificada cuando se trabaja con AppleTalk, bridging, IP, IPX, VINES, o XNS. El argumento <i>list-number</i> es el número de la lista de acceso como se especificó en el comando de configuración global <b>access-list</b> , para el <i>protocol-name</i> especificado. Por ejemplo, si el protocolo es AppleTalk, <i>list-number</i> debe ser un número de lista de acceso válido para AppleTalk.
<b>lt</b> <i>byte-count</i>	Especifica un contador "menor que". El nivel de prioridad asignado entra en acción cuando el tamaño de un paquete es menor que el valor especificado por el argumento <i>byte-count</i> . Este tamaño debe también incluir unos bytes

	adicionales debido a la encapsulación de la MAC sobre la interfaz de salida.
<b>tcp port</b>	Asigna un nivel de prioridad definido para segmentos TCP que se estén originando o estén destinados hacia un puerto especificado (para uso con IP solamente).
<b>udp port</b>	Asigna un nivel de prioridad definido para segmentos UDP que se estén originando o estén destinados hacia un puerto especificado (para uso con IP solamente).

### Especificando el tamaño máximo de las colas de prioridad

Para especificar el máximo número de paquetes permitidos en cada una de las colas de prioridad, se usa el siguiente comando en modo de configuración global:

Command	Purpose
Router(config)# <b>priority-list</b> <i>list-number</i> <b>queue-limit</b> [ <i>high-limit</i> [ <i>medium-limit</i> [ <i>normal-limit</i> [ <i>low-limit</i> ]]]]	Specifies the maximum number of packets allowed in each of the priority queues.

El comando **priority-list queue-limit** se usa para cada lista de prioridad. Los argumentos queue limit por defecto son listados en la siguiente tabla:

Priority Queue Argument	Packet Limits
<i>high-limit</i>	20
<i>medium-limit</i>	40
<i>normal-limit</i>	60
<i>low-limit</i>	80

### 2.1.1.2 ASIGNANDO LA LISTA DE PRIORIDAD A UNA INTERFAZ

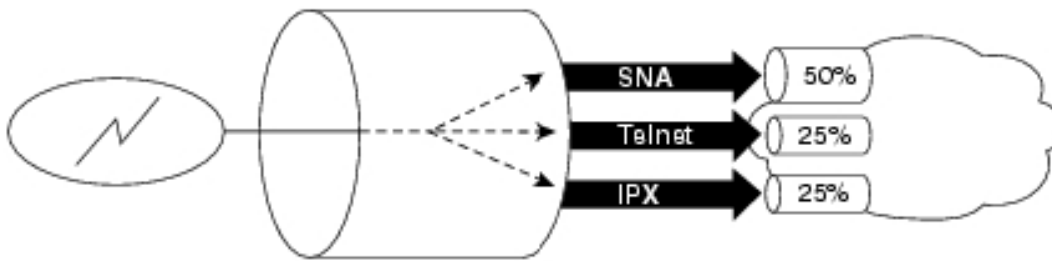
Se puede asignar un número de lista de prioridad a una interfaz. Solo una lista puede ser asignada por interfaz. Para asignar un grupo de prioridad a una interfaz, se usan los siguientes comandos comenzando en modo de configuración global:

	<b>Command</b>	<b>Purpose</b>
<b>Step 1</b>	Router(config)# <b>interface</b> <i>interface-type</i> <i>interface-number</i>	Specifies the interface, and then enters interface configuration mode.
<b>Step 2</b>	Router(config-if)# <b>priority-group</b> <i>list-number</i>	Assigns a priority list number to the interface.

**Paso 1:** Especifica la interfaz, y luego entra al modo de configuración de interfaz.

**Paso 2:** Asigna un numero de lista de prioridad a una interfaz.

## 2.2 DEFINICIÓN DE CUSTOM QUEUEING (CQ)



El algoritmo de encolamiento personalizado (CQ) fue creado como una mejora para evadir la rigidez de PQ, y permite al administrador priorizar el tráfico sin los efectos laterales de inanición de las colas de baja prioridad, especificando el número de paquetes o bytes que deben ser atendidos para cada cola. Se pueden crear hasta 16 colas para categorizar el tráfico, donde cada cola es atendida al estilo Round-Robin. CQ ofrece un mecanismo más refinado de encolamiento, pero no asegura una prioridad absoluta como PQ. Se utiliza CQ para proveer a tráficos particulares de un ancho de banda garantizado en un punto de posible congestión, asegurando para este tráfico una porción fija del ancho de banda y permitiendo al resto del tráfico utilizar los recursos disponibles.

### 2.2.1 CONFIGURANDO CQ

Para configurar CQ, desarrolle las tareas descritas en las siguientes secciones. La primera tarea y la tercera son requeridas, las demás son opcionales:

- Definir la lista de colas personalizadas (Requerido)
- Especificar el tamaño máximo de las colas personalizadas (Opcional)

- Asignar paquetes a las colas personalizadas (Requerido)
- Monitorear las listas de colas personalizadas (Opcional)

### 2.2.1.1 DEFINIENDO LA LISTA DE COLAS PERSONALIZADAS

Para asignar una lista de colas personalizadas a una interfaz, se usan los siguientes comandos comenzando en modo de configuración global:

	Command	Purpose
Step 1	Router(config)# <b>interface</b> interface-type interface-number	Specifies the interface, and then enters interface configuration mode.
Step 2	Router(config-if)# <b>custom-queue-list</b> list	Assigns a custom queue list to the interface. The list argument is any number from 1 to 16. There is no default assignment.

**Paso 1:** Especifica la interfaz, y luego entra al modo de configuración de interfaz.

**Paso 2:** Asigna una lista de colas personalizadas a una interfaz. El argumento *list* es cualquier numero en el rango de 1 a 16. No existe ningún valor por defecto que se asigne.

El comando **custom-queue-list** se utiliza en lugar de **priority-list**. Solo una lista de colas puede ser asignada por cada interfaz.

CQ permite una igualdad que no proveen métodos como priority queueing (PQ). Con CQ se puede controlar el ancho de banda disponible sobre una interfaz cuando esta no puede acomodar el tráfico encolado agregado. Hay un contador configurable de bytes asociado con cada cola de salida, el cual especifica cuantos bytes de datos deben ser entregados por el sistema desde la cola actual, antes que el sistema se mueva hacia la siguiente cola. Cuando una cola en particular está siendo



procesada, los paquetes son enviados hasta que el número de bytes enviados exceda el contador de bytes definido por el comando **queue-list queue byte-count**, o hasta que la cola se vacía completamente.

### 2.2.1.2 ESPECIFICANDO EL TAMAÑO MÁXIMO DE LAS COLAS PERSONALIZADAS

Se puede especificar el número máximo de paquetes permitidos en cada una de las colas personalizadas. El valor por defecto es de 20 entradas.

También es posible especificar el número aproximado de bytes que van a ser reenviados desde cada cola durante su turno en el ciclo. Este número es usado como promedio, puesto que todos los paquetes completos deben ser reenviados.

Para especificar este número aproximado de bytes, se usan los siguientes comandos en modo de configuración global, según se necesite:

Command	Purpose
Router(config)# <b>queue-list</b> <i>list-number</i> <b>queue</b> <i>queue-number</i> <b>limit</b> <i>limit-number</i>	Specifies the maximum number of packets allowed in each of the custom queues. The <i>limit-number</i> argument specifies the number of packets that can be queued at any one time. The range is from 0 to 32767.
Router(config)# <b>queue-list</b> <i>list-number</i> <b>queue</b> <i>queue-number</i> <b>byte-count</b> <i>byte-count-number</i>	Designates the average number of bytes forwarded per queue. The <i>byte-count-number</i> argument specifies the average number of bytes the system allows to be delivered from a given queue during a particular cycle.

**Opción 1:** Especifica el máximo número de paquetes permitidos en cada una de las colas personalizadas. El argumento *limit-number* especifica el número de paquetes que pueden ser encolados en cualquier momento. El rango va de 0 a 32767.

**Opción 2:** Designa el número promedio de bytes reenviados por cola. El argumento *byte-count-number* especifica el número promedio de bytes que el sistema permite para ser entregados a una cola determinada durante un ciclo particular.

### 2.2.1.3 ASIGNANDO PAQUETES A LAS COLAS PERSONALIZADAS

Se pueden asignar paquetes a las colas personalizadas basándose en el tipo de protocolo o interfaz por donde los paquetes entran al router. Adicionalmente, se puede determinar una cola por defecto para aquellos paquetes que no concuerden con las otras reglas de asignación. Se pueden especificar también múltiples reglas. Para definir las listas CQ, se usan los siguientes comandos en modo de configuración global, según se necesite:

Command	Purpose
Router(config)# <b>queue-list</b> list-number <b>protocol</b> protocol-name queue-number queue-keyword keyword-value	Establishes queuing priorities based on the protocol type.
Router(config)# <b>queue-list</b> list-number <b>interface</b> interface-type interface-number queue-number	Establishes CQ based on packets entering from a given interface.
Router(config)# <b>queue-list</b> list-number <b>default</b> queue-number	Assigns a queue number for those packets that do not match any other rule in the custom queue list.

**Opción 1:** Establece prioridades de encolado basándose en el tipo de protocolo.

**Opción 2:** Establece CQ basándose en los paquetes que entran al router por una interfaz dada.

**Opción 3:** Asigna un número de cola para aquellos paquetes que no concuerden con ninguna otra regla en la lista de colas personalizadas.

Todos los protocolos soportados por Cisco son permitidos. El argumento *queue-keyword* provee opciones adicionales incluyendo contador de bytes, asignaciones de servicios TCP y sus números de puerto, asignaciones de lista de acceso para XNS, AppleTalk, IP, IPX o VINES.

Cuando se utilizan múltiples reglas, hay que recordar que el sistema lee los comandos **queue-list** en orden de aparición. Cuando se clasifica un paquete, el sistema busca la lista de reglas especificada por los comandos **queue-list** para encontrar un tipo de protocolo o interfaz que concuerde. Cuando una coincidencia es encontrada, el paquete es asignado a la cola apropiada. La lista es examinada en el orden que tiene especificado, y la primera regla que concuerde termina la búsqueda.

#### 2.2.1.4 MONITOREANDO LAS LISTAS DE COLAS PERSONALIZADAS

Para mostrar información acerca de las colas de entrada y salida cuando CQ es habilitado en una interfaz, se usan los siguientes comandos en modo EXEC, según se necesite:

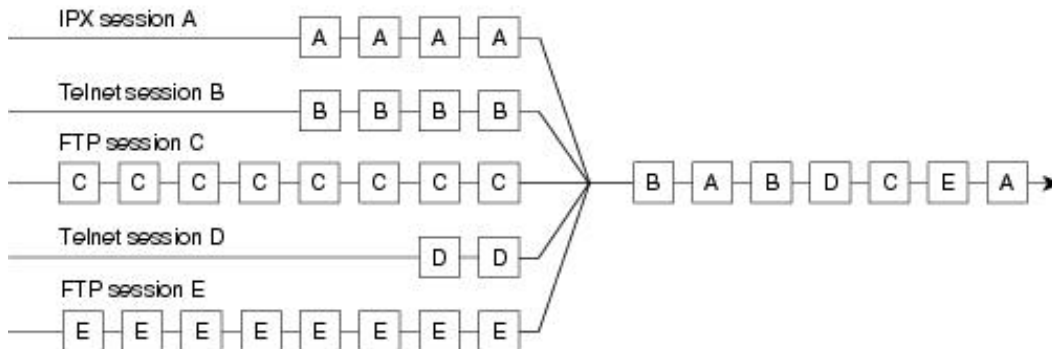
Command	Purpose
Router# <b>show queue</b> <i>interface-type interface-number</i>	Displays the contents of packets inside a queue for a particular interface or virtual circuit (VC).
Router# <b>show queueing custom</b>	Displays the status of the CQ lists.
Router# <b>show interfaces</b> <i>interface-type interface-number</i>	Displays the current status of the custom output queues when CQ is enabled.

**Opción 1:** Muestra los contenidos de paquetes dentro de una cola para una interfaz particular o circuito virtual (VC).

**Opción 2:** Muestra el estado de las listas CQ.

**Opción 3:** Muestra el estado actual de las colas personalizadas de salida cuando CQ es habilitado.

### 2.3 DEFINICIÓN DE WEIGHTED FAIR QUEUEING (WFQ)



El algoritmo de encolado justo o equitativo por pesos (WFQ) es un método automatizado que provee una justa asignación de ancho de banda para todo el tráfico de la red, utilizado habitualmente para enlaces de velocidades menores a 2048 Mbps. WFQ provee un manejo de prioridad de tráfico que automáticamente realiza una clasificación entre flujos de tráfico individuales, sin requerir que primero se definan listas de acceso. Los paquetes con la misma ip de origen, ip de destino, puerto TCP/UDP de origen, o puerto TCP/UDP de destino, pertenecen al mismo flujo. Una vez distinguidos estos flujos, el enrutador determina cuáles son de uso intensivo o sensibles al retardo, priorizándolos y asegurando que los flujos de alto volumen sean empujados al final de la cola, y los volúmenes bajos, sensibles al retardo, sean empujados al principio de la cola. Para esto el algoritmo determina un número de secuencia en cada cola, marcando cada paquete de acuerdo a su valor de precedencia ip en el encabezado y calculando su peso. Las ecuaciones que se utilizan en la mayoría de routers Cisco son:

**Weight** = 4096 / (IP Prec. + 1)

**Sequence number** = Weight \* NR

**NR:** Numero de rondas byte por byte que utiliza el algoritmo para leer los paquetes.

De esta manera los paquetes con el número de secuencia menor son transmitidos primero.

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

### 2.3.1 CONFIGURANDO WFQ

Para configurar WFQ sobre una interfaz, se usan los siguientes comandos en Modo de Configuración de Interfaz:

Command	Purpose
<code>Router(config-if)# fair-queue [congestive-discard-threshold [dynamic-queues [reservable-queues]]]</code>	Configures an interface to use WFQ.

- *congestive-discard-threshold*: (Opcional) Número de mensajes permitidos en cada cola. Por defecto son 64 mensajes, y un nuevo umbral de congestión (threshold) debe tener un poder de 2 en el rango de 16 a 4096 colas. Cuando una conversación alcanza este umbral máximo (threshold), los paquetes de mensajes nuevos son descartados.
- *dynamic-queues*: (Opcional) Número de colas dinámicas usadas para conversaciones de “mejor esfuerzo” (son conversaciones normales que no

requieren ningún servicio especial de red). Los valores son 16, 32, 64, 128, 256, 512, 1024, 2048, y 4096.

- *reservable-queues*: (Opcional) Numero de colas reservables usadas para conversaciones reservadas en el rango de 0 a 1000. El valor default es 0. Las colas reservables se utilizan en interfaces configuradas para determinadas funciones o características, como el Protocolo de Reserva de Recursos (RVSP).

Si no se configuran estos parámetros, el algoritmo utiliza unos parámetros establecidos de acuerdo al ancho de banda en la interfaz, como se describen en la siguiente tabla:

Bandwidth Range	Number of Dynamic Queues
Less than or equal to 64 kbps	16
More than 64 kbps and less than or equal to 128 kbps	32
More than 128 kbps and less than or equal to 256 kbps	64
More than 256 kbps and less than or equal to 512 kbps	128
More than 512 kbps	256

WFQ usa un servicio de registro de discriminación de flujo de tráfico de datos, para determinar a qué flujo de tráfico pertenece un mensaje.

Los valores defaults para el umbral de la congestión (threshold) son provistos después que los mensajes para conversaciones de ancho de banda alto son descartados, y para el número de colas reservables y dinámicas. Sin embargo, se puede mejorar el desempeño de la red cambiando estos estándares.

### **CAPITULO 3. ANALISIS DE CONGESTION EN UNA RED LAN Y WAN**

En las eventuales congestiones de enlaces que son parte del recorrido del tráfico entre dos equipos (host, o terminal) de distintas redes, cada paquete de información compite por un poco de ancho de banda disponible para poder alcanzar su destino. Típicamente, las redes operan en la base de entrega del mejor, donde todo el tráfico espera ser entregado a tiempo. Un usuario experimenta o genera tráfico en una red a través de programas de mensajería instantánea, conexión remota, operaciones a una base de datos y muchos otros dependiendo de los requerimientos del mismo y las condiciones de la red. Cuando ocurre la congestión, todo este tráfico tiene la misma probabilidad de ser descartado.

En ciertos tipos de datos que circulan por las redes hoy en día, por ejemplo, tráficos con requerimientos de tiempo real (voz o video), es deseable que no ocurra pérdida de información, que exista un gran ancho de banda disponible, y que los retrasos en los envíos de estos paquetes de datos sean mínimos. Es por ello que surge la necesidad de aplicar Calidad de Servicio (QoS) en el nivel del transporte de datos, métodos de diferenciación y encolamiento de tráficos particulares con el fin de otorgar preferencia a estos datos sensibles.

En el transcurso de este capítulo se describirá cada clase de tráfico para un usuario de tipo común o empresarial, y se analizará el desempeño de los algoritmos de encolamiento descritos anteriormente sobre situaciones de congestión en la red, configurándolos sobre el router y estableciendo prioridades.

### **3.1 TIPOS DE TRÁFICO EN LA RED**

De acuerdo a los requerimientos de cada usuario se puede determinar qué tráfico va a circular comúnmente en una red. Un administrador de red debe velar por el uso adecuado de los recursos de la red, y garantizar la entrega de cada paquete a su destino. Para definir políticas de prioridad en cada caso, es necesario conocer cada tipo de usuario y los requerimientos asociados a él.

#### **3.1.1 USUARIO LAN SIN ACCESO A INTERNET**

Normalmente son usuarios de tipo empresarial o institucional, cuyas actividades se basan generalmente en la comunicación y acceso a datos en una red privada LAN o WAN. Estas podrían ser:

- Comunicaciones de audio y video (videoconferencias, juegos en línea)
- Mensajería instantánea interna
- Operaciones de lectura y escritura sobre una base de datos
- Conexión remota
- Envío de correos electrónicos a un servidor SMTP interno
- Transferencias de archivos desde un servidor FTP interno



### 3.1.2 USUARIO LAN CON ACCESO A INTERNET

Generalmente son usuarios comunes cuyas necesidades pueden coincidir con las de un usuario empresarial o institucional, pero además requieren acceso a otros tipos de datos que solo circulan a través de internet. Este medio les proporciona una comunicación mucho más amplia, hacia redes WAN no privadas más allá de límites geográficos. Las actividades que realizan pueden ser de tipo:

- Comunicaciones de audio y video (videoconferencias, juegos en línea)
- Mensajería instantánea interna o a través de internet
- Operaciones de lectura y escritura sobre una base de datos
- Conexión remota
- Envío y recepción de correos electrónicos a través de internet
- Transferencias de archivos desde un servidor FTP interno o de internet
- Navegación a través de páginas web

## **3.2 DESCRIPCIÓN DE PROTOCOLOS Y SERVICIOS MÁS COMUNES EN EL TRÁFICO DE UNA RED**

Para que un usuario pueda comunicarse a través de la red es necesario que interactúe mediante la capa de Aplicación en el modelo OSI, la cual le provee una interfaz final más sencilla y de uso amigable. Es la capa que proporciona la interfaz entre las aplicaciones que utilizamos para comunicarnos y la red subyacente en la cual se transmiten los mensajes. Consiste en aplicaciones y servicios, los cuales manejan ciertas reglas o protocolos para enviar y recibir paquetes de datos a través de las demás capas, y por lo tanto, de la red. A continuación describiremos los protocolos y servicios más comunes o estándares en las comunicaciones de datos.

### **3.2.1 SMTP (SIMPLE MAIL TRANSFER PROTOCOL)**

El protocolo simple de transferencia de correo (SMTP) rige la transferencia de correos salientes desde el cliente emisor al servidor de e-mail SMTP, así también como el transporte de correos entre servidores de e-mail SMTP y POP. En otras palabras, permite transportar correos por las redes de datos entre diferentes tipos de software de cliente y servidor, y hace posible el intercambio de correos en internet.

El formato de mensajes del protocolo SMTP utiliza un conjunto rígido de comandos y respuestas. Estos comandos admiten los procedimientos utilizados en el SMTP, como inicio de sesión, transacción de correo, reenvío de correo, verificación de nombres de buzones, apertura y cierre de intercambios, y se comunica a través de TCP puerto 25.

### **3.2.2 POP (POST OFFICE PROTOCOL)**

El protocolo de oficina de correos (POP) es aquel utilizado por un servidor de e-mail para reenviar un correo a la bandeja de entrada del destinatario. En otras palabras, es el protocolo que interviene para la recepción de mensajes hacia la cuenta de correo de un usuario o cliente. Actualmente se utiliza la versión 3 de este protocolo, conocido como POP3.

Para establecer una conexión a un servidor POP, el cliente de correo abre una conexión TCP en el puerto 110 del servidor. Cuando la conexión se ha establecido, el servidor POP envía al cliente POP una invitación y después las dos máquinas se envían entre sí otras órdenes y respuestas que se especifican en el protocolo. Como parte de esta comunicación, al cliente POP se le pide que se autentifique (Estado de autenticación), donde el nombre de usuario y la contraseña del usuario se envían al servidor POP. Si la autenticación es correcta, el cliente POP pasa al Estado de transacción, en este estado se pueden utilizar órdenes LIST, RETR y DELE para mostrar, descargar y eliminar mensajes del servidor, respectivamente. Los mensajes definidos para su eliminación no se quitan realmente del servidor hasta que el cliente POP envía la orden QUIT para terminar la sesión.

### **3.2.3 FTP (FILE TRANSFER PROTOCOL)**

El Protocolo de transferencia de archivos (FTP) se desarrolló para permitir las transferencias de archivos entre un cliente y un servidor. Un cliente FTP es una aplicación que se ejecuta en una computadora y se utiliza para cargar y descargar archivos desde un servidor que ejecuta el daemon FTP (FTPd).

Para transferir los archivos en forma exitosa, el FTP requiere de dos conexiones entre cliente y servidor: una para comandos y respuestas, otra para la transferencia real de archivos.

El cliente establece la primera conexión con el servidor en TCP puerto 21. Esta conexión se utiliza para controlar el tráfico, que consiste en comandos del cliente y respuestas del servidor.

El cliente establece la segunda conexión con el servidor en TCP puerto 20. Esta conexión es para la transferencia real de archivos y se crea cada vez que se transfiere un archivo.

La transferencia de archivos puede producirse en ambas direcciones. El cliente puede descargar (bajar) un archivo desde el servidor o el cliente puede cargar (subir) un archivo en el servidor.<sup>5</sup>

### **3.2.4 HTTP (HIPERTEXT TRANSFER PROTOCOL)**

Es el protocolo utilizado por un servicio web para establecer la conexión con un explorador y transferir datos, como respuesta a la solicitud de navegación en una página web. Cuando un cliente, generalmente un explorador Web, envía un mensaje de solicitud a un servidor, el protocolo HTTP define los tipos de mensajes que el cliente utiliza para solicitar la página Web y envía los tipos de mensajes que el servidor utiliza para responder. Los tres tipos de mensajes más comunes son GET, POST y PUT.

GET es una solicitud de datos del cliente. Un explorador Web envía el mensaje GET para solicitar las páginas desde un servidor Web. Como se muestra en la figura, una vez que el servidor recibe la solicitud GET , responde con una línea de estado, como HTTP/1.1 200 OK, y un mensaje solo, cuyo cuerpo puede ser el archivo solicitado, un mensaje de error o alguna otra información.

POST y PUT se utilizan para enviar mensajes que cargan los datos al servidor Web. Por ejemplo, cuando el usuario ingresa datos en un formulario incorporado en una página Web, POST incluye los datos en el mensaje enviado al servidor. PUT carga los recursos o el contenido al servidor Web.

---

<sup>5</sup> CISCO CCNA EXPLORATION 4.0: *Aspectos Básico de Interworking*

Actualmente para las comunicaciones seguras de este tipo se utiliza una variación de este protocolo, conocida como HTTPS. De esta forma se puede utilizar encriptación u autenticación para asegurar los datos cuando viajan entre un cliente y un servidor.

### **3.2.5 SQL (STRUCTURED QUERY LANGUAGE)**

Es un lenguaje de consulta estructurado desarrollado para el acceso y comunicación con las bases de datos, mediante el cual se pueden realizar operaciones de lectura y escritura sobre las mismas. El servicio asociado con SQL depende de los diversos sistemas de gestión o aplicaciones de bases de datos que se han creado en la actualidad, y así también los puertos utilizados por cada uno. Los de uso más generalizado son MySQL Server, Oracle y PostgreSQL.

### **3.2.6 IM (INSTANT MESSAGING)**

Es un servicio que se basa en la transmisión instantánea de mensajes escritos de un cliente a otro a través de la red. Se diferencia del correo electrónico en que las conversaciones se realizan en tiempo real. Cada mensaje es enviado solo cuando el usuario da una orden de confirmación, lo cual da pie a que se puedan realizar correcciones y el cliente receptor no lea el mensaje hasta ese preciso momento. La mayoría de las aplicaciones que usan este servicio hoy en día admiten también transferencias de archivos de un cliente a otro, tráficos de voz y video como videoconferencias, conversaciones entre varios usuarios al tiempo y notificaciones de estados.

### 3.2.7 TELNET

Se define como un protocolo o servicio utilizado para conectarse en forma remota a sistemas informáticos distantes o cercanos.

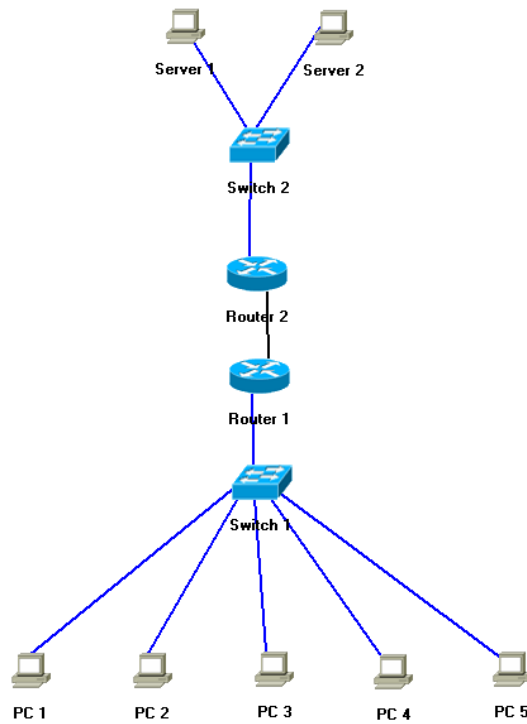
Mucho antes de que existieran las computadoras de escritorio con interfaces gráficas sofisticadas, las personas utilizaban sistemas basados en textos que eran simplemente terminales conectadas físicamente a una computadora central. Una vez que las redes estuvieran disponibles, las personas necesitaban acceder en forma remota a los sistemas informáticos de la misma manera en que lo hacían con las terminales conectadas en forma directa. Telnet se desarrolló para satisfacer esta necesidad. Proporciona un método estándar de emulación de dispositivos de terminal basados en texto en la red de datos. En lugar de utilizar un dispositivo físico para conectar al servidor, Telnet utiliza software para crear un dispositivo virtual que proporciona las mismas funciones que una sesión terminal con acceso a la Interfaz de línea de comandos (CLI) del servidor.<sup>6</sup> La conexión se realiza a través de TCP puerto 23.

Aunque el protocolo Telnet admite autenticación de usuario, no admite el transporte de datos encriptados. Todos los datos intercambiados durante una sesión Telnet se transportan como texto sin formato por la red.

---

<sup>6</sup> CISCO CCNA EXPLORATION 4.0: *Aspectos Básico de Interworking*

### 3.3 MONITOREO EN UNA TOPOLOGÍA DE RED WAN



Para el desarrollo de este capítulo definimos una topología cliente-servidor conformada por 2 redes LAN. En la primera consideramos que debían quedar todos los clientes necesarios para generar congestión, y así emular el comportamiento de un ambiente de trabajo empresarial o institucional. Y la segunda se compone de dos servidores que administrarían el acceso a sus datos. Para realizar el monitoreo en toda la WAN fue necesario configurar Netflow en la interfaz fastethernet del Router 1, como mecanismo para supervisar el procesamiento de todos los paquetes en él y verificar el funcionamiento de los algoritmos de encolado empleados.

### 3.3.1 PROTOCOLO NETFLOW

Se trata de un protocolo propietario de Cisco soportado en la actualidad por las líneas más recientes de switches y routers Cisco. Este protocolo permite a los dispositivos recolectar información referida a todo tipo de tráfico que atraviesa los enlaces y enviar la información referida a ese tráfico utilizando UDP a un dispositivo final o PC que recibe el nombre de NetFlow Collector.

La ventaja de NetFlow respecto de SNMP es que este protocolo brinda, fundamentalmente, información y funcionalidades de administración, y además le agrega la referida al tráfico que provoca el uso de cada dispositivo. Por lo tanto se pueden obtener estadísticas y gráficas sobre el tráfico generado filtrando por IP de origen, IP de destino, números de puertos de origen y destino, y protocolos.

Entre las aplicaciones posibles de NetFlow se pueden contar el monitoreo de la red, el monitoreo de aplicaciones específicas, el monitoreo de usuarios, el planeamiento de actualizaciones o modificaciones de la red, el análisis de seguridad, etc.

Para que el router pueda recolectar y enviar esta información hacia la IP del Netflow Collector, es necesario realizar una configuración sobre la interfaz Ethernet del router conectada al switch, como se detalla a continuación:

```
router#enable
router#configure terminal
router(config)#interface FastEthernet 0/1
router(config-if)#ip route-cache flow*
router(config-if)#exit
router(config)#ip flow-export destination <ip-address> <port>
router(config)#ip flow-export source FastEthernet 0/1
router(config)#ip flow-export version 5
router(config)#ip flow-cache timeout active 1
router(config)#ip flow-cache timeout inactive 15
router(config)#snmp-server ifindex persist
router#show ip flow export
router#show ip cache flow
```



<b>Comando</b>	<b>Descripción</b>
<i>interface {interface} {interface_number}</i> <i>ip route-cache flow</i>	Habilita Netflow en una interfaz particular del router.
<i>ip flow-export destination{hostname ip_address}</i> 9996	Especifica la dirección IP del PC configurado como Netflow Collector, hacia donde se exportan los flujos de datos Netflow. El puerto utilizado por defecto es 9996.
<i>ip flow-export source</i> <i>{interface} {interface_number}</i>	Especifica la dirección IP de origen, que corresponde a la interfaz configurada del router. La aplicación analizadora de Netflow hará peticiones SNMP al dispositivo en esta interfaz.
<i>ip flow-export version 5 [peer-as   origin-as]</i>	Define la versión de los comandos asociados a Netflow Export. La versión 5 es la más utilizada en la mayoría de los routers Cisco.
<i>ip flow-cache timeout active 1</i>	Define intervalos de tiempo de 1 minuto en el envío de flujos de datos del router hacia el PC. Se puede elegir cualquier número en el rango de 1 a 60 minutos. El valor por default es de 30 minutos.
<i>ip flow-cache timeout inactive 15</i>	Este comando garantiza que los flujos obtenidos sean exportados periódicamente. El valor por defecto son 15 segundos, pero se puede elegir cualquier número entre 10 y 600 segundos. Sin embargo, si se escoge un valor mayor a 250 segundos, los niveles de tráfico exportados podrían reflejarse afectados y verse muy bajos.

### Tabla comparativa entre Netflow y SNMP

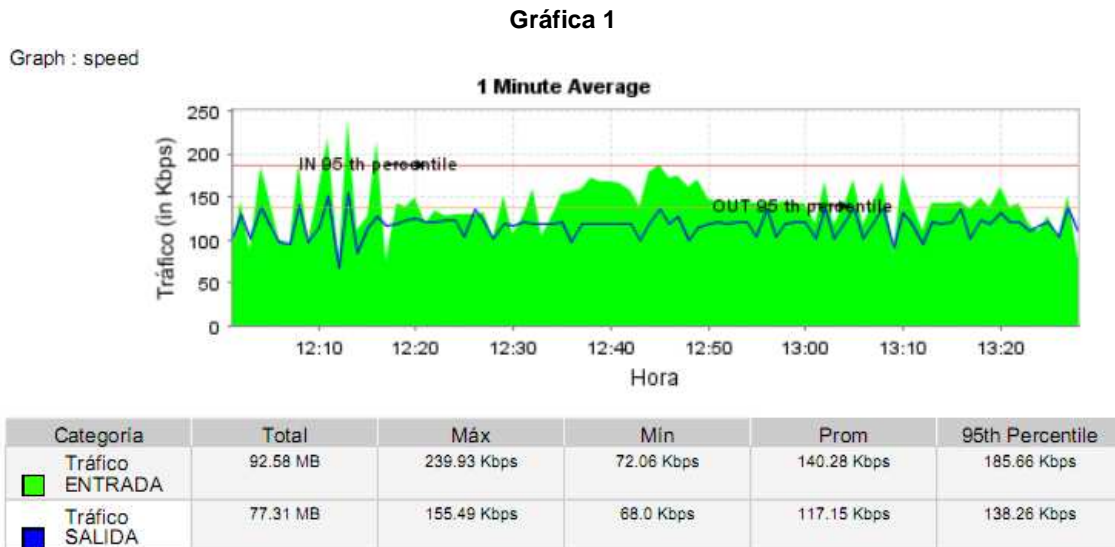
Función	SNMP	Netflow
Instalación	Fácil, a nivel de sistema operativo.	Puede ser un proceso complejo, ya que requiere configurar el switch, router o dispositivo de capa 3.
Filtro de tráfico	No	Si
Diferenciación de uso de ancho de banda por protocolos o direcciones IP	No	Si
Generación de listas de clasificación por parámetros como conexiones, direcciones IP, protocolos, etc.	No	Si
Filtro de uso de ancho de banda por direcciones IP	No	Si
Filtro de uso de ancho de banda por puerto	Si	Si
Monitoreo de parámetros de red diferentes al uso de ancho de banda	Si	No

Debido a todas estas ventajas que presenta el uso del protocolo netflow como método de monitoreo en la red, fue el que utilizamos en el presente trabajo para analizar la actividad y flujos de información en el router. Dado que necesitamos estudiar el comportamiento de cada algoritmo de encolado aplicado para cada tipo de tráfico o protocolo, fue el método que mejor se adaptó y nos facilitó el análisis de las estadísticas generadas.

### 3.3.2 ANÁLISIS DE RESULTADOS EN LA WAN SIN CONEXIÓN A INTERNET

Durante las pruebas de laboratorio como primera instancia se sometió a la WAN a un tráfico de audio y video constante, generado por una aplicación de juego en línea (Unreal Tournament) para crear un ambiente de alta congestión. A su vez se hicieron consultas y actualizaciones a una base de datos, transferencias de archivos desde un servidor FTP, envío de correos hacia un servidor SMTP y se utilizó una aplicación de mensajería instantánea LAN, lo cual se describe detalladamente en el informe de prácticas. Con base en las estadísticas generadas se analizó el comportamiento de la red frente a situaciones de alta congestión, y el desempeño de cada uno de los algoritmos de encolamiento priorizando diferentes tipos de tráfico.

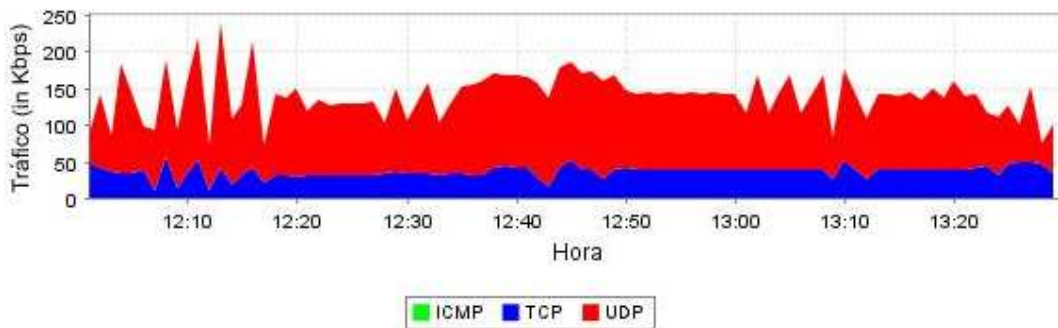
#### 3.3.2.1 TRAFICO SIN PRIORIZACIÓN



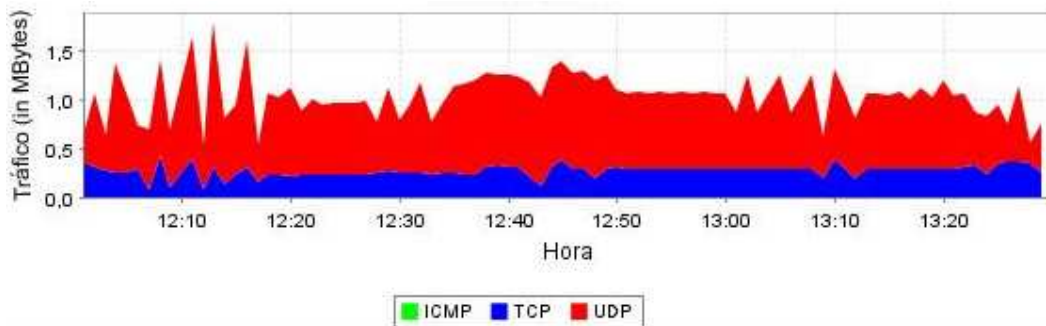
Durante esta prueba se sometió la WAN a un tráfico constante descrito anteriormente sin configurar ningún algoritmo de encolado sobre los enrutadores.

La gráfica general muestra un mayor volumen de tráfico de entrada con respecto al de salida, con velocidades promedio de 140.28 Kbps y 117.15 Kbps. Esto evidencia la carencia de políticas de priorización definidas dentro de la LAN para protocolos TCP, lo cual conlleva a que la transmisión de tráfico UDP generado por los clientes hacia los servidores sea notable y se vea reflejada tanto en el tráfico de entrada como en el de salida. Si se realiza una diferenciación en la gráfica por grupos de protocolos, se tendría lo siguiente:

**Gráfica 2**



**Gráfica 3**



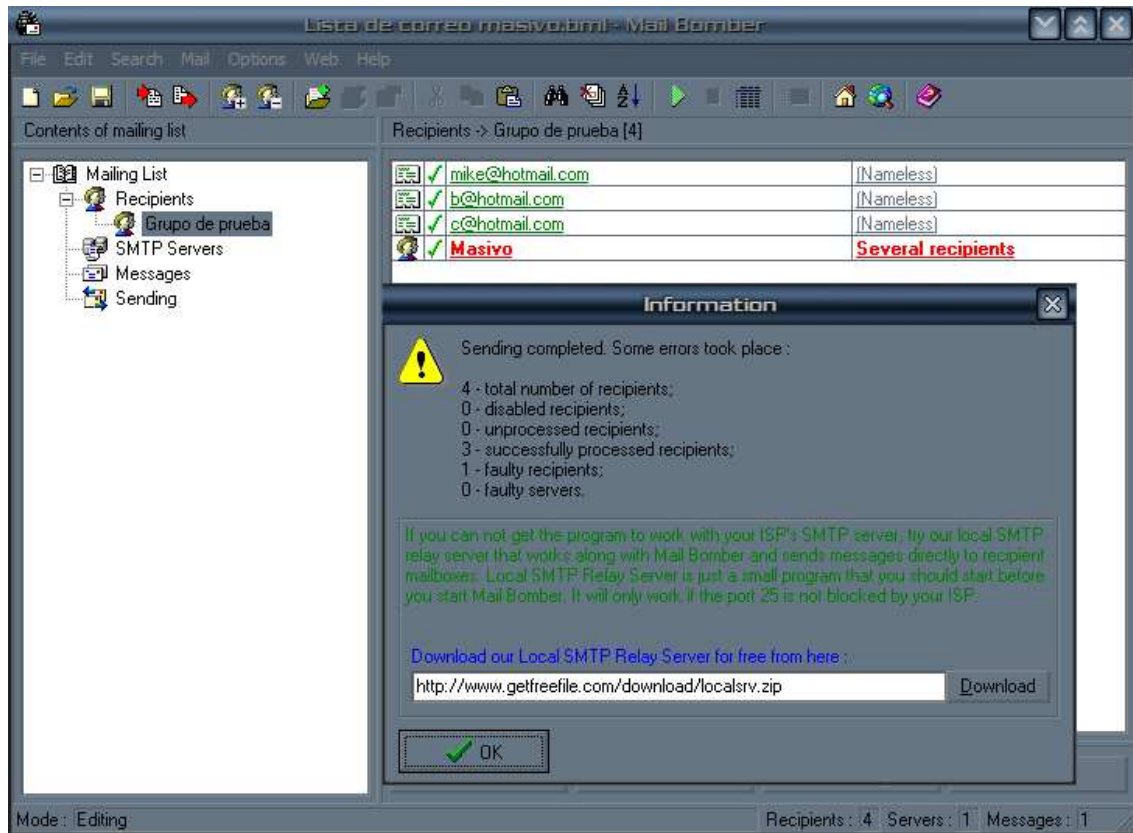
**Gráfica 4**

Aplicación	Tráfico (92.46 MB)	Porcentaje de tráfico total
UDP_App	59.24 MB	64%
mysql	21.82 MB	24%
ms-sql-s	7.04 MB	8%
TCP_App	940.52 KB	1%
smtp	553.53 KB	1%
bootps	514.6 KB	1%
netbios-ns	64.88 KB	<1%
icmp	29.33 KB	<1%
snmp	28.53 KB	<1%
netbios-dgm	24.91 KB	<1%
telnet	14.27 KB	<1%
ftp	2.39 KB	<1%
http	1.64 KB	<1%
skinny	204,00 Bytes	<1%
pptp	120,00 Bytes	<1%
sql-server	120,00 Bytes	<1%
Unaccounted	2.17 MB	2%

Como se puede observar en los reportes obtenidos, el porcentaje de tráfico UDP corresponde al generado por la aplicación de audio y video, y difiere mucho en cuanto a velocidad y volumen al tráfico generado por el grupo de protocolos TCP.

Las tasas de valores que muestran las gráficas 2 y 3 correspondientes al tráfico UDP oscilan entre los 160 kbps y 1.2 MBytes generados, respectivamente. Mientras que el grupo de protocolos TCP muestran un menor consumo del ancho de banda, con una velocidad promedio de 40 kbps y un volumen aproximado de 0.35 MBytes.

El tratamiento que le da el router a los paquetes en este caso es muy similar al del algoritmo FIFO (First In First Out), puesto que les da servicio conforme van entrando independientemente de su naturaleza o protocolo. Este punto se evidencia en la gráfica 3 al analizar los picos y los valles coincidentes para cada tipo de tráfico, los cuales se generan en los mismos intervalos de tiempo y corresponden a momentos de mayor actividad en la red. Al no definirse políticas de prioridad, algunos paquetes sensibles a retardo como correos masivos enviados al servidor SMTP fueron descartados y nunca entregados a su destino, como se puede observar en el siguiente mensaje de error:



### 3.3.2.2 SQL IMPLEMENTANDO PQ

Para esta práctica de laboratorio se configuró el algoritmo Priority Queueing (PQ) sobre la interfaz fastethernet del **Router 1**, priorizando el tráfico TCP por puerto 3306 que corresponde al utilizado por mysql server.

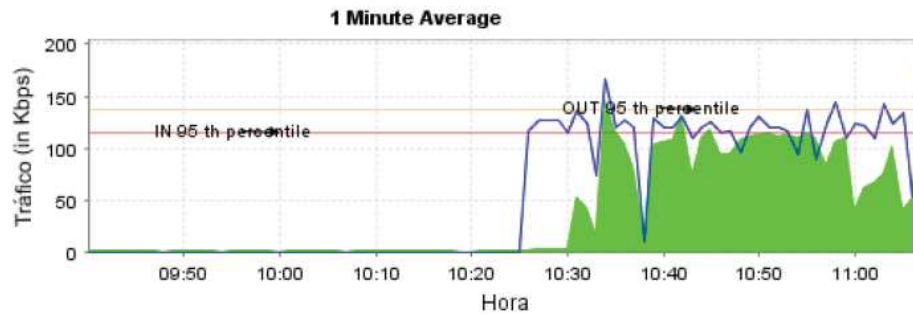
```

router#enable
router#configure terminal
router(config)#priority-list 1 protocol ip high tcp 3306
router(config)#interface fastethernet 0/1
router(config-if)#priority-group 1

```

Gráfica 5

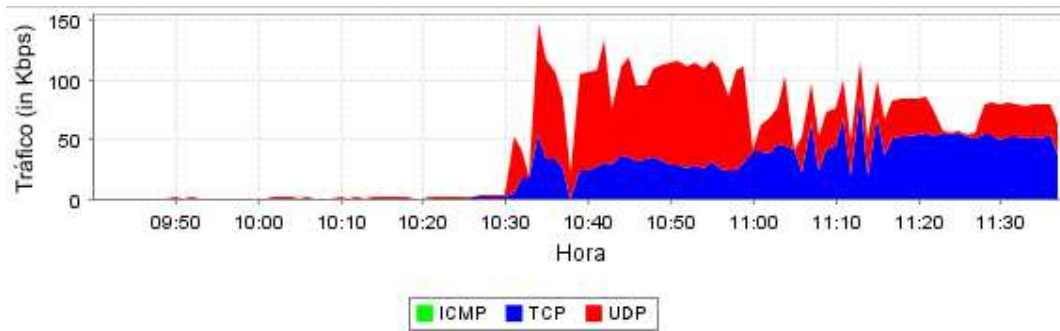
Graph : speed



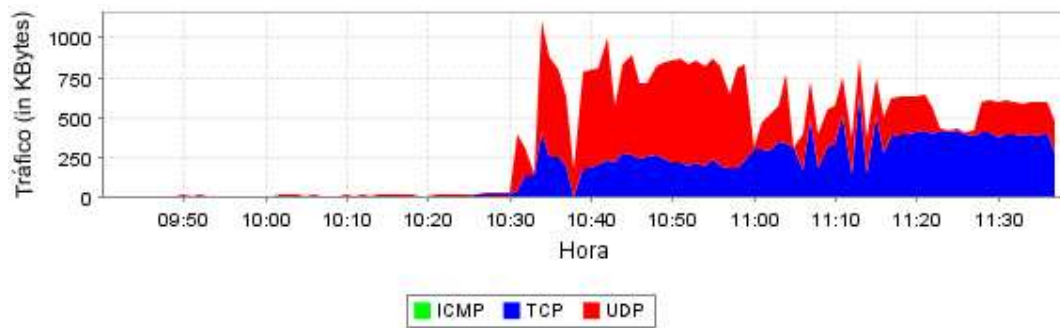
Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	25.85 MB	146.71 Kbps	58,13 bps	39.16 Kbps	114.82 Kbps
Tráfico SALIDA	37.32 MB	194.48 Kbps	0.00	56.55 Kbps	136.39 Kbps

La gráfica general muestra un alto porcentaje de tráfico de salida a lo largo de todo el tiempo de monitoreo con un promedio de **56.55 Kbps**, cifra que respecto al tráfico de entrada que es de **39.16 Kbps** evidencia el comportamiento de PQ, el cual le otorga mayor prioridad a la operación de salida de datos desde los servidores que están fuera de la LAN manteniendo las conexiones mysql con la base de datos sobre la que se realizan inserciones de registros. Esto se debe a que el proceso de encolamiento demanda más procesamiento en el router, y al ser mayor el flujo de tráfico de entrada se presenta más congestión que en el caso de salida de datos hacia los clientes. Por lo tanto, se presentan retardos en general en el número de paquetes enviados, pero no se descartan o pierden los de mysql. Además, se visualiza en la gráfica que en los momentos en que la red llega a puntos donde no hay congestión, siempre el tráfico de salida es mucho mayor al tráfico de entrada, lo que permite afirmar que aún para protocolos diferentes al protocolo priorizado (TCP), la tasa de salida de estos también se incrementa de forma notoria. Por esta razón se puede considerar a PQ como un algoritmo que tiene un buen manejo para la salida de datos del router cuando se implementa con protocolos TCP.

Gráfica 6



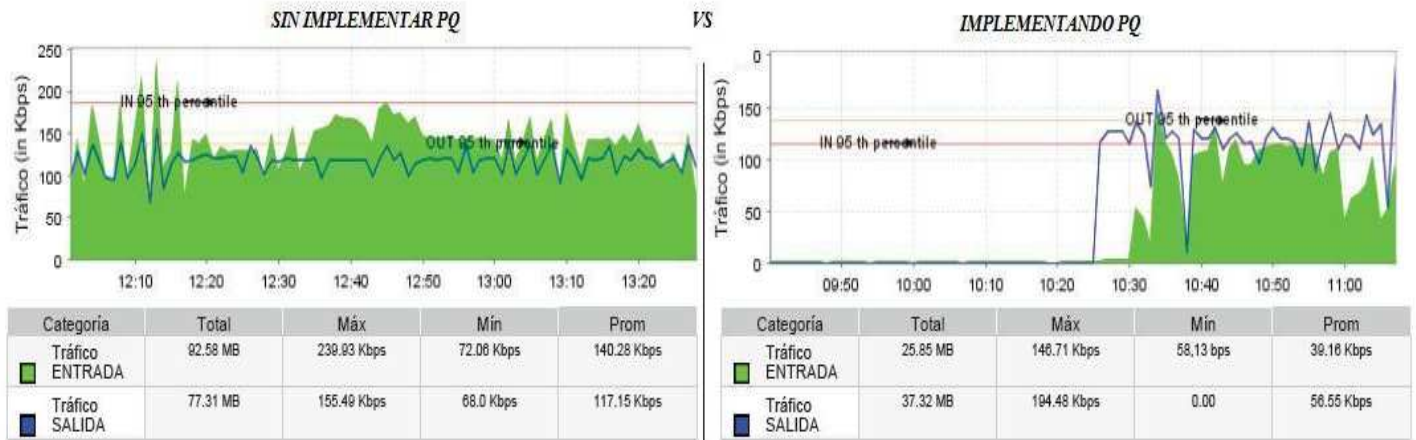
Gráfica 7



En las gráficas 6 y 7 podemos observar que en el intervalo de 10:30 a 11:00 hubo un incremento notorio en el tráfico UDP en comparación al generado por el grupo de protocolos TCP. Esto se debe a que intencionalmente se inició primero el intercambio de audio y video entre los clientes y el servidor de la aplicación, mientras que las operaciones SQL sobre la base de datos fueron llevándose a cabo por los distintos usuarios o clientes paulatinamente. De esta forma se demuestra como de 11:00 a 11:30 el tráfico TCP experimenta un alza y demanda un mayor porcentaje de la capacidad del canal de comunicación, y se estabiliza hasta el final del monitoreo. Mientras que el consumo de la red del tráfico de audio y video generado por el protocolo UDP disminuye un poco. A partir de estos resultados se puede afirmar que PQ se adapta al tráfico generado por protocolos utilizados por el servicio SQL, y lo prioriza eficientemente.



## No PQ versus PQ implementado sobre SQL



Realizando una comparación general entre el comportamiento de la red cuando se ha implementado PQ para puertos usados por MYSQL y en donde no se implementa encolado de prioridad podemos resaltar los siguientes puntos:

- Las tasas de velocidad de entrada y salida de datos con PQ son relativamente bajas con respecto a las obtenidas sin implementar PQ.
- El volumen de tráfico de entrada sin implementar PQ es mucho mayor al volumen de tráfico de entrada implementándolo.
- El volumen de tráfico de salida sin implementar PQ es mucho mayor al volumen de tráfico de salida implementándolo.
- La velocidad del tráfico de entrada sin implementar PQ es mucho mayor al tráfico de salida, caso contrario ocurre implementando PQ.
- En situaciones de implementación o no el tráfico de salida depende del tráfico de entrada, es decir, si disminuye el uno, el otro también lo hace, y viceversa.

## Porcentaje de tráfico de entrada no PQ versus PQ implementado sobre SQL

*SIN IMPLEMENTAR PQ*

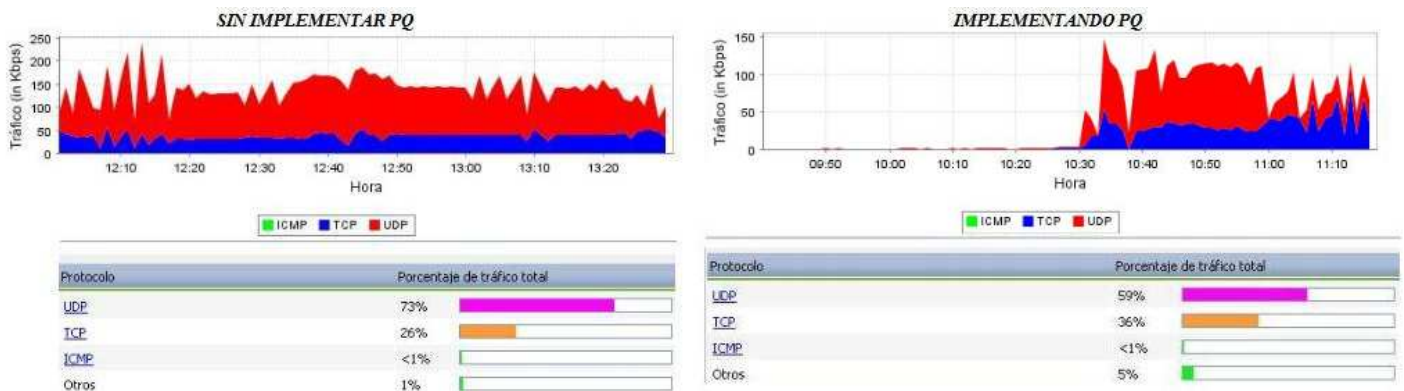
Aplicación	Tráfico (92.46 MB)	Porcentaje de tráfico total
UDP_App	59.24 MB	64%
mysql	21.82 MB	24%
ms-sql-s	7.04 MB	8%
TCP_App	940.52 KB	1%
smtp	553.53 KB	1%
bootps	514.6 KB	1%
netbios-ns	64.88 KB	<1%
iomp	29.33 KB	<1%
snmp	28.53 KB	<1%
netbios-dgm	24.91 KB	<1%
telnet	14.27 KB	<1%
ftp	2.39 KB	<1%
http	1.64 KB	<1%
skinny	204.00 Bytes	<1%
ppptp	120.00 Bytes	<1%
sql-server	120.00 Bytes	<1%
Unaccounted	2.17 MB	2%

*IMPLEMENTANDO PQ*

Aplicación	Tráfico (25.66 MB)	Porcentaje de tráfico total
UDP_App	16.12 MB	63%
mysql	7.32 MB	29%
bootps	508.74 KB	2%
TCP_App	227.91 KB	1%
smtp	88.01 KB	<1%
netbios-ns	37.82 KB	<1%
netbios-dgm	36.33 KB	<1%
http	29.03 KB	<1%
iomp	23.67 KB	<1%
snmp	18.16 KB	<1%
telnet	8.81 KB	<1%
ftp	1.66 KB	<1%
skinny	204.00 Bytes	<1%
miaactivation	144.00 Bytes	<1%
mregistry	144.00 Bytes	<1%
citrix_metaframe	120.00 Bytes	<1%
wins	120.00 Bytes	<1%
Oracle	120.00 Bytes	<1%
ingres	120.00 Bytes	<1%
orbix-locator	120.00 Bytes	<1%
orbix-config	120.00 Bytes	<1%
HTTP-Proxy	120.00 Bytes	<1%
Unaccounted	1.23 MB	5%

A simple vista se puede observar la diferencia de porcentajes utilizados en cada gráfica. Independientemente del tráfico generado, el porcentaje de tráfico total utilizado por mysql es mucho mayor implementando PQ, el cual representa un 29% del tráfico total con respecto al 24% del tráfico generado sin implementarse el encolado de prioridad.

## No PQ vs PQ implementado sobre mysql: diferenciación por protocolos



Al comparar las dos gráficas se pueden notar los siguientes comportamientos:

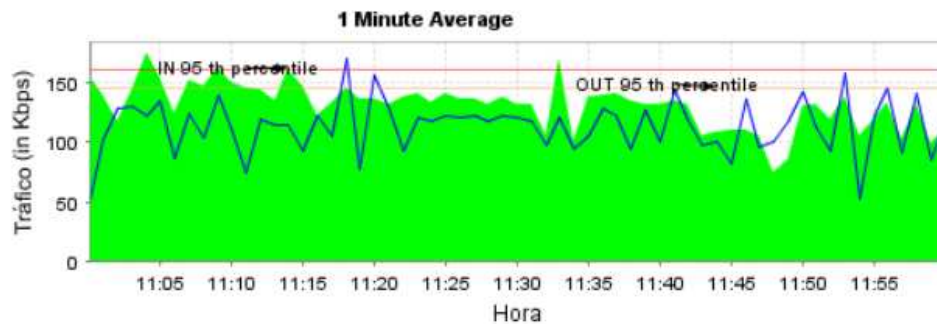
- El porcentaje de tráfico total usado donde se ha implementado PQ (TCP 36%) es mucho mayor al caso donde no se ha implementado el encolado de prioridad (TCP 26%).
- Con PQ se observan cifras que no alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo.
- Al no implementarse PQ, el router tiene menos carga para procesar los paquetes, lo cual eleva el volumen de tráfico que puede circular por la red. Se habla de un tráfico total de 94.53 MB que circuló sin implementación de PQ contra un 33.66 MB implementándose, lo que deja en evidencia la gran demanda de procesamiento por parte del router al utilizarse encolamiento de prioridad.

### 3.3.2.3 FTP IMPLEMENTANDO PQ

Para este laboratorio se configuró el algoritmo Priority Queueing (PQ) sobre la interfaz fastethernet del **Router 1**, priorizando el tráfico TCP por puerto 21 que corresponde al utilizado por el servidor FTP.

```
router#enable
router#configure terminal
router(config)#priority-list 1 protocol ip high tcp 21
router(config)#interface fastethernet 0/1
router(config-if)#priority-group 1
```

Gráfica 8



Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	59.76 MB	174.93 Kbps	74.29 Kbps	130.62 Kbps	160.68 Kbps
Tráfico SALIDA	51.82 MB	169.09 Kbps	51.98 Kbps	113.28 Kbps	145.81 Kbps

Si se observa con detenimiento la gráfica general, muestra que en este monitoreo se obtuvieron mayores promedios y valores de velocidad y volumen para el tráfico de entrada. Esto se debe a que al priorizar las comunicaciones por el puerto 21, solo se tratan aquellas peticiones de autenticación y modificación de archivos hacia el servidor FTP, más no las transferencias reales de archivos. Al encontrarse dicho servidor detrás de un router solo acepta conexiones en modo **pasivo**, puesto que no existe una conexión directa punto a punto o a través de un switch como la que si se podría presentar dentro de una misma LAN. Para las conexiones remotas de este

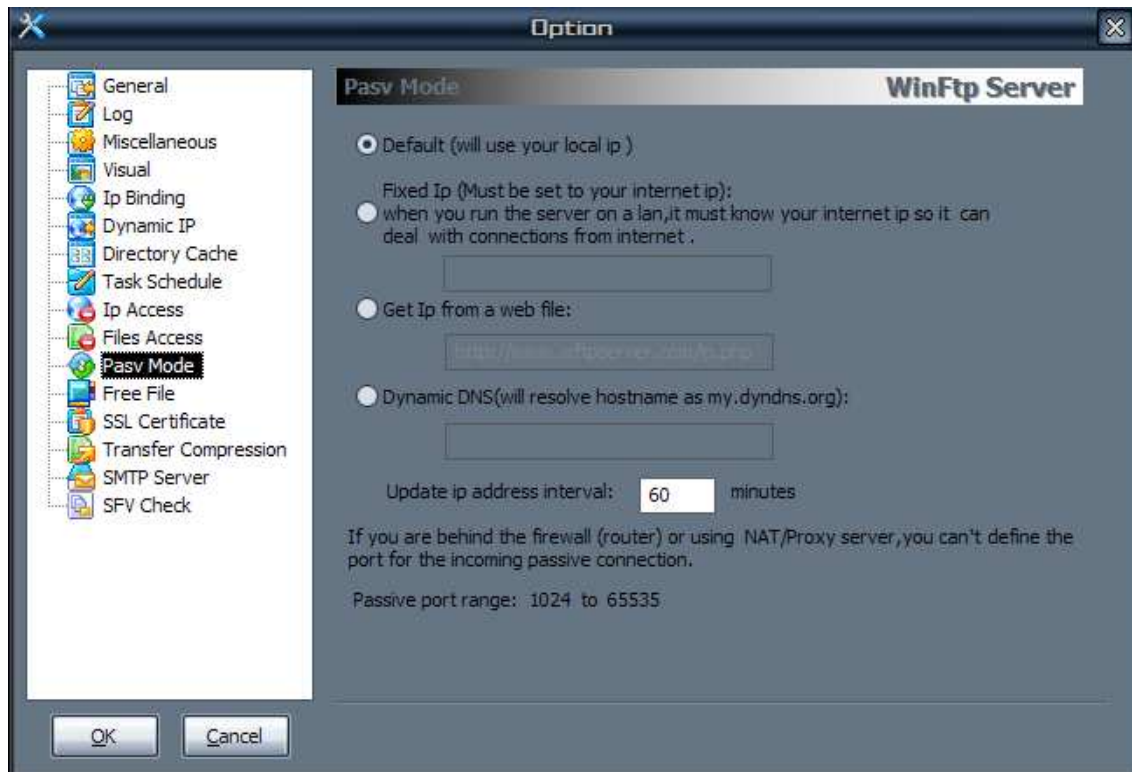
tipo a través de un router, el servidor entra en modo pasivo y realiza las transferencias de archivos mediante puertos variables en un rango de 1024 a 65535.

La mayoría de los routers utilizan este método de seguridad debido a que el puerto 20 puede ser objeto de ataques, dada la naturaleza del protocolo FTP. Aun si se encuentran abiertos todos los puertos en el dispositivo se utiliza el modo pasivo.

Por tal razón las peticiones que se realizan por el puerto 21 pueden ser muy pocas y de menor volumen, no se genera tanto trafico por esta vía y el router no demanda tanto procesamiento para atenderlas. El comportamiento de esta gráfica resulta siendo similar al obtenido cuando no se implementó ningún mecanismo de priorización.

La siguiente grafica es una captura de la configuración de la aplicación WinFTP Server, utilizada para administrar el servidor FTP y en donde se describe el uso del modo pasivo:

Gráfica 9



Gráfica 10

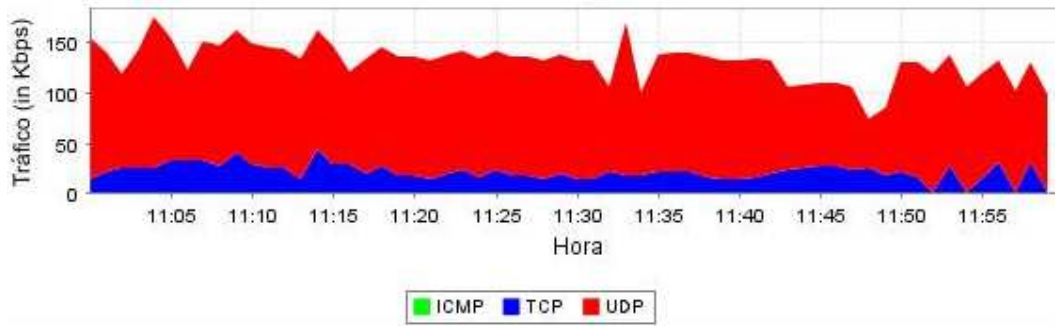
TRAFICO DE ENTRADA

Aplicación	Tráfico (59.76 MB)	Porcentaje de tráfico total
UDP_App	49.12 MB	82%
mysql	7.6 MB	13%
smtp	1.09 MB	2%
TCP_App	646.28 KB	1%
bootps	382.4 KB	1%
netbios-ns	26.59 KB	<1%
icmp	21.68 KB	<1%
snmp	20.7 KB	<1%
netbios-dgm	11.24 KB	<1%
telnet	10.32 KB	<1%
ftp	4.28 KB	<1%
http	896.00 Bytes	<1%
lotusnote	120.00 Bytes	<1%
direct_connect	120.00 Bytes	<1%
ms-sql-s	120.00 Bytes	<1%
wins	120.00 Bytes	<1%
Oracle	120.00 Bytes	<1%
ingres	120.00 Bytes	<1%
Unaccounted	835.79 KB	1%

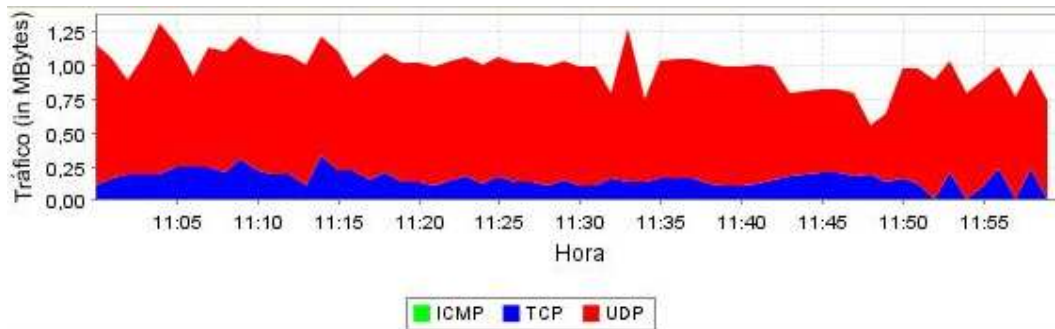
TRAFICO DE SALIDA

Aplicación	Tráfico (60.03 MB)	Porcentaje de tráfico total
TCP_App	30.07 MB	50%
UDP_App	27.74 MB	46%
smtp	1.16 MB	2%
mysql	1.03 MB	2%
ftp	5.07 KB	<1%
icmp	1.22 KB	<1%
lotusnote	144.00 Bytes	<1%
ms-sql-s	144.00 Bytes	<1%
direct_connect	144.00 Bytes	<1%
ingres	144.00 Bytes	<1%
Oracle	144.00 Bytes	<1%
wins	144.00 Bytes	<1%
msmq	144.00 Bytes	<1%

Gráfica 11



Gráfica 12



Mediante la grafica 10 se evidencia el poco volumen de tráfico generado a partir de peticiones hacia el servidor por el puerto 21. Para las operaciones de entrada y salida se obtuvo un tráfico total de 4.28 Kb y 5.07 Kb por este protocolo, lo cual no representa prácticamente ninguna carga para el router.

También podemos observar que para el tráfico de salida, **TCP\_App** generó un porcentaje de tráfico total de 50% respecto al resto de protocolos. Esto representa una agrupación de protocolos TCP realizada por el programa al no asociarse a ninguna aplicación conocida. En nuestro caso corresponde a todo aquel tráfico realizado entre servidores y clientes por puertos aleatorios en el rango de 1024 a 65535, lo cual sucede cuando el servidor FTP entra en modo pasivo para realizar la transferencia real de archivos. Si bien es cierto su volumen fue mucho mayor que el del resto de protocolos, no se podría decir que tenga algo que ver con la priorización que se realizó puesto que solamente se tuvo en cuenta el puerto 21.

La distribución de protocolos para el tráfico de entrada obtenida en las gráficas 10 y 11 demuestra una vez más la gran diferencia entre el tráfico de tipo audio y video y el resto de actividades, además de alcanzar valores máximos de 150 Kbps y 1.25 Mb que solo se obtuvieron cuando no se configuró ningún algoritmo de encolamiento. Al no realizar peticiones de autenticación y de archivos al servidor tan seguido, el router no necesita realizar muchas diferenciaciones de tráfico y el desempeño es similar.



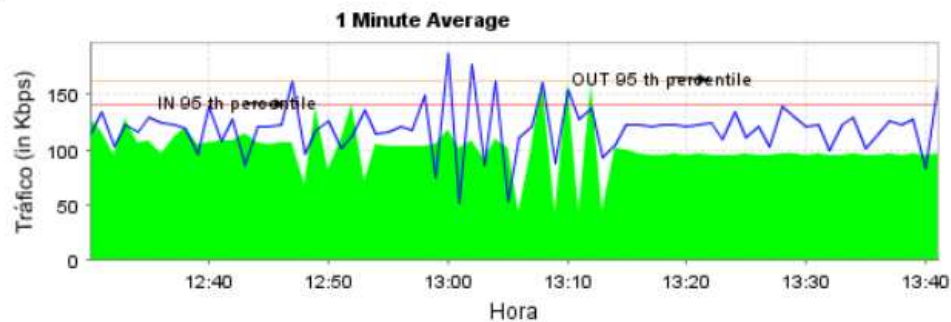
### 3.3.2.4 SMTP IMPLEMENTANDO PQ

Para la configuración de PQ priorizando el protocolo SMTP se definió una prioridad alta para el tráfico de TCP por el puerto 25, que es el estándar en este caso.

```
router#enable
router#configure terminal
router(config)#priority-list 1 protocol ip high tcp 25
router(config)#interface fastethernet 0/1
router(config-if)#priority-group 1
```

Gráfica 13

Graph : speed



Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	54.33 MB	160.14 Kbps	42.56 Kbps	100.62 Kbps	140.43 Kbps
Tráfico SALIDA	64.63 MB	187.85 Kbps	51.71 Kbps	119.68 Kbps	162.98 Kbps

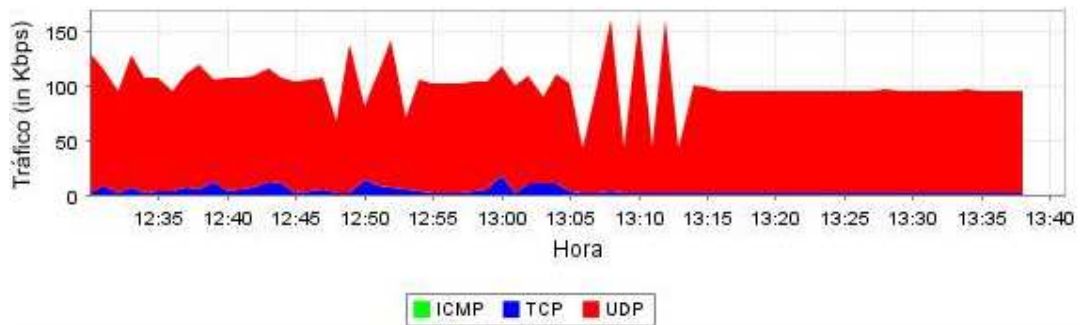
En la gráfica 13 podemos observar un mayor porcentaje de tráfico de salida que de entrada, con tasas de 119.68 Kbps y 100.62 Kbps respectivamente, tal como sucedió en el caso de SQL. A pesar de que el tráfico generado por el protocolo SMTP, como el envío esporádico de correos masivos no es constante, al configurar PQ sobre la interfaz fastethernet del Router 1 se incrementa la tasa de salida para protocolos diferentes al que fue priorizado.



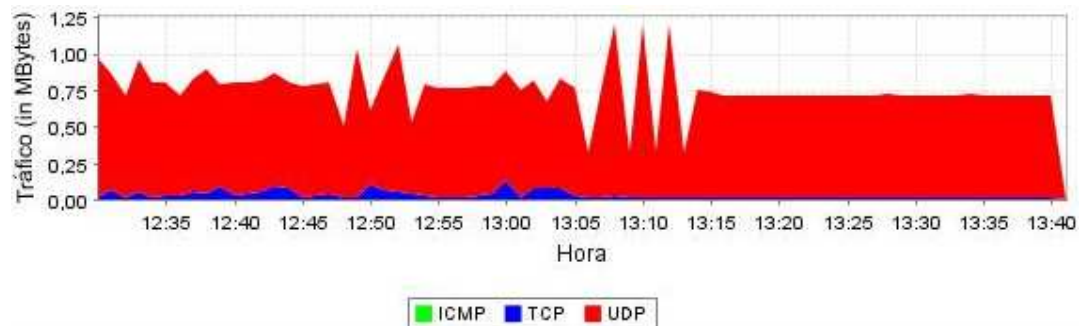
Gráfica 14

Aplicación	Tráfico (55.09 MB)	Porcentaje de tráfico total
UDP_App	33.17 MB	60%
nessus	17.57 MB	32%
smtp	1.05 MB	2%
TCP_App	677.39 KB	1%
mysql	590.94 KB	1%
bootps	433.06 KB	1%
netbios-ns	35.47 KB	<1%
icmp	24.99 KB	<1%
snmp	24.49 KB	<1%
netbios-dgm	13.51 KB	<1%
telnet	13.02 KB	<1%
ftp	8.89 KB	<1%
http	608,00 Bytes	<1%
skinny	120,00 Bytes	<1%
knetd	120,00 Bytes	<1%
soulseek	120,00 Bytes	<1%
Unaccounted	1.46 MB	3%

Gráfica 15



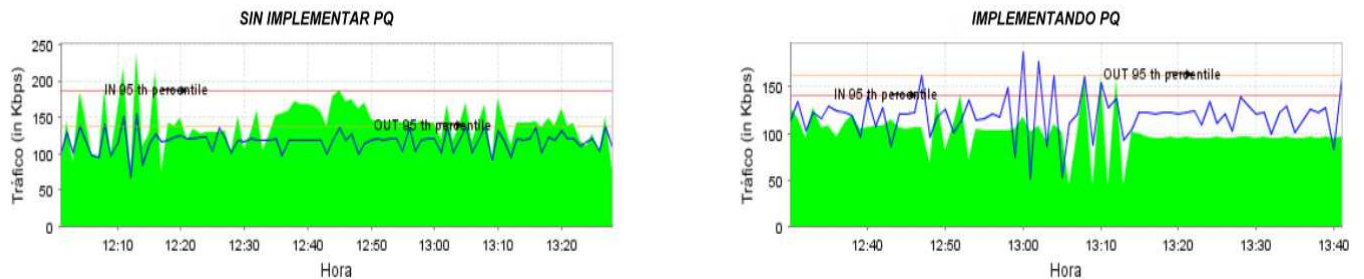
Gráfica 16



La gráfica 14 muestra el porcentaje de tráfico de entrada del protocolo SMTP, que fue de 2% con un valor de 1.05 MB. Esto es razonable considerando que en un

ambiente empresarial o institucional el envío de correos electrónicos solo se efectúa cuando es necesario y no permanentemente. Sin embargo, se observa un incremento del mismo en comparación con otras pruebas de laboratorio donde se perdían constantemente paquetes y el tráfico generado no pasaba de los 600 KB. En las gráficas 15 y 16 se puede evidenciar la forma como se impuso notoriamente el tráfico generado mediante el protocolo UDP (audio y video) sobre el tráfico del grupo de protocolos TCP, debido a que su volumen es mayor y causa mucha más congestión en la red. Esto permite afirmar una vez más la eficiencia de PQ al priorizar un tipo de datos sensible a retardos como ha demostrado ser el de SMTP.

### No PQ versus PQ implementado sobre SMTP



Categoría	Total	Máx	Min	Prom	95th Percentile	Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	92.58 MB	239.93 Kbps	72.06 Kbps	140.28 Kbps	185.66 Kbps	Tráfico ENTRADA	54.33 MB	160.14 Kbps	42.56 Kbps	100.62 Kbps	140.43 Kbps
Tráfico SALIDA	77.31 MB	155.49 Kbps	68.0 Kbps	117.15 Kbps	138.26 Kbps	Tráfico SALIDA	64.63 MB	187.85 Kbps	51.71 Kbps	119.68 Kbps	162.98 Kbps

Realizando una comparación general entre el comportamiento de la red cuando se ha implementado PQ para el protocolo SMTP y en donde no se implementa encolado de prioridad podemos resaltar los siguientes puntos:

- Las tasas de velocidad de entrada y salida de datos con PQ siguen relativamente bajas con respecto a las obtenidas sin implementar PQ.
- El volumen de tráfico de entrada sin implementar PQ se mantiene mucho más alto en comparación al volumen de tráfico de entrada implementándolo.

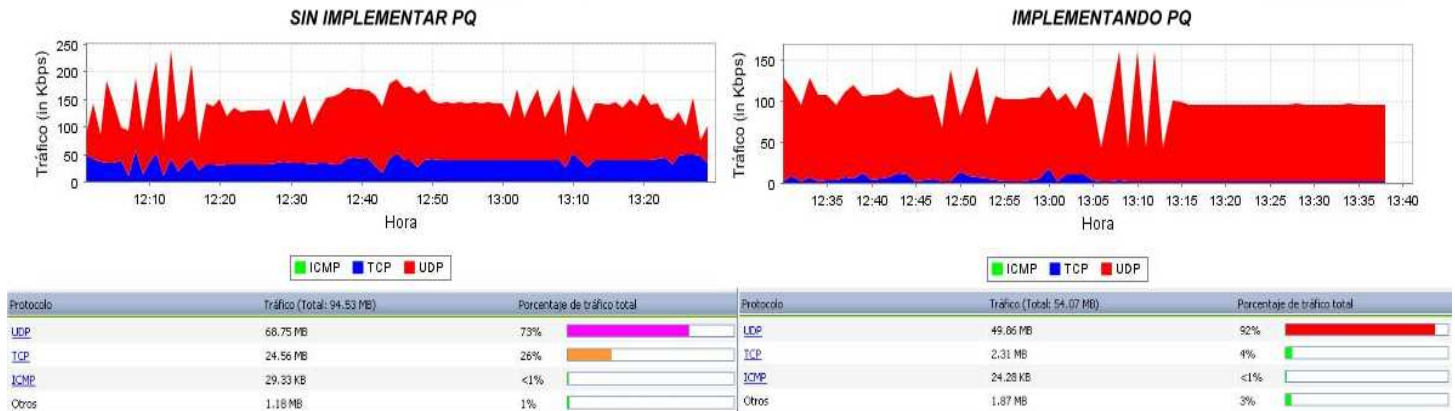
- El volumen de tráfico de salida sin implementar PQ es mucho mayor al volumen de tráfico de salida implementándolo.
- La velocidad del tráfico de entrada sin implementar PQ es mucho mayor al tráfico de salida, caso contrario ocurre implementando PQ.
- En ambas pruebas el tráfico de salida depende del tráfico de entrada, es decir, si disminuye el uno, el otro también lo hace, y viceversa.

### Porcentaje de tráfico de entrada no PQ versus PQ implementado sobre SMTP

SIN IMPLEMENTAR PQ			IMPLEMENTANDO PQ		
Aplicación	Tráfico (92.46 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (55.09 MB)	Porcentaje de tráfico total
UDP_App	59.24 MB	64%	UDP_App	33.17 MB	60%
mysql	21.82 MB	24%	nessus	17.57 MB	32%
ms-sql-s	7.04 MB	8%	smtp	1.05 MB	2%
TCP_App	940.52 KB	1%	TCP_App	677.39 KB	1%
smtp	553.53 KB	1%	mysql	590.94 KB	1%
bootps	514.6 KB	1%	bootps	433.06 KB	1%
netbios-ns	64.88 KB	<1%	netbios-ns	35.47 KB	<1%
icmp	29.33 KB	<1%	icmp	24.99 KB	<1%
snmp	28.53 KB	<1%	snmp	24.49 KB	<1%
netbios-dgm	24.91 KB	<1%	netbios-dgm	13.51 KB	<1%
telnet	14.27 KB	<1%	telnet	13.02 KB	<1%
ftp	2.39 KB	<1%	ftp	8.89 KB	<1%
http	1.64 KB	<1%	http	608.00 Bytes	<1%
skinny	204.00 Bytes	<1%	skinny	120.00 Bytes	<1%
pptp	120.00 Bytes	<1%	knetd	120.00 Bytes	<1%
sql-server	120.00 Bytes	<1%	souseek	120.00 Bytes	<1%
Unaccounted	2.17 MB	2%	Unaccounted	1.46 MB	3%

Como se puede observar, el porcentaje de tráfico generado hacia el servidor SMTP es mucho mayor implementando PQ, tal como se mencionó anteriormente. En este caso fue de 2%, con un volumen total de 1.05 MB. Mientras que en las pruebas de laboratorio donde no se utilizó PQ se obtuvo un porcentaje de 1% con un volumen total de 553.53 KB.

## No PQ vs PQ implementado sobre SMTP: diferenciación por protocolos



Al comparar las dos gráficas se observan los siguientes comportamientos:

- El porcentaje de tráfico total usado donde se ha implementado PQ (TCP 4%) es mucho menor al caso donde no se ha implementado el encolado de prioridad (TCP 26%).
- Al igual que en las pruebas priorizando SQL, con PQ se observan cifras que no alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo.
- Al no implementarse PQ, el router tiene menos carga para procesar los paquetes, lo cual eleva el volumen de tráfico que puede circular por la red. Se habla de un tráfico total de 94.53 MB que circuló sin implementación de PQ contra un 53.25 MB implementándose, lo que deja en evidencia una vez más la gran demanda de procesamiento por parte del router al utilizarse encolamiento de prioridad.

### 3.3.2.5 IM IMPLEMENTANDO PQ

Para esta prueba de laboratorio se define prioridad alta sobre el protocolo UDP puerto 7550, utilizado por la herramienta Borgchat de mensajería instantánea LAN y WAN.

```
router#enable
router#configure terminal
router(config)#priority-list 1 protocol ip high udp 7550
router(config)#interface fastethernet 0/1
router(config-if)#priority-group 1
```

Gráfica 17

Origen	Destino	Protocolo	Puerto de origen	Puerto de dest	Tráfico	Añadir
192.168.100.6	192.168.101.2	UDP	1386	7777	4.05 MB	<a href="#">Add Mapping</a>
192.168.100.4	192.168.101.2	UDP	1095	7777	4.03 MB	<a href="#">Add Mapping</a>
192.168.100.5	192.168.101.2	UDP	2414	7777	3.84 MB	<a href="#">Add Mapping</a>
192.168.100.12	192.168.101.2	UDP	61455	7777	2.9 MB	<a href="#">Add Mapping</a>
192.168.100.3	192.168.101.2	UDP	1903	7777	1.87 MB	<a href="#">Add Mapping</a>
192.168.100.3	192.168.101.4	UDP	7550	7550	118.16 KB	<a href="#">Add Mapping</a>
192.168.100.3	192.168.101.0	UDP	7550	7550	117.82 KB	<a href="#">Add Mapping</a>
192.168.100.6	192.168.101.4	UDP	7550	7550	94.14 KB	<a href="#">Add Mapping</a>
192.168.100.6	255.255.255.255	UDP	7550	7550	85.14 KB	<a href="#">Add Mapping</a>
192.168.100.3	255.255.255.255	UDP	7550	7550	78.82 KB	<a href="#">Add Mapping</a>
192.168.100.5	255.255.255.255	UDP	7550	7550	32.0 KB	<a href="#">Add Mapping</a>
192.168.100.3	192.168.101.4	UDP	0	0	16.02 KB	<a href="#">Add Mapping</a>
192.168.100.3	192.168.101.0	UDP	0	0	16.02 KB	<a href="#">Add Mapping</a>
192.168.100.6	192.168.101.4	UDP	0	0	2.04 KB	<a href="#">Add Mapping</a>

Como se pudo analizar en la figura anterior, el tráfico generado por este protocolo está sujeto a las líneas de texto y conversaciones que se mantengan en todo el tiempo que demore un monitoreo. Dada su naturaleza de transmisión en tiempo real y su pequeño volumen, son datos no sensibles a retardos y no requieren un mecanismo de priorización o encolado para garantizar la entrega de paquetes a su

destino. Para cada PC o cliente, el tráfico generado por la aplicación fue solo de 118.16 KB, 117.82 KB y 94.14 KB.

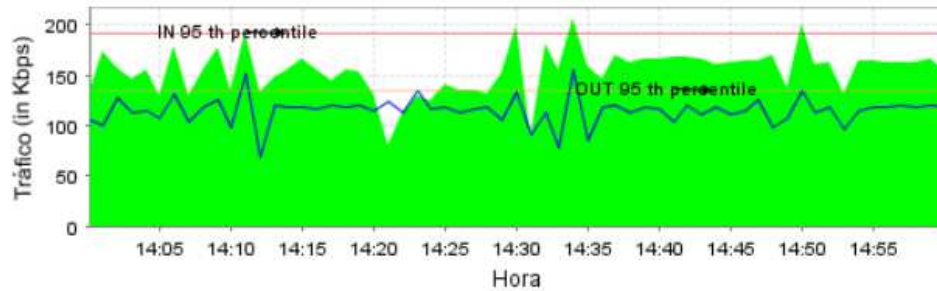
### 3.3.2.6 UDP (UNREAL) IMPLEMENTANDO PQ

Durante estas prácticas se priorizó el tráfico generado por la aplicación de juego en línea Unreal Tournament, el cual es de tipo audio y video y se transmite por UDP puerto 7777.

```

router#enable
router#configure terminal
router(config)#priority-list 1 protocol ip high udp 7777
router(config)#interface fastethernet 0/1
router(config-if)#priority-group 1
    
```

Gráfica 18



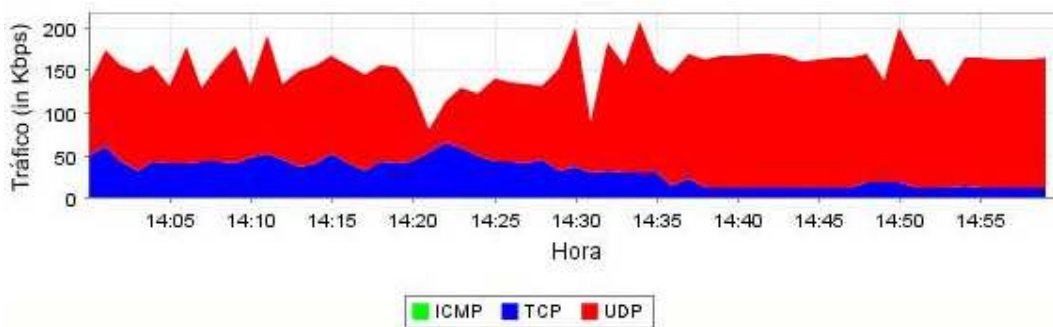
Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	70.26 MB	206.24 Kbps	79.97 Kbps	153.58 Kbps	190.93 Kbps
Tráfico SALIDA	52.52 MB	155.08 Kbps	68.49 Kbps	114.8 Kbps	134.04 Kbps

Los resultados de esta gráfica muestran un comportamiento diferente al obtenido en las pruebas de laboratorio anteriores. Como se puede observar, el volumen de tráfico de entrada es superior con respecto al de salida, con tasas de velocidades de

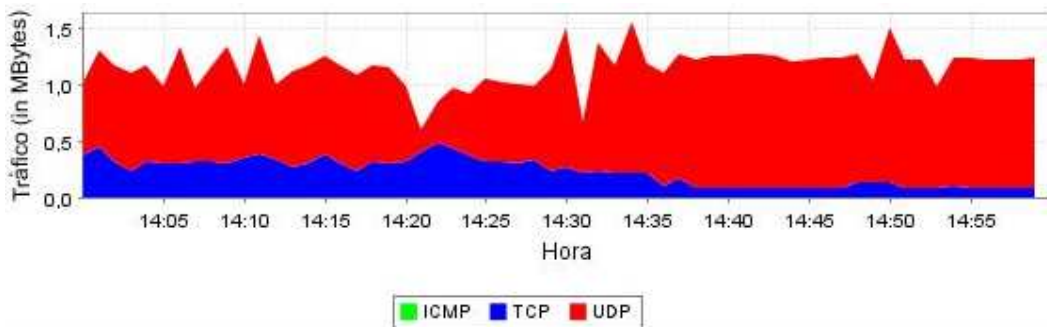


153.58 Kbps y 114.8 Kbps respectivamente. Esto se debe a que las condiciones de un juego en línea hacen que se produzcan grandes flujos de audio y video en ambas direcciones, desde los clientes hacia el servidor y viceversa. En general cualquier aplicación que genere intercambio de datos de esta clase presenta esta característica. Por lo tanto, al aplicar PQ se da prioridad al protocolo UDP con respecto a los demás protocolos utilizados por el resto de aplicaciones, cuyo flujo de datos es más unidireccional dependiendo de la naturaleza de las peticiones.

Gráfica 19



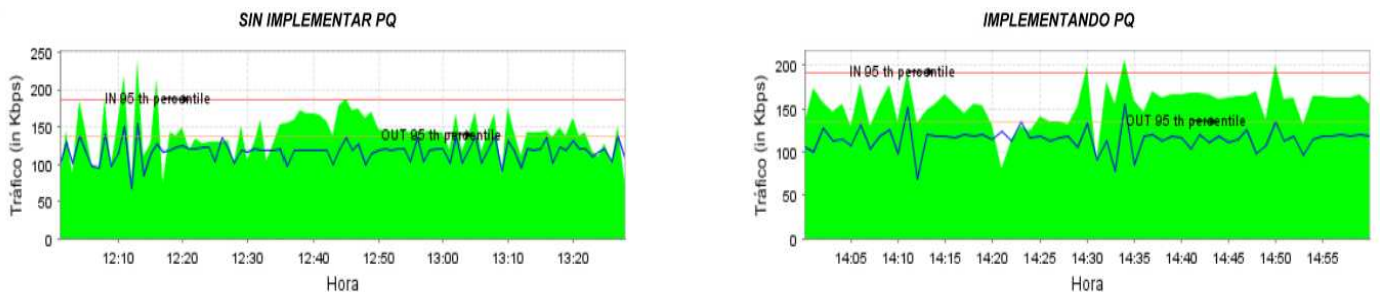
Gráfica 20



Analizando las graficas de diferenciación por protocolos, se puede notar que en el intervalo de 14:00 a 14:20 hubo una mayor congestión en la red y el tráfico de UDP se mantuvo cercano a los 1.5 MB. A las 14:22 aproximadamente se presentó una caída en el tráfico de UDP, lo que provocó la única alza significativa para el grupo de protocolos TCP en todo el tiempo de monitoreo. A partir de las 14:25 en adelante se observa cómo comienza a disminuir poco a poco el consumo de la red por parte de los protocolos TCP, lo cual conlleva a que el tráfico de UDP alcance picos de 1.6 MB y velocidades mayores a los 200Kbps. Estos resultados evidencian como el

tráfico de UDP es priorizado frente a los demás protocolos, puesto que en ningún momento del monitoreo las aplicaciones utilizadas dejaron de generar tráfico por tanto tiempo. Conforme la aplicación de audio y video iba pidiendo recursos de la red, así fue su procesamiento por el router, a diferencia de TCP.

### No PQ versus PQ implementado sobre Unreal



Categoría	Total	Máx	Min	Prom	95th Percentile	Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	92.58 MB	239.93 Kbps	72.06 Kbps	140.28 Kbps	185.66 Kbps	Tráfico ENTRADA	70.26 MB	206.24 Kbps	79.97 Kbps	153.58 Kbps	190.93 Kbps
Tráfico SALIDA	77.31 MB	155.49 Kbps	68.0 Kbps	117.15 Kbps	138.26 Kbps	Tráfico SALIDA	52.52 MB	155.08 Kbps	68.49 Kbps	114.8 Kbps	134.04 Kbps

Comparando las gráficas generales de tráfico total en el caso de priorización del protocolo UDP con respecto a cuando no se implementó PQ, se determinan los siguientes puntos:

- La tasa de velocidad de entrada obtenida implementando PQ es mucho mayor a la que se presentó al no configurarse dicho algoritmo, con una diferencia promedio de 13.3 Kbps. Por otro lado, la tasa de velocidad de salida de tráfico configurando PQ es menor que en el caso contrario, con una diferencia promedio de 2.35 Kbps.
- El volumen de tráfico de entrada sin implementar PQ se mantiene mucho más alto en comparación al volumen de tráfico de entrada implementándolo.



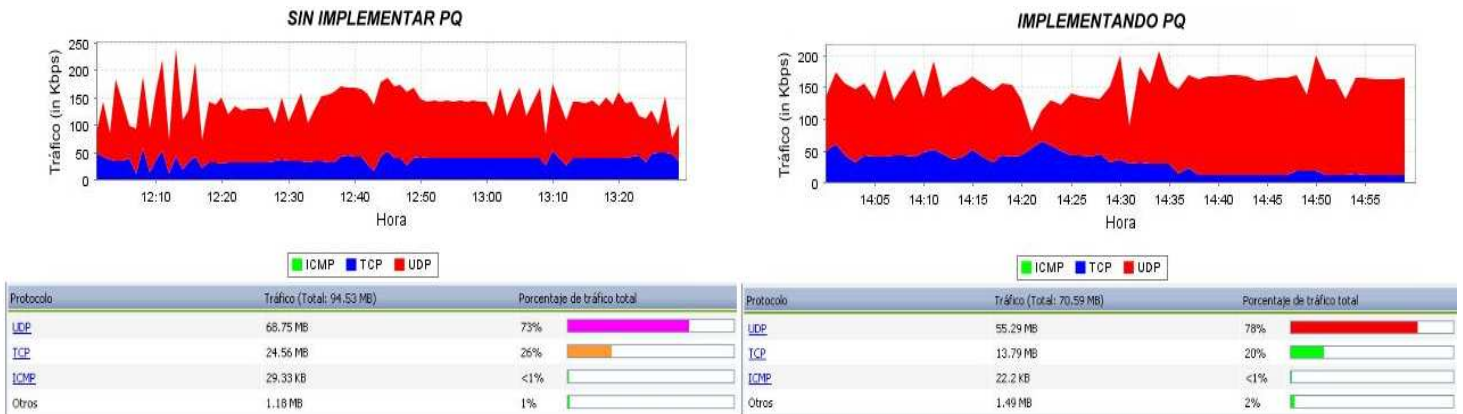
- El volumen de tráfico de salida sin implementar PQ es mucho mayor al volumen de tráfico de salida implementándolo.
- La velocidad del tráfico de entrada es mucho mayor al tráfico de salida en ambos casos.
- En ambas pruebas el tráfico de salida depende del tráfico de entrada, es decir, si disminuye el uno, el otro también lo hace, y viceversa.

### Porcentaje de tráfico de entrada no PQ versus PQ implementado sobre Unreal

SIN IMPLEMENTAR PQ			IMPLEMENTANDO PQ		
Aplicación	Tráfico (92.46 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (73.07 MB)	Porcentaje de tráfico total
UDP_App	59.24 MB	64%	UDP_App	54.87 MB	75%
mysql	21.82 MB	24%	mysql	12.13 MB	17%
ms-sql-s	7.04 MB	8%	smtp	1.12 MB	2%
TCP_App	940.52 KB	1%	TCP_App	524.47 KB	1%
smtp	553.53 KB	1%	bootps	360.58 KB	<1%
bootps	514.6 KB	1%	netbios-ns	27.22 KB	<1%
netbios-ns	64.88 KB	<1%	icmp	22.2 KB	<1%
icmp	29.33 KB	<1%	snmp	20.73 KB	<1%
snmp	28.53 KB	<1%	netbios-dgm	11.24 KB	<1%
netbios-dgm	24.91 KB	<1%	telnet	10.32 KB	<1%
telnet	14.27 KB	<1%	http	608.00 Bytes	<1%
ftp	2.39 KB	<1%	skinny	68.00 Bytes	<1%
http	1.64 KB	<1%	Unaccounted	3.97 MB	5%
skinny	204.00 Bytes	<1%			
pptp	120.00 Bytes	<1%			
sql-server	120.00 Bytes	<1%			
Unaccounted	2.17 MB	2%			

En esta comparación de tablas podemos ver cómo se ve afectado el tráfico del protocolo UDP implementando PQ, cuyo porcentaje sobre el tráfico total fue de 75% en contraste con un 64% que se obtuvo sin implementar algoritmos de encolamiento. A pesar de que en el 2º caso hubo menor consumo de la red o menor tráfico generado, la diferencia en cuanto a volúmenes es muy poca en comparación con otras aplicaciones cuyo protocolo es TCP (59.24 MB frente a 54.87 MB).

## No PQ vs PQ implementado sobre Unreal: diferenciación por protocolos



Al comparar las dos gráficas se pueden observar los siguientes comportamientos:

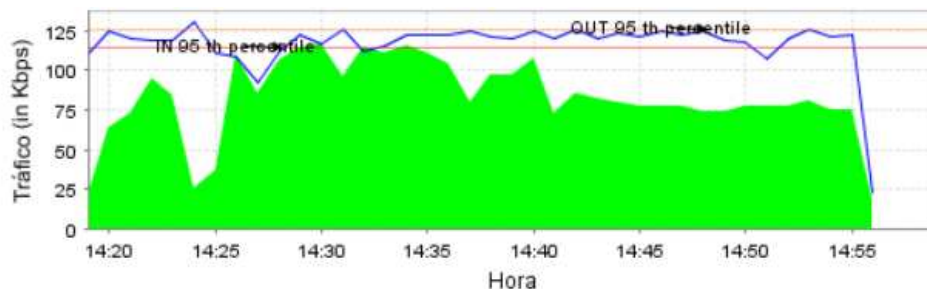
- El porcentaje de tráfico total usado donde se ha implementado PQ (UDP 78%) es un poco mayor al presentado cuando no se configuró el encolado de prioridad (UDP 73%).
- Al igual que en las anteriores pruebas, con PQ se observan cifras que no alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo. Sin embargo, esta vez hubo intervalos de tiempo donde la velocidad de transmisión se mostró cercana a los 200 Kbps, a diferencia de los casos en que se priorizaron protocolos TCP donde dichos valores se mantuvieron por debajo de los 150 Kbps.
- Al no implementarse PQ, el router tiene menos carga para procesar los paquetes, lo cual eleva el volumen de tráfico que puede circular por la red. Se habla de un tráfico total de 94.53 MB que circuló sin implementación de PQ contra un 70.58 MB implementándose, lo que deja en evidencia también en este caso la gran demanda de procesamiento por parte del router al utilizarse encolamiento de prioridad.

### 3.3.2.7 SQL IMPLEMENTANDO CQ

En el transcurso de este laboratorio se utilizó CQ para priorizar el tráfico generado mediante mysql hacia un servidor de bases de datos. Para su configuración se define una cola personalizada 1 donde se atenderán en primera instancia los paquetes de protocolo tcp puerto 3306, y una cola por defecto 5 que atenderá todos los demás paquetes que lleguen al router.

```
router#enable
router#configure terminal
router(config)#interface fastethernet 0/1
router(config-if)#custom-queue-list 1
router(config-if)#exit
router(config)#queue-list 1 protocol ip 1 tcp 3306
router(config)#queue-list 1 default 5
```

Gráfica 21



Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	23.56 MB	116.94 Kbps	19.14 Kbps	76.62 Kbps	114.02 Kbps
Tráfico SALIDA	33.25 MB	130.79 Kbps	23.17 Kbps	108.15 Kbps	125.5 Kbps

En la grafica 21 se puede notar que se presentan características similares al comportamiento del algoritmo PQ. En este caso se ve un porcentaje mayor de tráfico de salida que de entrada, con tasas de velocidad promedio de 108.15 kbps y 76.62 kbps respectivamente. También se observa un volumen más alto de tráfico de salida, de 33.25 MB respecto al de entrada que es de 23.56 MB. A partir de esto

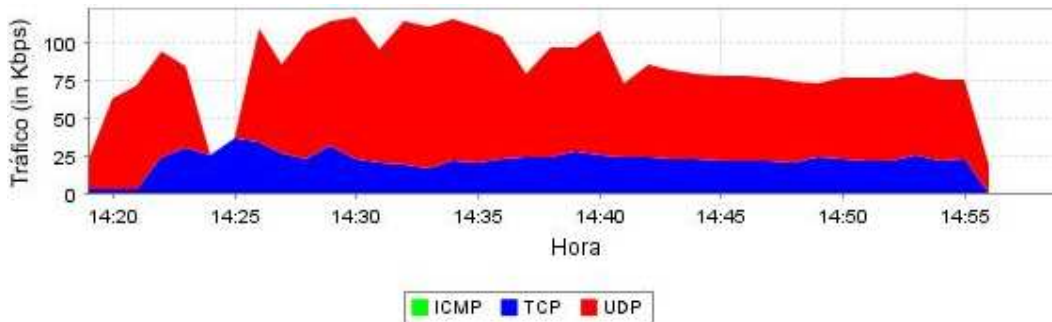
podemos decir que CQ también le otorga mayor prioridad a la operación de salida de datos desde los servidores que están fuera de la LAN manteniendo las conexiones mysql con la base de datos sobre la que se realizan inserciones de registros, debido a la congestión presentada en la operación de entrada y un mayor procesamiento en el router.

Si se analiza la gráfica que ilustra el tráfico de salida, es fácil notar que se mantiene mucho más alto que el de entrada independientemente de las caídas o valles que se formen, como el que se observa a las 14:25. De esta manera puede determinarse que CQ afecta el comportamiento de todos los protocolos TCP y UDP ubicados en la LAN, realizando un buen manejo a la salida de datos del router.

**Gráfica 22**

Aplicación	Tráfico (23.56 MB)	Porcentaje de tráfico total
UDP_App	16.28 MB	69%
mysql	5.2 MB	22%
radius	981.3 KB	4%
smtp	583.62 KB	2%
TCP_App	277.54 KB	1%
http	80.81 KB	<1%
snmp	74.02 KB	<1%
icmp	31.37 KB	<1%
epmap	28.64 KB	<1%
netbios-ns	11.42 KB	<1%
netbios-dgm	5.8 KB	<1%
ftp	2.37 KB	<1%
lockd	120,00 Bytes	<1%
skinny	68,00 Bytes	<1%
Unaccounted	2,00 Bytes	<1%

**Gráfica 23**



Al realizar la distribución por protocolos y analizar la gráfica 23, se observa que a las 14:25 el tráfico generado por puertos TCP no se afecta por la caída de UDP y la disminución de la congestión en la red. A pesar de que este fenómeno no duró mucho tiempo, lo anterior indica que al implementar CQ los tiempos de reacción del router se ven afectados al no detectar la caída y atender más rápidamente los paquetes del protocolo SQL, aunque en ningún momento dejó de hacerlo.

### No CQ versus CQ implementado sobre SQL



Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	92.58 MB	239.93 Kbps	72.06 Kbps	140.28 Kbps	185.66 Kbps
Tráfico SALIDA	77.31 MB	155.49 Kbps	68.0 Kbps	117.15 Kbps	138.26 Kbps

































Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	23.56 MB	116.94 Kbps	19.14 Kbps	76.62 Kbps	114.02 Kbps
Tráfico SALIDA	33.25 MB	130.79 Kbps	23.17 Kbps	108.15 Kbps	125.5 Kbps

Realizando una comparación general entre el comportamiento de la red cuando se ha implementado CQ para puertos usados por MySQL y en donde no se implementa encolado de prioridad podemos afirmar lo siguiente:

- Cuando se implementó CQ para priorizar los puertos utilizados por MySQL el tráfico de salida va a ser mayor que el tráfico de entrada, a diferencia del tráfico sin priorización, donde el tráfico de entrada es mayor que el tráfico de salida.
- El volumen del tráfico de entrada implementando CQ es mucho menor al volumen del tráfico de entrada sin implementación.
- La velocidad del tráfico de entrada sin implementar CQ es mucho mayor al tráfico de salida, caso contrario ocurre implementando CQ.

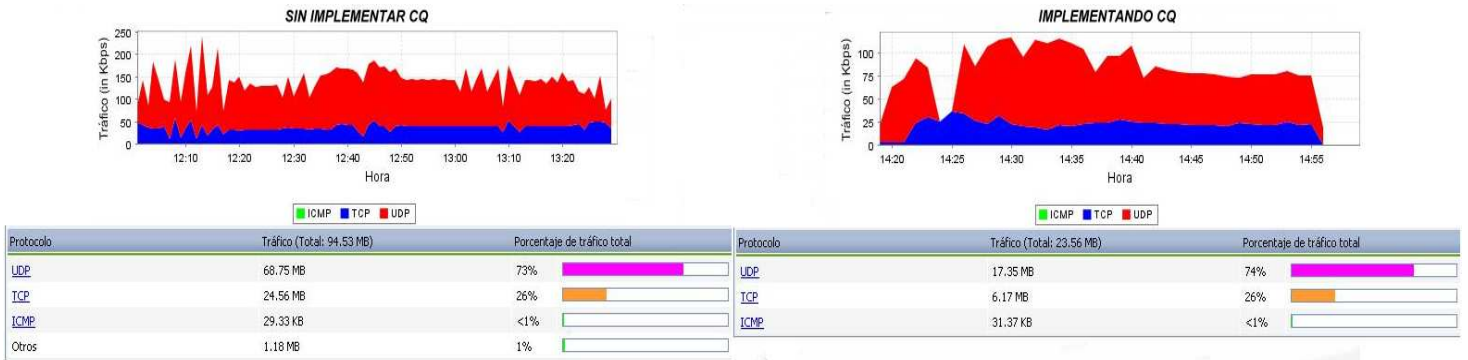
- El volumen de tráfico de salida sin implementar CQ es mayor al volumen de tráfico de salida implementando CQ pero la diferencia no es muy amplia.

### Porcentaje de tráfico de entrada no CQ versus CQ implementado sobre SQL

SIN IMPLEMENTAR CQ				IMPLEMENTANDO CQ			
Aplicación	Tráfico (92.46 MB)	Porcentaje de tráfico total		Aplicación	Tráfico (23.56 MB)	Porcentaje de tráfico total	
UDP_App	59.24 MB	64%		UDP_App	16.28 MB	69%	
mysql	21.82 MB	24%		mysql	5.2 MB	22%	
ms-sql-s	7.04 MB	8%		radius	981.3 KB	4%	
TCP_App	940.52 KB	1%		smtp	583.62 KB	2%	
smtp	553.53 KB	1%		TCP_App	277.54 KB	1%	
bootps	514.6 KB	1%		http	80.81 KB	<1%	
netbios-ns	64.88 KB	<1%		snmp	74.02 KB	<1%	
icmp	29.33 KB	<1%		icmp	31.37 KB	<1%	
snmp	28.53 KB	<1%		epmap	28.64 KB	<1%	
netbios-dgm	24.91 KB	<1%		netbios-ns	11.42 KB	<1%	
telnet	14.27 KB	<1%		netbios-dgm	5.8 KB	<1%	
ftp	2.39 KB	<1%		ftp	2.37 KB	<1%	
http	1.64 KB	<1%		lockd	120.00 Bytes	<1%	
skinnny	204.00 Bytes	<1%		skinnny	68.00 Bytes	<1%	
pptp	120.00 Bytes	<1%		Unaccounted	2.00 Bytes	<1%	
sql-server	120.00 Bytes	<1%					
Unaccounted	2.17 MB	2%					

Por lo que se puede observar en la gráfica se puede deducir que aunque el porcentaje en la implementación del algoritmo CQ haya sido menor (22%) con respecto al tráfico sin priorización (24%), no hay una gran diferencia entre estos valores. Por lo tanto no se puede afirmar a ciencia cierta que el algoritmo para esta aplicación no funcione.

## No CQ vs CQ implementado sobre SQL: diferenciación por protocolos



Al comparar las dos gráficas pueden notarse los siguientes comportamientos:

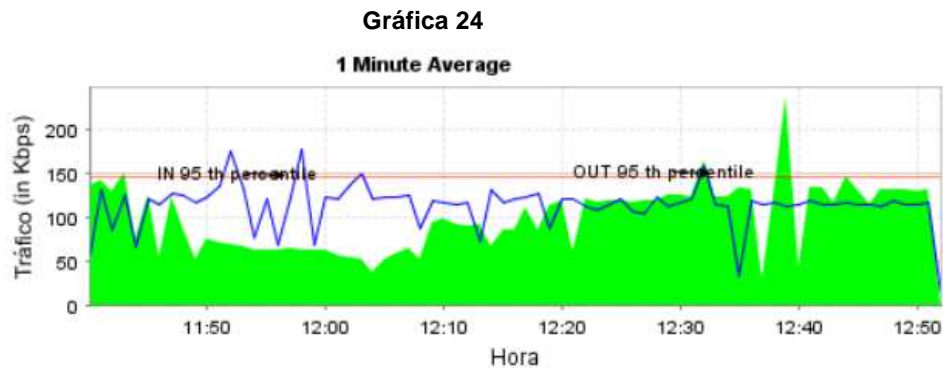
- El porcentaje del tráfico total utilizado por el grupo de protocolos TCP cuando se ha implementado CQ es igual al del tráfico sin priorización (26%).
- En este caso también se observan cifras que no alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo. Sin embargo, esta vez hubo intervalos de tiempo donde la velocidad de transmisión se mostró cercana a los 100 Kbps, un valor menor al que se presentó al configurar PQ y priorizar SQL que fue de 150 Kbps.
- El volumen general del tráfico sin priorización es mucho mas alto que el volumen general cuando se implementa el algoritmo CQ, lo que nuevamente deja en evidencia la gran demanda de procesamiento por parte del router al utilizarse encolamiento de prioridad.



### 3.3.2.8 FTP IMPLEMENTANDO CQ

En el transcurso de este laboratorio se utilizó CQ para priorizar el tráfico generado mediante peticiones de descarga de archivos a un servidor FTP. Para su configuración se define una cola personalizada 1 donde se atenderán en primera instancia los paquetes de protocolo tcp puerto 21, y una cola por defecto 5 que atenderá todos los demás paquetes que lleguen al router.

```
router#enable
router#configure terminal
router(config)#interface fastethernet 0/1
router(config-if)#custom-queue-list 1
router(config-if)#exit
router(config)#queue-list 1 protocol ip 1 tcp 21
router(config)#queue-list 1 default 5
```



Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	53.81 MB	237.07 Kbps	371.20 bps	98.28 Kbps	147.31 Kbps
Tráfico SALIDA	62.36 MB	177.49 Kbps	10.27 Kbps	113.91 Kbps	149.99 Kbps

En esta prueba se obtuvo un mayor promedio de velocidad para el tráfico de salida del router, con un valor de 113.91 Kbps respecto al tráfico de entrada que fue de 98.28 Kbps. De igual manera, el volumen total resultante de tráfico de salida fue

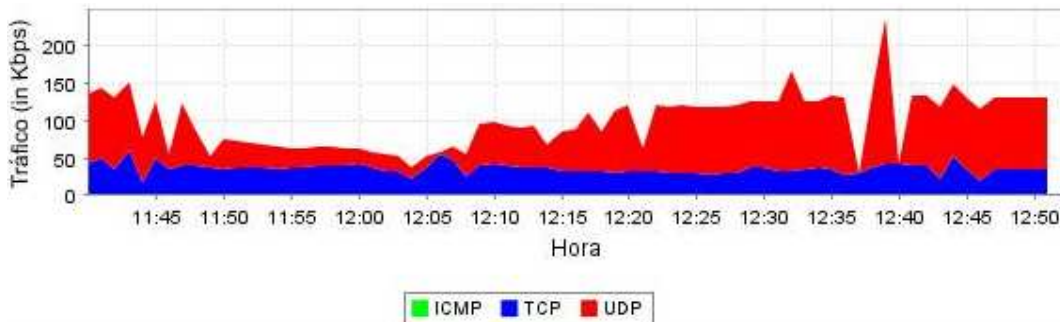


mayor al de entrada, con valores de 62.36 Mb y 53.81 Mb respectivamente. Todo esto demuestra que hay un flujo de paquetes muy superior proveniente de cualquiera de los servidores hacia los clientes, que correspondería a la salida de la interfaz fastethernet configurada y por lo tanto a la transferencia de archivos vía FTP, entre otros tipos de tráfico. Sin embargo, al igual que en las pruebas realizadas con PQ, el servidor FTP entra en modo pasivo al encontrarse detrás de un router, lo cual hace imposible la priorización de la transferencia real de archivos y solo deja el puerto 21 para las peticiones y autenticaciones. En este caso también se presentó un comportamiento similar a cuando no se realizó ningún tipo de priorización, al presentarse niveles paralelos de tráfico de entrada y salida. La diferencia que se puede observar es la disminución del tráfico de entrada en el intervalo de 11:50 a 12:10, que bien se podría deber a situaciones de reconexión con los servidores, peticiones de nuevos archivos o cambios de escenario en la aplicación de audio y video.

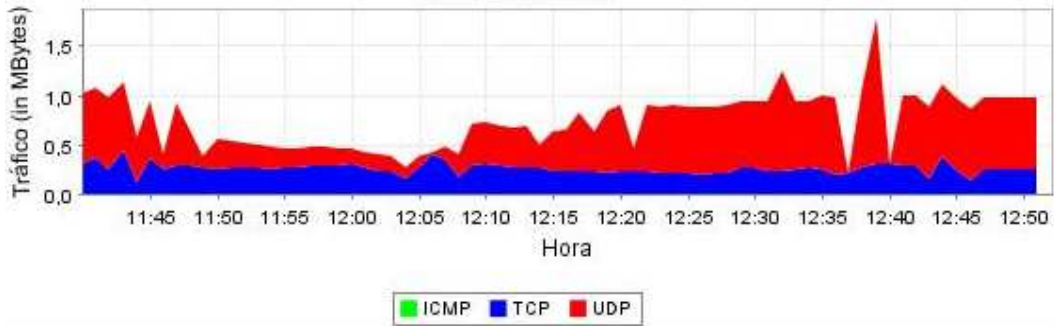
**Gráfica 25**

TRAFICO DE ENTRADA			TRAFICO DE SALIDA		
Aplicación	Tráfico (54.61 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (69.03 MB)	Porcentaje de tráfico total
UDP_App	33.28 MB	61%	TCP_App	39.35 MB	57%
mysql	15.91 MB	29%	UDP_App	25.44 MB	37%
smtp	1.17 MB	2%	mysql	2.4 MB	3%
TCP_App	912.34 KB	2%	smtp	1.42 MB	2%
http	145.87 KB	<1%	http	254.86 KB	<1%
snmp	131.46 KB	<1%	snmp	63.33 KB	<1%
epmap	59.98 KB	<1%	epmap	38.48 KB	<1%
icmp	54.65 KB	<1%	icmp	37.69 KB	<1%
netbios-ns	21.22 KB	<1%	netbios-ns	9.08 KB	<1%
netbios-dgm	11.65 KB	<1%	ftp	3.47 KB	<1%
ftp	2.51 KB	<1%	jrun_ejb	144.00 Bytes	<1%
skinny	204.00 Bytes	<1%	webmethods	144.00 Bytes	<1%
jrun_ejb	120.00 Bytes	<1%	gds_db	144.00 Bytes	<1%
webmethods	120.00 Bytes	<1%			
Unaccounted	2.89 MB	5%			

**Gráfica 26**



Gráfica 27



Mediante la gráfica 25 se evidencia el poco volumen de tráfico generado a partir de peticiones hacia el servidor por el puerto 21. Para las operaciones de entrada y salida se obtuvo un tráfico total de 2.51 Kb y 3.47 Kb por este protocolo, lo cual no representa prácticamente ninguna carga para el router.

También podemos observar que para el tráfico de salida, **TCP\_App** generó un porcentaje de tráfico total de 57% respecto al resto de protocolos. Esto corresponde a todo aquel tráfico realizado entre servidores y clientes por puertos aleatorios en el rango de 1024 a 65535, lo cual sucede cuando el servidor FTP entra en modo pasivo para realizar la transferencia real de archivos. Si bien es cierto su volumen fue mucho mayor que el del resto de protocolos, no se podría decir que tenga algo que ver con la priorización realizada puesto que solamente se tuvo en cuenta el puerto 21.

En la distribución de protocolos realizada sobre el tráfico de entrada, las gráficas 26 y 27 muestran un tráfico TCP de velocidad y volumen constante respecto a un tráfico muy variable del protocolo UDP. Esto es algo normal considerando que CQ solo funcionó en aquellos casos donde los clientes necesitaron pedir archivos y autenticarse con el servidor FTP, y los demás protocolos TCP se vieron la mayoría del tiempo superados por el tráfico de audio y video. Este último alcanzó velocidades de 200 kbps y volúmenes de 1.5 Mbps, asemejándose a un tráfico general sin priorización.

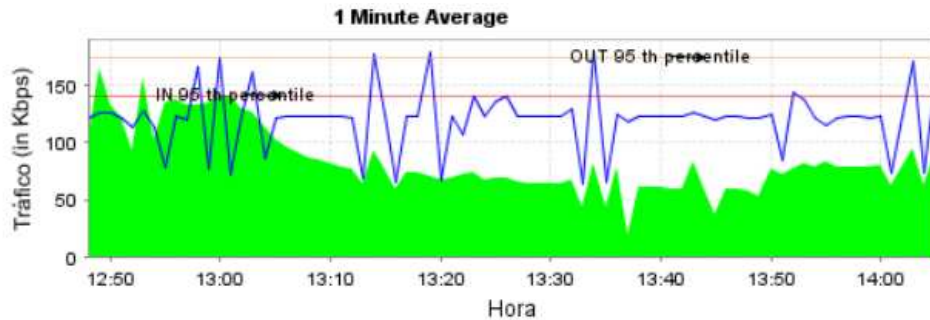
### 3.3.2.9 SMTP IMPLEMENTANDO CQ

En este laboratorio se utilizó CQ para priorizar el tráfico correspondiente al envío de correos electrónicos de los clientes hacia un servidor SMTP, ya sea interno o remoto. Para su configuración se define una cola personalizada 1 donde se atenderán en primera instancia los paquetes de protocolo tcp puerto 25, y una cola por defecto 5 que atenderá todos los demás paquetes que lleguen al router.

```

router#enable
router#configure terminal
router(config)#interface fastethernet 0/1
router(config-if)#custom-queue-list 1
router(config-if)#exit
router(config)#queue-list 1 protocol ip 1 tcp 25
router(config)#queue-list 1 default 5
    
```

Gráfica 28



Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	48.48 MB	163.66 Kbps	18.18 Kbps	82.87 Kbps	139.61 Kbps
Tráfico SALIDA	70.38 MB	180.28 Kbps	63.95 Kbps	120.31 Kbps	172.75 Kbps

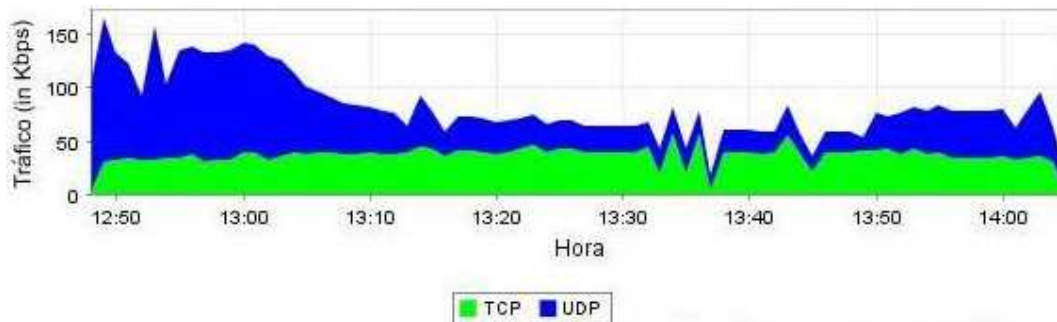
Como podemos observar en la gráfica general, se presenta un mayor volumen de tráfico de salida con un total de 70.38 MB y una velocidad de transferencia promedio de 120.31 Kbps, con respecto al tráfico de entrada que mostró valores de 48.48 MB

y 82.87 Kbps. Al ser el tráfico del protocolo SMTP de poco volumen, a pesar de generarse masivamente, no representa una gran parte del tráfico de entrada. Esto se puede evidenciar en la siguiente tabla, la cual muestra un volumen generado por SMTP de 761.91 KB, mucho mayor a los casos donde se priorizaron otros protocolos.

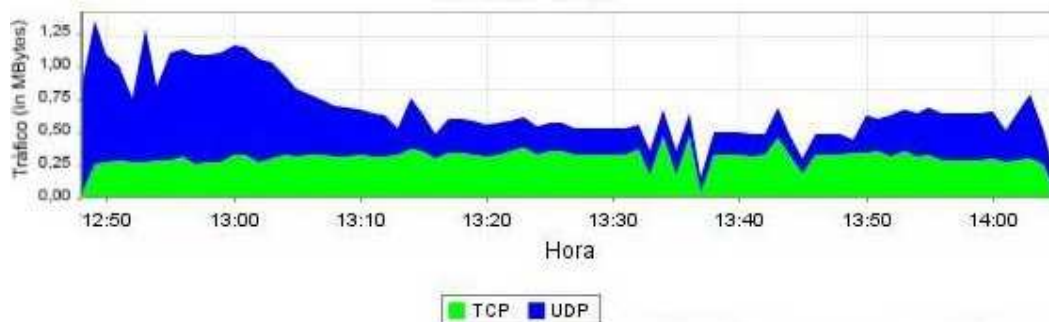
**Gráfica 29**

Aplicación	Tráfico (51.19 MB)	Porcentaje de tráfico total
UDP_App	26.09 MB	51%
mysql	18.99 MB	37%
TCP_App	1.46 MB	3%
smtp	761.91 KB	1%
bootps	463.87 KB	1%
netbios-ns	101.08 KB	<1%
netbios-dgm	13.5 KB	<1%
snmp	2.98 KB	<1%
Unaccounted	3.3 MB	6%

**Gráfica 30**

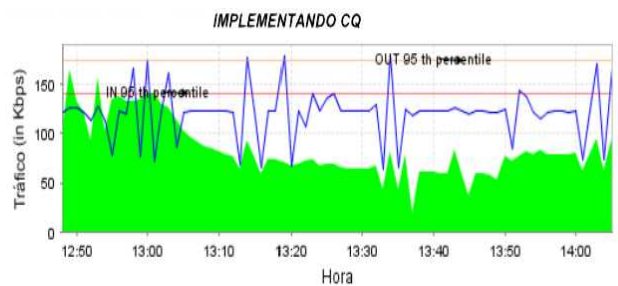


**Gráfica 31**



En las graficas 30 y 31 podemos ver que a pesar de haber sido mucho mayor el volumen de audio y video comenzando el monitoreo, las tasas de transferencia de TCP y UDP llegan a hacerse constantes y casi iguales la mayoría del tiempo, lo cual indica el funcionamiento del encolado personalizado. Teniendo en cuenta que el tráfico de audio y video siempre va a ser mayor a los demás, y que el envío de correos electrónicos representa en nuestro caso un tráfico de poco volumen pero constante, el router necesita encolarlo y priorizarlo frente al de mayor volumen. Es por esto que el tráfico de audio y video tiende a reducirse pero manteniéndose mayor, atendiéndose sin llegar a descartarse en ningún momento.

### No CQ versus CQ implementado sobre SMTP



Categoría	Total	Máx	Min	Prom	95th Percentile	Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	92.58 MB	239.93 Kbps	72.06 Kbps	140.28 Kbps	185.66 Kbps	Tráfico ENTRADA	48.48 MB	163.66 Kbps	18.18 Kbps	82.87 Kbps	139.61 Kbps
Tráfico SALIDA	77.31 MB	165.49 Kbps	68.0 Kbps	117.15 Kbps	138.26 Kbps	Tráfico SALIDA	70.38 MB	180.28 Kbps	63.95 Kbps	120.31 Kbps	172.75 Kbps

Realizando una comparación general entre el comportamiento de la red cuando se ha implementado CQ para puertos usados por SMTP y en donde no se implementa encolado de prioridad podemos afirmar lo siguiente:

- Cuando se implementó CQ para priorizar los puertos utilizados por SMTP el tráfico de salida resulta siendo mayor que el tráfico de entrada, a diferencia del tráfico sin priorización, donde el tráfico de entrada es mayor que el tráfico de salida.
- El volumen del tráfico de entrada implementando CQ es mucho menor al volumen del tráfico de entrada sin implementación.

- La velocidad del tráfico de entrada sin implementar CQ es mucho mayor al tráfico de salida, caso contrario ocurre implementando CQ.
- El volumen de tráfico de salida sin implementar CQ es mayor al volumen de tráfico de salida implementando CQ pero la diferencia no es muy amplia.

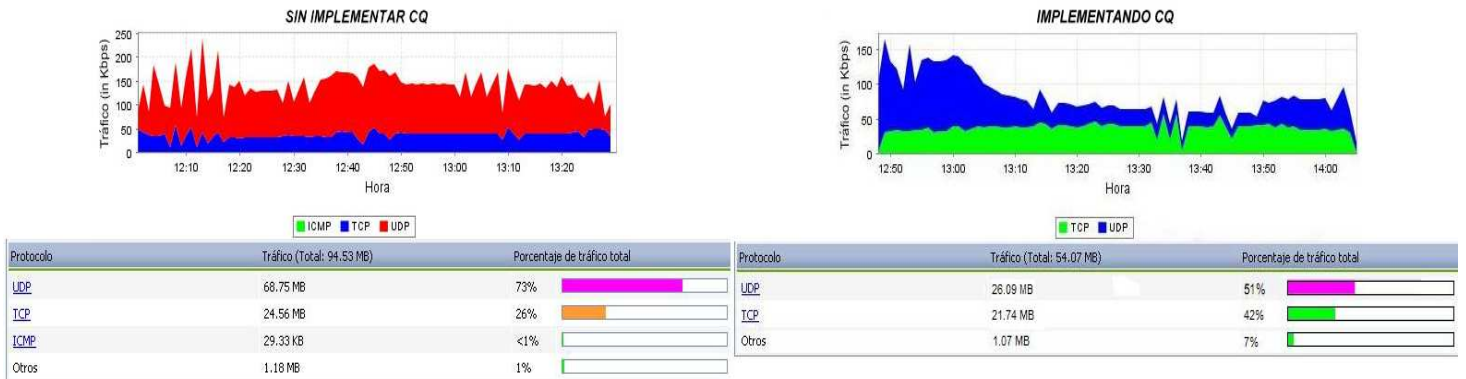
### Porcentaje de tráfico de entrada no CQ versus CQ implementado sobre SMTP

SIN IMPLEMENTAR CQ			IMPLEMENTANDO CQ		
Aplicación	Tráfico (92.46 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (51.19 MB)	Porcentaje de tráfico total
UDP_App	59.24 MB	64%	UDP_App	26.09 MB	51%
mysql	21.82 MB	24%	mysql	18.99 MB	37%
ms-sql-s	7.04 MB	8%	TCP_App	1.46 MB	3%
TCP_App	940.52 KB	1%	smtp	761.91 KB	1%
smtp	553.53 KB	1%	bootps	463.87 KB	1%
bootps	514.6 KB	1%	netbios-ns	101.08 KB	<1%
netbios-ns	64.88 KB	<1%	netbios-dgm	13.5 KB	<1%
icmp	29.33 KB	<1%	snmp	2.98 KB	<1%
snmp	28.53 KB	<1%	Unaccounted	3.3 MB	6%
netbios-dgm	24.91 KB	<1%			
telnet	14.27 KB	<1%			
ftp	2.39 KB	<1%			
http	1.64 KB	<1%			
skinny	204.00 Bytes	<1%			
pptp	120.00 Bytes	<1%			
sql-server	120.00 Bytes	<1%			
Unaccounted	2.17 MB	2%			

En esta tabla comparativa podemos observar que en ambos casos se obtuvo un 1% de tráfico de entrada para el protocolo SMTP, pero el mayor volumen generado se presenta cuando se configuró CQ sobre la interfaz. En este caso se obtuvo 761.91 KB, en comparación con un valor de 553.53 KB obtenido al no implementar CQ.



## No CQ vs CQ implementado sobre SMTP: diferenciación por protocolos



Al comparar las dos gráficas pueden notarse los siguientes comportamientos:

- El porcentaje del tráfico total utilizado por el grupo de protocolos TCP cuando se ha implementado CQ (42%) es mayor al del tráfico sin priorización (26%).
- En este caso también se observan cifras que no alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo.
- El volumen general del tráfico sin priorización es mucho más alto que el volumen general cuando se implementa el algoritmo CQ, lo que nuevamente deja en evidencia la gran demanda de procesamiento por parte del router al utilizarse encolamiento de prioridad.

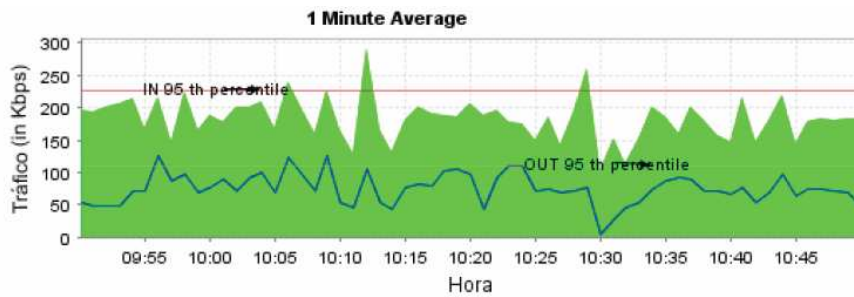
### 3.3.2.10 UDP (UNREAL) IMPLEMENTANDO CQ

Para esta práctica de laboratorio se configuró el algoritmo Custom Queueing (CQ) para priorizar el tráfico generado por el juego en línea Unreal Tournament, el cual es de tipo audio y video y se transmite por UDP puerto 7777. Para su configuración se define una cola personalizada 1 donde se atenderán en primera instancia los paquetes del protocolo UDP puerto 7777, y una cola por defecto 5 que atenderá por defecto los demás paquetes que lleguen al router.

```

router#enable
router#configure terminal
router(config)#interface fastethernet 0/1
router(config-if)#custom-queue-list 1
router(config-if)#exit
router(config)#queue-list 1 protocol ip 1 udp 7777
router(config)#queue-list 1 default 5
    
```

**Gráfica 32**



Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	83.41 MB	290.93 Kbps	101.68 Kbps	182.31 Kbps	226.12 Kbps
Tráfico SALIDA	34.42 MB	125.87 Kbps	5.09 Kbps	75.23 Kbps	110.75 Kbps

En la grafica podemos observar que el tráfico de entrada a la interfaz del router es mucho mayor al tráfico de salida. Este último es de 34.42 Mb, mientras que el de entrada es de 83.41Mb. Debido a la naturaleza de este tipo de tráfico, se produce una mayor congestión en la red al fluir paquetes desde todos los clientes hacia el

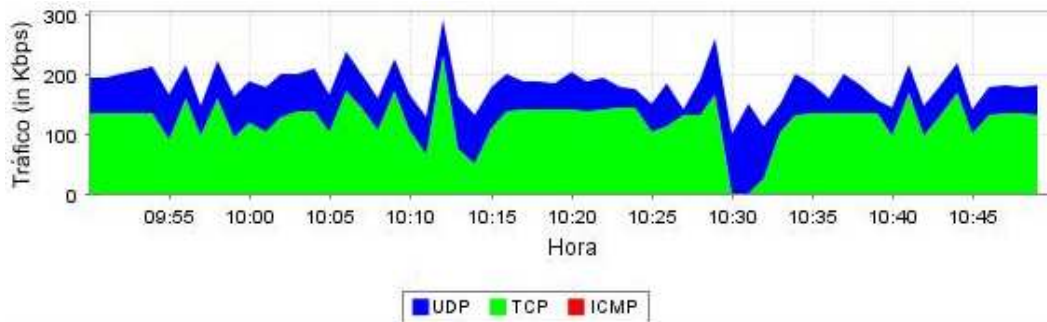


servidor (entrada) que de un solo servidor hacia todos los clientes (salida). Por lo tanto, siempre va a existir una diferencia realmente notoria durante la comunicación entre ambas vías.

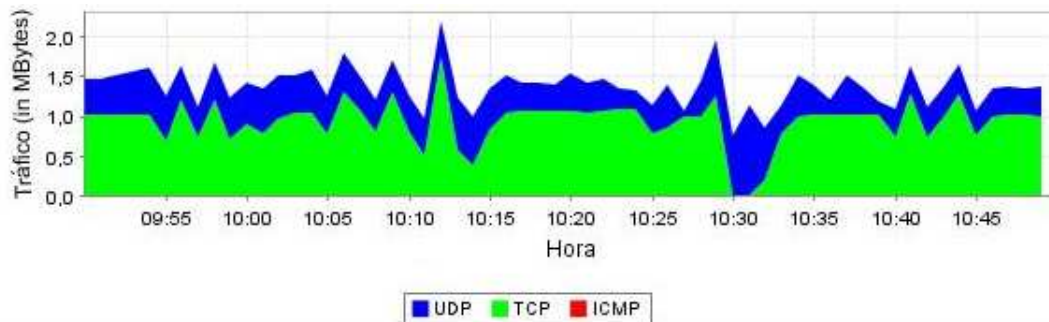
**Gráfica 33**

Aplicación	Tráfico (92.69 MB)	Porcentaje de tráfico total
UDP_App	54.96 MB	59%
mysql	13.77 MB	15%
TCP_App	10.78 MB	12%
smtp	1.87 MB	2%
bootps	360.58 KB	<1%
netbios-ns	215.2 KB	<1%
http	28.94 KB	<1%
ftp	14.75 KB	<1%
netbios-dgm	10.3 KB	<1%
icmp	1.32 KB	<1%
msmq	144,00 Bytes	<1%
knetd	144,00 Bytes	<1%
Unaccounted	10.65 MB	11%

**Gráfica 34**



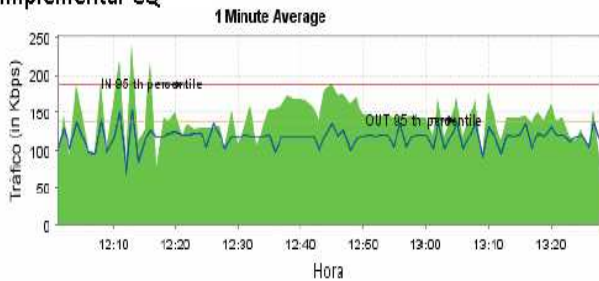
**Gráfica 35**



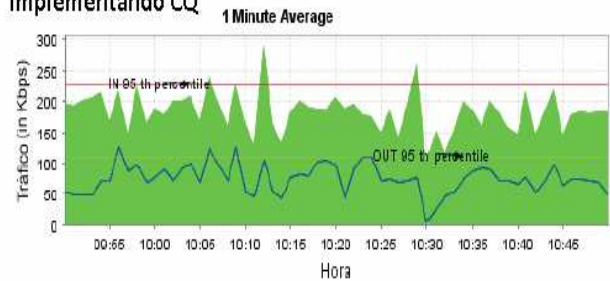
Analizando las gráficas de diferenciación de protocolos puede notarse que en términos generales hubo congestión en la red durante todo el monitoreo, los picos y valles de ambos protocolos fueron coincidentes en casi todos los puntos. Pero si se observa con detenimiento, a las 10:30 se presenta una caída del protocolo TCP mas eso no ocurre con el protocolo UDP que mantiene su tráfico, por lo que se puede afirmar la presencia del algoritmo CQ. Después de haberse presentado esto el protocolo TCP vuelve a aumentar su tráfico y se vuelve a evidenciar un comportamiento similar al que hubo antes de la caída. En promedio el tráfico UDP tuvo un volumen de 1.3 MB con picos de hasta 2 MB y en promedio velocidades cercanas a 200 Kbps con picos de 300 Kbps.

### No CQ versus CQ implementado sobre Unreal

Sin implementar CQ



Implementando CQ



Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	82.58 MB	238.93 Kbps	72.08 Kbps	140.28 Kbps	185.86 Kbps
Tráfico SALIDA	77.31 MB	155.49 Kbps	68.0 Kbps	117.15 Kbps	138.26 Kbps

Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	83.41 MB	280.93 Kbps	101.88 Kbps	182.31 Kbps	228.12 Kbps
Tráfico SALIDA	34.42 MB	125.87 Kbps	5.09 Kbps	75.23 Kbps	110.75 Kbps

Comparando las gráficas generales de tráfico total en el caso de priorización del protocolo UDP con respecto a cuando no se implementó CQ, se determinan los siguientes puntos:

- La tasa de velocidad de entrada obtenida implementando CQ es mucho mayor a la que se presentó al no configurarse dicho algoritmo, con una diferencia promedio de 42.03 Kbps. Por otro lado, la tasa de velocidad de

salida de tráfico configurando CQ es menor que en el caso contrario, con una diferencia promedio de 41.92 Kbps.

- El volumen de tráfico de entrada sin implementar CQ se mantiene más alto en comparación al volumen de tráfico de entrada implementándolo.
- El volumen de tráfico de salida sin implementar CQ es mucho mayor al volumen de tráfico de salida implementándolo.
- La velocidad del tráfico de entrada es mucho mayor al tráfico de salida en ambos casos.
- En ambas pruebas el tráfico de salida depende del tráfico de entrada, es decir, si disminuye el uno, el otro también lo hace, y viceversa.

### Porcentaje de tráfico de entrada no CQ versus CQ implementado sobre Unreal

Sin implementar CQ

Aplicación	Tráfico (92.46 MB)	Porcentaje de tráfico total
UDP_App	59.24 MB	64%
mysql	21.82 MB	24%
ms-sql-s	7.04 MB	8%
TCP_App	040.52 KB	1%
smtp	553.53 KB	1%
bootps	514.8 KB	1%
netbios-ns	84.88 KB	<1%
icmp	29.33 KB	<1%
snmp	28.53 KB	<1%
netbios-dgm	24.81 KB	<1%
telnet	14.27 KB	<1%
ftp	2.39 KB	<1%
http	1.84 KB	<1%
skinny	204,00 Bytes	<1%
pptp	120,00 Bytes	<1%
sql-server	120,00 Bytes	<1%
Unaccounted	2.17 MB	2%

Implementando CQ

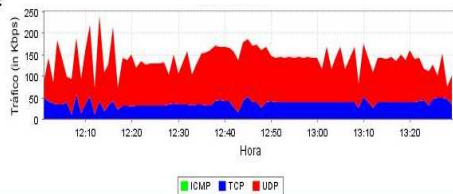
Aplicación	Tráfico (92.69 MB)	Porcentaje de tráfico total
UDP_App	54.98 MB	59%
mysql	13.77 MB	15%
TCP_App	10.78 MB	12%
smtp	1.87 MB	2%
bootps	380.58 KB	<1%
netbios-ns	215.2 KB	<1%
http	28.94 KB	<1%
ftp	14.75 KB	<1%
netbios-dgm	10.3 KB	<1%
icmp	1.32 KB	<1%
msmq	144,00 Bytes	<1%
knetd	144,00 Bytes	<1%
Unaccounted	10.65 MB	11%

En esta comparación de tablas puede observarse que el porcentaje del tráfico generado por el protocolo UDP implementando CQ es menor (59%) que el

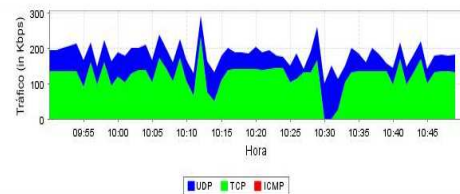
porcentaje del tráfico generado por este mismo protocolo sin implementar algoritmos de encolamiento (64%). Aunque el tráfico implementando el algoritmo de encolamiento es menor que el tráfico sin implementar el algoritmo no se puede afirmar que el algoritmo no funcione. Si se compara el porcentaje de tráfico de las demás aplicaciones en cada una de las tablas se nota que los porcentajes de tráfico son menores implementando el algoritmo, por lo cual se afirma que CQ si le da prioridad al protocolo UDP.

### No CQ vs CQ implementado sobre Unreal: diferenciación por protocolos

Sin implementar CQ



Implementando CQ



Protocolo	Tráfico (Total: 94.53 MB)	Porcentaje de tráfico total
UDP	68.75 MB	73%
TCP	24.56 MB	26%
ICMP	29.33 KB	<1%
Otros	1.18 MB	1%

Protocolo	Tráfico (Total: 95.41 MB)	Porcentaje de tráfico total
UDP	55.55 MB	58%
TCP	26.48 MB	28%
ICMP	1.32 KB	<1%
Otros	13.37 MB	14%

Al comparar las dos gráficas se observan los siguientes comportamientos:

- El porcentaje de tráfico total usado donde se ha implementado CQ (UDP 58%) es menor al presentado cuando no se configuró el encolado de prioridad (UDP 75%).
- A diferencia del tráfico sin implementar ningún tipo de algoritmo de encolamiento, en el caso en el que se ha implementado CQ podemos observar picos de velocidades más altos cercanos a los 300 Kbps y valles de velocidades que no descienden de 100 Kbps.

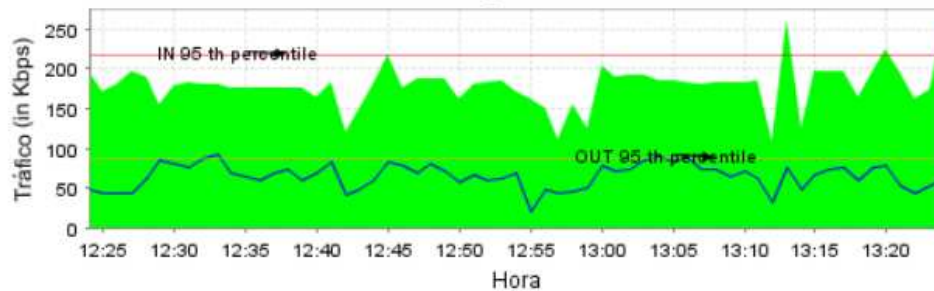
### 3.3.2.11 IMPLEMENTACIÓN DE WFQ PARA TODOS LOS PROTOCOLOS

Para este monitoreo se configura WFQ sobre la interfaz fastethernet 0/1 del router, clasificando y priorizando de forma automática aquellos protocolos que el algoritmo determine como sensibles a retardos. Para ello se definen como parámetros un tamaño para cada cola de 64 mensajes permitidos y un número máximo de 512 colas dinámicas que puede crear el algoritmo, como se puede ver a continuación:

```
router#enable
router#configure terminal
router(config)#interface fastethernet 0/1
router(config-if)#fair-queue 64 512
```

**Gráfica 36**

**1 Minute Average**



Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	81.32 MB	261.21 Kbps	108.52 Kbps	177.76 Kbps	217.36 Kbps
Tráfico SALIDA	29.86 MB	91.52 Kbps	20.14 Kbps	65.28 Kbps	87.94 Kbps

La gráfica general muestra un comportamiento muy similar a los casos donde se priorizó UDP con los demás algoritmos, ya que se presentan mayores volúmenes y velocidades promedio de tráfico de entrada que de salida. Esto se debe a que en nuestro caso el protocolo UDP agrupa servicios de audio y video y mensajería instantánea, los cuales son sensibles a retardos en caso de congestión. WFQ es un algoritmo basado en flujos que clasifica el tráfico de acuerdo a lo anterior, y pone al principio de la cola todo aquel tráfico de tiempo real relativo a conversaciones de

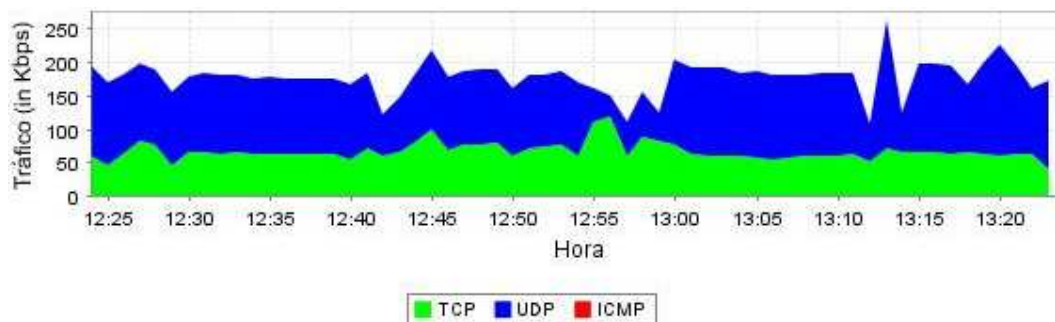
chat o videoconferencias. Por ello, al fluir un mayor volumen de este tipo de tráfico a la entrada que a la salida de la interfaz del router, se realiza un mejor encolado y el volumen de entrada que se obtiene también es mayor al de salida.

Para el resto de protocolos hace una clasificación de acuerdo a este principio para su encolado, y reparte equitativamente el ancho de banda restante entre ellos.

**Gráfica 37**

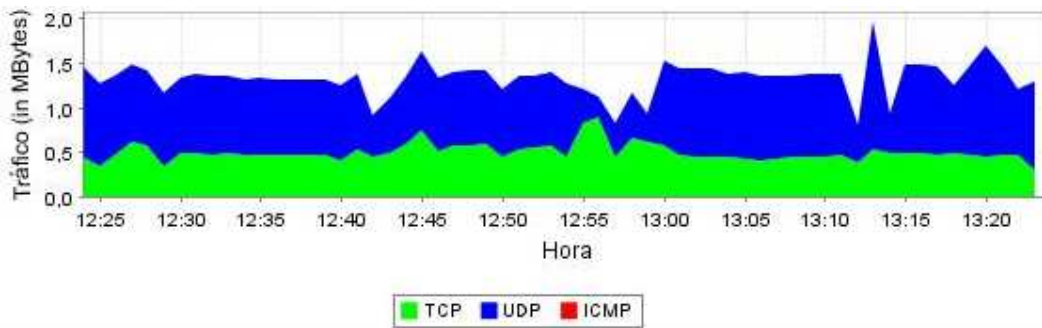
Aplicación	Tráfico (88.16 MB)	Porcentaje de tráfico total
UDP_App	48.94 MB	56%
mysql	18.43 MB	21%
TCP_App	9.73 MB	11%
smtp	1.73 MB	2%
bootps	360.58 KB	<1%
netbios-ns	220.66 KB	<1%
http	16.84 KB	<1%
netbios-dgm	10.1 KB	<1%
ftp	8.85 KB	<1%
icmp	4.03 KB	<1%
gds_db	144,00 Bytes	<1%
HTTP-Proxy	144,00 Bytes	<1%
netmeeting	144,00 Bytes	<1%
skinny	68,00 Bytes	<1%
Unaccounted	8.69 MB	10%

**Gráfica 38**



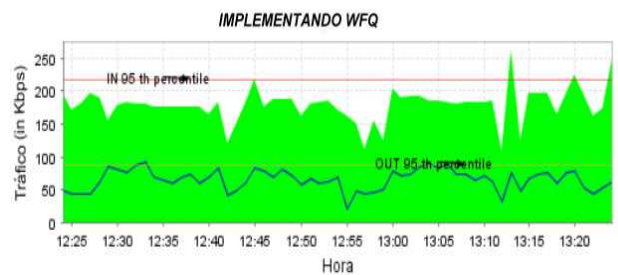
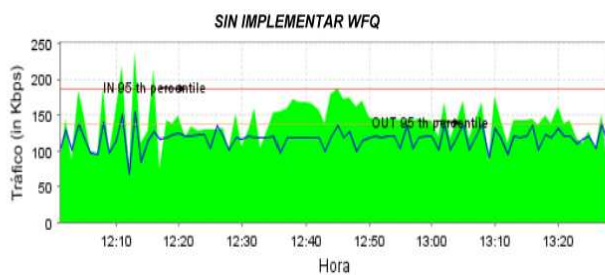


**Gráfica 39**



En la gráfica 37 se puede observar como el grupo de protocolos UDP representan un 56% del tráfico total atendido a la entrada del router y entregado a su destino, el cual fue mucho mayor respecto a otras pruebas de laboratorio. En las gráficas de distribución por protocolos se obtienen valores máximos de tasas de transferencia de hasta 250 Kbps y 2.0 Mbps, cifras que solo se obtienen en aquellos casos donde se priorizó tráfico de audio y video. También se observa que pese a esto, el tráfico de protocolos TCP no se ve afectado en gran medida y alcanza tasas de transferencia por encima de los 100 Kbps y 1.0 Kbps, lo cual deja en evidencia un manejo del ancho de banda justo entre todos los tipos de tráfico.

**No WFQ versus WFQ sobre tráfico en general**



Categoría	Total	Máx	Min	Prom	95th Percentile	Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	92.58 MB	239.93 Kbps	72.06 Kbps	140.28 Kbps	185.66 Kbps	Tráfico ENTRADA	81.32 MB	261.21 Kbps	108.52 Kbps	177.76 Kbps	217.36 Kbps
Tráfico SALIDA	77.31 MB	155.49 Kbps	68.0 Kbps	117.15 Kbps	138.26 Kbps	Tráfico SALIDA	29.86 MB	91.52 Kbps	20.14 Kbps	65.28 Kbps	87.94 Kbps

Realizando una comparación general entre el comportamiento de la red cuando se ha implementado WFQ sobre la interfaz del router y cuando no se implementa encolado de prioridad, podemos afirmar lo siguiente:

- Cuando se implementa WFQ el tráfico de entrada va a ser mayor que el tráfico de salida, al igual que cuando no se realizó ningún tipo de priorización.
- El volumen del tráfico de entrada implementando WFQ es algo menor al volumen del tráfico de entrada sin implementación.
- El volumen de tráfico de salida sin implementar WFQ es mucho mayor al volumen de tráfico de salida implementando WFQ.
- La velocidad promedio del tráfico de entrada sin implementar WFQ es mucho mayor a la del tráfico de salida, al igual que cuando se implementó el algoritmo.
- La velocidad promedio del tráfico de entrada implementando WFQ es mayor a cuando no se efectuó priorización. Caso contrario ocurre para el tráfico de salida, cuya velocidad promedio fue menor cuando se implementó WFQ que al no configurarse el algoritmo.

### Porcentaje de tráfico de entrada no WFQ versus WFQ

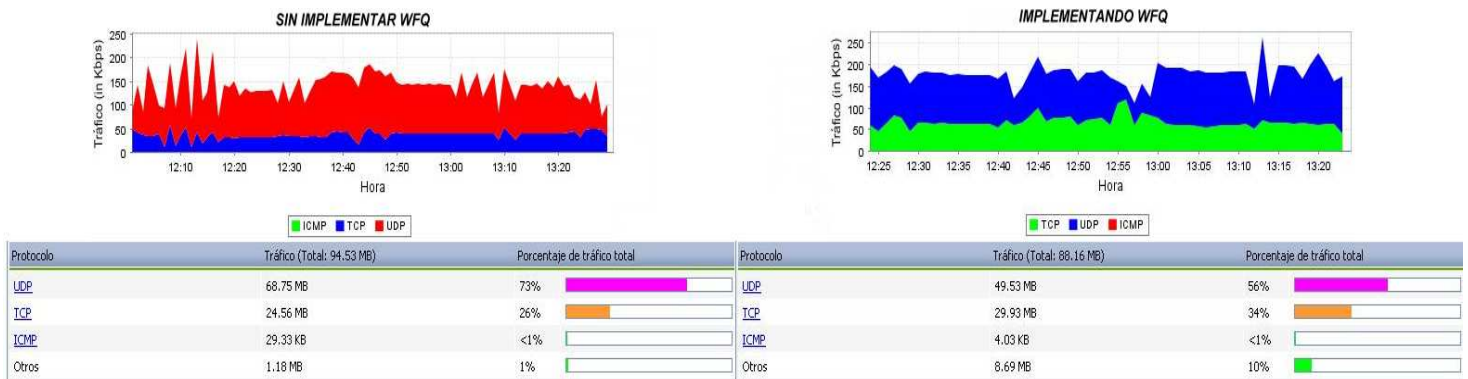
SIN IMPLEMENTAR WFQ			IMPLEMENTANDO WFQ		
Aplicación	Tráfico (92.46 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (88.16 MB)	Porcentaje de tráfico total
UDP_App	59.24 MB	64%	UDP_App	48.94 MB	56%
mysql	21.82 MB	24%	mysql	18.43 MB	21%
ms-sql-s	7.04 MB	8%	TCP_App	9.73 MB	11%
TCP_App	940.52 KB	1%	smtp	1.73 MB	2%
smtp	553.53 KB	1%	bootps	360.58 KB	<1%
bootps	514.6 KB	1%	netbios-ns	220.66 KB	<1%
netbios-ns	64.88 KB	<1%	http	16.84 KB	<1%
icmp	29.33 KB	<1%	netbios-dgm	10.1 KB	<1%
snmp	28.53 KB	<1%	ftp	8.85 KB	<1%
netbios-dgm	24.91 KB	<1%	icmp	4.03 KB	<1%
telnet	14.27 KB	<1%	gds_db	144,00 Bytes	<1%
ftp	2.39 KB	<1%	HTTP-Proxy	144,00 Bytes	<1%
http	1.64 KB	<1%	netmeeting	144,00 Bytes	<1%
skinny	204,00 Bytes	<1%	skinny	68,00 Bytes	<1%
pptp	120,00 Bytes	<1%	Unaccounted	8.69 MB	10%
sql-server	120,00 Bytes	<1%			
Unaccounted	2.17 MB	2%			

En esta tabla comparativa se nota cómo al implementar el algoritmo WFQ se atendieron volúmenes de tráfico considerables para cada uno de los protocolos y



servicios, teniendo en cuenta que en ambos casos se generó el mismo tráfico desde igual número de clientes. Esto permite dar una idea de cómo el algoritmo realiza el encolado justo y comparte equitativamente el ancho de banda para cada uno de ellos, evitando que sucedan pérdidas de paquetes.

### No WFQ vs WFQ: diferenciación por protocolos



Al comparar las dos gráficas pueden notarse los siguientes comportamientos:

- El porcentaje del tráfico total generado por el grupo de protocolos UDP cuando se ha implementado WFQ (56%) es menor al del tráfico sin priorización (73%). Por otro lado, el porcentaje de tráfico total correspondiente a los protocolos TCP (34%) fue mayor al del tráfico sin priorización (26%), aunque la diferencia no fue enorme.
- En este caso se observan cifras que alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo. En el caso de TCP las tasas de transferencia fueron mayores cuando se implementó WFQ.
- El volumen general del tráfico sin priorización es mucho más alto que el volumen general cuando se implementa el algoritmo WFQ, lo que deja en

evidencia que aún para este tipo de encolado de prioridad se requiere una gran demanda de procesamiento en el router.

### 3.3.3 ANÁLISIS DE RESULTADOS EN LA WAN CON CONEXIÓN A INTERNET

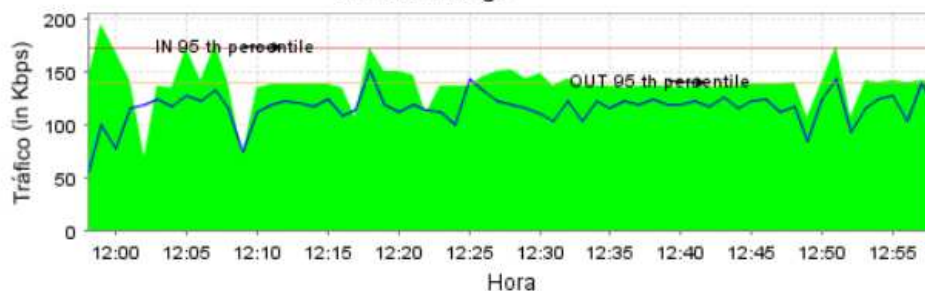
Durante esta parte del trabajo como primera instancia se simuló el acceso a internet en una LAN empresarial o institucional, definiendo servidores web y POP3 remotos y realizando peticiones en ellos. Una WAN como la que representa internet puede estar compuesta por múltiples redes externas y dispositivos de capa 3, pero en el caso de un laboratorio resulta más factible realizar una topología de 2 o 3 redes remotas. En nuestro caso dichos servidores se ubicaron en la red definida por los Router y Switch número 2.

Para llevar a cabo esta parte del análisis nos limitaremos a observar el comportamiento de los algoritmos de encolado configurándolos sobre la interfaz fastethernet del router 1 y priorizando los protocolos HTTP y POP3. De esta manera estudiaremos el tratamiento que se le da a estos tipos de tráfico, usuales en cualquier tipo de red, frente a situaciones de alta congestión como las anteriores. Los clientes, además de realizar las mismas actividades y generar un tráfico de igual magnitud, descargarán correos electrónicos en sus bandejas de entrada personales y navegarán por la web, escuchando música en línea, viendo videos o descargando archivos en caso de que lo necesiten. Para ello tendremos más en cuenta los resultados obtenidos para el tráfico de salida de la interfaz, puesto que son paquetes que fluyen de la WAN hacia los clientes. El caso contrario ocurre sólo cuando se realizan peticiones HTTP y de correos hacia los servidores, lo cual resulta siendo mínimo. Por lo tanto, funcionará mejor el encolado a la salida del router y se analizará mejor su desempeño.

### 3.3.3.1 TRÁFICO SIN PRIORIZACIÓN

Gráfica 40

1 Minute Average



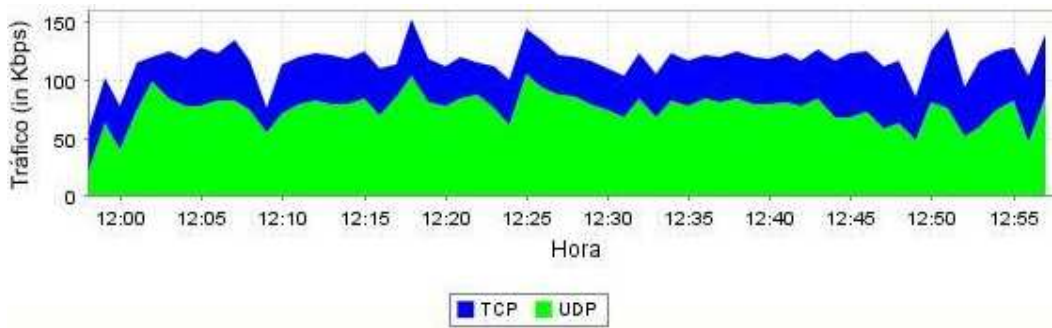
Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	63.62 MB	195.9 Kbps	69.64 Kbps	139.07 Kbps	172.51 Kbps
Tráfico SALIDA	53.26 MB	152.38 Kbps	54.87 Kbps	116.42 Kbps	139.18 Kbps

Durante esta prueba de laboratorio se somete la WAN al mismo tráfico constante generado anteriormente, con la diferencia de que se agregó tráfico de protocolos HTTP y POP3 característicos en una red con acceso a internet. La gráfica general obtenida muestra un tráfico de entrada mayor de 63.62 MB con respecto al de salida, que fue de 53.26 MB. También se obtuvieron velocidades promedio de 139.07 Kbps y 116.42 Kbps respectivamente.

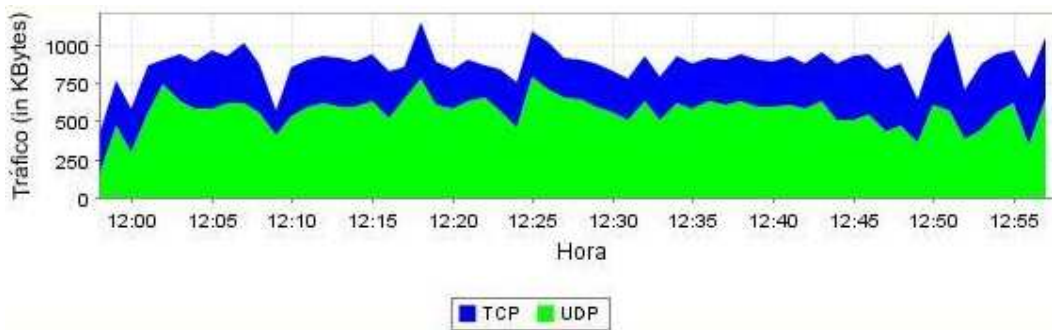
Gráfica 41

TRAFICO DE ENTRADA			TRAFICO DE SALIDA		
Aplicación	Tráfico (64.68 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (53.26 MB)	Porcentaje de tráfico total
UDP_App	57.83 MB	89%	TCP_App	19.05 MB	36%
mysql	1.89 MB	3%	UDP_App	18.66 MB	35%
http	1.17 MB	2%	http	11.77 MB	22%
TCP_App	378.71 KB	1%	pop3	1.78 MB	3%
bootps	377.41 KB	1%	mysql	762.26 KB	1%
netbios-ns	366.13 KB	1%	smtp	365.9 KB	1%
smtp	328.69 KB	1%	ftp	3.48 KB	<1%
pop3	186.97 KB	<1%	Unaccounted	850.98 KB	2%
domain	13.77 KB	<1%			
netbios-dgm	11.07 KB	<1%			
ftp	2.68 KB	<1%			
orbix-config	144.00 Bytes	<1%			
Oracle	144.00 Bytes	<1%			
skinny	136.00 Bytes	<1%			
Unaccounted	2.1 MB	3%			

Gráfica 42



Gráfica 43



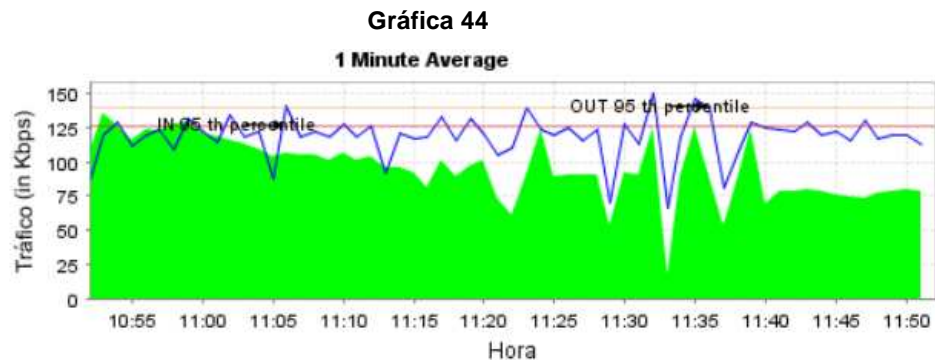
La gráfica 41 muestra un porcentaje de tráfico total de entrada del protocolo UDP abrumador con respecto a los demás protocolos y servicios, que fue de 89%. Para el tráfico de salida se obtuvo un porcentaje de 39% generado por puertos aleatorios utilizados por FTP en modo pasivo, un porcentaje de 35% generado por UDP, 22% generado por HTTP y 3% generado por POP3. Esto indica una gran congestión en el router y la falta de políticas de priorización, ya que los paquetes fueron atendidos tal como iban llegando al dispositivo.

En las posteriores gráficas de distribución de protocolos para el tráfico de salida se presentó un comportamiento similar entre los protocolos TCP y UDP a todo lo largo del tiempo de monitoreo. La única diferencia que se observa es que para TCP se presentaron mayores tasas de velocidad y volumen que para UDP, dado que el tráfico de audio y video generado por el servidor del juego hacia los clientes es siempre menor al tráfico que representan las descargas de archivos, navegación web y descargas de correos.

### 3.3.3.2 HTTP IMPLEMENTANDO PQ

En esta prueba se configuró PQ sobre la interfaz fastethernet 0/1 del router, dando prioridad alta a aquel tráfico de tipo TCP de puerto 80 que correspondería al protocolo HTTP. De esta manera se priorizan todas aquellas peticiones que se realicen al servidor web y las descargas de texto, imágenes, videos y música que se efectúen al visualizar un sitio.

```
router#enable
router#configure terminal
router(config)#priority-list 1 protocol ip high tcp 80
router(config)#interface fastethernet 0/1
router(config-if)#priority-group 1
```



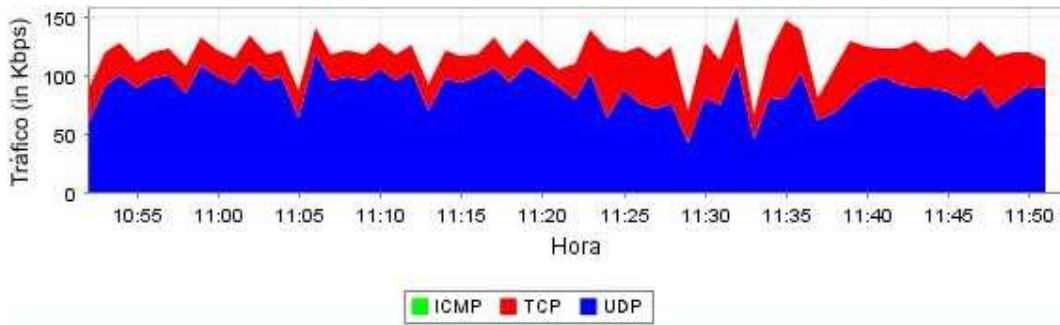
Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	42.83 MB	134.62 Kbps	16.84 Kbps	93.62 Kbps	125.82 Kbps
Tráfico SALIDA	53.25 MB	150.31 Kbps	65.79 Kbps	116.41 Kbps	139.46 Kbps

En la gráfica de tráfico general obtenemos un mayor volumen y tasa de velocidad promedio para el tráfico de salida, con valores de 116.41 Kbps y 53.25 MB respectivamente. Esto debido a que en las pruebas dicho tráfico de salida aumentó al realizar descargas del servidor web, visualizar videos, escuchar música y consultar el correo electrónico. Además se priorizó el tráfico de protocolo HTTP, el cual fue de gran volumen.

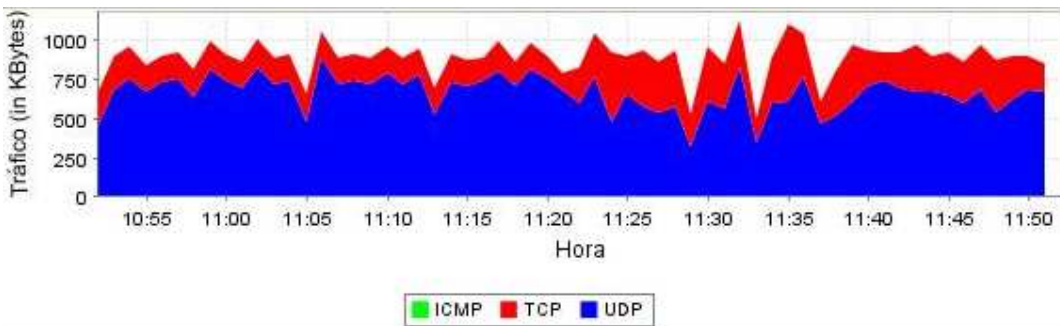
Gráfica 45

TRAFICO DE ENTRADA			TRAFICO DE SALIDA		
Aplicación	Tráfico (43.43 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (54.18 MB)	Porcentaje de tráfico total
UDP_App	38.03 MB	88%	http	18.7 MB	35%
http	2.06 MB	5%	TCP_App	16.19 MB	30%
mysql	868.62 KB	2%	UDP_App	13.59 MB	25%
pop3	409.72 KB	1%	pop3	4.2 MB	8%
TCP_App	314.02 KB	1%	mysql	354.58 KB	1%
smtp	268.36 KB	1%	smtp	191.37 KB	<1%
netbios-ns	254.9 KB	1%	icmp	3.63 KB	<1%
domain	19.52 KB	<1%	ftp	1.74 KB	<1%
netbios-dgm	9.81 KB	<1%	Unaccounted	930.76 KB	2%
ftp	1.35 KB	<1%			
snmp	350.00 Bytes	<1%			
icmp	180.00 Bytes	<1%			
orbix-locator	144.00 Bytes	<1%			
souiseek	144.00 Bytes	<1%			
microsoft-ds	144.00 Bytes	<1%			
Unaccounted	1.18 MB	3%			

Gráfica 46



Gráfica 47



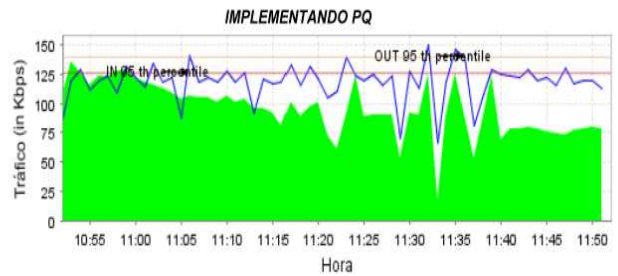
La gráfica 45 muestra los porcentajes de tráfico por cada protocolo a la entrada y salida del router una vez hecho el encolado. Como podemos observar, a la entrada se obtuvo un 5% (2.06 MB) de tráfico HTTP correspondiente a las peticiones GET que se hicieron al servidor web, lo cual representa una carga muy mínima pero que fue uno de los mayores volúmenes del grupo de protocolos TCP. A la salida del



router se obtuvo un 35% de tráfico HTTP, el cual fue mucho mayor al del resto de protocolos. Todo esto evidencia el funcionamiento de PQ y la priorización de este tipo de tráfico.

Las gráficas 46 y 47 muestran cómo a la salida del router se obtuvieron tasas de transferencia de 150 Kbps y volúmenes de 1 MB por parte del grupo de protocolos TCP, el cual se mantuvo mayor en todo momento debido al aumento de congestión y dado que el servidor de audio y video siempre generará un tráfico menor hacia los clientes que de los clientes hacia el servidor.

### No PQ versus PQ implementado sobre HTTP



Categoría	Total	Máx	Min	Prom	95th Percentile	Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	63.62 MB	195.9 Kbps	69.64 Kbps	139.07 Kbps	172.51 Kbps	Tráfico ENTRADA	42.83 MB	134.62 Kbps	16.84 Kbps	93.62 Kbps	125.82 Kbps
Tráfico SALIDA	53.26 MB	152.38 Kbps	54.87 Kbps	116.42 Kbps	139.18 Kbps	Tráfico SALIDA	53.25 MB	150.31 Kbps	65.79 Kbps	116.41 Kbps	139.46 Kbps

Realizando una comparación general entre el comportamiento de la red cuando se ha implementado PQ sobre la interfaz del router y cuando no se implementa encolado de prioridad, podemos afirmar lo siguiente:

- Cuando se implementó PQ en este caso el tráfico de salida va a ser mayor que el tráfico de entrada, a diferencia de cuando no se realizó ningún tipo de priorización.
- El volumen del tráfico de entrada implementando PQ es mucho menor al volumen del tráfico de entrada sin implementación.



- El volumen de tráfico de salida sin implementar PQ es prácticamente igual al volumen de tráfico de salida implementando PQ.
- La velocidad promedio del tráfico de entrada sin implementar PQ es mucho mayor a la del tráfico de salida, caso contrario ocurre cuando se implementó el algoritmo.
- La velocidad promedio del tráfico de entrada implementando PQ es menor a cuando no se efectuó priorización. En cambio para el tráfico de salida, la velocidad promedio fue prácticamente igual cuando se implementó PQ que al no configurarse el algoritmo.

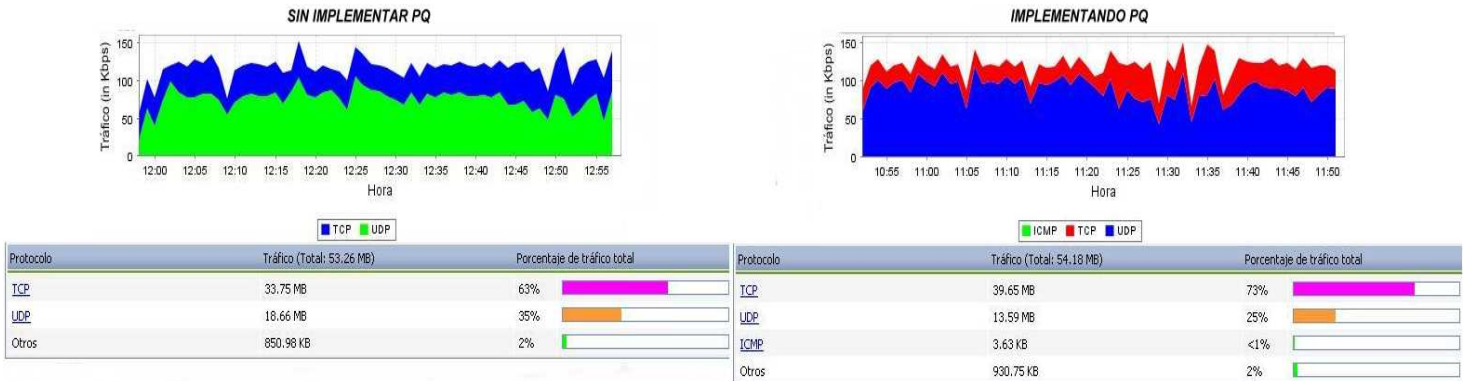
### Porcentaje de tráfico de salida no PQ versus PQ implementado sobre HTTP

SIN IMPLEMENTAR PQ			IMPLEMENTANDO PQ		
Aplicación	Tráfico (53.26 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (54.18 MB)	Porcentaje de tráfico total
TCP_App	19.05 MB	36%	http	18.7 MB	35%
UDP_App	18.66 MB	35%	TCP_App	16.19 MB	30%
http	11.77 MB	22%	UDP_App	13.59 MB	25%
pop3	1.78 MB	3%	pop3	4.2 MB	8%
mysql	762.26 KB	1%	mysql	354.58 KB	1%
smtp	365.9 KB	1%	smtp	191.37 KB	<1%
ftp	3.48 KB	<1%	icmp	3.63 KB	<1%
Unaccounted	850.98 KB	2%	ftp	1.74 KB	<1%
			Unaccounted	930.76 KB	2%

Como se aclaró anteriormente, para el caso de priorización de estos protocolos se consideró en mayor medida el tráfico de salida dado que su volumen mayor fluye de los servidores hacia los clientes, y para la entrada solo se realizan peticiones que no representan una gran carga para el router. Por lo tanto, funcionará mejor el encolado a la salida del router y se analizará mejor su desempeño.

Al observar la tabla comparativa, vemos que al implementar PQ se obtuvo un porcentaje de tráfico mayor del protocolo HTTP (35%) respecto al que se obtuvo al no realizarse ningún tipo de priorización (22%). Además de esto, cuando se implementó PQ fue el porcentaje de tráfico más alto de todos.

## No PQ vs PQ implementado sobre HTTP: diferenciación por protocolos



Al comparar las dos gráficas pueden notarse los siguientes comportamientos:

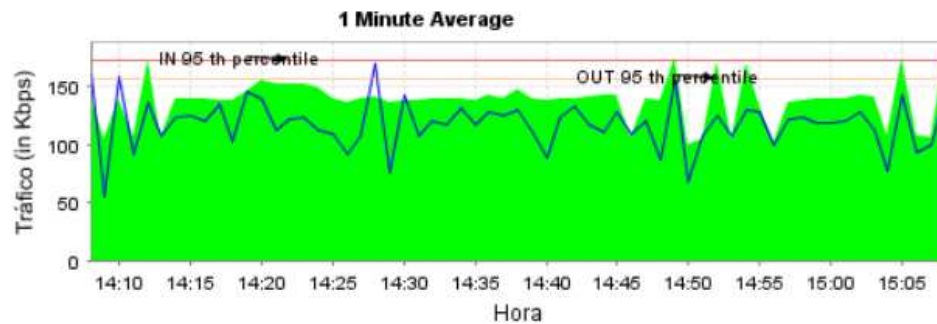
- El porcentaje del tráfico total generado por el grupo de protocolos TCP cuando se ha implementado PQ (73%) es mayor al del tráfico sin priorización (63%). Por otro lado, el porcentaje de tráfico total correspondiente a los protocolos UDP (25%) fue menor al del tráfico sin priorización (35%).
- En este caso se observan cifras que alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo.

### 3.3.3.3 POP IMPLEMENTANDO PQ

En esta prueba se configuró PQ sobre la interfaz fastethernet 0/1 del router, dando prioridad alta a aquel tráfico de tipo TCP de puerto 110 que correspondería al protocolo POP3. De esta manera se priorizó toda aquella actividad consistente en inicio de sesión en el servidor POP3, peticiones y posteriores descargas de correos a la correspondiente bandeja de entrada personal del usuario o cliente.

```
router#enable
router#configure terminal
router(config)#priority-list 1 protocol ip high tcp 110
router(config)#interface fastethernet 0/1
router(config-if)#priority-group 1
```

Gráfica 48



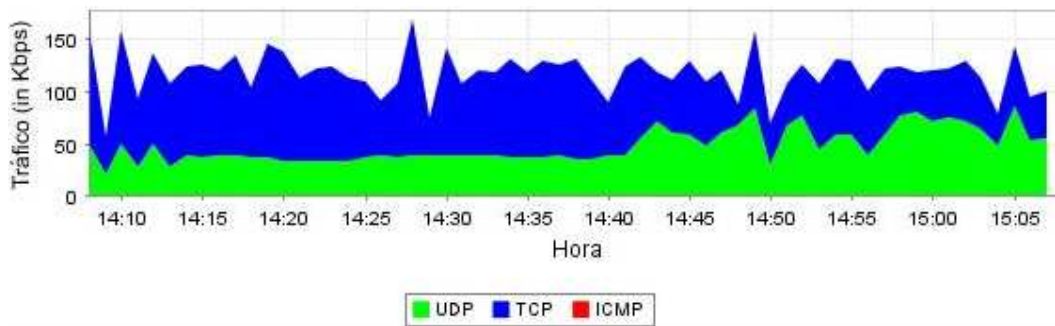
Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	62.89 MB	179.25 Kbps	100.35 Kbps	137.48 Kbps	172.87 Kbps
Tráfico SALIDA	53.87 MB	168.6 Kbps	56.02 Kbps	117.74 Kbps	157.12 Kbps

Durante este monitoreo se obtuvo resultados muy similares al caso en el que no se configuró ningún algoritmo, como podemos observar en la gráfica general. El tráfico de entrada fue mayor al de salida, tanto en volumen como en velocidad promedio. Esto indica que al priorizar el protocolo POP3, que representa un tráfico poco voluminoso, no se va a afectar en gran medida el tráfico de salida y por lo tanto no va a resultar mayor como en el caso anterior en que se priorizó HTTP.

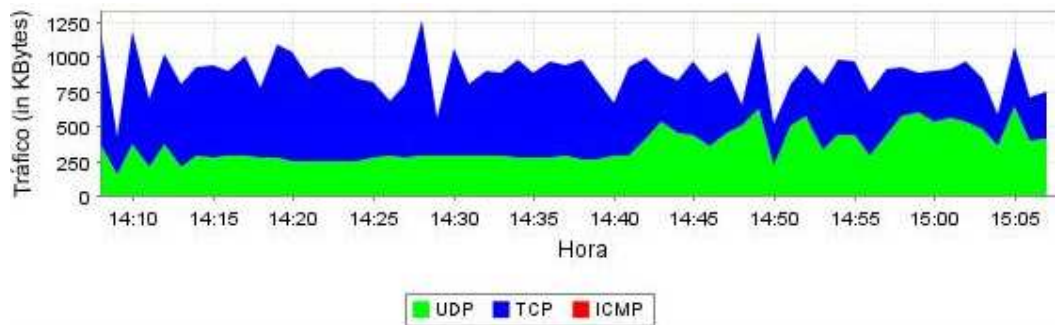
Gráfica 49

TRAFICO DE ENTRADA			TRAFICO DE SALIDA		
Aplicación	Tráfico (63.71 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (53.87 MB)	Porcentaje de tráfico total
UDP_App	58.22 MB	91%	UDP_App	21.45 MB	40%
http	1.09 MB	2%	http	15.65 MB	29%
mysql	1.04 MB	2%	TCP_App	14.43 MB	27%
bootps	374.28 KB	1%	pop3	610.16 KB	1%
TCP_App	296.44 KB	<1%	mysql	412.36 KB	1%
netbios-ns	232.59 KB	<1%	smtp	243.23 KB	<1%
smtp	195.48 KB	<1%	ftp	790.00 Bytes	<1%
pop3	67.84 KB	<1%	icmp	240.00 Bytes	<1%
netbios-dgm	11.07 KB	<1%	Unaccounted	1.06 MB	2%
domain	6.96 KB	<1%			
ftp	1.3 KB	<1%			
icmp	240.00 Bytes	<1%			
jun_ejb	144.00 Bytes	<1%			
Unaccounted	2.15 MB	3%			

Gráfica 50



Gráfica 51

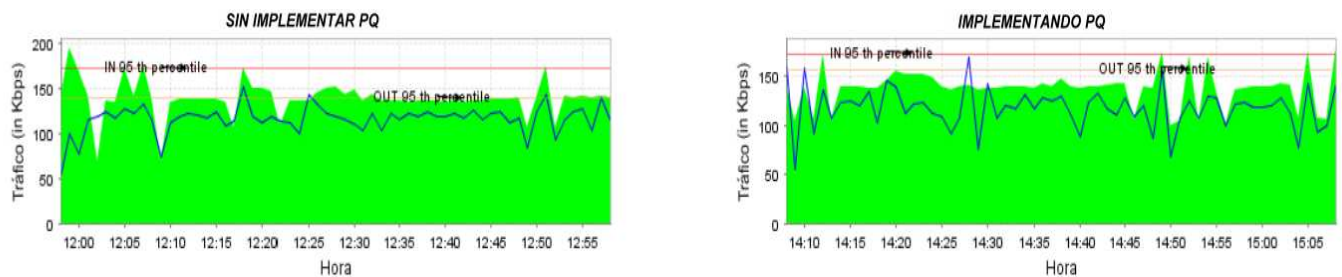


En la gráfica 49 podemos observar que el volumen de tráfico de entrada atendido y generado por el protocolo POP3 fue de 67.84 Kb, que corresponde al inicio de sesión en el servidor. Y para la salida del router se obtuvo un volumen de este mismo protocolo de 610.16 Kb. Dado que en otras pruebas se obtuvieron volúmenes mayores de este protocolo sin priorizarlo, y habiendo sido generado para igual número de clientes en las mismas condiciones de congestión, al parecer es de características muy variables dependiendo de momentos críticos de congestión en

la red que pueden presentarse en algunos casos y en otros no. Por lo tanto, no podemos afirmar que PQ priorice eficientemente este tipo de protocolo. Cabe aclarar que esto solo indica grandes demoras o retardos por unidad de tiempo en la entrega de paquetes o correos a la bandeja de entrada de cada destinatario, mas no la pérdida de ellos.

Las gráficas de distribución de protocolos para el tráfico de salida muestran valores de velocidad y volumen muy superiores para los protocolos TCP, alcanzando en algunos casos picos de más 150 Kbps y 1.2 MB. Dichas tasas de transferencia mostraron un comportamiento muy similar a cuando no se implementó ningún algoritmo de encolado.

### No PQ versus PQ implementado sobre POP3



Categoría	Total	Máx	Min	Prom	95th Percentile	Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	63.62 MB	195.9 Kbps	69.64 Kbps	139.07 Kbps	172.51 Kbps	Tráfico ENTRADA	62.89 MB	179.25 Kbps	100.35 Kbps	137.48 Kbps	172.87 Kbps
Tráfico SALIDA	53.26 MB	152.38 Kbps	54.87 Kbps	116.42 Kbps	139.18 Kbps	Tráfico SALIDA	53.87 MB	168.6 Kbps	56.02 Kbps	117.74 Kbps	157.12 Kbps

Realizando una comparación general entre el comportamiento de la red cuando se ha implementado PQ sobre la interfaz del router y cuando no se implementa encolado de prioridad, podemos afirmar lo siguiente:

- Cuando se implementó PQ en este caso el tráfico de salida va a ser menor que el tráfico de entrada, al igual que cuando no se realizó ningún tipo de priorización.
- El volumen del tráfico de entrada implementando PQ es un poco mayor al volumen del tráfico de entrada sin implementación.

- El volumen de tráfico de salida sin implementar PQ es un poco menor al volumen de tráfico de salida implementando PQ.
- La velocidad promedio del tráfico de entrada sin implementar PQ es mucho mayor a la del tráfico de salida, al igual que cuando se implementó el algoritmo.
- Los valores de velocidad promedio del tráfico de entrada para ambos casos difieren en cantidades muy mínimas, lo que reafirma un comportamiento muy similar.

### Porcentaje de tráfico de salida no PQ versus PQ implementado sobre POP3

SIN IMPLEMENTAR PQ			IMPLEMENTANDO PQ		
Aplicación	Tráfico (53.26 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (53.87 MB)	Porcentaje de tráfico total
TCP_App	19.05 MB	36%	UDP_App	21.45 MB	40%
UDP_App	18.68 MB	35%	http	15.65 MB	29%
http	11.77 MB	22%	TCP_App	14.43 MB	27%
pop3	1.78 MB	3%	pop3	610.16 KB	1%
mysql	782.26 KB	1%	mysql	412.36 KB	1%
smtp	365.9 KB	1%	smtp	243.23 KB	<1%
ftp	3.48 KB	<1%	ftp	790.00 Bytes	<1%
Unaccounted	850.98 KB	2%	icmp	240.00 Bytes	<1%
			Unaccounted	1.06 MB	2%

En esta tabla comparativa puede observarse cómo se obtiene un tráfico de salida mayor para POP3 cuando no se configuró ningún algoritmo de encolado. Para este caso el porcentaje de tráfico total fue de 3% y el volumen de 1.78 MB, en contraste con el caso donde se implementó PQ donde se obtuvieron valores de 1% y 610.16 KB. Como se dijo anteriormente, esto demuestra que el algoritmo PQ no es muy útil para priorizar este tipo de tráfico durante el mismo tiempo de monitoreo. Si bien es cierto se presentan retardos durante momentos de congestión y no son reducidos adecuadamente, puede que el encolado pueda mostrar buen desempeño al evitar la pérdida de paquetes de protocolo POP3.



## No PQ vs PQ implementado sobre POP3: diferenciación por protocolos



Protocolo	Tráfico (Total: 53.26 MB)	Porcentaje de tráfico total
TCP	33.75 MB	63%
UDP	18.66 MB	35%
Otros	650.98 KB	2%

Protocolo	Tráfico (Total: 53.87 MB)	Porcentaje de tráfico total
TCP	31.35 MB	58%
UDP	21.45 MB	40%
ICMP	240,00 Bytes	<1%
Otros	1.06 MB	2%

Al comparar las dos gráficas pueden notarse los siguientes comportamientos:

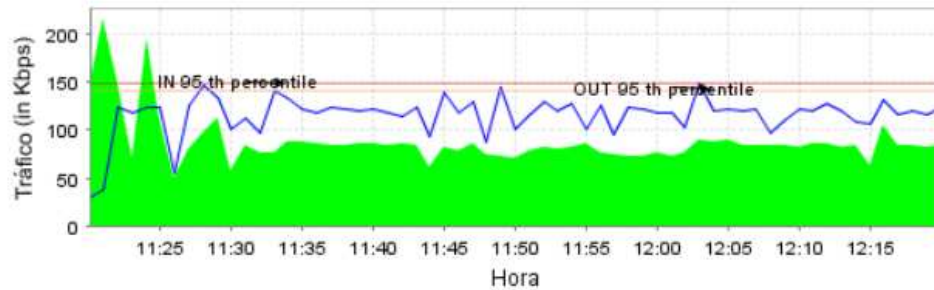
- El porcentaje del tráfico total generado por el grupo de protocolos TCP cuando se ha implementado PQ (58%) es menor al del tráfico sin priorización (63%). Por otro lado, el porcentaje de tráfico total correspondiente a los protocolos UDP (40%) fue mayor al del tráfico sin priorización (35%).
- En este caso se observan cifras que alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo. En el caso de TCP las tasas de transferencia fueron mayores cuando se implementó PQ.

### 3.3.3.4 HTTP IMPLEMENTANDO CQ

Para esta práctica de laboratorio se configura el algoritmo Custom Queueing (CQ) para priorizar aquel tráfico de tipo TCP de puerto 80 que correspondería al protocolo HTTP. Para su configuración se define una cola personalizada 1 donde se atenderán en primera instancia los paquetes del protocolo TCP puerto 80, y una cola por defecto 5 que atenderá por defecto los demás paquetes que lleguen al router.

```
router#enable
router#configure terminal
router(config)#interface fastethernet 0/1
router(config-if)#custom-queue-list 1
router(config-if)#exit
router(config)#queue-list 1 protocol ip 1 tcp 80
router(config)#queue-list 1 default 5
```

**Gráfica 52**  
**1 Minute Average**



Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	39.96 MB	215.66 Kbps	49.94 Kbps	87.35 Kbps	147.89 Kbps
Tráfico SALIDA	52.92 MB	148.81 Kbps	30.64 Kbps	115.68 Kbps	141.56 Kbps

Durante esta prueba se obtuvo un tráfico de salida mucho mayor al de entrada, tal como sucedió cuando se priorizó utilizando PQ. En este caso el volumen de tráfico de salida fue de 52.92 MB, con una velocidad promedio de 115.68 Kbps. Mientras

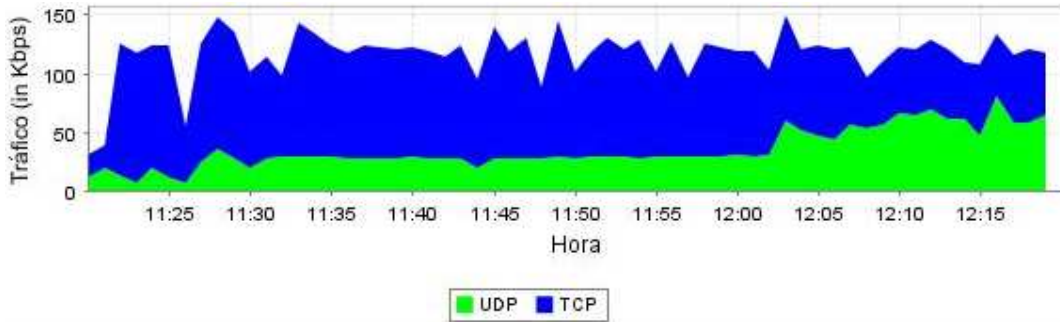


que el tráfico de entrada presentó valores de 39.98 MB y 87.3 Kbps, respectivamente. Dicho esto se puede concluir que al priorizar el protocolo HTTP aumentan considerablemente las tasas de transferencia para la operación de salida de la interfaz del router respecto al de entrada, debido a que algoritmo demuestra un encolado eficiente para el protocolo en cuestión.

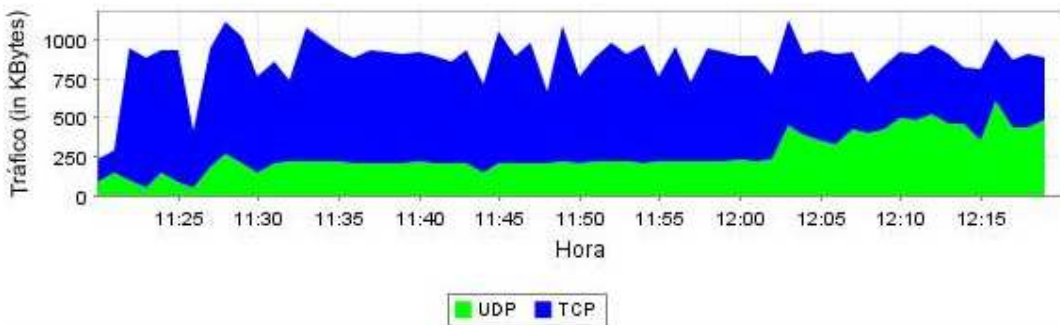
**Gráfica 53**

TRAFICO DE ENTRADA			TRAFICO DE SALIDA		
Aplicación	Tráfico (41.09 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (52.92 MB)	Porcentaje de tráfico total
UDP_App	34.29 MB	83%	http	20.08 MB	38%
http	2.01 MB	5%	UDP_App	15.73 MB	30%
mysql	1.69 MB	4%	TCP_App	15.14 MB	29%
bootps	382.4 KB	1%	mysql	881.7 KB	1%
netbios-ns	337.19 KB	1%	smtp	188.5 KB	<1%
TCP_App	298.46 KB	1%	pop3	169.35 KB	<1%
smtp	296.26 KB	1%	ftp	3.69 KB	<1%
pop3	22.09 KB	<1%	Unaccounted	921.42 KB	2%
netbios-dgm	9.85 KB	<1%			
ftp	3.81 KB	<1%			
domain	3.19 KB	<1%			
microsoft-ds	432.00 Bytes	<1%			
snmp	420.00 Bytes	<1%			
kazaa	96.00 Bytes	<1%			
Unaccounted	1.75 MB	4%			

**Gráfica 54**



**Gráfica 55**

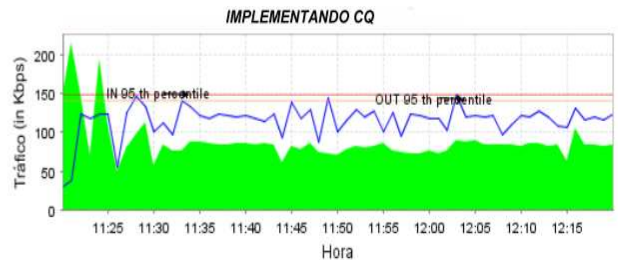
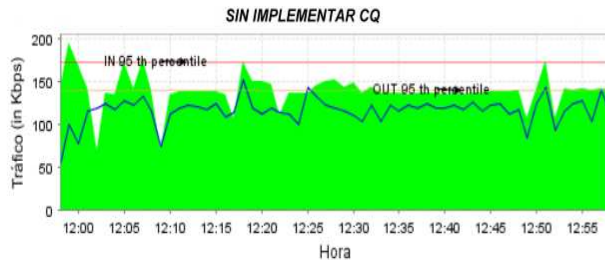


La gráfica 53 permite observar los porcentajes de tráfico total de entrada y salida para cada protocolo de aplicación y servicio. El volumen de tráfico de entrada generado por HTTP fue de 2.01 MB, el cual corresponde a todas aquellas peticiones GET realizadas al servidor web para visualizar contenidos. A pesar de que representa muy poco volumen, fue el mayor entre todos los protocolos TCP.

Para la operación de salida de la interfaz se observa que HTTP generó un porcentaje de tráfico total de 38%, con un valor de 20.08 MB. Lo cual evidencia su eficiente encolado y priorización respecto a los demás tipos de tráfico.

En las gráficas de distribución de protocolos para el tráfico de salida, UDP muestra un comportamiento constante y tasas de transferencia mucho más bajas que el protocolo TCP, el cual mostró valores muy altos de uso del canal en todo el tiempo de monitoreo. Incluso llegó a mantener los niveles en aquellas ocasiones en que el protocolo UDP aumentaba su actividad y congestión.

### No CQ versus CQ implementado sobre HTTP



Categoría	Total	Máx	Min	Prom	95th Percentile	Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	63.62 MB	195.9 Kbps	69.64 Kbps	139.07 Kbps	172.51 Kbps	Tráfico ENTRADA	39.96 MB	215.66 Kbps	49.94 Kbps	87.35 Kbps	147.89 Kbps
Tráfico SALIDA	53.26 MB	152.38 Kbps	54.87 Kbps	116.42 Kbps	139.18 Kbps	Tráfico SALIDA	52.92 MB	148.81 Kbps	30.64 Kbps	115.68 Kbps	141.56 Kbps

Realizando una comparación general entre el comportamiento de la red cuando se ha implementado CQ sobre la interfaz del router y cuando no se implementa encolado de prioridad, podemos afirmar lo siguiente:

- Cuando se implementó CQ en este caso el tráfico de salida va a ser mayor que el tráfico de entrada, a diferencia de cuando no se realizó ningún tipo de priorización.

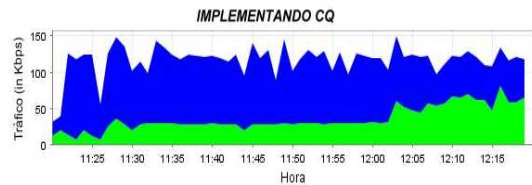
- El volumen del tráfico de entrada implementando CQ es mucho menor al volumen del tráfico de entrada sin implementación.
- El volumen de tráfico de salida sin implementar CQ es prácticamente igual al volumen de tráfico de salida implementando CQ.
- La velocidad promedio del tráfico de entrada sin implementar CQ es mucho mayor a la del tráfico de salida, caso contrario ocurre cuando se implementó el algoritmo.
- La velocidad promedio del tráfico de entrada implementando CQ es menor a cuando no se efectuó priorización. En cambio para el tráfico de salida, la velocidad promedio fue prácticamente igual cuando se implementó CQ que al no configurarse el algoritmo.

### Porcentaje de tráfico de salida no CQ versus CQ implementado sobre HTTP

SIN IMPLEMENTAR CQ			IMPLEMENTANDO CQ		
Aplicación	Tráfico (53.26 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (52.92 MB)	Porcentaje de tráfico total
TCP_App	19.05 MB	36%	http	20.08 MB	38%
UDP_App	18.66 MB	35%	UDP_App	15.73 MB	30%
http	11.77 MB	22%	TCP_App	15.14 MB	29%
pop3	1.78 MB	3%	mysql	681.7 KB	1%
mysql	762.26 KB	1%	smtp	188.5 KB	<1%
smtp	365.9 KB	1%	pop3	169.35 KB	<1%
ftp	3.48 KB	<1%	ftp	3.69 KB	<1%
Unaccounted	850.98 KB	2%	Unaccounted	921.42 KB	2%

Como se puede observar, al implementar el algoritmo CQ y priorizar HTTP se obtuvo un porcentaje de tráfico total de este protocolo de 38% y un volumen de 20.08 MB, respecto a solo un 22% o 11.77 MB que se obtuvo sin implementar ningún mecanismo de priorización. Una vez más, fue el mayor tráfico que se obtuvo entre todos los demás protocolos y servicios.

## No CQ vs CQ implementado sobre HTTP: diferenciación por protocolos



Protocolo	Tráfico (Total: 53.26 MB)	Porcentaje de tráfico total	Protocolo	Tráfico (Total: 52.92 MB)	Porcentaje de tráfico total
TCP	33.75 MB	63%	TCP	36.27 MB	69%
UDP	18.66 MB	35%	UDP	15.73 MB	30%
Otros	850.98 KB	2%	Otros	921.42 KB	2%

Al comparar las dos gráficas pueden notarse los siguientes comportamientos:

- El porcentaje del tráfico total generado por el grupo de protocolos TCP cuando se ha implementado CQ (69%) es mayor al del tráfico sin priorización (63%). Por otro lado, el porcentaje de tráfico total correspondiente a los protocolos UDP (30%) fue menor al del tráfico sin priorización (35%).
- En este caso se observan cifras que alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo.

### 3.3.3.5 POP IMPLEMENTANDO CQ

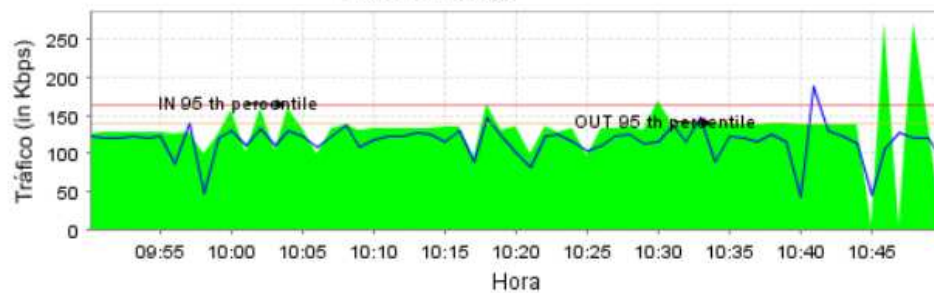
Para esta prueba de laboratorio se configuró el algoritmo Custom Queueing (CQ) para priorizar aquel tráfico de tipo TCP de puerto 110 que correspondería al protocolo POP3. Para su configuración se define una cola personalizada 1 donde se atenderán en primera instancia los paquetes del protocolo TCP puerto 110, y una cola por defecto 5 que atenderá por defecto los demás paquetes que lleguen al router.

```

router#enable
router#configure terminal
router(config)#interface fastethernet 0/1
router(config-if)#custom-queue-list 1
router(config-if)#exit
router(config)#queue-list 1 protocol ip 1 tcp 110
router(config)#queue-list 1 default 5
  
```

**Gráfica 56**

**1 Minute Average**



Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	59.25 MB	272.33 Kbps	3.22 Kbps	129.5 Kbps	162.48 Kbps
Tráfico SALIDA	53.08 MB	186.53 Kbps	43.99 Kbps	116.02 Kbps	138.68 Kbps

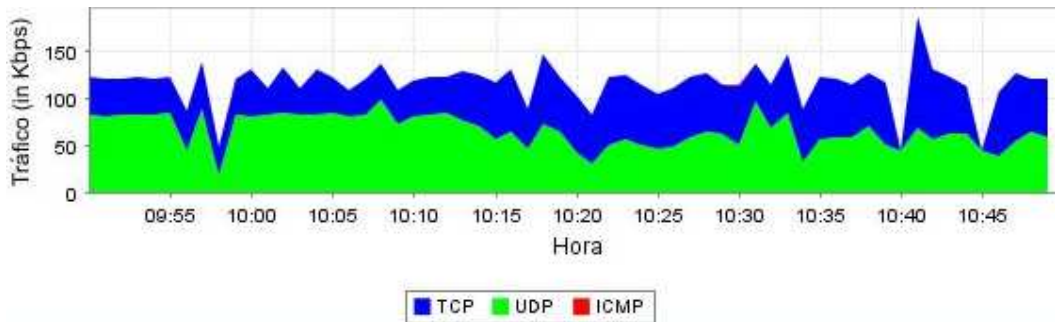
Para este caso los resultados de tráfico general vuelven a mostrar un comportamiento muy parecido al obtenido cuando se utilizó PQ para priorizar este protocolo, o cuando no se configuró ningún algoritmo de encolamiento. El volumen

del tráfico de entrada fue mayor al de salida, con un valor de 59.25 MB y una velocidad promedio de 129.5 Kbps. Lo cual indica que a pesar de configurar el algoritmo, el router no experimenta mucha carga ni se retarda en gran medida el tráfico de entrada. Esto debido al poco tráfico que genera el protocolo POP3 y que además es muy variable, como se mencionaba anteriormente.

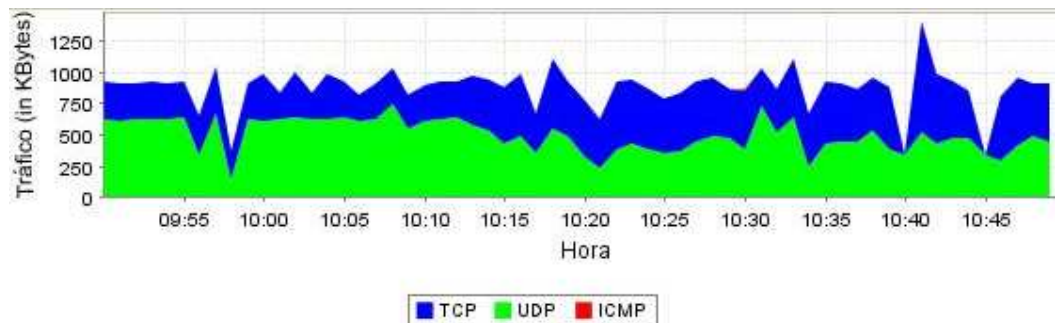
Gráfica 57

TRAFICO DE ENTRADA			TRAFICO DE SALIDA		
Aplicación	Tráfico (64.34 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (53.08 MB)	Porcentaje de tráfico total
UDP_App	55.96 MB	87%	UDP_App	22.79 MB	43%
mysql	950.83 KB	1%	TCP_App	15.54 MB	29%
smtp	931.44 KB	1%	http	12.96 MB	24%
http	365.4 KB	1%	pop3	483.72 KB	1%
bootps	382.4 KB	1%	mysql	380.28 KB	1%
TCP_App	317.96 KB	<1%	smtp	233.6 KB	<1%
netbios-ns	238.44 KB	<1%	icmp	6.34 KB	<1%
pop3	86.35 KB	<1%	ftp	2.93 KB	<1%
netbios-dgm	12.19 KB	<1%	Unaccounted	674.32 KB	1%
domain	9.17 KB	<1%			
ftp	2.25 KB	<1%			
icmp	480,00 Bytes	<1%			
snmp	420,00 Bytes	<1%			
Unaccounted	5.11 MB	8%			

Gráfica 58



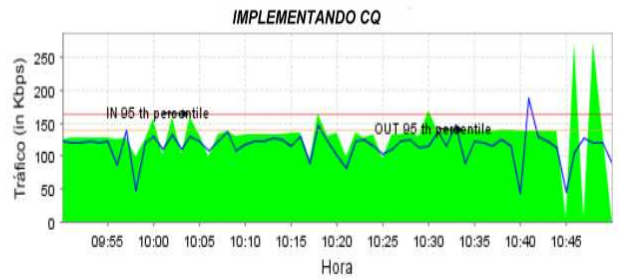
Gráfica 59





En la gráfica 57 se observa que para el tráfico de entrada, POP3 generó un porcentaje de tráfico menor al 1% y un volumen de apenas 66.35 KB, correspondiente al inicio de sesión y peticiones de correos al servidor. Para el tráfico de salida, su porcentaje fue del 1% con un volumen de 483.72 KB, un valor menor incluso al que se obtuvo priorizando con el algoritmo PQ en anteriores pruebas. Las gráficas 58 y 59 muestran un tráfico de salida de TCP no tan mayor al de UDP, alcanzando picos y tasas de transferencia muy similares a las que se obtuvieron al no implementarse ningún algoritmo de encolamiento.

### No CQ versus CQ implementado sobre POP3



Categoría	Total	Máx	Mín	Prom	95th Percentile	Categoría	Total	Máx	Mín	Prom	95th Percentile
Tráfico ENTRADA	63.62 MB	195.9 Kbps	69.64 Kbps	139.07 Kbps	172.51 Kbps	Tráfico ENTRADA	59.25 MB	272.33 Kbps	3.22 Kbps	129.5 Kbps	162.48 Kbps
Tráfico SALIDA	53.26 MB	152.38 Kbps	54.87 Kbps	116.42 Kbps	139.18 Kbps	Tráfico SALIDA	53.08 MB	186.53 Kbps	43.99 Kbps	116.02 Kbps	138.68 Kbps

Realizando una comparación general entre el comportamiento de la red cuando se ha implementado CQ sobre la interfaz del router y cuando no se implementa encolado de prioridad, podemos afirmar lo siguiente:

- Cuando se implementó CQ en este caso el tráfico de salida va a ser menor que el tráfico de entrada, al igual que cuando no se realizó ningún tipo de priorización.
- El volumen del tráfico de entrada implementando CQ es menor al volumen del tráfico de entrada sin implementación.

- El volumen de tráfico de salida sin implementar CQ es un poco mayor al volumen de tráfico de salida implementando CQ.
- La velocidad promedio del tráfico de entrada sin implementar CQ es mucho mayor a la del tráfico de salida, al igual que cuando se implementó el algoritmo.
- La velocidad promedio del tráfico de entrada implementando CQ es menor a la que se presentó sin configurar el algoritmo.

### Porcentaje de tráfico de salida no CQ versus CQ implementado sobre POP3

SIN IMPLEMENTAR CQ			IMPLEMENTANDO CQ		
Aplicación	Tráfico (53.26 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (53.08 MB)	Porcentaje de tráfico total
TCP_App	19.05 MB	36%	UDP_App	22.79 MB	43%
UDP_App	18.66 MB	35%	TCP_App	15.54 MB	29%
http	11.77 MB	22%	http	12.96 MB	24%
pop3	1.78 MB	3%	pop3	483.72 KB	1%
mysql	762.26 KB	1%	mysql	380.28 KB	1%
smtp	365.9 KB	1%	smtp	233.6 KB	<1%
ftp	3.48 KB	<1%	icmp	6.34 KB	<1%
Unaccounted	850.98 KB	2%	ftp	2.93 KB	<1%
			Unaccounted	674.32 KB	1%

En este caso también se observa que se obtiene un volumen de tráfico de salida de POP3 mayor cuando no se implementó ningún algoritmo de encolado, con un valor de 1.78 MB respecto al que se presentó configurando el algoritmo, que fue de 483.72 Kb. Esto deja en evidencia que la naturaleza de este protocolo es muy variable y su encolado no garantiza una buena priorización, aunque si puede evitar pérdidas de paquetes.



## No CQ vs CQ implementado sobre POP3: diferenciación por protocolos



Protocolo	Tráfico (Total: 53.26 MB)	Porcentaje de tráfico total
TCP	33.75 MB	63%
UDP	18.66 MB	35%
Otros	850.98 KB	2%

Protocolo	Tráfico (Total: 53.08 MB)	Porcentaje de tráfico total
TCP	29.61 MB	56%
UDP	22.79 MB	43%
ICMP	6.34 KB	<1%
Otros	674.32 KB	1%

Al comparar las dos gráficas pueden notarse los siguientes comportamientos:

- El porcentaje del tráfico total generado por el grupo de protocolos TCP cuando se ha implementado CQ (56%) es menor al del tráfico sin priorización (63%). Por otro lado, el porcentaje de tráfico total correspondiente a los protocolos UDP (43%) fue mayor al del tráfico sin priorización (35%).
- En este caso se observan cifras que alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo.

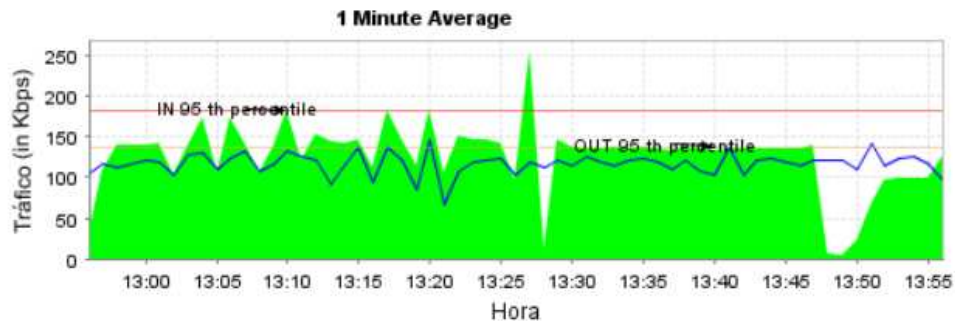
### 3.3.3.6 IMPLEMENTACIÓN DE WFQ PARA TODOS LOS PROTOCOLOS

Para este monitoreo se configuró WFQ sobre la interfaz fastethernet 0/1 del router, clasificando y priorizando de forma automática aquellos protocolos que el algoritmo determine como sensibles a retardos. Con la diferencia en este caso que habrá más congestión debido al tráfico generado por los protocolos HTTP y POP3.

Para ello se definen los mismos parámetros de las pruebas anteriores, un tamaño para cada cola de 64 mensajes permitidos y un número máximo de 512 colas dinámicas que puede crear el algoritmo, como se puede ver a continuación:

```
router#enable
router#configure terminal
router(config)#interface fastethernet 0/1
router(config-if)#fair-queue 64 512
```

Gráfica 60



Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	57.41 MB	254.57 Kbps	4.24 Kbps	125.5 Kbps	180.9 Kbps
Tráfico SALIDA	53.47 MB	145.08 Kbps	68.2 Kbps	116.87 Kbps	137.46 Kbps

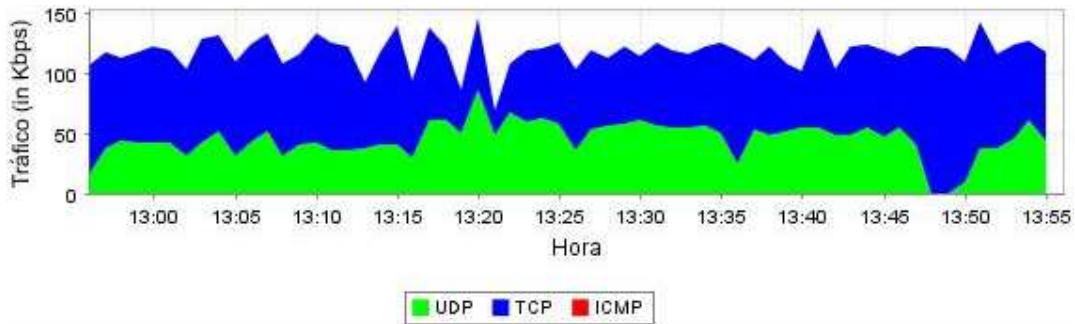
En la gráfica 60 se observó también un comportamiento muy similar al que se presentó cuando no se realizó ningún tipo de priorización, ya que el tráfico de entrada es mucho mayor al de salida. Esto debido a que WFQ es un algoritmo que prioriza automáticamente aquellos protocolos que detecte como sensibles a retardos, y al ser el tráfico de audio y video el de mayor volumen a la entrada de la

interfaz del router y de transmisión en tiempo real, siempre lo va a encolar y atender primero. Como consecuencia, el volumen y velocidad promedio del tráfico de entrada va a resultar mayor en todos los casos que se utilice este algoritmo. Por ello, para analizar su funcionamiento para cada tipo de tráfico o protocolo de aplicación, se estudiaron las siguientes gráficas:

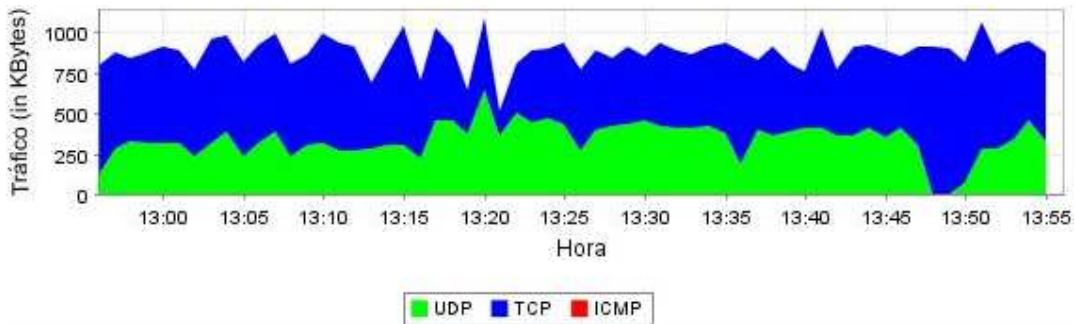
**Gráfica 61**

TRAFICO DE ENTRADA			TRAFICO DE SALIDA		
Aplicación	Tráfico (58.85 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (53.47 MB)	Porcentaje de tráfico total
UDP_App	53.1 MB	90%	UDP_App	20.42 MB	38%
mysql	1.0 MB	2%	TCP_App	19.07 MB	36%
smtp	707.22 KB	1%	http	12.3 MB	23%
http	639.68 KB	1%	mysql	403.53 KB	1%
TCP_App	371.97 KB	1%	pop3	317.49 KB	1%
bootps	360.58 KB	1%	smtp	218.61 KB	<1%
netbios-ns	227.67 KB	<1%	icmp	6.0 KB	<1%
pop3	40.87 KB	<1%	ftp	1.6 KB	<1%
netbios-dgm	10.06 KB	<1%	netbios-ns	468.00 Bytes	<1%
domain	4.9 KB	<1%	Unaccounted	722.82 KB	1%
ftp	1.26 KB	<1%			
icmp	480.00 Bytes	<1%			
snmp	420.00 Bytes	<1%			
HTTP-Proxy	144.00 Bytes	<1%			
Unaccounted	2.37 MB	4%			

**Gráfica 62**



**Gráfica 63**



Como se dijo anteriormente, WFQ prioriza como primera instancia aquellos tráficos que requieran conversaciones en tiempo real, como el de audio y video que en este caso es de protocolo UDP. Por ello, para este protocolo se obtuvo un porcentaje de tráfico de entrada de 90% respecto a los demás, con un volumen de 53.1 MB. Los protocolos HTTP y POP3 generaron volúmenes de tráfico de 639.68 Kb y 40.87 Kb, cifras algo menores a las obtenidas cuando se implementaron algoritmos de encolado diferentes. A la salida del router, se obtuvo nuevamente un tráfico de UDP mayor a los demás, con porcentaje de 38% y un volumen de 20.42 MB. HTTP generó un tráfico de 12.3 MB, el 23% del tráfico total. Y POP3 generó un volumen de 317.49 Kb, el 1% del total. Con base a esto puede concluirse que WFQ prioriza protocolos que considera más sensibles a retardos que los utilizados por aplicaciones y servicios web, sin retardarlos en gran medida y asignándoles el ancho de banda que les corresponde para manejar de mejor manera la congestión. Las gráficas 62 y 63 muestran un comportamiento muy variable para los protocolos TCP y UDP en el tráfico de salida, siendo el de TCP mayor y estable durante el periodo de monitoreo.

### No WFQ versus WFQ sobre tráfico en general



Categoría	Total	Máx	Min	Prom	95th Percentile	Categoría	Total	Máx	Min	Prom	95th Percentile
Tráfico ENTRADA	63.62 MB	195.9 Kbps	69.64 Kbps	139.07 Kbps	172.51 Kbps	Tráfico ENTRADA	57.41 MB	254.57 Kbps	4.24 Kbps	125.5 Kbps	180.9 Kbps
Tráfico SALIDA	53.26 MB	152.38 Kbps	54.87 Kbps	116.42 Kbps	139.18 Kbps	Tráfico SALIDA	53.47 MB	145.08 Kbps	68.2 Kbps	116.87 Kbps	137.46 Kbps

Realizando una comparación general entre el comportamiento de la red cuando se ha implementado WFQ sobre la interfaz del router y cuando no se implementa encolado de prioridad, se puede afirmar lo siguiente:

- Cuando se implementa WFQ el tráfico de salida va a ser menor que el tráfico de entrada, al igual que cuando no se realizó ningún tipo de priorización.
- El volumen del tráfico de entrada implementando WFQ es menor al volumen del tráfico de entrada sin implementación.
- El volumen de tráfico de salida sin implementar WFQ es prácticamente igual al volumen de tráfico de salida implementando WFQ.
- La velocidad promedio del tráfico de entrada sin implementar WFQ es mucho mayor a la del tráfico de salida, al igual que cuando se implementó el algoritmo.
- La velocidad promedio del tráfico de entrada implementando WFQ es menor a la que se presentó sin configurar el algoritmo.

### Porcentaje de tráfico de salida no WFQ versus WFQ

SIN IMPLEMENTAR WFQ			IMPLEMENTANDO WFQ		
Aplicación	Tráfico (53.26 MB)	Porcentaje de tráfico total	Aplicación	Tráfico (53.47 MB)	Porcentaje de tráfico total
TCP_App	19.05 MB	36%	UDP_App	20.42 MB	38%
UDP_App	18.86 MB	35%	TCP_App	19.07 MB	36%
http	11.77 MB	22%	http	12.3 MB	23%
pop3	1.78 MB	3%	mysql	403.53 KB	1%
mysql	762.26 KB	1%	pop3	317.49 KB	1%
smtp	365.9 KB	1%	smtp	218.61 KB	<1%
ftp	3.48 KB	<1%	icmp	6.0 KB	<1%
Unaccounted	850.98 KB	2%	ftp	1.6 KB	<1%
			netbios-ns	468.00 Bytes	<1%
			Unaccounted	722.82 KB	1%

Como se puede observar en la gráfica, al haber configurado WFQ se obtuvo un porcentaje de tráfico de salida de 23% y volumen de 12.3 MB para HTTP, mayor al que se obtuvo sin priorización que fue de 22% con un volumen de 11.77 MB. En el caso del protocolo POP3 fue todo lo contrario, ya que su porcentaje fue mayor al no haberse configurado ningún algoritmo de encolado.

## No WFQ vs WFQ: diferenciación por protocolos



Protocolo	Tráfico (Total: 53.26 MB)	Porcentaje de tráfico total
TCP	33.75 MB	63%
UDP	18.66 MB	35%
Otros	850.98 KB	2%

Protocolo	Tráfico (Total: 53.47 MB)	Porcentaje de tráfico total
TCP	32.32 MB	60%
UDP	20.42 MB	38%
ICMP	6.0 KB	<1%
Otros	722.82 KB	1%

Al comparar las dos gráficas pueden notarse los siguientes comportamientos:

- El porcentaje del tráfico total generado por el grupo de protocolos TCP cuando se ha implementado WFQ (60%) es menor al del tráfico sin priorización (63%). Por otro lado, el porcentaje de tráfico total correspondiente a los protocolos UDP (38%) fue mayor al del tráfico sin priorización (35%).
- En este caso se observan cifras que alcanzan picos tan altos de uso del canal como los que se presentaron al no configurar dicho algoritmo.



### 3.3.3.7 CUADRO COMPARATIVO GENERAL DE ALGORITMOS DE ENCOLAMIENTO

Análisis de velocidad promedio de tráfico en la LAN									
Protocolos	Sin priorización		Priority Queueing (PQ)		Custom Queueing (CQ)		Weighted Fair Queueing (WFQ)		
	Tráfico de entrada general	Tráfico de salida general	Tráfico de entrada general	Tráfico de salida general	Tráfico de entrada general	Tráfico de salida general	Tráfico de entrada general	Tráfico de salida general	Tráfico de salida general
MYSQL			39.16 Kbps	56.55 Kbps	76.62 Kbps	108.15 Kbps			
SMTP			100.62 Kbps	119.68 Kbps	82.87 Kbps	120.31 Kbps			
FTP	140.28 Kbps	117.15 Kbps	N/A	N/A	N/A	N/A	177.76 Kbps	65.28 Kbps	
IM (UDP)			N/A	N/A	N/A	N/A			
UNREAL (UDP)			153.58 Kbps	114.8 Kbps	182.31 Kbps	75.23 Kbps			
Análisis de velocidad promedio de tráfico en la WAN									
Protocolos	Sin priorización		Priority Queueing (PQ)		Custom Queueing (CQ)		Weighted Fair Queueing (WFQ)		
	Tráfico de entrada general	Tráfico de salida general	Tráfico de entrada general	Tráfico de salida general	Tráfico de entrada general	Tráfico de salida general	Tráfico de entrada general	Tráfico de salida general	Tráfico de salida general
HTTP	139.07 Kbps	116.42 Kbps	93.62 Kbps	116.41 Kbps	87.35 Kbps	115.68 Kbps	125.5 Kbps	116.87 Kbps	
POP3			137.48 Kbps	117.74 Kbps	129.5 Kbps	116.02 Kbps			

• N/A: No aplicable. Esto quiere decir que para el tipo de protocolo o tráfico en cuestión no es necesario aplicar mecanismos de priorización.

### 3.3.3.8 CUADRO COMPARATIVO DE ALGORITMOS DE ENCOLAMIENTO POR PROTOCOLOS

Análisis de volumen tráfico en la LAN										
Protocolos	Sin priorización			Priority Queueing (PQ)			Custom Queueing (CQ)		Weighted Fair Queueing (WFQ)	
	Tráfico de entrada	Tráfico de salida		Tráfico de entrada	Tráfico de salida		Tráfico de entrada	Tráfico de salida	Tráfico de entrada	Tráfico de salida
MYSQL	21.82 MB (24%)	6.54 MB (9%)		7.32 MB (29%)	19.45 MB (50%)		5.2 MB (22%)	15.25 MB (47%)	18.43 MB (21%)	7.45 MB (25%)
SMTP	553.53 KB (1%)	20.0 KB (<1%)		1.05 MB (2%)	1.03 MB (1%)		761.91 KB (1%)	858.42 KB (1%)	1.73 MB (2%)	2.07 MB (7%)
FTP	N/A	N/A		N/A	N/A		N/A	N/A	N/A	N/A
IM (UDP)	N/A	N/A		N/A	N/A		N/A	N/A	N/A	N/A
UNREAL (UDP)	59.24 MB (64%)	34.3 MB (48%)		54.87 MB (75%)	33.0 MB (55%)		54.96 MB (59%)	26.2 MB (76%)	48.94 MB (56%)	19.62 MB (66%)
Análisis de volumen de tráfico en la WAN										
Protocolos	Sin priorización			Priority Queueing (PQ)			Custom Queueing (CQ)		Weighted Fair Queueing (WFQ)	
	Tráfico de entrada	Tráfico de salida		Tráfico de entrada	Tráfico de salida		Tráfico de entrada	Tráfico de salida	Tráfico de entrada	Tráfico de salida
HTTP	1.89 MB (2%)	11.77 MB (22%)		2.06 MB (5%)	18.7 MB (35%)		2.01 MB (5%)	20.08 MB (38%)	639.68 KB (1%)	12.3 MB (23%)
POP3	186.97 KB (<1%)	1.78 MB (3%)		67.84 KB (<1%)	610.16 KB (1%)		66.35 KB (<1%)	483.72 KB (1%)	40.87 KB (<1%)	317.49 KB (1%)

- N/A: No aplicable. Esto quiere decir que para el tipo de protocolo o tráfico en cuestión no es necesario aplicar mecanismos de priorización.



## CONCLUSIONES

► En el ambiente de congestión creado a partir de las aplicaciones utilizadas en las prácticas de laboratorio para evaluar el desempeño de los algoritmos de encolamiento, se observaron los siguientes comportamientos creados a partir de los servicios que circulan por las redes actualmente (SQL, SMTP, FTP, HTTP, IM, POP3, audio y video):

En cuanto a velocidad promedio general se observaron los siguientes comportamientos:

- Cuando se configuró CQ sobre la interfaz del router en la LAN se obtuvo la mayor tasa de velocidad general de salida en todas las pruebas de laboratorio realizadas, con un valor de 120.31 Kbps al priorizar el protocolo SMTP. Además, se obtuvo la mayor tasa de velocidad general de entrada con un valor de 182.31 Kbps al priorizar el protocolo UDP que corresponde a la aplicación de audio y video. Esto debido a que en el caso de CQ el administrador define colas personalizadas para cada tipo de tráfico o protocolo, y el router se ahorra el tiempo y la capacidad de procesamiento para crearlas. Lo que al final se refleja en estos altos valores respecto a los otros algoritmos de encolamiento.
- Cuando se configuró PQ sobre la interfaz del router en la WAN se obtuvo la mayor tasa de velocidad general de entrada en todas las pruebas de laboratorio realizadas, con un valor de 137.48 Kbps al priorizar el protocolo POP3. Así mismo, se obtuvo la mayor tasa de velocidad general de salida con un valor de 117.74 Kbps al priorizar este mismo protocolo. Este comportamiento se debe a que POP3 representa un tráfico de paquetes poco voluminoso, y el router no consume tanta capacidad de procesamiento para analizarlos.

En cuanto a volumen promedio por protocolos de aplicación y servicios se observaron los siguientes comportamientos:

- Para el protocolo SQL se obtuvo un mayor tráfico de entrada configurando el algoritmo WFQ, con un volumen de 18.43 MB. Por otra parte, se obtuvo el mayor tráfico de salida implementando el algoritmo PQ y priorizando este protocolo, con un volumen de 19.45 MB.
- Para el protocolo SMTP se obtuvieron mayores volúmenes de tráfico de entrada y salida configurando el algoritmo WFQ con respecto a los demás, con valores de 1.73 MB y 2.07 MB respectivamente.
- Para el protocolo UDP utilizado por la aplicación de audio y video se presentó un mayor tráfico de entrada configurando el algoritmo CQ, con un volumen de 54.96 MB. Por otro lado, implementando PQ se obtuvo un mayor volumen de tráfico de salida, que fue de 33 MB. Esto debido a que tanto CQ como PQ se utilizaron para priorizar este tipo de tráfico en especial, mientras que WFQ realizó un buen manejo de este protocolo pero tomando en cuenta además otros servicios sensibles a retardos y asignándoles colas y ancho de banda diferente.
- Para el protocolo HTTP se obtuvo un mayor tráfico de entrada configurando el algoritmo PQ, con un volumen de 2.06 MB. Por otra parte, se obtuvo el mayor tráfico de salida implementando el algoritmo CQ y priorizando este protocolo, con un volumen de 20.08 MB.
- Para el protocolo POP3 se obtuvieron mayores volúmenes de tráfico de entrada y salida configurando el algoritmo PQ con respecto a los demás, con valores de 67.84 KB y 610.16 KB respectivamente. Estos resultados, en comparación a cuando no se realizó ningún tipo de priorización, nos permiten afirmar que ningún algoritmo de encolamiento es adaptable a este tipo de

tráfico, ya que no realizan el manejo adecuado de paquetes que requiere este protocolo. No se presentan buenos tiempos de respuesta, aunque puede que cumplan su función de entrega de paquetes evitando la pérdida de ellos.

► Al comparar la eficiencia de cada algoritmo frente a las mismas condiciones de congestión y flujos de datos, analizamos que PQ presenta una ventaja notoria respecto a CQ y WFQ. De los 3 algoritmos, es el único que lleva a cabo su función sin desmejorar en gran medida el desempeño de la WAN. Presenta las mayores tasas de volumen y velocidades de tráfico de entrada y salida durante todas las pruebas de laboratorio. Por otro lado, utilizar WFQ supone dotar de más “inteligencia” al router al ser un algoritmo totalmente automatizado, clasificar y determinar por si solo los flujos de datos en cualquier momento de la congestión y construir las colas en base a esto. Por ello demanda más carga para el router pero menos trabajo para el administrador de red, es menos personalizado y más autónomo. Además presenta un manejo más adecuado particularmente para tráfico de audio y video sin retardarlo en gran medida. Es un algoritmo adaptativo a los cambios eventuales del tráfico en la red.

► Al monitorear protocolos como los utilizados para mensajería instantánea, concluimos que no es necesario utilizar algoritmos de encolado que prioricen estos flujos de paquetes en especial. Son líneas de texto que se intercambian en tiempo real entre emisores y receptores, y dado su poco volumen son datos no sensibles a retardos que llegan en el mismo instante en que son enviados. Una vez que la sesión entre varios usuarios es creada o abierta, la conversación es iniciada y no se cae como pudimos observar en todos los monitoreos que se realizaron.

Si se utilizan aplicaciones de mensajería instantánea para la transferencia de archivos o conferencias de audio y video, el comportamiento de los datos ya pierde esta naturaleza para asemejarse a la de cualquier otro tipo de tráfico monitoreado, como el generado por FTP o UDP. En este caso si se hace factible el uso de algoritmos de encolamiento para su tratamiento.

► Los mismos resultados se presentan al analizar el comportamiento de protocolos utilizados por Telnet. Durante las pruebas de laboratorio observamos que dicho

servicio no requiere una gran actividad en la red para conectarse remotamente a un servidor o cliente. Los datos obtenidos acerca de su volumen de tráfico son mínimos en comparación a otros tipos de tráfico, y sugieren que tampoco requiere métodos de priorización para evitar la pérdida de sus paquetes.

► Para finalizar, concluimos que en general todos los algoritmos de encolamiento demandan una gran capacidad de procesamiento, y por ende uso de la memoria y buffers de los routers. Esto trae como consecuencia que se presente menor volumen de tráfico en la red.

## RECOMENDACIONES

Para un óptimo aprovechamiento de los resultados de este trabajo de grado, se deben tener en cuenta los siguientes puntos:

- Ser conscientes de los problemas o conflictos que está presentando la red LAN o WAN y las consecuencias presentadas a raíz de ellos.
- Identificar cada tipo de problema y analizar si la mejor solución es implementar calidad de servicio en la red o es suficiente con ampliar la capacidad de los canales de ancho de banda. En otras palabras, antes que todo debe hacerse un balance de necesidades, recursos y gastos.
- Examinar qué clase de equipos intermedios hacen parte de la red a mejorar (routers, switches, hubs), y si estos son aptos o cumplen con los requerimientos necesarios para aplicar mecanismos de priorización que controlen el tráfico que circula mediante ellos. Si estas características no se dan, realizar un estudio de factibilidad para determinar la posibilidad de adquirir equipos adecuados para su implementación.
- Tomando como base la topología de red utilizada en el presente trabajo para realizar el estudio, se recomienda profundizar el estudio de los algoritmos seleccionados utilizando en la LAN switches de capa 3 en adelante, puesto que tienen capacidad de enrutamiento y por lo tanto es posible configurar algoritmos de encolado para el manejo de la congestión.
- En base a los resultados expuestos en el presente trabajo, seleccionar qué tipo de algoritmo se amolda más eficientemente a su red para solucionar los conflictos que se presentan. Se deben seguir las indicaciones presentes en el mismo para su configuración y correspondiente administración.

- Para el caso específico en que se desee priorizar tráfico generado por el protocolo SQL, se recomienda que el administrador realice un estudio previo de las necesidades y tipo de operaciones que más se efectúen en su red, como consultas e inserción de registros en la base de datos interna. Con base en ello, podrá determinar qué algoritmo se adecua más eficientemente a sus requerimientos de acuerdo a su desempeño mostrado en los resultados del presente trabajo para la entrada y salida de la interfaz del router.
- En el caso que se requiera priorizar tráfico de protocolo FTP, se recomienda utilizar aplicaciones de administración del servidor en las que se pueda especificar un rango más pequeño de puertos aleatorios cuando el servidor entre en modo pasivo al estar detrás de un router. Esto con el fin de poder priorizar aquellos puertos que se encuentren dentro del rango especificado, y así garantizar el encolado para la transferencia real de paquetes.
- Monitorear periódicamente el desempeño o actividad de la red y determinar que el mecanismo de priorización escogido siga siendo el óptimo, debido a la naturaleza cambiante de los flujos de datos que circulan por la red y los cambios de necesidades de los usuarios a lo largo del tiempo.
- Los algoritmos de encolamiento seleccionados solo deben ser utilizados en ambientes donde estrictamente se requieran métodos que aseguren el envío masivo de datos sensibles a retardos o de poco volumen con menores tiempos de respuesta, sin dar pie a la existencia de márgenes de error. Deben ser utilizados bajo la responsabilidad del usuario o administrador de la red, la cual pensamos que debe ser en el mejor de los casos de tipo empresarial.

# MANUAL DE USUARIO

## DESCRIPCIÓN DE APLICACIONES UTILIZADAS

### ▶ Generación de tráfico tipo SQL:

- mysql Essentials v5.1: Servicio de conexión y administración al servidor de la base de datos.
- EMS mysql Manager 2007: Manejador o administrador de bases de datos.
- EMS Data Generator 2005: Generador de datos aleatorios hacia una base de datos local o remota.

### ▶ Generación de tráfico tipo FTP:

- WinFTP Server: Servidor FTP de datos.

### ▶ Generación de tráfico tipo SMTP:

- OCloud Mail Direct Pro v2.6: Servidor SMTP de correos.
- Mail Bomber v9.1: Envío masivo de correos.

### ▶ Generación de tráfico tipo IM:

- BORGChat v1.0: Cliente de mensajería instantánea LAN.

### ▶ Generación de tráfico tipo audio y video:

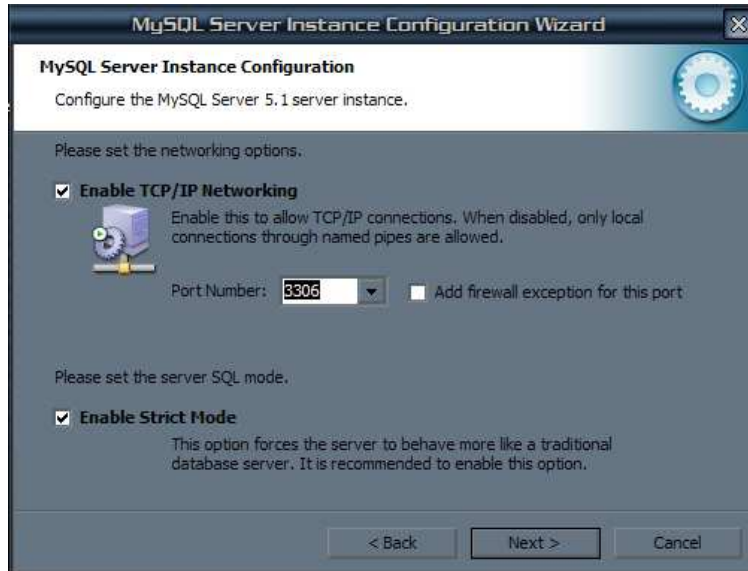
- Unreal Tournament: Juego interactivo en línea

### ▶ Monitoreo de tráfico en la red:

- Netflow Analyzer v7: Analizador de flujos netflow y administrador de la red

# MYSQL ESSENTIALS V5.1: CONFIGURACION

## Paso 1



Se debe configurar el puerto de comunicación TCP entre el servidor y los clientes. El puerto por defecto comúnmente es el 3306.

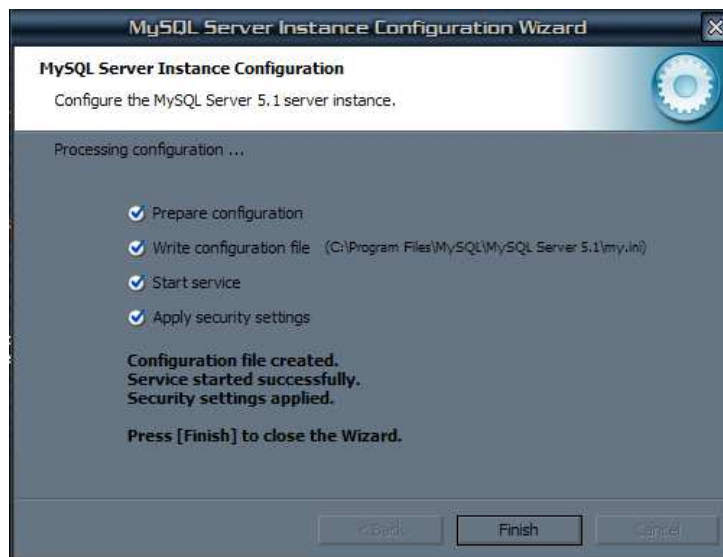
## Paso 2





Se configura un nombre de usuario y contraseña, en el caso que se quiera implementar un método de autenticación. Además debe seleccionarse la opción “Habilitar acceso desde maquinas remotas”, para casos en que los clientes se encuentren fuera de la LAN.

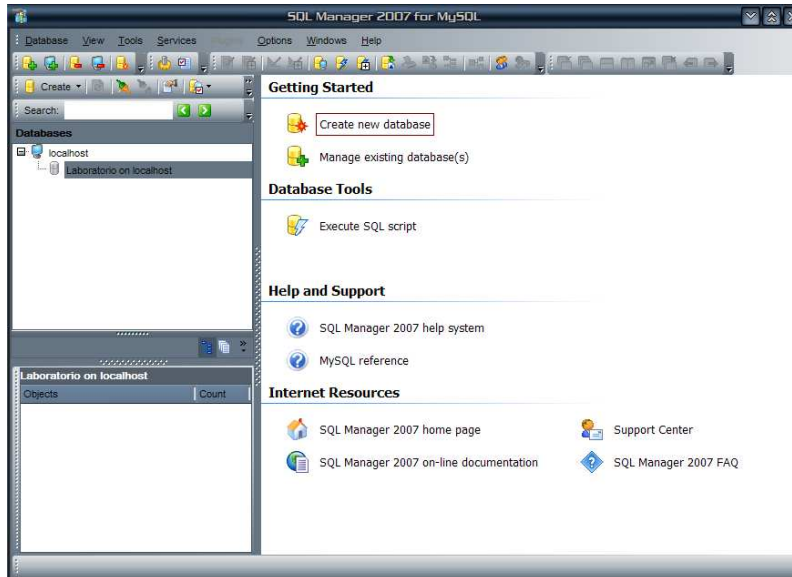
### Paso 3



Una vez realizada la configuración, el asistente debe mostrar la ejecución satisfactoria del servicio mysql y la aplicación de las opciones de seguridad.

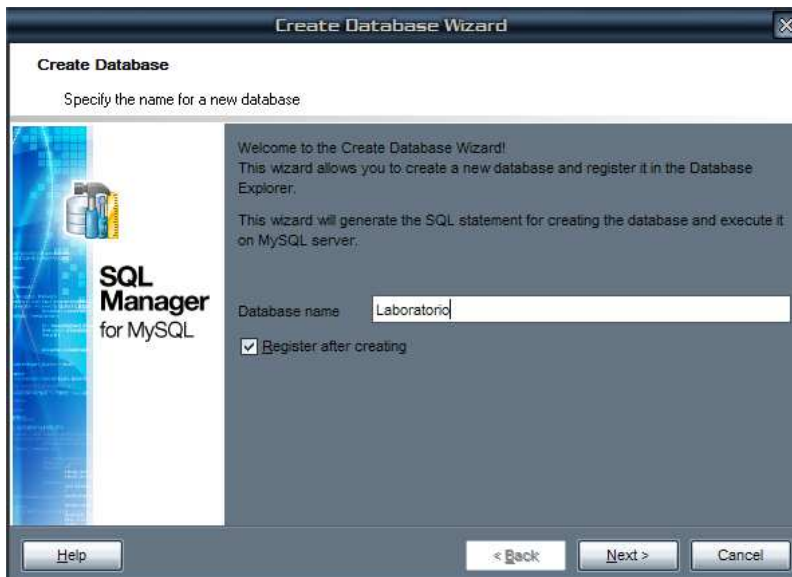
# EMS MYSQL MANAGER: CREACIÓN DE BASES DE DATOS

## Paso 1



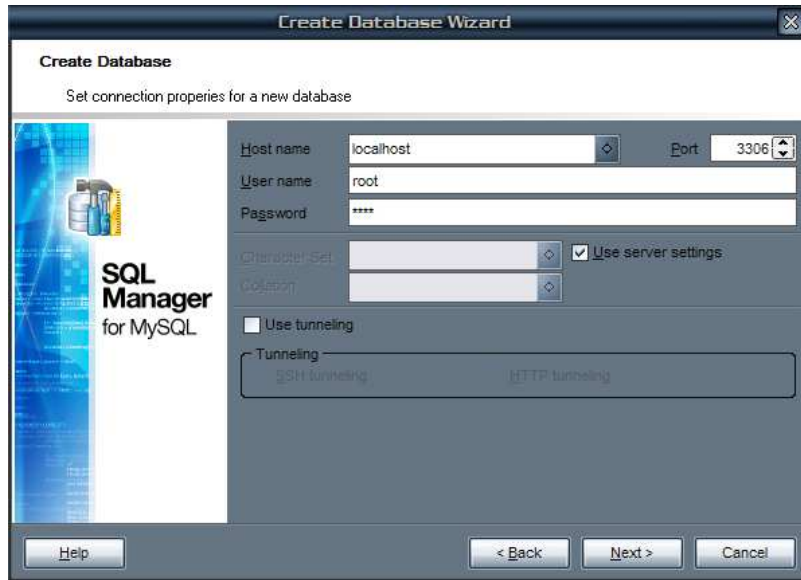
Se selecciona la opción "Create Database" para entrar en modo creación de base de datos.

## Paso 2



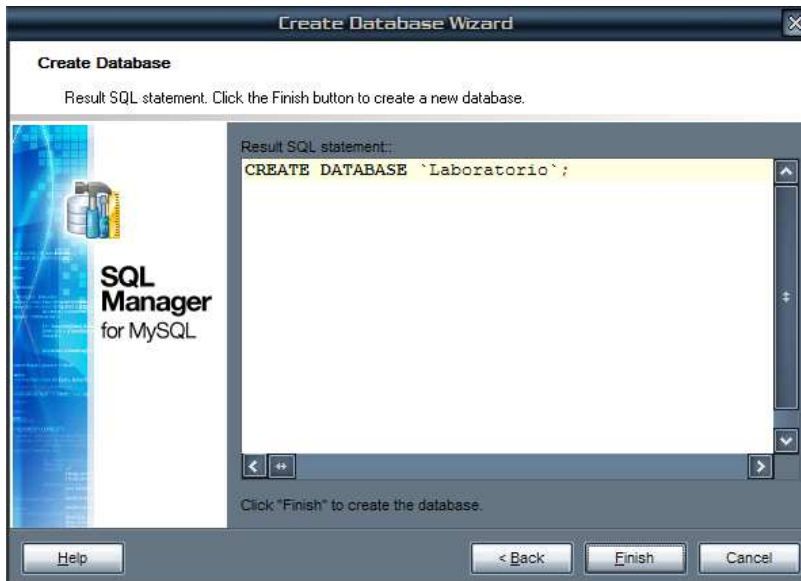
Se escoge un nombre para la nueva base de datos.

### Paso 3



En este paso se especifican los parámetros de conexión con el servidor mysql, tanto los de autenticación como los de número de puerto.

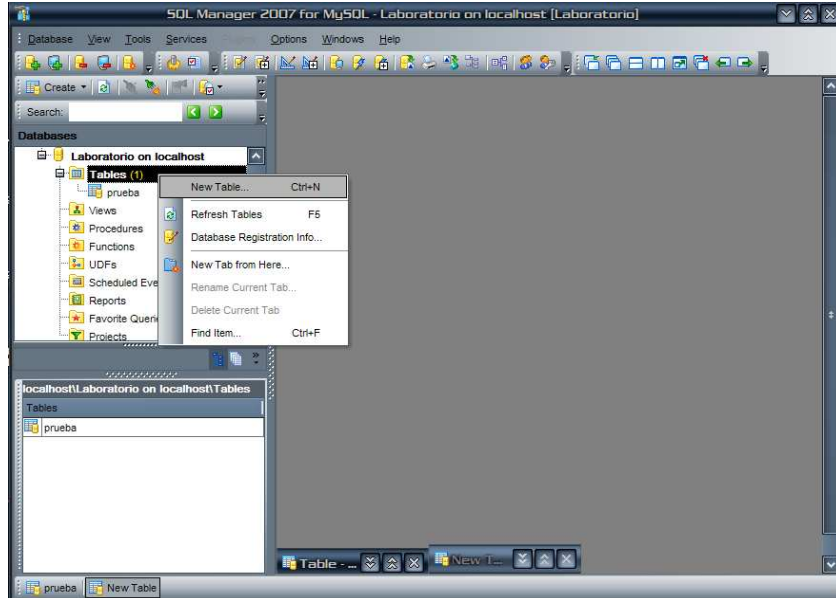
### Paso 4



Como paso final se generan los comandos SQL para la creación de la base de datos, y se elige la opción Finalizar.

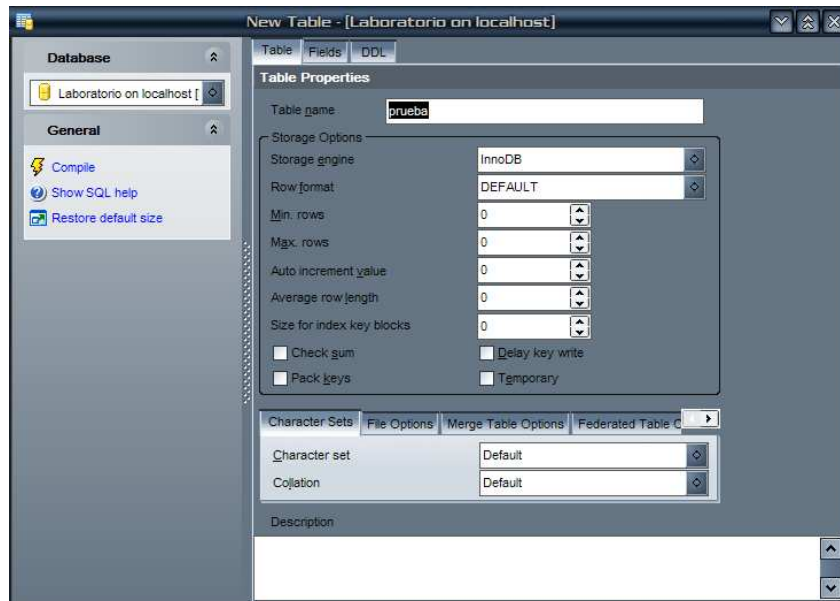
# EMS MYSQL MANAGER: CREACIÓN DE TABLAS

## Paso 1



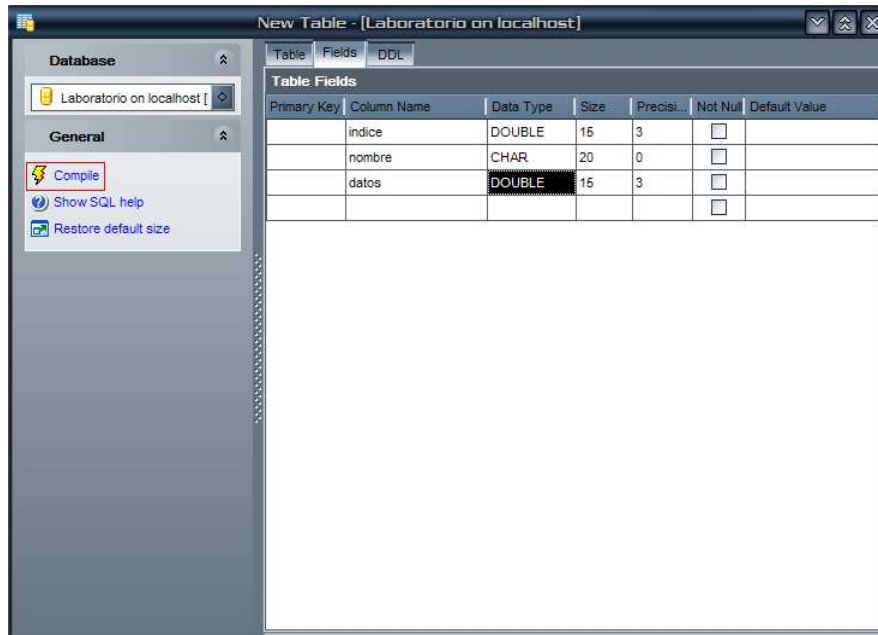
Como primera instancia, en la opción Tables se hace click derecho con el mouse y se elige la opción "New Table".

## Paso 2



Inmediatamente aparece un cuadro de dialogo, en el cual se elige la pestaña Table para nombrar la tabla de la base de datos.

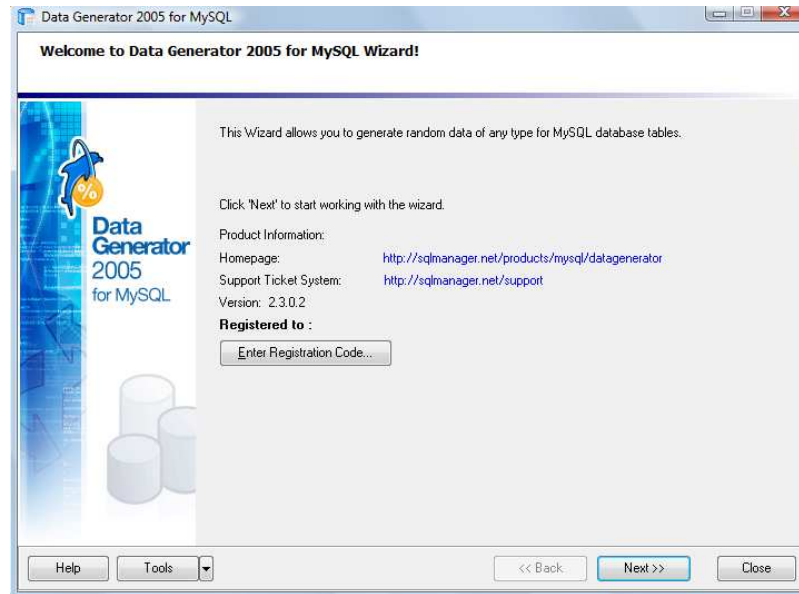
### Paso 3



En la pestaña Fields, se especifican las columnas de las cuales se compondrá la tabla con sus respectivos tipos de datos y tamaños. Luego se elige la opción "Compile", para compilar el script de la tabla y pueda crearse.

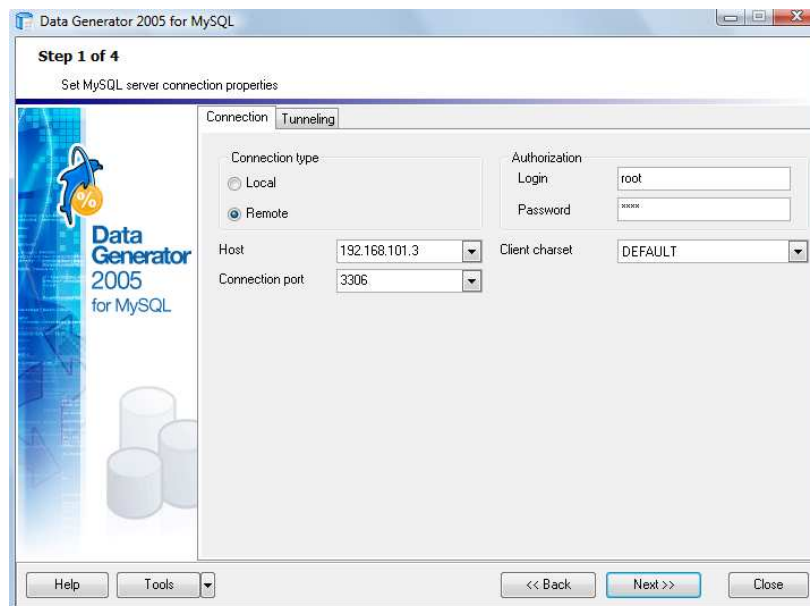
# EMS DATA GENERATOR: CONFIGURACION

## Paso 1



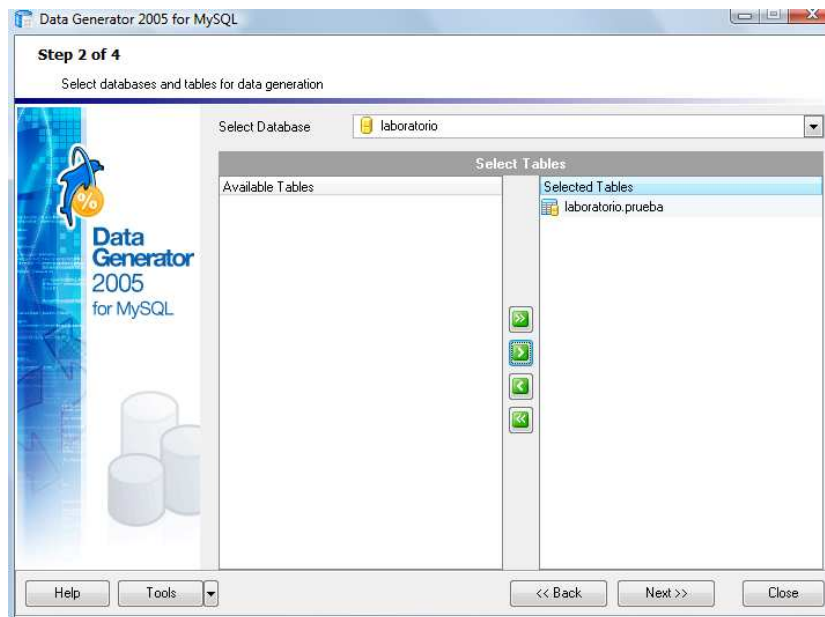
Al instalarse e iniciarse por primera vez, aparece un asistente indicando que se generarán datos aleatorios en una base de datos indicada.

## Paso 2



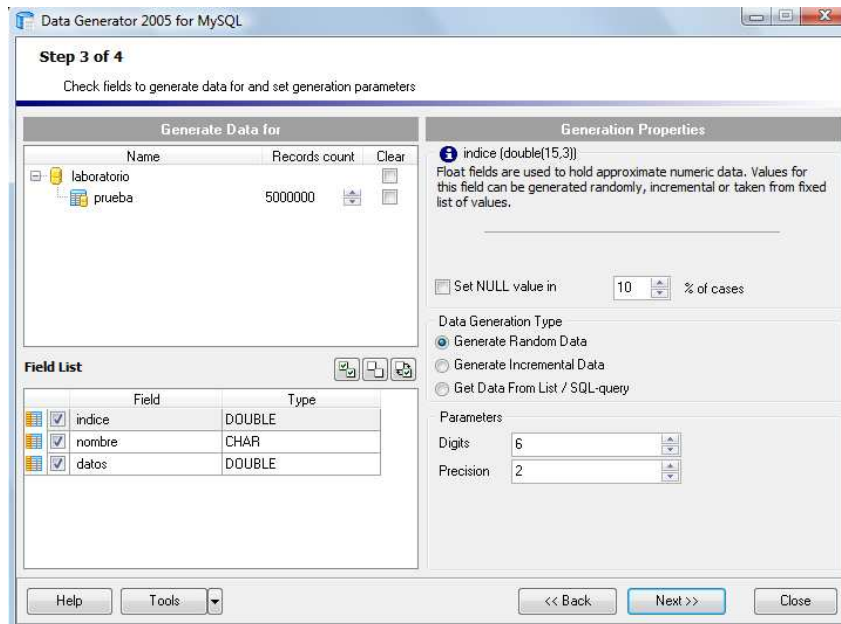
En este paso se especifican los parámetros de conexión con el servidor de la base de datos, para lo cual se elegirá el tipo de conexión remota, las variables de autenticación y la IP del servidor.

### Paso 3



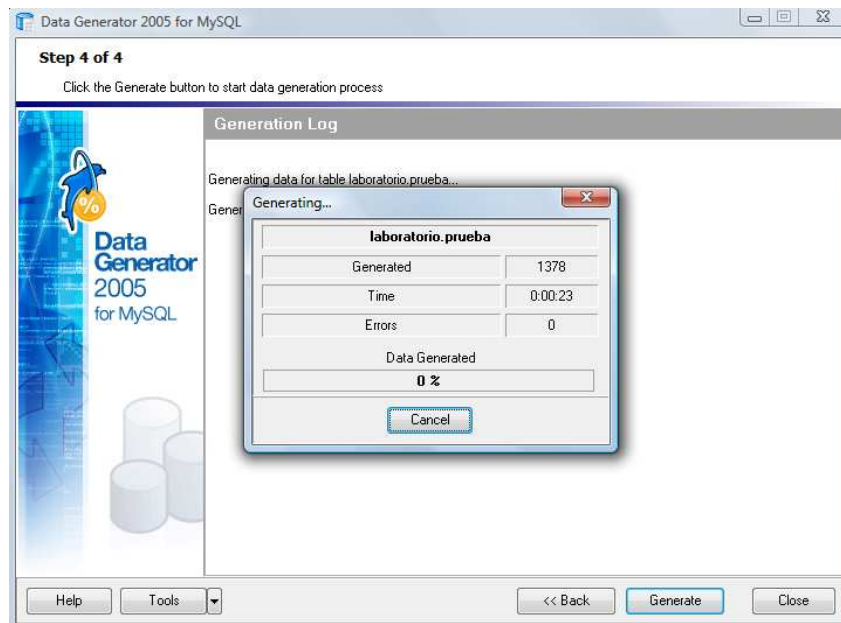
Una vez se conecta con el servidor, se elige la base de datos y se selecciona la tabla con la cual se trabajará para la generación de tráfico SQL.

## Paso 4



Aquí se especificará el número de registros que se generarán en la tabla de la base de datos, y se podrá escoger si los registros son aleatorios o incrementales.

## Paso 5

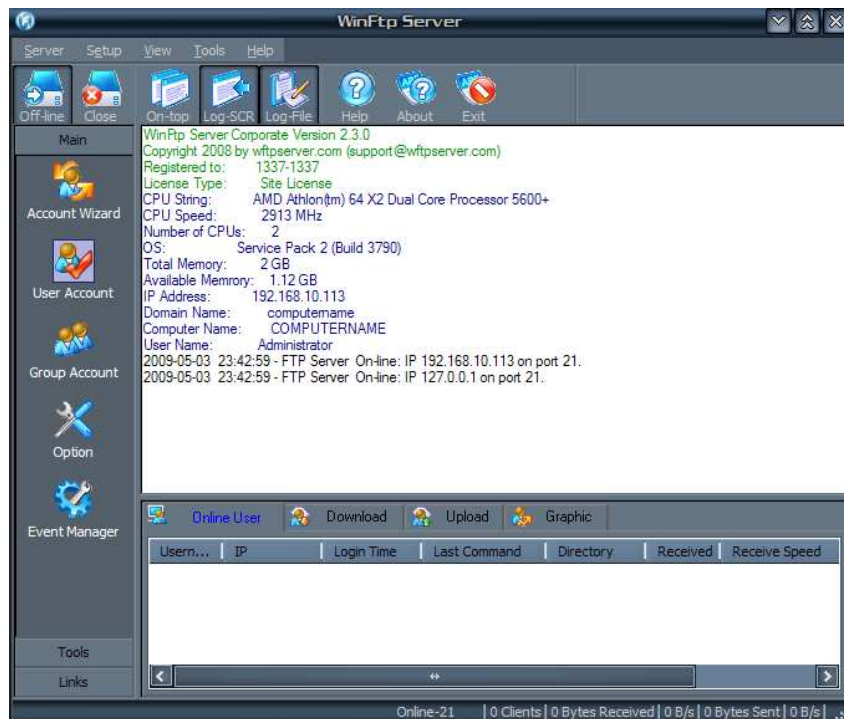




Una vez se hace click en el botón Generate se comienza el proceso de generación de registros aleatorios en la base de datos remota, y se llevará una cuenta del número de datos insertados.

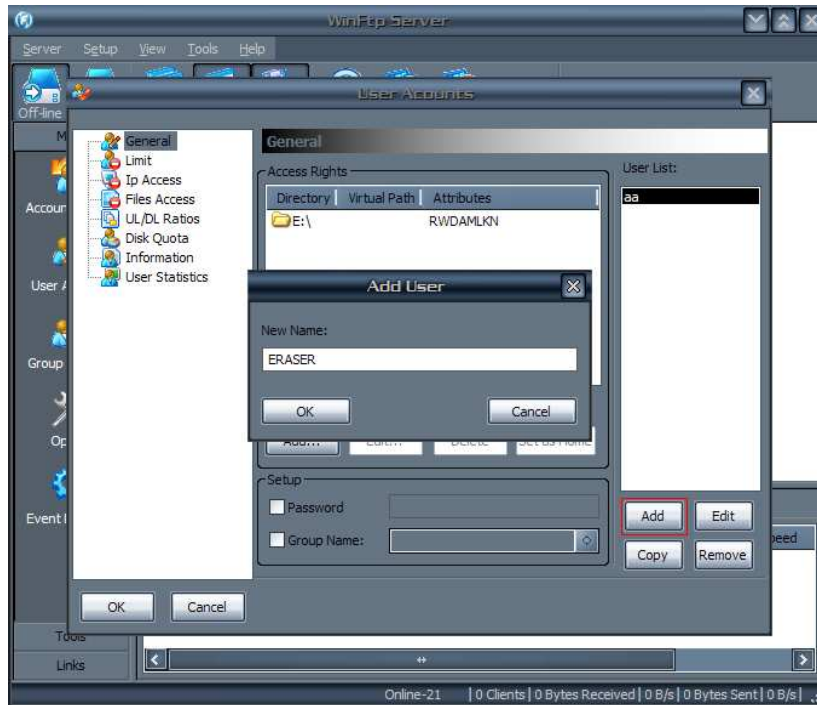
## WINFTP SERVER: CONFIGURACIÓN

### Paso 1



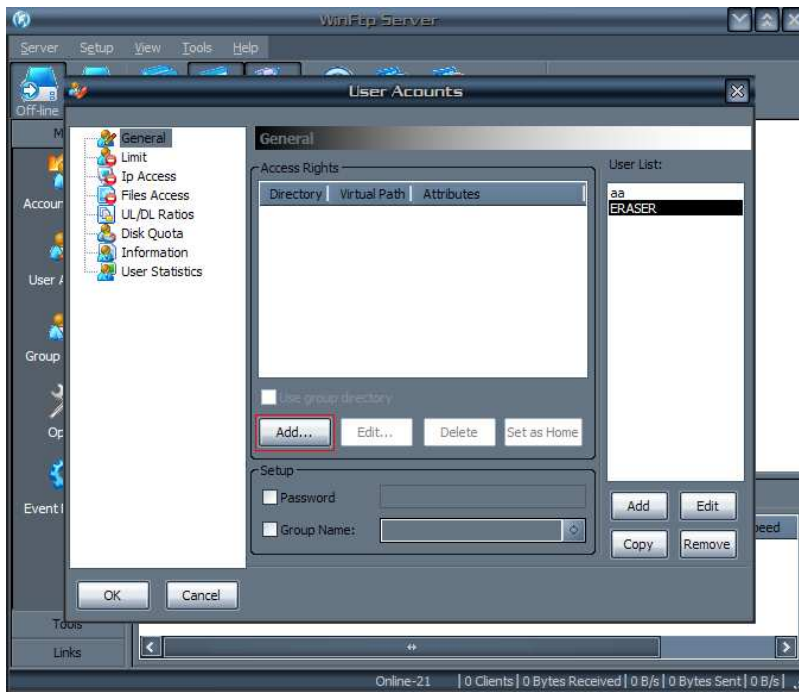
Se presiona el botón Online para que el servidor escuche las comunicaciones entrantes y abra los puertos de conexión. Para el siguiente paso se entrará a las opciones de creación de usuario mediante el botón User Account.

## Paso 2



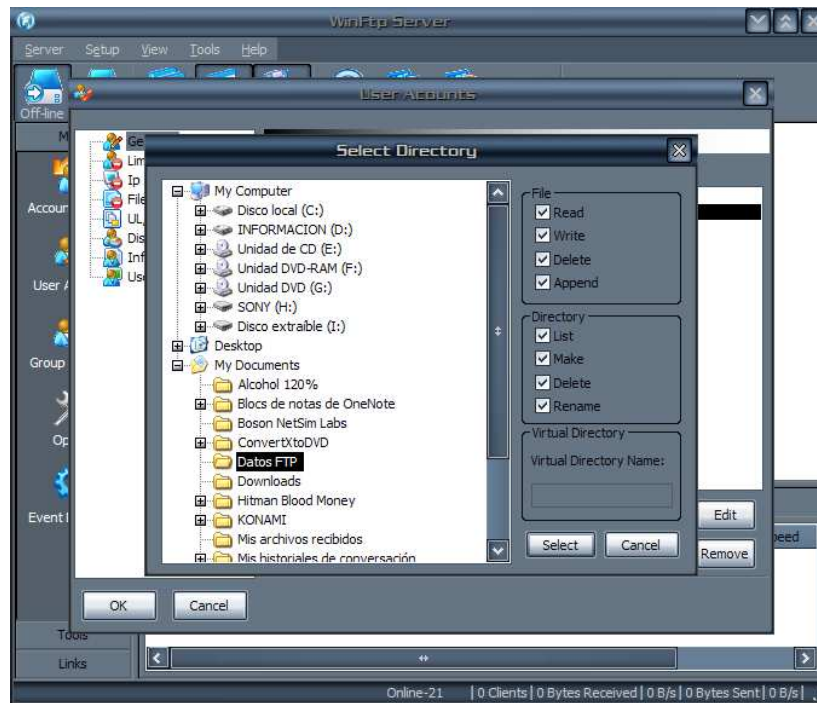
Se presiona el botón Add para añadir un usuario nuevo, se especifica un nombre de usuario para la cuenta y una contraseña en caso de que se prefiera una autenticación con el servidor. En estos casos es lo mas recomendado debido a los ataques que se presentan utilizando el protocolo FTP.

### Paso 3



Se agregan los directorios que se visualizaran cuando los clientes conecten con el servidor, carpetas donde se compartirán archivos de cualquier tipo.

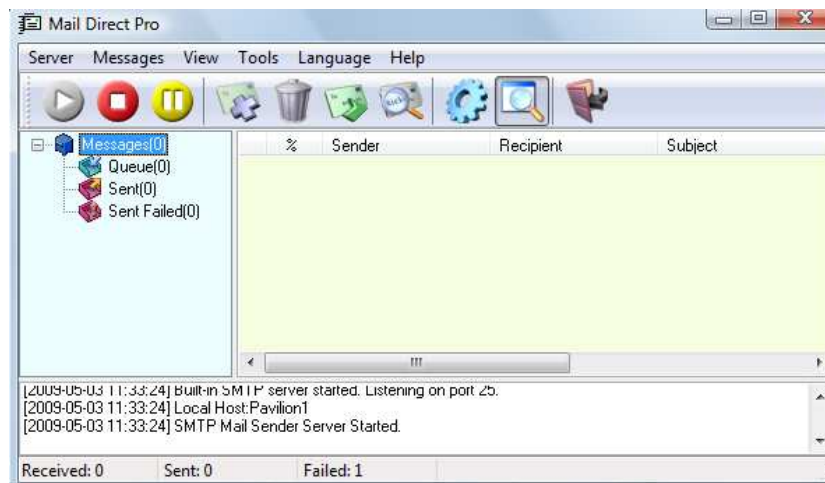
### Paso 4



Luego de haberse agregado cada directorio, deben especificarse los permisos de lectura y escritura correspondientes sobre archivos y carpetas internas.

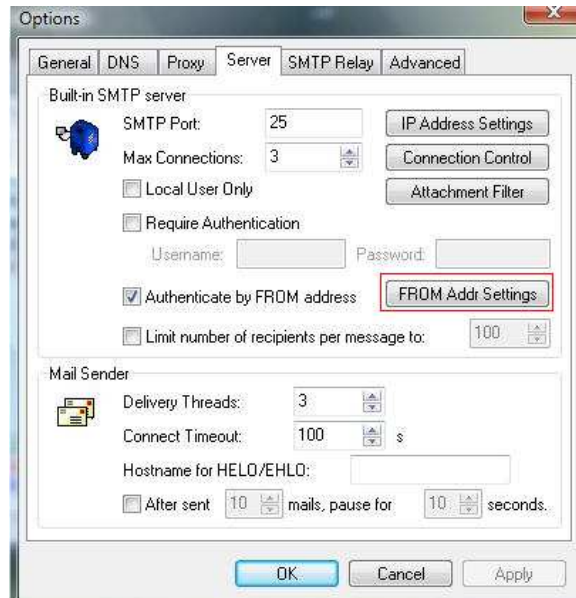
## OCCLOUD MAIL DIRECT PRO: CONFIGURACIÓN

### Paso 1



Al iniciar el programa, surge la ventana principal en la cual aparece el total de correos recibidos, enviados y aquellos cuyo envío resultó fallido. Además contiene un log en la parte inferior donde muestra la información de la actividad de la conexión, direcciones IP que se conectan al servidor y los correos que van llegando en tiempo real. Para entrar a las opciones del programa se entra al menú Tools y se selecciona Options.

## Paso 2



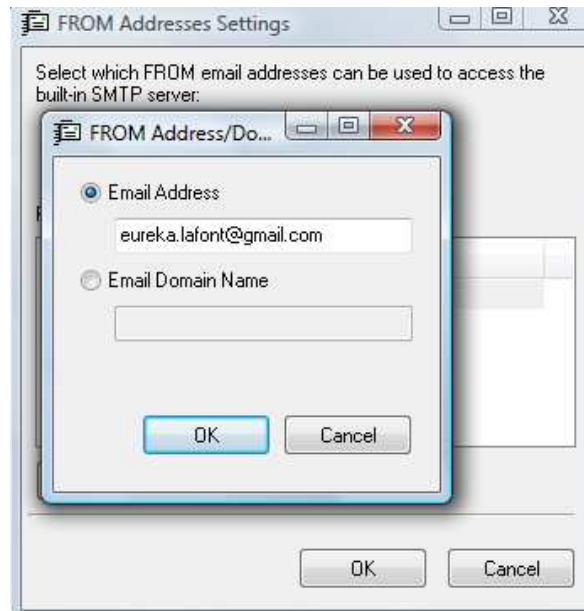
En la pestaña Server se especifica el número de puerto del protocolo SMTP si se desea cambiar. Además para dotar de seguridad al servidor se selecciona la opción de Autenticar por dirección de correo, para que solo accedan a él aquellas cuentas o clientes que se le especifiquen.

## Paso 3



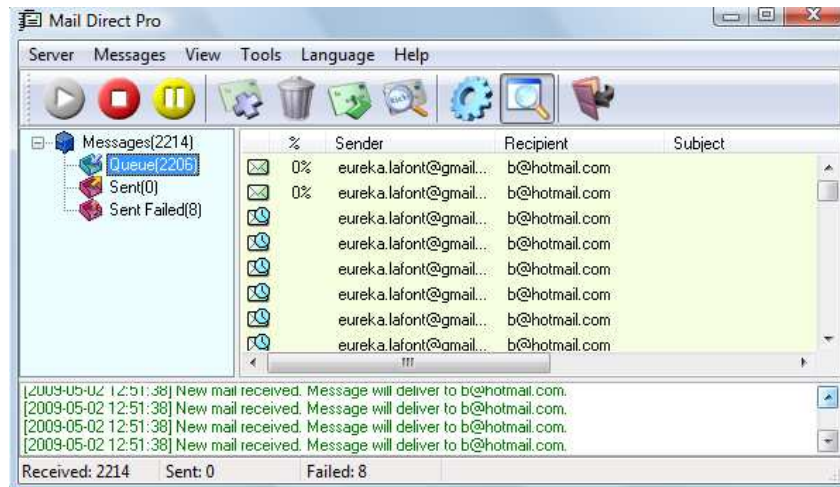
Para añadir direcciones de correo entrantes, se presiona el botón Add.

## Paso 4



Aparece un cuadro de dialogo en el cual se ingresa cada una de las direcciones de correo. Este paso debe realizarse tantas veces como correos se quieran agregar. Al final se aceptan todas las opciones dando Ok.

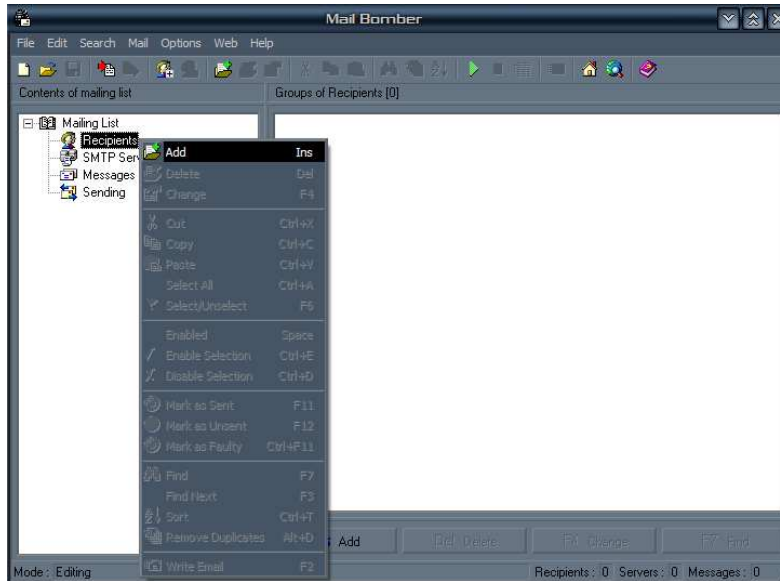
## Paso 5



Se inicia el servidor haciendo click sobre el botón de Play. Una vez comienzan a enviarse correos hacia el servidor, este comienza a mostrarlos y a reenviarlos a su correspondiente servidor POP3 para que puedan ser descargados por sus destinatarios.

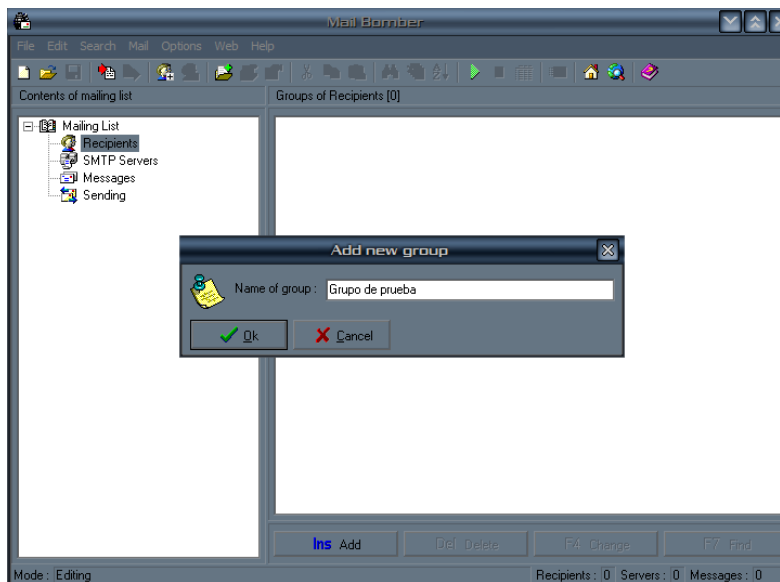
## MAIL BOMBER: CREANDO GRUPOS DE DESTINATARIOS

### Paso 1



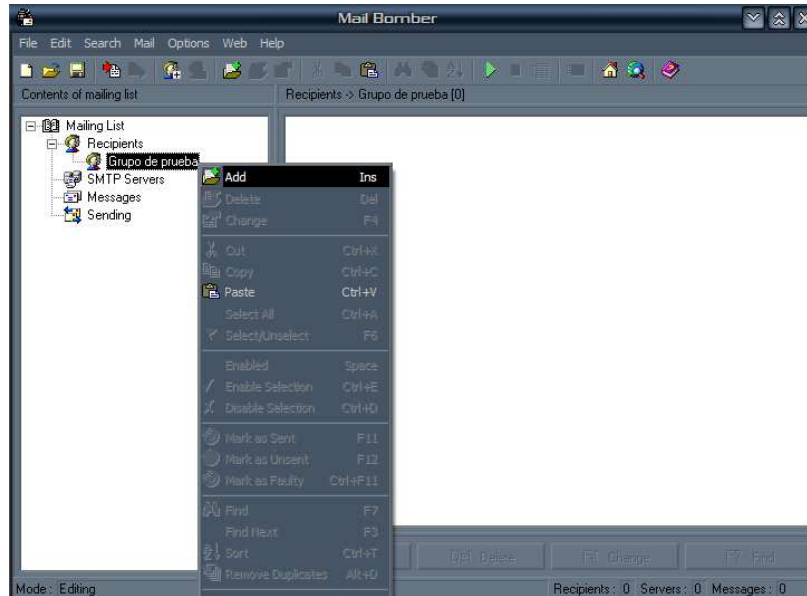
Sobre la opción Recipients, se hace click con el botón derecho del mouse y se selecciona Add o añadir.

### Paso 2



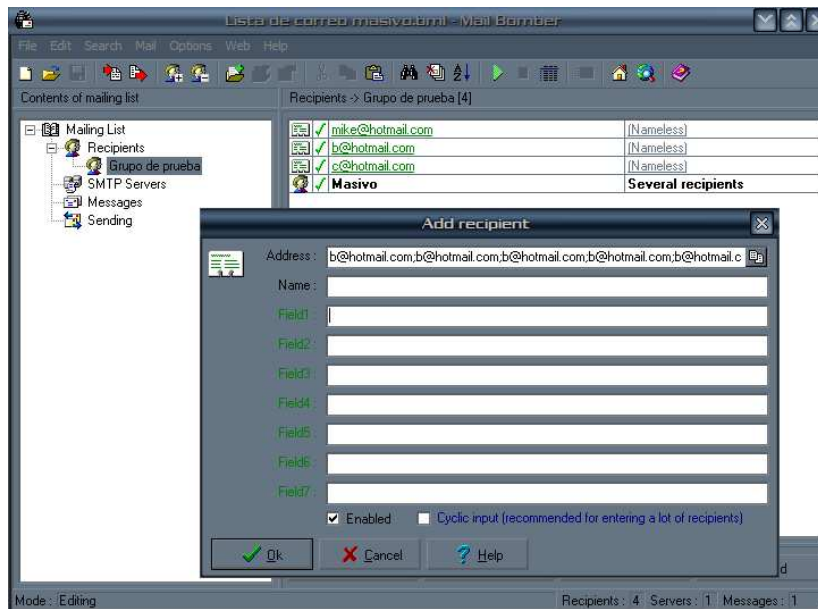
Se especifica un nombre para el grupo de destinatarios a crear.

### Paso 3



Sobre el grupo de destinatarios creado, en este caso “Grupo de prueba”, se hace click derecho y se elige Add.

### Paso 4

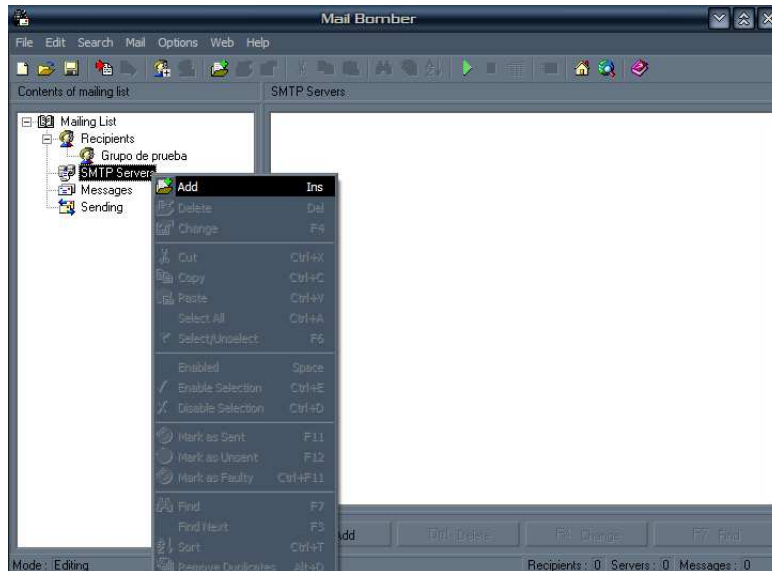




En el campo de Address se escriben todos los destinatarios hacia los cuales se van a enviar los correos masivos, que pueden ser tantos como se deseen sin límite de número.

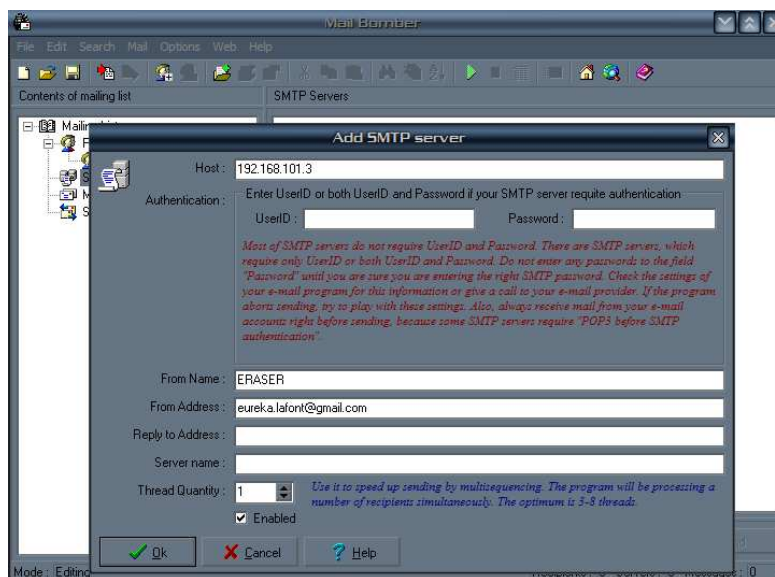
## MAIL BOMBER: CONFIGURANDO CONEXIÓN

### Paso 1



Sobre la opción SMTP Server, se hace click derecho y se selecciona Add.

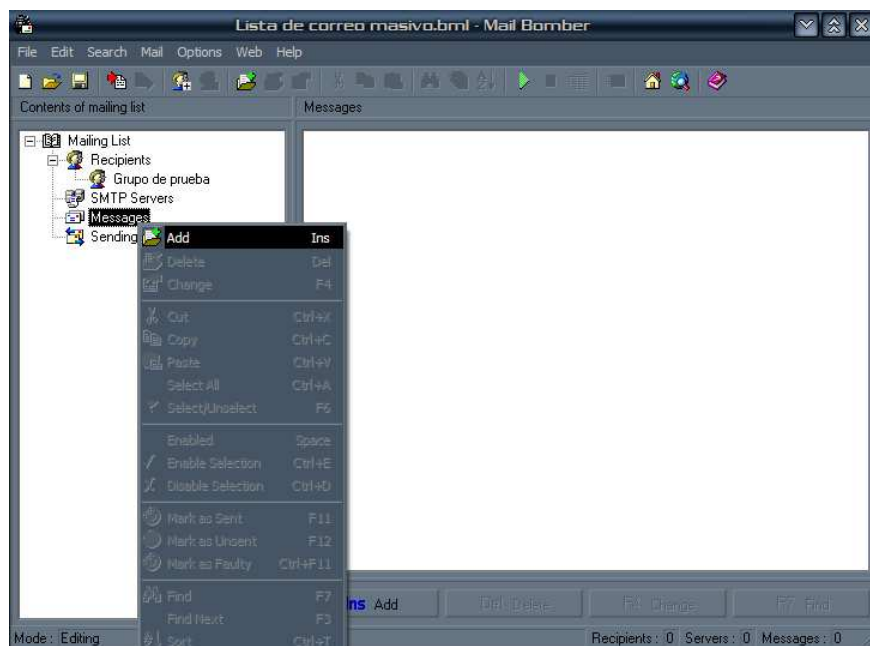
### Paso 2



En el campo Host especificamos la dirección IP del equipo donde se encuentra en línea el servidor SMTP. Luego colocamos un nombre de usuario y la dirección de correo que aparecerá como origen en los mensajes enviados. Esta debe ser cualquiera de las añadidas anteriormente al servidor, para poder acceder a él.

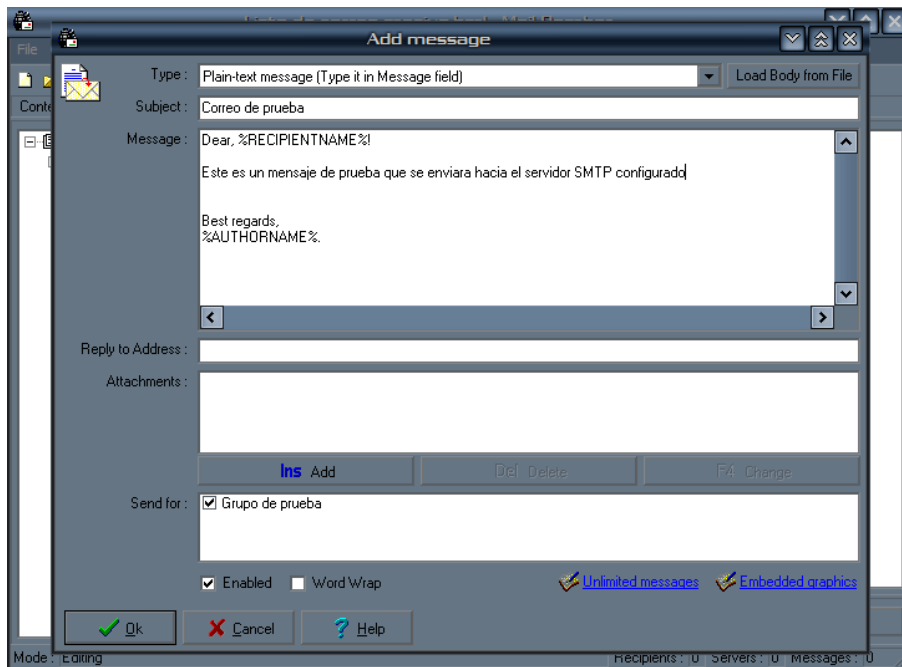
## MAIL BOMBER: CREANDO MENSAJES DE PRUEBA

### Paso 1



Sobre la opción Messages, se hace click derecho y se selecciona Add.

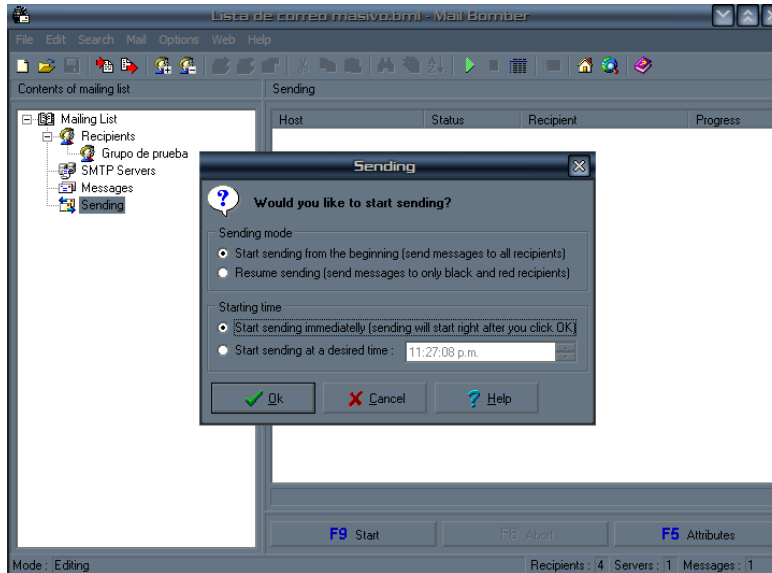
## Paso 2



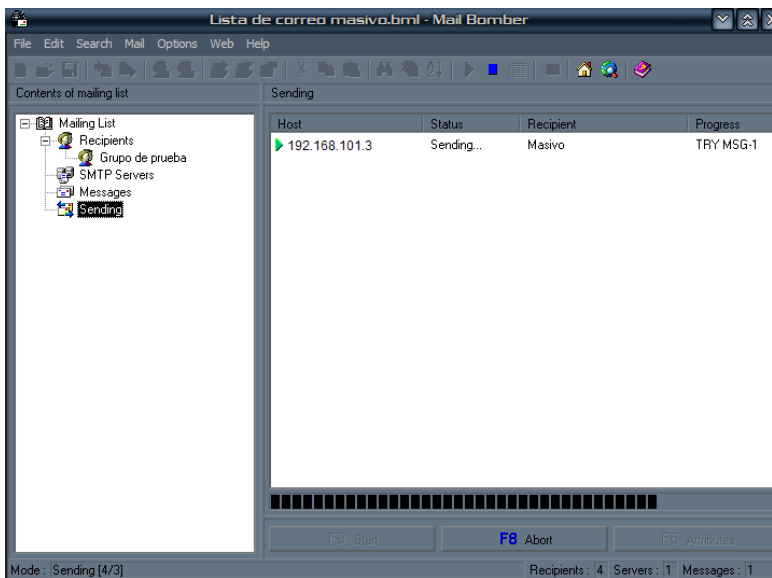
Aparece una ventana en la cual se especifica el asunto y el cuerpo del mensaje. Además se podrán especificar los grupos de destinatarios creados, en el caso de haber sido varios.

# MAIL BOMBER: ENVIANDO CORREOS

## Paso 1

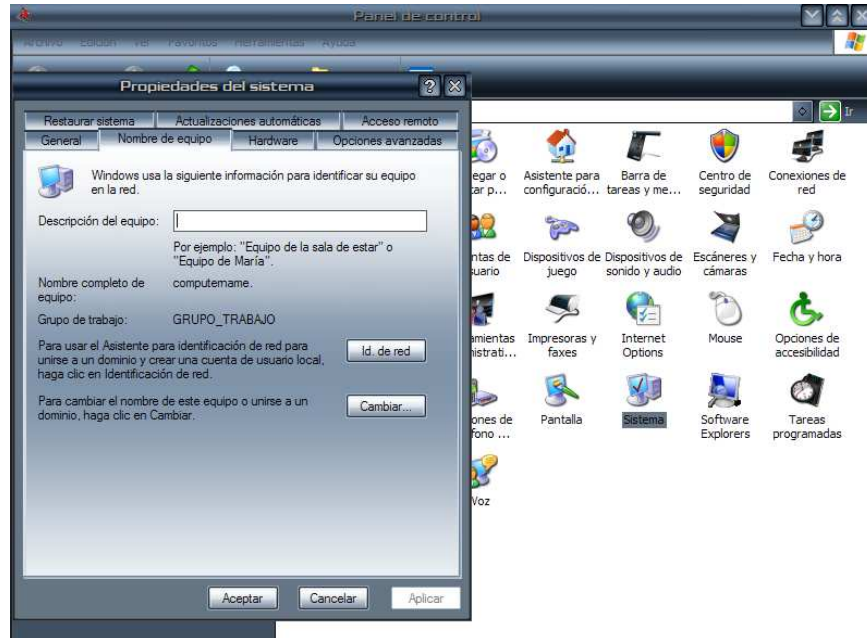


Se hace click en el botón de Enviar, cuyo icono es el que comúnmente se conoce como Reproducir, de color verde. En la ventana que aparece, hay que especificar que los correos se enviarán inmediatamente. Luego que se preciona Ok, comenzará el proceso de envío.



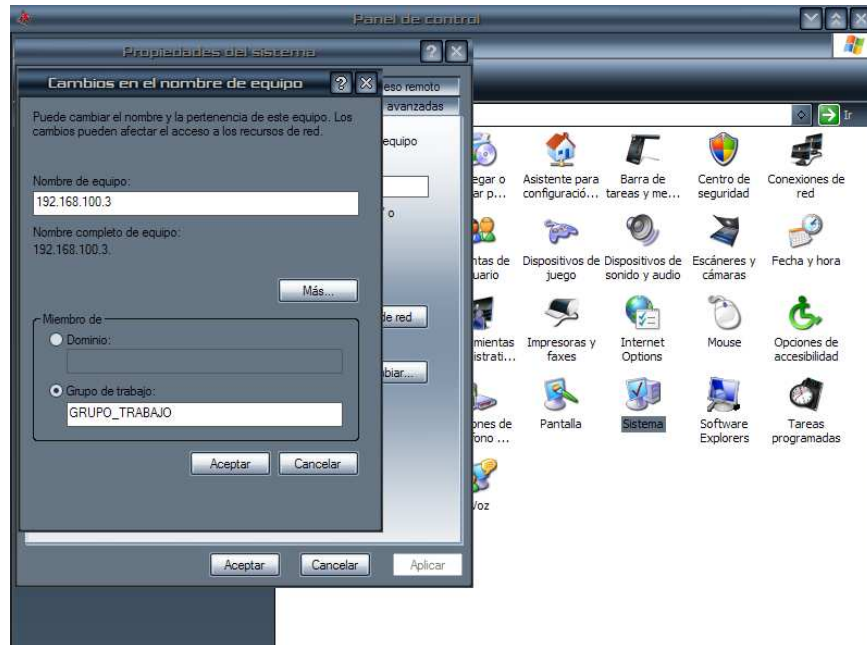
# BORGCHAT: CONFIGURANDO CONEXIÓN LAN

## Paso 1



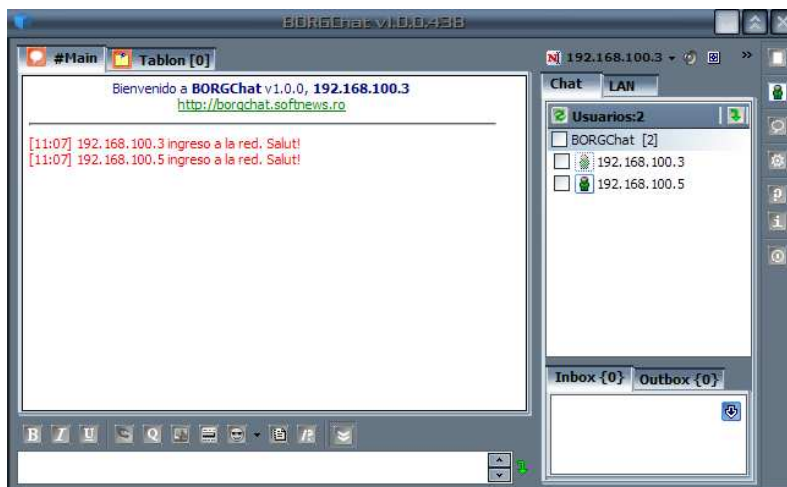
Antes de empezar a configurar el programa de mensajería instantánea, es necesario entrar al Panel de control, seleccionar la opción Sistema y cambiar el grupo de trabajo para todos los clientes que se encuentran en la misma LAN. Esto con el fin de que puedan pertenecer a la misma red y verse mutuamente. Para ello se presiona en el botón Cambiar.

## Paso 2



Se especifica como nombre de equipo la dirección IP que le fue asignada, para poder diferenciarlo una vez conectado en Borgchat. Además se le asigna el mismo nombre de grupo de trabajo en el que están todos los demás hosts.

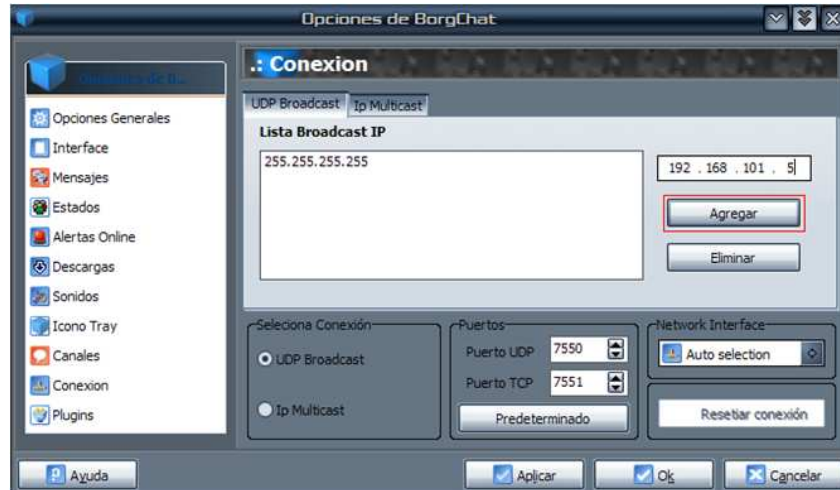
## Paso 3



Al iniciar por primera vez Borgchat en cada uno de los hosts, automáticamente se conectan y reconocen dado que se encuentran en la misma red y grupo de trabajo.

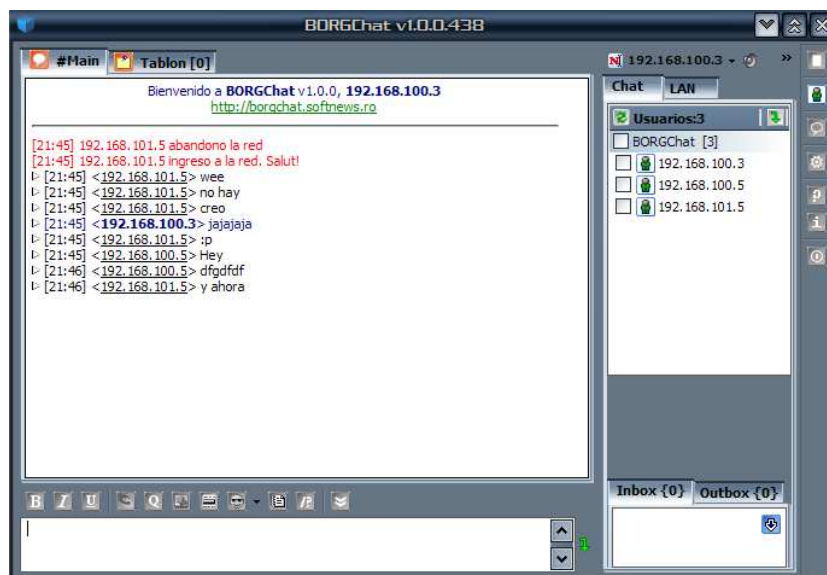
# BORGCHAT: CONFIGURANDO CONEXIÓN WAN

## Paso 1



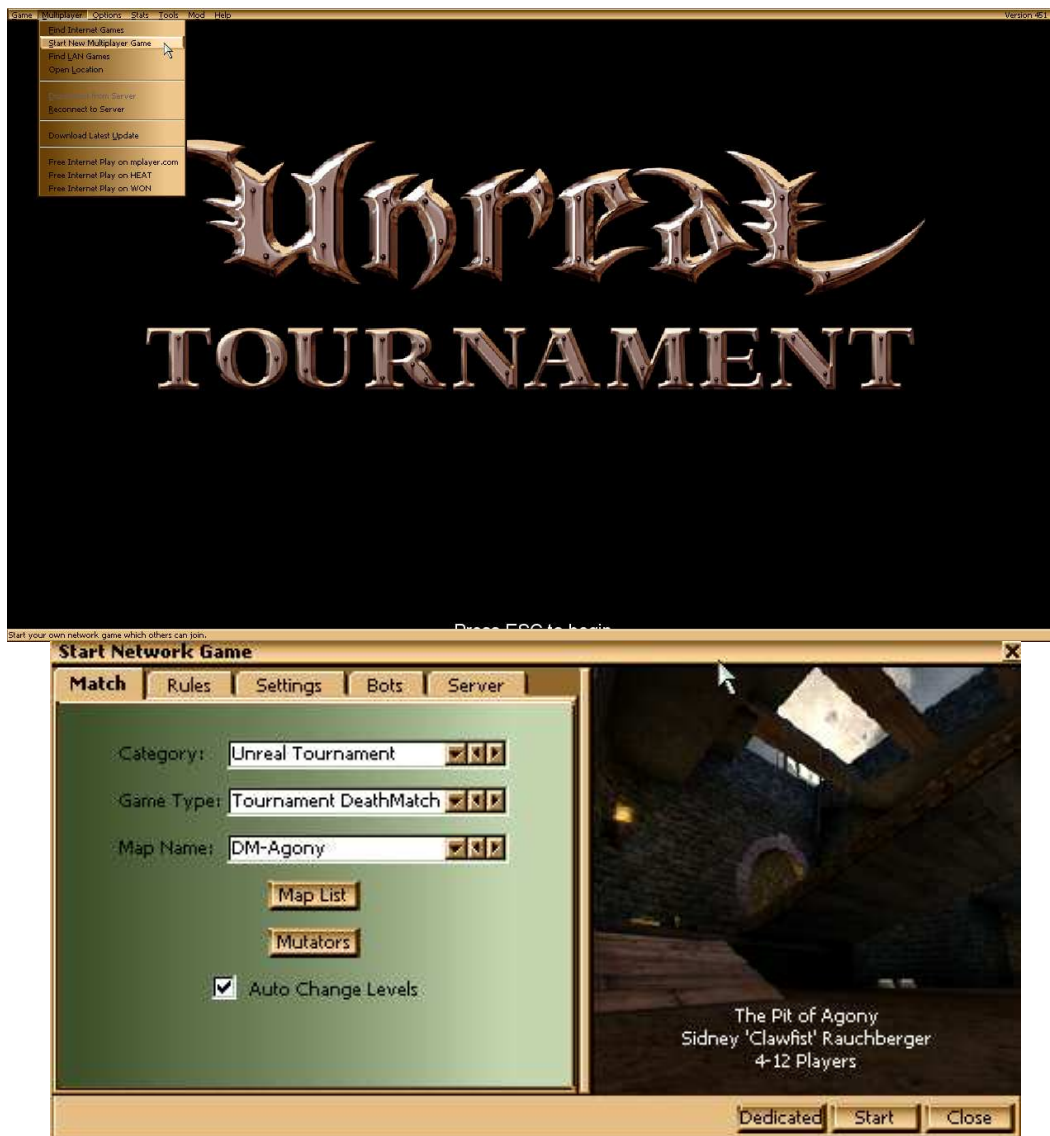
En las opciones de Borgchat, se selecciona Conexión y se agregan manualmente las IPs que se encuentran fuera de la LAN, para ser encontradas por Broadcast. Luego se aplican y aceptan las configuraciones para que tomen efecto.

## Paso 2



Una vez hecho lo anterior, los equipos remotos comienzan a aparecer en línea y disponibles para chat.

## UNREAL TOURNAMENT: CONFIGURACIÓN DE SERVIDOR



En el menú Multiplayer, se selecciona la opción Start Multiplayer Game. Luego se eligen los mapas de juego, y se da click en Start para comenzar.



## UNREAL TOURNAMENT: CONFIGURACIÓN DE CLIENTE



En el menú Multiplayer, se selecciona la opción Open Location. Luego se especifica la dirección IP del servidor con el cual se realizará la conexión.

## NETFLOW ANALIZER: AUTENTICACIÓN

ManageEngine®  
NetFlow Analyzer 7 Professional *Plus*

Netflow analyzer uses Cisco Netflow™, Cisco NBAR™, sFlow™ and Cisco CBQoS MIB to provide valuable information on what applications are using bandwidth, who is using them and the priority of the traffic. This information is useful for network analysis, troubleshooting, application performance monitoring, QoS policy validation and more.

[www.netflowanalyzer.com](http://www.netflowanalyzer.com)

**Sign In**

Nombre de usuario: admin

Contraseña: \*\*\*\*\*

Keep me signed in

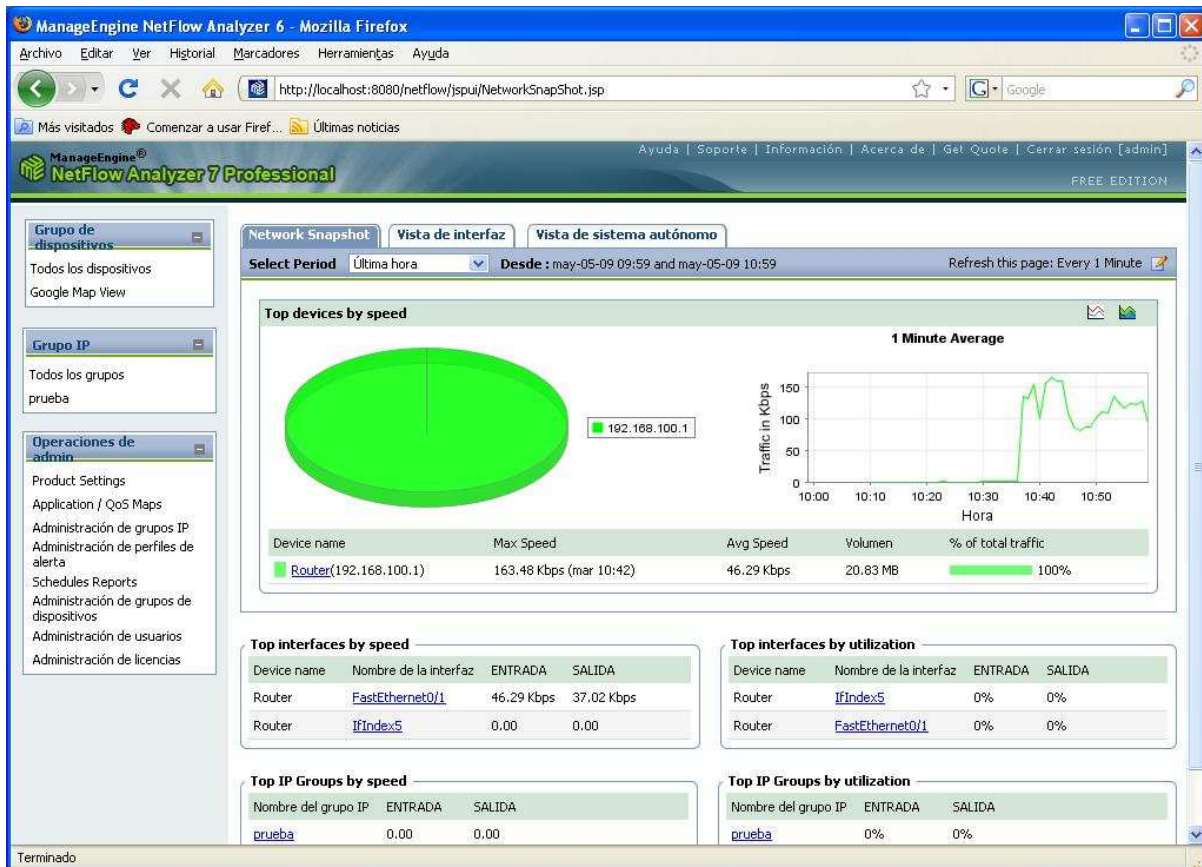
**Login**

Copyright 2004-2008 AdventNet Inc. All rights reserved

Support : [support@netflowanalyzer.com](mailto:support@netflowanalyzer.com) Toll free : 1 888 720 9500

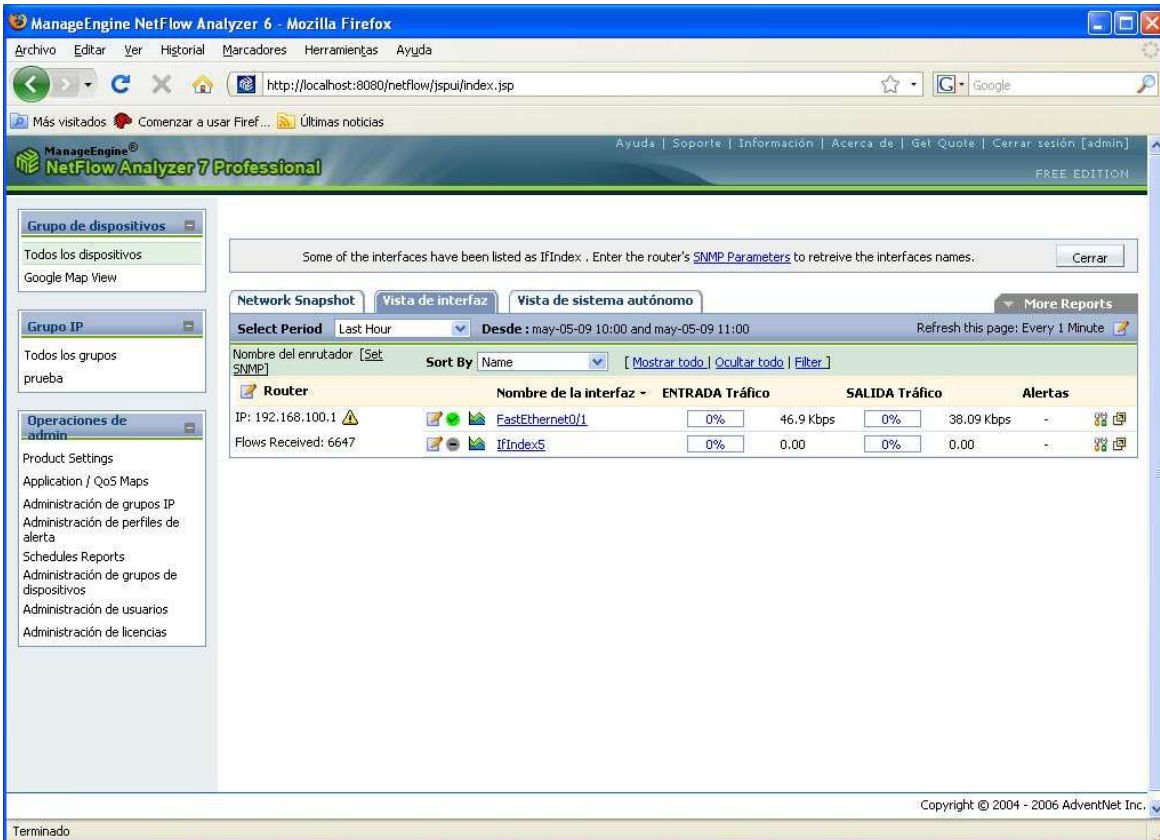
Para entrar a esta aplicación debe hacerse mediante cualquier explorador de internet y digitar la dirección <http://localhost:8080>, ya que el equipo se conectara mediante el puerto 8080 a la misma aplicación local. Además el administrador debe autenticarse con un nombre de usuario y contraseña, que por defecto son definidos como admin y admin respectivamente.

## NETFLOW ANALIZER: VISUALIZACIÓN DE DISPOSITIVOS



Una vez iniciada la aplicación, reconoce todos los dispositivos de capa 3 conectados a la red. Además muestra las interfaces Ethernet y seriales que posee el router, así también como algunas estadísticas preliminares. Para ello debe hacerse click sobre la opción "Todos los dispositivos".

## NETFLOW ANALIZER: DETECCIÓN DE INTERFACES



The screenshot displays the ManageEngine NetFlow Analyzer 7 Professional web interface. The browser window title is "ManageEngine NetFlow Analyzer 6 - Mozilla Firefox". The address bar shows "http://localhost:8080/netflow/jspui/index.jsp". The interface includes a navigation menu on the left with sections like "Grupo de dispositivos", "Grupo IP", and "Operaciones de admin". The main content area shows a "Network Snapshot" view for a router with IP 192.168.100.1. A message at the top states: "Some of the interfaces have been listed as IfIndex. Enter the router's [SNMP Parameters](#) to retrieve the interfaces names..". Below this, a table lists detected interfaces with their traffic statistics.

Router	Nombre de la interfaz	ENTRADA Tráfico	SALIDA Tráfico	Alertas		
IP: 192.168.100.1	FastEthernet0/1	0%	46.9 Kbps	0%	38.09 Kbps	-
Flows Received: 6647	IfIndex5	0%	0.00	0%	0.00	-

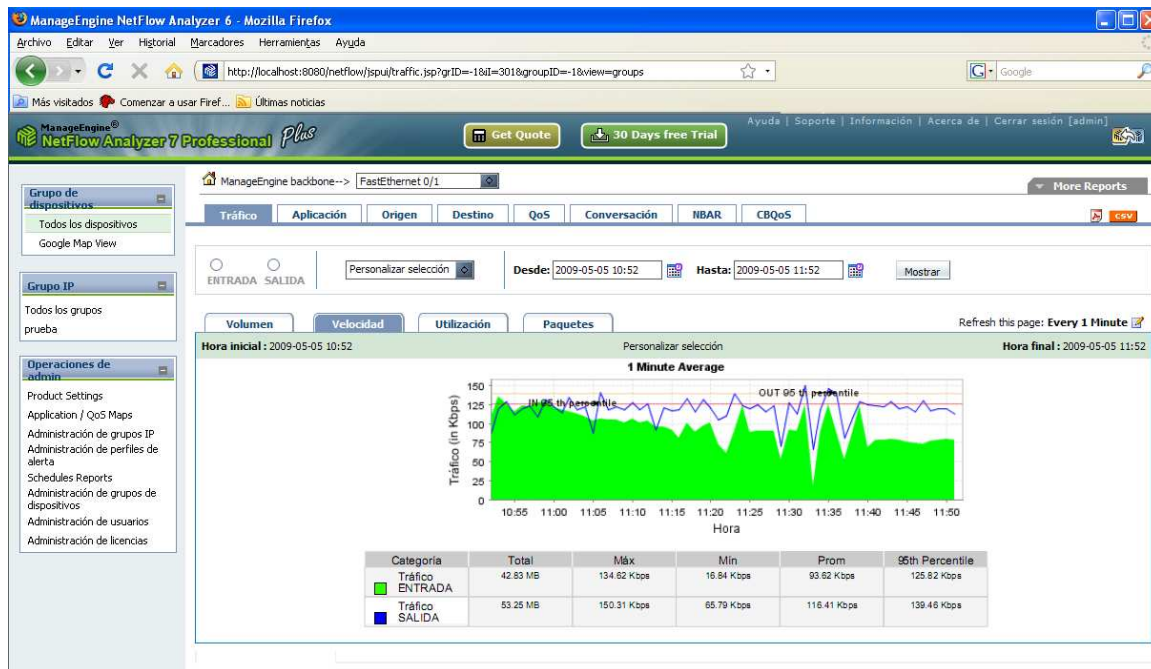
Al hacer click sobre el dispositivo, la aplicación mostrará las interfaces detectadas e indicará con un icono de estado cuál de ellas esta enviando flujos de protocolo netflow hacia el equipo. Para comenzar a analizar y visualizar estos flujos, es necesario entrar a la interfaz en cuestión.

## NETFLOW ANALIZER: PERSONALIZACIÓN DE TIEMPO DE MONITOREO

Categoría	Total	Máx	Mín	Prom	95th Percentile
IN	21.39 MB	163.48 kbps	0.00	46.76 Kbps	154.9 Kbps
OUT	17.5 MB	193.41 kbps	0.00	38.25 Kbps	127.96 Kbps

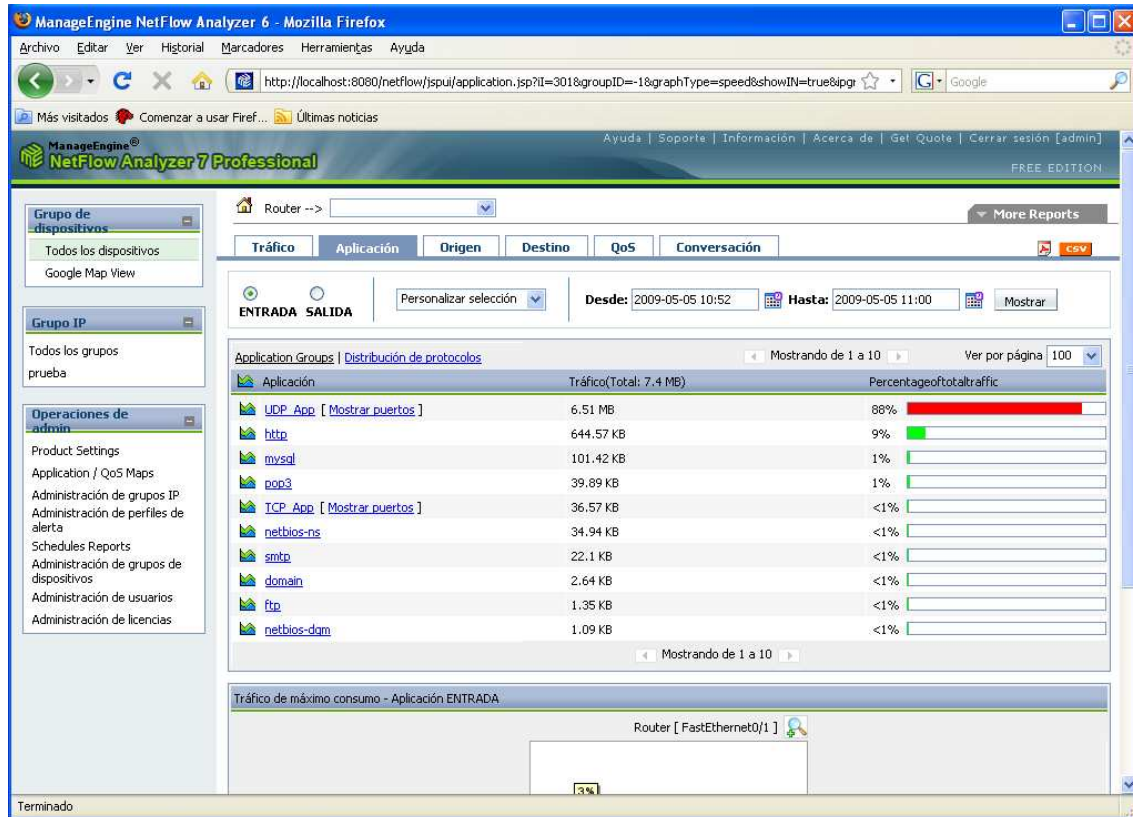
Para definir rangos de tiempo se hace click sobre el combo y se elige la opción Personalizar Selección. Inmediatamente se habilitan los campos a la derecha, donde se podrá especificar la hora de principio y finalización del monitoreo.

## NETFLOW ANALIZER: TRÁFICO GENERAL



Para observar las graficas generadas de tráfico general, se hace click sobre la pestaña de Tráfico y se seleccionan las opciones de Volumen o Velocidad, según se desee.

## NETFLOW ANALIZER: TRÁFICO POR GRUPOS DE PROTOCOLOS O APLICACIONES



Se hace click sobre la pestaña Aplicación y se visualiza el tráfico de las aplicaciones y servicios de acuerdo a una asociación que hace el programa, identificando el número de puerto de los paquetes que circulan por la interfaz del router. Además se da la opción de analizar el tráfico tanto de entrada como de salida.



# NETFLOW ANALIZER: TRÁFICO POR PUERTOS

## Paso 1

The screenshot displays the ManageEngine NetFlow Analyzer 6 web interface. The browser window title is "ManageEngine NetFlow Analyzer 6 - Mozilla Firefox". The URL is "http://localhost:8080/netflow/jsps/application.jsp?nI=301&groupID=-1&graphType=speed&showIN=true&ipg". The interface shows a navigation menu on the left with sections like "Grupo de dispositivos", "Grupo IP", and "Operaciones de admin". The main content area is titled "Router -->" and includes tabs for "Tráfico", "Aplicación", "Origen", "Destino", "QoS", and "Conversión". Below these tabs, there are filters for "ENTRADA" and "SALIDA", a "Personalizar selección" dropdown, and date range filters for "Desde: 2009-05-05 10:52" and "Hasta: 2009-05-05 11:00". A table titled "Application Groups | Distribución de protocolos" shows traffic data for various applications. The table has columns for "Aplicación", "Tráfico(Total: 7.4 MB)", and "Percentageoftotaltraffic". The data is as follows:

Aplicación	Tráfico(Total: 7.4 MB)	Percentageoftotaltraffic
UDP App [Mostrar puertos]	6.51 MB	88%
http	644.57 KB	9%
mysql	101.42 KB	1%
pop3	39.89 KB	1%
TCP App [Mostrar puertos]	36.57 KB	<1%
netbios-ns	34.94 KB	<1%
smtp	22.1 KB	<1%
domain	2.64 KB	<1%
ftp	1.35 KB	<1%
netbios-dgm	1.09 KB	<1%

Below the table, there is a section titled "Tráfico de máximo consumo - Aplicación ENTRADA" showing a graph for "Router [ FastEthernet0/1 ]". The status bar at the bottom indicates "Terminado".

En aquellos casos donde se presente tráfico proveniente de puertos aleatorios y la aplicación los agrupe por protocolos de capa de transporte, aparece una opción al lado que dice “Mostrar puertos”.



## Paso 2

The screenshot shows the ManageEngine NetFlow Analyzer 6 web interface in Mozilla Firefox. A modal window titled 'Mostrar puertos' is open, displaying a table of traffic data for unmapped applications categorized as 'UDP\_App'. The table includes columns for Origin, Destination, Protocol, Source Port, Destination Port, and Traffic volume. Below the table, there is a 'Terminado' status indicator.

Origen	Destino	Protocolo	Puerto de origen	Puerto de dest	Tráfico	Añadir
192.168.100.3	192.168.101.2	UDP	1627	7777	2.0 MB	<a href="#">Add Mapping</a>
192.168.100.4	192.168.101.2	UDP	3737	7777	1.91 MB	<a href="#">Add Mapping</a>
192.168.100.5	192.168.101.2	UDP	3016	7777	1.3 MB	<a href="#">Add Mapping</a>
192.168.100.6	192.168.101.2	UDP	2494	7777	1.29 MB	<a href="#">Add Mapping</a>
192.168.100.5	255.255.255.255	UDP	7550	7550	312,00 Bytes	<a href="#">Add Mapping</a>

Al hacer click en la opción “Mostrar puertos”, aparece una ventana en la cual se observa el volumen de tráfico generado por cada puertos y direcciones IP de origen y destino.

# NETFLOW ANALYZER: DISTRIBUCIÓN GRÁFICA DE PROTOCOLOS

## Paso 1

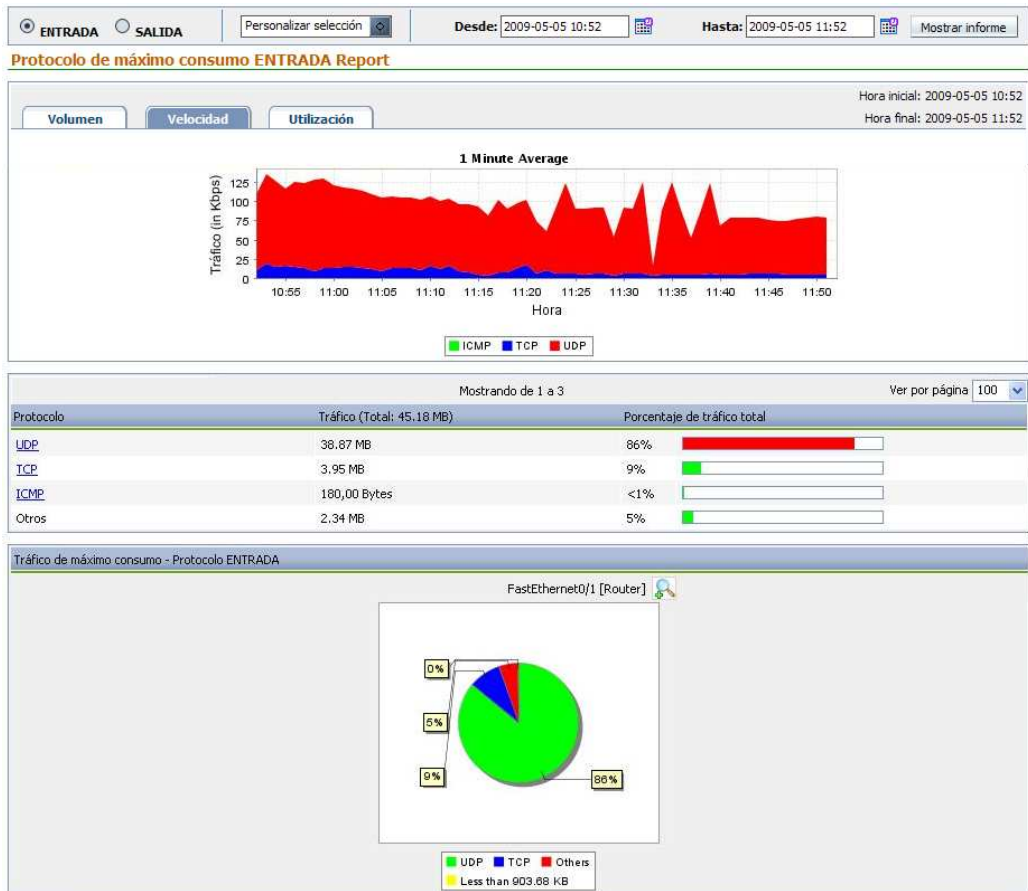
The screenshot displays the ManageEngine NetFlow Analyzer 7 Professional interface. The main content area shows a report titled 'Distribución de protocolos' (Protocol Distribution) for the 'ENTRADA' (Ingress) direction. The report is filtered for the date range 'Desde: 2009-05-05 10:52' to 'Hasta: 2009-05-05 11:00'. The total traffic is 7.4 MB. The table below lists the top applications and their respective traffic volumes and percentages.

Aplicación	Tráfico(Total: 7.4 MB)	Percentageoftotaltraffic
UDP_App [Mostrar puertos]	6.51 MB	88%
http	644.57 KB	9%
mysql	101.42 KB	1%
pop3	39.89 KB	1%
TCP_App [Mostrar puertos]	36.57 KB	<1%
netbios-ns	34.94 KB	<1%
smtp	22.1 KB	<1%
domain	2.64 KB	<1%
ftp	1.35 KB	<1%
netbios-dgm	1.09 KB	<1%

Below the table, there is a section for 'Tráfico de máximo consumo - Aplicación ENTRADA' (Maximum Traffic Consumption - Application ENTRADA) for Router [FastEthernet0/1], showing a 3% consumption.

Se hace click sobre la opción “Distribución de protocolos”.

## Paso 2



Se inicia una ventana emergente donde el administrador tendrá la opción de visualizar la grafica de protocolos para la entrada y la salida del router, así como analizarla por volumen, velocidad y utilización.

## BIBLIOGRAFIA

- [1] *Bandwidth Management and Queueing*. Febrero 2002, Cisco Systems.  
<http://www.cisco.com/warp/public/cc/pd/ibsw/ibdlsw/prodlit/dlsw5 rg.htm>
- [2] *Weighted Fair Queueing (WFQ)*, Cisco Systems.  
<http://www.cisco.com/warp/public/732/Tech/wfq/>
- [3] *Class-Based Weighted Fair Queueing*. Agosto 2003, Cisco Systems.
- [4] *Low Latency Queueing*. Enero 2003, Cisco Systems.
- [5] *Congestion Management Overview*. Enero 2003, Cisco Systems.
- [6] *Diferenciación de servicios por clase en redes inalámbricas*. Marzo 2003, Departamento de Ingeniería Telemática - Universidad Politécnica de Cataluña.
- [7] *Protocolos de Encolamiento para la Administración de Ancho de Banda con Routers Cisco*. Septiembre 2003, Patricio Denzer (Departamento de Electrónica - Universidad Técnica Federico Santa María).
- [8] *Data Communications and Networks (Hardcover)*, David Miller - McGraw-Hill/Irwin; 1 edition (January 14, 2005).
- [9] *Data Networks (2nd Edition) (Pie) (Paperback)*, Dimitri P. Bertsekas and Gallager - Prentice Hall; 2 edition (January 5, 1992).
- [10] *Quality of Service Control in High-Speed Networks (Hardcover)*, H. Jonathan Chao and Xiaolei Guo - Wiley-Interscience; 1st edition (November 15, 2001).
- [11] *High Performance Switches and Routers (Hardcover)*, H. Jonathan Chao and Bin Liu - Wiley-IEEE Press (April 6, 2007).
- [12] *Cisco CCNA Exploration 4.0*. Abril 2008, Cisco Systems.