

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



Trabajo Fin de Máster

Análisis e implementación de un sistema real de medida de ancho de banda

Autor:
Francisco Javier Ramos de Santiago

Director:
Dr. Javier Aracil Rico

Madrid, 2010

Análisis e implementación de un sistema real de medida de ancho de banda

TRABAJO FIN DE MÁSTER: Análisis e implementación de un sistema
real de medida de ancho de banda

AUTOR: Javier Ramos de Santiago

DIRECTOR: Dr. Javier Aracil Rico

El tribunal para la defensa de este trabajo fin de máster está compuesto por:

PRESIDENTE: Dr. Javier Aracil Rico

VOCALES: Dr. Jorge E. López de Vergara Méndez

Dr. Francisco Gómez Arribas

Dr. Sergio López Buedo

Agradecimientos a:

Mis padres, por apoyarme y hacer de mi lo que soy.

Mi familia y amigos por su ayuda.

Victoria por su apoyo y comprensión.

Mi tutor Javier así como a las otras personas que han participado de una u otra forma en este trabajo: Jorge, Pedro, Sergio y Paco. Gracias por guiar este trabajo con ideas, sugerencias y consejos.

Todos las personas del grupo HPCN de la EPS por todas las cosas que me han enseñado y los buenos momentos que hemos pasado.

Índice

1. Introducción	3
2. Estado del arte	5
2.1. Conceptos generales	5
2.1.1. ¿Qué es y qué se entiende por ancho de banda?	5
2.1.2. Capacidad	5
2.1.3. Ancho de banda disponible	6
2.1.4. Throughput TCP	6
2.2. Traffic Shaping y Traffic Policing	7
2.2.1. Token Bucket	10
2.2.2. Policing usando Token Bucket	11
2.2.2.1. Single Rate Three Color Marker (srTCM)	11
2.2.2.2. Two Rate Three Color Marker (trTCM)	12
2.2.3. Shaping usando Token Bucket	12
2.3. Disciplinas de encolado	13
2.3.1. FIFO (First In First Out)	13
2.3.2. PQ (Priority Queueing)	13
2.3.3. FQ (Fair Queueing)	14
2.3.4. WFQ (Weight Fair Queuing)	15
2.3.5. WRR (Weighted Round Robin)	15
2.3.6. SRR (Shaped Round Robin)	16
2.4. Limitaciones físicas: el caso de ADSL	16
2.5. Problemas derivados de las limitaciones de ancho de banda	19
2.6. Métodos de medida de ancho de banda basados en descarga de fichero	21
2.7. Métodos de medida de ancho de banda basados en pares de paquetes	23

Trabajo de Fin de Máster: Análisis e implementación de un sistema real de
medida de ancho de banda

2.7.1. Packet-Pair	23
2.7.2. Packet-Pair Train	25
3. Evaluación de entornos de medida	26
3.1. Accesos limitados físicamente: el caso de ADSL	26
3.2. Accesos limitados lógicamente	27
4. Factores que influyen en la medida del ancho de banda	29
4.1. Descripción del entorno de laboratorio	30
4.2. CPU	31
4.3. Memoria	35
4.4. Sistema Operativo y Hardware	39
4.5. Tráfico interferente	40
4.6. Interrupt Coalescence	43
5. Implementación de un sistema real: GoSPoll	45
5.1. Arquitectura	45
5.2. Diseño	46
5.3. Cálculo de la calidad de la medida	50
6. Validación del sistema	52
6.1. Entorno de laboratorio	53
6.2. Entorno Real	55
7. Conclusiones	58
8. Trabajo Futuro	59
Apéndices	61

1. Introducción

Con el desarrollo de nuevos servicios multimedia y la implantación de las redes de banda ancha en los hogares se ha producido una creciente demanda de métodos que permitan medir parámetros de calidad de servicio (QoS). Esta demanda proviene tanto de los usuarios finales como de operadoras de telecomunicaciones y organismos reguladores.

En el 2005 se recibieron 43,286 peticiones en la oficina de atención a usuarios de telecomunicaciones y 234,708 en 2008. La mayoría de las peticiones son quejas de usuarios a causa de degradaciones del servicio contratado por parte del operador. Es por ello que tanto usuarios como operadores están interesados en medir de manera precisa y estándar los parámetros de calidad de servicio de las conexiones de banda ancha.

Uno de los parámetros de calidad de servicio que tiene especial relevancia es el ancho de banda. Este parámetro es importante tanto para usuarios domésticos que pueden contar con accesos de ADSL de baja velocidad como para empresas que contratan líneas dedicadas sujetas un acuerdo de nivel de servicio (SLA). Aparte, tanto operadoras como organismos oficiales tienen interés en esta métrica ya que su medida constituye un proceso de verificación del cumplimiento del contrato entre un operador y un cliente.

Para medir este parámetro muchos usuarios utilizan programas que se ejecutan a nivel de usuario [25] y que deben competir por los recursos (CPU, Memoria, Red) con otras aplicaciones concurrentes. Estas aplicaciones a menudo pueden generar tráfico que influye en la propia medida de ancho de banda. Es por ello que a la hora de realizar medidas de ancho de banda se debe tener en cuenta no sólo factores puramente relacionados con las comunicaciones si no también factores relacionados con el rendimiento global de un equipo o sistema.

En el trabajo se analizan las diferentes técnicas de medida de ancho de banda así como los métodos de limitación de ancho de banda para profundizar en los aspectos que pueden ser más importantes a la hora de realizar una medida. Estas técnicas de limitación de ancho de banda engloban tanto limitaciones físicas, como las impuestas en el caso de las líneas ADSL, como limitaciones lógicas debidas a Shaping, Policing o técnicas de encolado. Además se aborda el problema de la sobre-suscripción que causa los cuellos de botella que hacen que se incumplan los niveles especificados entre clientes y operadoras en los

contratos o acuerdos de nivel de servicio (Service Level Agreement -SLA-)

Este trabajo analiza también la influencia de los factores externos sobre los métodos de medida basados tanto en descarga de fichero como en pares de paquetes. Para cada uno de ellos se analiza la influencia de la carga de CPU y memoria así como del tráfico interferente auto-inducido. Para ello se usó una metodología formal para el diseño de experimentos.

Adicionalmente se propone un método de medida de ancho de banda que es inmune a parámetros relacionados con el rendimiento (carga CPU y memoria) y que, además, es capaz de analizar el tráfico interferente auto-inducido por otras aplicaciones y evaluar la validez del mismo.

El método ha sido implementado en una plataforma de medida de ancho de banda llamada QoSPoll que usa varios métodos de medida y que además permite obtener una estimación de la calidad de la medida analizando los datos de rendimiento más relevantes de los equipos en los cuales se ejecuta. Además, se ha realizado una solicitud de patente de invención para este método.

El trabajo se organiza de la siguiente manera: en la sección 2 se analiza el estado del arte en cuanto a lo que a métodos de medida de ancho de banda se refiere, analizando las diversas opciones y haciendo un repaso a las diferentes técnicas de limitación del ancho de banda tanto físicas como lógicas y a los problemas que presentan. En la sección 3 se evalúan los diferentes escenarios en los que el método de medida aplica así como sus problemas e inconvenientes. En la sección 4 se tratan los parámetros que influyen en una medida de ancho de banda así como su medida y valoración. En la sección 5 se propone y analiza la implementación real del sistema QoSPoll seguido de los resultados de los experimentos tanto controlados como en entornos reales y la validación del método usado en la sección 6. En la sección 7 se comentan las conclusiones que se han obtenido de este trabajo para finalmente exponer ideas sobre trabajos futuros en la sección 8.

2. Estado del arte

2.1. Conceptos generales

2.1.1. ¿Qué es y qué se entiende por ancho de banda?

Antes de profundizar en los diferentes métodos de medida del ancho de banda es necesario definir una serie de conceptos básicos que engloban las diferentes acepciones del término “Ancho de banda ” ya que al ser un término extendido en varios entornos su uso puede llevar a confusiones.

Desde el punto de vista de un usuario final el ancho de banda es la velocidad que percibe al navegar por Internet o hacer uso de un servicio concreto. Generalmente esta visión se puede relacionar con el concepto throughput de una conexión TCP.

Desde el punto de vista de un operador el ancho de banda es la capacidad de un enlace o enlaces a través de los cuales proporciona un acceso o servicio a un usuario. Por tanto, su visión del ancho de banda es a más bajo nivel que la de un usuario final.

Por último también se puede hacer referencia al término “Ancho de banda” cuando en realidad se habla de ancho de banda disponible. Este caso puede corresponder al de un distribuidor de contenidos en la red que percibe el ancho de banda como la cantidad de información por unidad de tiempo que puede enviar en cada momento o la cantidad de “enlace” que puede usar en cada momento.

2.1.2. Capacidad

La capacidad de un enlace de nivel 2 se define como la tasa de transmisión constante. Esta tasa de transmisión constante está limitada por las características físicas del medio de transmisión y por las características eléctricas/ópticas del hardware de los emisores y receptores.

En la capa IP se define la capacidad como la tasa de transmisión teniendo en cuenta el overhead introducido por las cabeceras del nivel de enlace. La capacidad en este caso puede describirse como:

$$C_{L3} = C_{L2} \frac{1}{1 + \frac{H_{L2}}{L_{L3}}} \quad (1)$$

donde C_{L2} es la capacidad de un enlace a nivel 2, H_{L2} es la longitud de las cabeceras y preámbulos del nivel 2 y L_{L3} es el tamaño del paquete IP en bytes.

Se define la capacidad C_i de un salto i como la máxima tasa de transmisión a nivel IP de ese salto. Por tanto, la capacidad de un salto se calcula siempre tomando el tamaño del paquete IP igual a la MTU del nivel 2.

La capacidad C de un camino end-to-end se define como el mínimo de las capacidades de los saltos que componen ese camino. El salto con la menor capacidad en un camino es conocido como el *narrow link*.

2.1.3. Ancho de banda disponible

El ancho de banda disponible de un camino end-to-end se define como la capacidad no usada durante un periodo de tiempo. Por tanto, esta métrica depende tanto de las características físicas como de la carga de tráfico en los enlaces a lo largo del tiempo. Para calcular el ancho de banda disponible es necesario saber la utilización de los enlaces. Como esta utilización instantánea no es muy útil, se define la utilización media en un periodo de tiempo como:

$$\bar{u}(t - \tau, t) = \frac{1}{\tau} \int_{t-\tau}^t u(x) dx. \quad (2)$$

donde $u(x)$ es la utilización instantánea del enlace que puede ser 0 o 1 y τ es el intervalo de tiempo o *averaging timescale*

A partir de esta definición se puede ver que el ancho de banda medio disponible de un salto i es:

$$\bar{A}_i = (1 - \bar{u}_i) C_i \quad (3)$$

y el ancho de banda disponible de un camino end-to-end es el mínimo de los anchos de banda disponibles en cada salto. El enlace con el mínimo ancho de banda disponible se conoce como *tight link*. Como el ancho de banda disponible cambia a lo largo del tiempo, es necesario medirlo rápidamente especialmente cuando la medida se usa para adaptar la distribución de contenido de una o varias aplicaciones.

2.1.4. Throughput TCP

Otra métrica importante a la hora de hablar de ancho de banda es el throughput de una conexión TCP. La desventaja de esta métrica es que depende de

diferentes factores como el tamaño de la transferencia de datos, el número de conexiones TCP activas, la congestión en los enlaces de retorno o la cantidad de tráfico TCP o UDP cruzado. Bulk Transfer Capacity [17] es una métrica que representa el throughput de una conexión TCP.

2.2. Traffic Shaping y Traffic Policing

Además de la diversidad de conceptos usados bajo un mismo término existe una diferencia clara, no explícita generalmente, que es la diferencia entre el ancho de banda medio o de pico. El ancho de banda medio especifica cual es el valor medio de la magnitud medida (capacidad, ancho de banda disponible o throughput TCP) en un intervalo determinado de tiempo. Esta magnitud no tiene que mantenerse forzosamente constante en el tiempo pudiendo alcanzar así máximos y mínimos que distan del valor medio. Por el contrario el ancho de banda de pico expresa el máximo valor de la magnitud a lo largo del tiempo. Es por ello que la magnitud podrá decrecer de valor pero nunca superar este tope.

Generalmente el uso de uno u otro término depende de la tecnología que subyace y de las limitaciones físicas o lógicas que asociadas a la misma.

Estas limitaciones tanto físicas como lógicas influyen a la hora de utilizar los diferentes métodos de medida de ancho de banda y es importante conocerlas y comprenderlas para saber qué se va a medir y de qué manera.

Traffic Shaping y Traffic Policing son métodos lógicos para la limitación y control del ancho de banda en redes de comunicaciones. Esta limitación se usa para controlar y asegurar que un paquete o una fuente de datos se adhiere a un determinado contrato o acuerdo de nivel de servicio. Ambos métodos intentan mantener la tasa de tráfico por debajo de un cierto nivel previamente determinado. La diferencia entre ellos radica en la manera de tratar el tráfico y en los ámbitos en los que son aplicables estos métodos.

Traffic Policing usa una política de eliminación o marcado de los paquetes que pertenecen a ráfagas que exceden la tasa de tráfico marcada como límite. Este método no modifica el tráfico de ningún otro modo de manera que se conserva el retardo y características del tráfico original. Se conservan por tanto las ráfagas de tráfico contenidas en el tráfico original siempre que no excedan el límite. Esta técnica puede ser aplicada tanto en sentido entrante como en sentido de

salida. Su principal desventaja consiste en la pérdida del tráfico que excede la tasa fijada lo cual puede penalizar mucho el rendimiento de los flujos TCP que estén activos debido a las retransmisiones que se deben hacer. En la figura 1 se observa el comportamiento de este método.

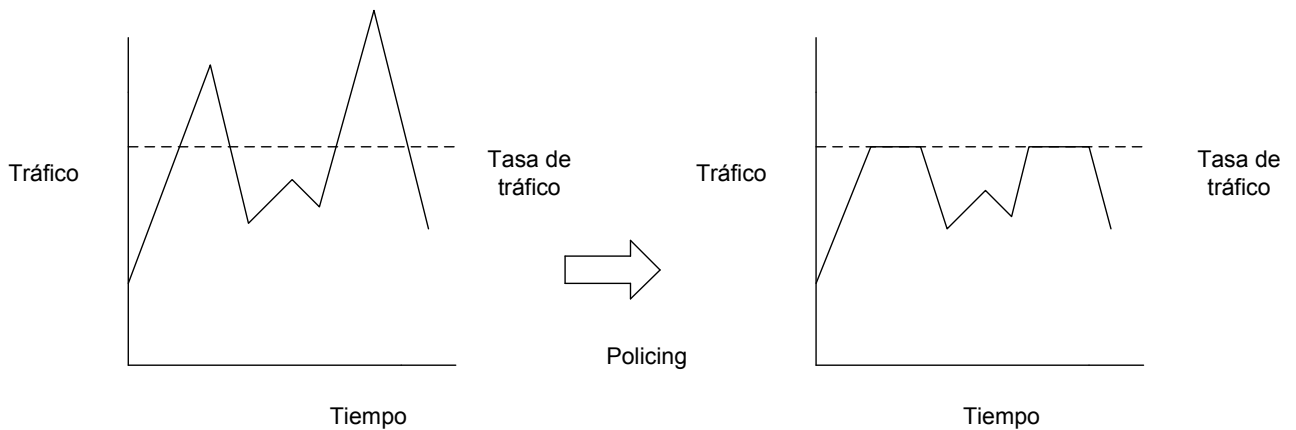


Figura 1: Traffic Policing

Traffic Shaping usa una política basada en el encolado y la posterior extracción de paquetes para mantener una tasa más o menos constante de tráfico. A diferencia de Traffic Policing las características y retardos originales del tráfico desaparecen debido al encolado. La ventaja principal de este método aparte de mantener el tráfico en el entorno de una cierta tasa de tráfico fijada es que permite controlar las ráfagas del tráfico entrante para obtener un tráfico más suave y continuo en sentido de salida. Aparte, este método reduce las pérdidas cuando el tráfico entrante supera la tasa de tráfico fijada reduciendo así, el número medio de retransmisiones en las comunicaciones. La principal desventaja de este método es el aumento en los retardos debido al encolado lo cual puede afectar negativamente a sistemas en tiempo real o multimedia. En la figura 2 se observa el comportamiento de este método.

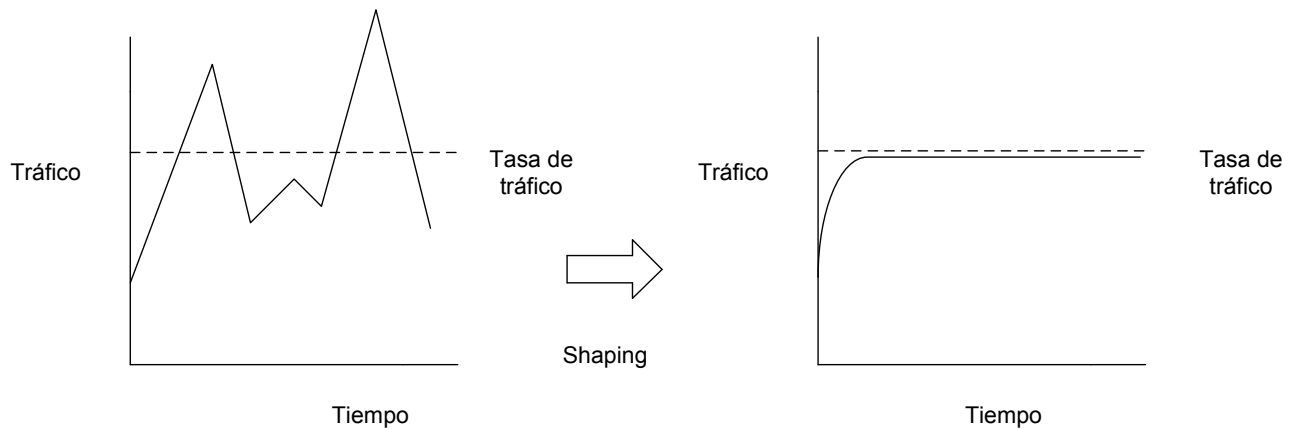


Figura 2: Traffic Shaping

En ambos casos siempre se toma en cuenta una serie de parámetros para ajustar la tolerancia y el funcionamiento de estos métodos. Estos parámetros son:

- PIR (Peak Information Rate): tasa máxima de emisión de un cliente expresada en bits/s y acordada previamente entre cliente y operador mediante algún tipo de contrato o acuerdo de nivel de servicio (SLA). El PIR nunca puede ser mayor que la capacidad de la línea que provee el operador.
- CIR (Committed Information Rate): tasa media de tráfico a largo plazo que el operador se compromete a dar a un cliente mediante un contrato o acuerdo de nivel de servicio (SLA). Este parámetro se mide en bits/s y generalmente es menor al PIR. En cualquier caso el CIR nunca puede ser mayor al PIR.
- CBS (Committed Burst Size): tamaño máximo de las ráfagas permitidas en la red. Especifica el número máximo de bytes que pueden ser transmitidos al PIR sin dejar de cumplir con el acuerdo del CIR.
- EBS (Excess Burst Size): umbral de tamaño de ráfagas superior al CBS. Cuando las ráfagas exceden el EBS los paquetes que contienen se marcan para eliminarlos.
- PBS (Peak Burst Size): parámetro similar a CBS pero definido respecto al PIR en vez de respecto al CIR.

2.2.1. Token Bucket

Para realizar tanto *Shaping* como *Policing* se usa generalmente un algoritmo que se encarga de controlar la tasa de transmisión en una red teniendo en cuenta ráfagas de datos. Este algoritmo se conoce como Token Bucket. El algoritmo se basa en la existencia de un “cubo” de *tokens*. Un *token* representa una cantidad de bytes o un paquete de tamaño determinado. Cada vez que se desea enviar un paquete o conjunto de paquetes se comprueba que haya *tokens* suficientes en el cubo como para enviar la cantidad deseada de paquetes o bytes. Una vez enviados el paquete o paquetes se eliminan los *tokens* correspondientes del “cubo”. Cuando el cubo queda vacío, si se desea enviar un paquete o paquetes se debe esperar a que se generen nuevos tokens. Es mediante este método de gasto y generación de *tokens* como se controla la tasa de salida de datos. A continuación se definen las siguientes variables:

- r : Tasa de generación de tokens
- d : Tamaño máximo del cubo
- C : Tasa máxima de transmisión del enlace de salida
- B : Tamaño del buffer

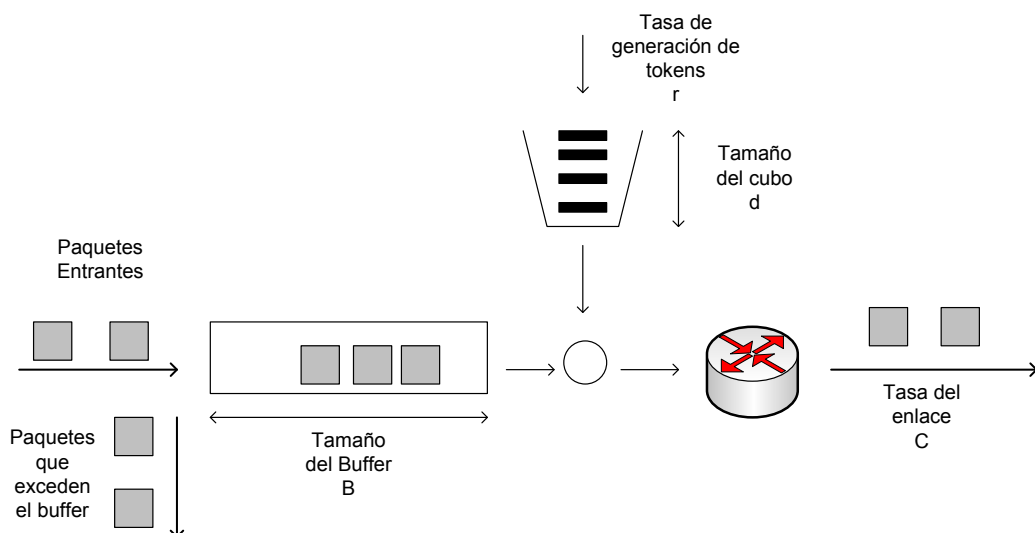


Figura 3: Diagrama del método Token Bucket

El funcionamiento del método es el siguiente. Se generan tokens a una tasa constante r y se almacenan en el cubo. En caso de que el cubo exceda su capacidad máxima d los nuevos tokens generados son eliminados. Por otra parte

un token permite el envío de un byte de un paquete. Cuando se quiere enviar un paquete se comprueba que haya tokens suficientes. En caso de que haya tokens disponibles el paquete se envía. En el caso contrario dependiendo de si se usa una cola o no, el paquete será encolado o eliminado respectivamente. Este método permite que se envíen ráfagas de tamaño máximo d a tasa C cuando el cubo está completamente lleno. Por el contrario si el cubo está completamente vacío, los paquetes se enviarán a una tasa r . De esta manera se puede especificar tanto el CIR (parámetro r), como el PIR (parámetro C) y el CBS (parámetro d) para una conexión.

Entender este método de limitación tiene importancia a la hora de medir ya que en muchos casos las líneas que se desean medir usan este mecanismo y el concepto a medir como “ancho de banda” es el CIR que se ha especificado en un contrato o acuerdo de nivel de servicio entre el cliente y el operador.

2.2.2. Policing usando Token Bucket

2.2.2.1. Single Rate Three Color Marker (srTCM)

Este método [12] se usa para realizar policing a una única tasa (CIR). El método mide la tasa de tráfico de los paquetes y en función de una serie de parámetros marca los paquetes con tres colores: verde, amarillo y rojo. A la hora de descartar paquetes, los rojos y amarillos serán los primeros en ser eliminados. Los parámetros que usa este método para limitar el ancho de banda son CIR, CBS y EBS.

El objetivo principal es asegurar que la tasa media de un flujo está en el entorno del CIR. Para ello se usan dos token buckets llamados C y E. El tamaño máximo del cubo C es CBS y el del cubo E es EBS. La tasa de generación de tokens para ambos cubos es el CIR (cada $1/\text{CIR}$ segundos se genera un token). Cada vez que se genera un token se añade a los cubos en caso de que no estén llenos.

Cuando llega un paquete de B bytes, se comprueba que en el cubo C hay suficientes tokens como para dejar pasar el paquete. Si C tiene suficientes tokens el paquete se marca como verde y se eliminan los tokens correspondientes al tamaño B del cubo C. En caso contrario, se comprueba si en el cubo E hay suficientes tokens como para que el paquete pase. Si en el cubo E hay tokens suficientes el paquete se marca como amarillo y se eliminan los tokens correspondientes al tamaño B del cubo E. Si ni el cubo C ni el cubo E hay tokens

suficientes el paquete se marca como rojo.

Usando este método se asegura que la tasa media está cercana al CIR permitiendo ciertas ráfagas en función de los valores de CBS y EBS.

2.2.2.2. Two Rate Three Color Marker (trTCM)

Este método [13] se usa para hacer policing sobre CIR y PIR de manera independiente. Al igual que el algoritmo anterior, se colorean los paquetes entrantes de verde, amarillo o rojo. Para este método se usan los parámetros PIR, CIR, PBS y CBS.

El método usa dos token bucket llamados C y P. El cubo C tiene una capacidad máxima de CBS tokens y se llena a una tasa CIR. El cubo P tiene una capacidad máxima de PBS tokens y se llena a una tasa constante PIR. Inicialmente los cubos se encuentran llenos y cada vez que se genera un token se añade al cubo correspondiente a no ser que esté lleno.

Cuando un paquete de B bytes llega, se comprueba si en el cubo P hay tokens suficientes para que el paquete pase. Si no hay tokens suficientes el paquete se marca como rojo. Si el cubo P tiene suficientes tokens se comprueba si en el cubo C hay suficientes tokens como para que pase el paquete. Si no hay suficientes tokens en el cubo C el paquete se marca como amarillo y se eliminan los tokens que ha gastado del cubo P. Si hay suficientes tokens en los dos cubos, el paquete se marca como verde y se elimina de ambos cubos los tokens usados para que el paquete pasase.

2.2.3. Shaping usando Token Bucket

Para realizar shaping se usa un token bucket similar al usado para hacer policing del CIR en los modelos srTCM y trTCM. Los tokens entran en el cubo a una tasa constante r si este está vacío. Si no, los tokens generados no se almacenan. Para realizar shaping es necesario que haya una cola previa donde el tráfico entrante es almacenado. Además de la tasa r existe una tasa C a la cual se sacan los tokens del cubo para permitir el paso de paquetes. De esta manera ajustando esta tasa C y la tasa r se consigue que el tráfico a ráfagas se regule y haya una salida más o menos constante a tasa C .

2.3. Disciplinas de encolado

Aparte de las limitaciones que pueden ser impuestas de manera física o lógica, a la hora de realizar medidas del ancho de banda se debe tener en cuenta que entre los dos puntos de la medida puede haber varios saltos en cada uno de los cuales probablemente existan colas que modifiquen el comportamiento de los paquetes en la red. En un sistema de colas existen diversas disciplinas de encolado que permiten organizar el tráfico y tratarlo de una u otra manera. Algunas de estas disciplinas de encolado pueden incluso en algunas situaciones actuar como limitadores de ancho de banda lo cual influiría claramente en la medida que se realizase. A continuación se describen brevemente algunas disciplinas de encolado comentando sus ventajas e inconvenientes así como la manera y condiciones en que pueden limitar el ancho de banda.

2.3.1. FIFO (First In First Out)

La disciplina FIFO es la disciplina de encolado usada por defecto si no especifica una. Es la disciplina más simple y consiste en que los paquetes se encolan en el orden en que llegan y salen de la cola en el mismo orden. Esta disciplina se conoce a veces como FCFS (First Come First Served). La principal ventaja de este método es la simplicidad a la hora de implementarlo.

FIFO trata de la misma manera a todos los paquetes lo cual hace que sea un buen método para los paquetes marcados para *Best effort*. Este hecho hace que no sea especialmente buena para el resto del tráfico ya que al no diferenciar clases de tráfico todos los flujos se ven perjudicados de la misma manera ante un estado de congestión. Por ejemplo la disciplina FIFO favorece más a los flujos UDP que a los TCP cuando hay congestión ya que TCP disminuye su tasa para minimizar las pérdidas y además usa reenvíos.

2.3.2. PQ (Priority Queueing)

PQ soluciona en cierta manera los problemas de la disciplina FIFO definiendo N colas con prioridad 1 a N de tal manera que se puede clasificar el tráfico en diferentes clases. El funcionamiento del método se basa en que sólo se envían los paquetes de la cola j si no hay paquetes encolados en las colas 1 a (j-1).

Este método mantiene la simplicidad de las colas FIFO aportando la posibilidad

de crear clases de tráfico diferencias. La principal desventaja de este método es que se puede producir *inanición* de tal manera que si siempre hay paquetes en las colas de prioridad alta, las colas de prioridad baja no son atendidas nunca. Esta disciplina está especialmente recomendada para escenarios en los que el tráfico de alta prioridad representa una pequeña fracción del tráfico total. Por ejemplo se puede usar esta disciplina para crear colas de tiempo real, vídeo o audio sobre IP. Si se usa TCP se debe tener cuidado ya que los mecanismos de TCP cuando hay congestión hacen que la inanición sea más grave para el resto del tráfico que hay en otras colas.

Para solucionar el problema de la inanición existe una modalidad de encolado llamada *rate controlled PQ* que determina umbrales de uso de la salida para cada cola. De esta manera mientras que las colas de mayor prioridad hagan un uso menor que el umbral predefinido, el método funciona como PQ. Cuando el umbral es superado se restringe el tráfico para favorecer a las colas de menor prioridad.

2.3.3. FQ (Fair Queueing)

Una manera de crear colas con clases de tráfico diferentes es FQ que realiza una clasificación basada en flujos. Cada paquete que llega al sistema se clasifica en una cola de 1 a N. Cada cola tiene asignado $1/N$ del ancho de banda total del puerto de salida. El planificador recorre las N colas en orden (*round-robin*) enviando un paquete de cada cola si ésta no esta vacía.

Esta disciplina es fácilmente implementable y escalable de manera que añadir una nueva cola no implica grandes cambios en el algoritmo. Una de las desventajas de este método es que al estar dividido el ancho de banda en $1/N$ partes iguales los flujos que requieren un mínimo de ancho de banda se ven perjudicados. Este puede ser el caso de los flujos dedicados a la medida del ancho de banda ya que se necesitaría tener disponible todo el ancho de banda del enlace de salida y solo se obtendría $1/N$ del ancho de banda.

Otro de los problemas de FQ es que al transmitirse un paquete de cada cola, los flujos con tamaños de paquete grande acabarán por obtener más de $1/N$ del ancho de banda del puerto de salida. Por ende, los flujos con tamaños de paquete pequeños obtendrán menos ancho de banda lo cual puede afectar negativamente si son flujos de medida de ancho de banda o servicios multimedia sobre IP.

2.3.4. WFQ (Weight Fair Queuing)

Esta disciplina de encolado que es una aproximación del método GPS (Generalized Processor Sharing) [21] divide el tráfico entrante en N colas. El ancho de banda del puerto de salida se divide entre las N colas en función de pesos asignados a cada cola. La suma de los pesos siempre tiene que ser 100%. En WFQ a diferencia de FQ el planificador saca de las N colas los paquetes en orden de su tiempo de finalización. Este tiempo de finalización se estima haciendo una aproximación al modelo teórico *Weighted bit-by-bit round robin*. Este modelo recorre las colas usando round robin sacando cada vez un bit. Cuando un paquete está completo se envía de manera que los paquetes grandes esperan más tiempo a ser enviados que los pequeños. Esta aproximación es sólo teórica y no es aplicable ya que reensamblar los bits de cada trama una vez extraídos de las colas requeriría un procesamiento mayor.

Para un enlace de salida de capacidad C, la tasa de salida de la clase i con el peso w_i viene dada por:

$$C_i = \frac{Cw_i}{w_1 + w_2 + \dots + w_N} \quad (4)$$

Esta disciplina solventa los problemas que presentaba FQ y permite un reparto diferenciado por clases del ancho de banda de salida. En una situación en la que todas las colas tienen tráfico, el ancho de banda de un flujo queda limitado por la fórmula 4 lo cual impone una limitación a la hora de medir el ancho de banda en sistemas que tengan este tipo de encolado.

2.3.5. WRR (Weighted Round Robin)

Para solucionar el problema de los flujos que tienen requisitos determinados de ancho de banda en FQ se usa la disciplina WRR o también conocida como CBQ(Class-Based Queueing). Este método divide los flujos de entrada en m clases diferentes en función de sus requisitos de ancho de banda. Cada una de estas clases lleva asociado un peso y la suma de todos los pesos de las clases es 100%. Dentro de cada clase i existen N_i colas FQ.

A la hora de sacar paquetes de las colas el planificador visita las m clases en orden round-robin. Además dentro de cada clase las colas FQ se visitan en orden round-robin también. El peso individual de cada cola FQ j dentro de la

clase i viene determinado por:

$$W_{ij} = W_i \times \frac{1}{N_i}; j = 0, 1, \dots, N_i \quad (5)$$

Al igual que en el caso anterior cuando todas las colas están llenas, el ancho de banda de un flujo puede verse limitado debido al peso de la cola que lo contiene e influir negativamente a la hora de realizar medidas del ancho de banda del enlace.

Aunque soluciona el problema de la distribución del ancho de banda de los flujos con requisitos, este método no soluciona el desequilibrio producido por la diferencia en los tamaños de los paquetes de los flujos.

2.3.6. SRR (Shaped Round Robin)

SRR es una disciplina adoptada por algunos fabricantes de routers como Cisco la cual se basa en especificar la tasa a la que los paquetes pueden ser extraídos de una cola. De esta manera se controla de manera estricta el ancho de banda asignado a cada cola. Para realizar la limitación se recorren las colas en orden round-robin y se establecen tiempos en los que se pueden extraer paquetes de las colas para ajustar el ancho de banda máximo para cada cola. SRR tiene un modo de funcionamiento más relajado que el *shaped* denominado *shared* que permite hacer uso del ancho de banda asignado a otras colas en caso de que estas estén vacías.

El modo *shared* se usa cuando se quiere obtener la máxima eficiencia en un sistema de colas ya que las colas que están ociosas pueden ser usadas para almacenar el exceso de tráfico de otras colas. El modo *shaped* se usa cuando se quiere establecer un límite estricto del ancho de banda máximo que puede usar una cola. Esta limitación puede ser usada junto con otras limitaciones de ancho de banda aplicadas anteriormente.

2.4. Limitaciones físicas: el caso de ADSL

Las limitaciones en los anchos de banda de un enlace pueden no ser sólo lógicas como anteriormente se ha comentado sino que pueden tener motivos físicos

reales que hacen que no se pueda superar una determinada tasa de transmisión. A la hora de medir un enlace se deben tener en cuenta estas limitaciones. Un ejemplo claro de estas limitaciones en el ámbito de este trabajo son las líneas ADSL que son usadas frecuentemente en hogares y pequeñas y medianas empresas.

Las líneas ADSL hacen uso del par de cobre empleado en la telefonía convencional para transmitir datos. Para ello los datos se modulan en una banda de frecuencias más altas que las utilizadas para la voz. Estas líneas son asimétricas y por tanto los anchos de banda de subida y bajada no son iguales. Por lo general, las líneas ADSL proveen más ancho de banda de bajada que de subida.

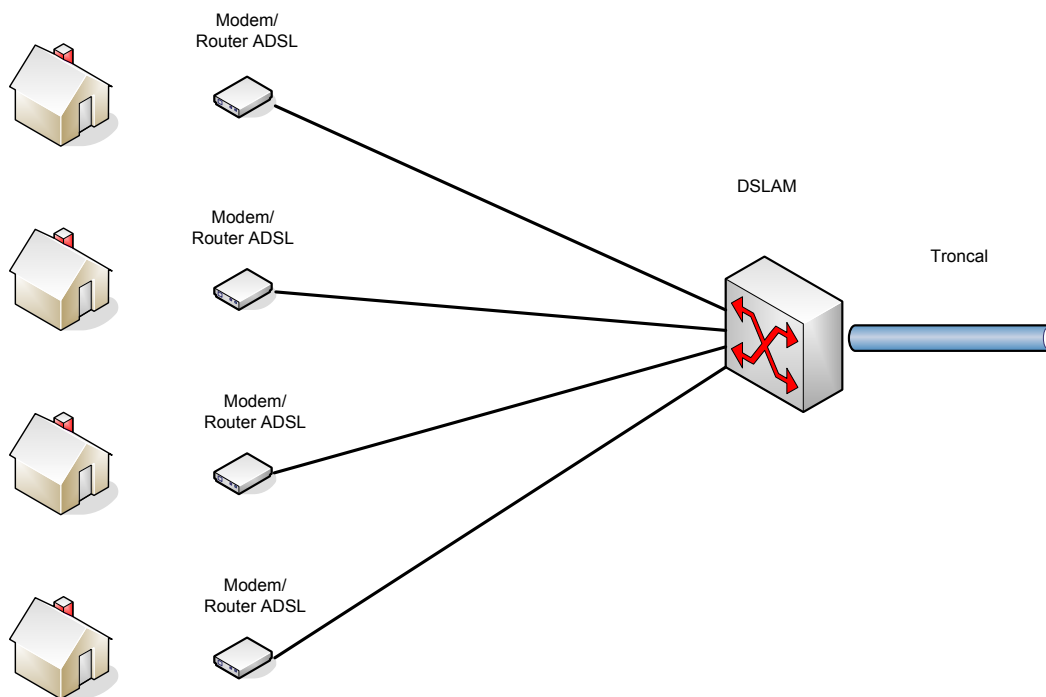


Figura 4: Esquema básico ADSL

En la figura 4 se puede observar un esquema básico en el que varios usuarios se conectan mediante líneas ADSL a un DSLAM (Digital Subscriber Line Access Multiplexer) que permite multiplexar el tráfico de varios usuarios a un troncal de mayor capacidad que puede usar tanto tecnología ATM como Ethernet. Observando este esquema se puede ver que el punto con mayor probabilidad de ser el cuello de botella en este tipo de líneas es el último salto desde el DSLAM al equipo final del usuario.

Para determinar la limitación física de la línea se realiza un proceso de sincronización entre el equipo del usuario (módem/router ADSL) y el puerto asignado al mismo en el DSLAM. Este proceso de sincronización analiza parámetros tanto físicos como de los equipos para determinar la velocidad máxima de transmisión de la línea. El proceso de inicialización se divide en 4 pasos fundamentales:

Handshake: En esta fase basada en la recomendación G.994.1 [11], se identifican los transceptores de ambos extremos así como sus características de manera que se acuerde un protocolo común entre ambos extremos para el resto de las fases de inicialización. Como en esta primera fase la comunicación es muy importante se usa una modulación BPSK para asegurar que será resistente ante fallo y ruido en el canal.

El funcionamiento básico consiste en el intercambio de mensajes entre los dos extremos que contienen el tipo de equipo final y una serie de subparámetros como el rango de frecuencias o el número de subportadoras DMT (Discrete Multitone Modulation) soportadas.

Entrenamiento de transceptores: Esta es la segunda fase del proceso de inicialización y en ella se obtiene el flujo de símbolos DMT, se ajustan las ganancias y ecualizadores en caso de que los haya. También se estiman las atenuaciones en los receptores y se ajustan las potencias de los emisores para asegurar un correcto funcionamiento.

Caracterización del canal: En esta fase los extremos intercambian información sobre sus capacidades y se realiza una caracterización detallada del canal. Parte de la información que intercambian es: margen mínimo SNR, posibilidad de cancelación de eco y posibilidad de usar codificación Trellis.

En esta fase se miden parámetros como número de subportadoras que pueden ser usadas, atenuación por cada subportadora y SNR. Usando esta información y las restricciones impuestas para cada línea en función del servicio contratado, el DSLAM ofrece al cliente hasta 4 tasas de transferencia en orden decreciente. El equipo situado en el lado del cliente decide cual usar aunque generalmente será la mayor de todas las tasas ofrecidas.

Aplicación de las tasas de transmisión: En esta fase se aplican las modulaciones a cada subportadora de manera que queden determinadas las tasas de transmisión tanto de subida como de bajada. El DSLAM asigna las ganancias y modulaciones a las subportadoras de bajada y el equipo del cliente las ganancias y modulaciones a las subportadoras de subida. En esta fase final se pasa de la modulación BPSK o QPSK usada para asegurar las transmisiones durante la fase inicial a las modulaciones QAM de orden superior que permiten las transmisión de datos a mayor velocidad.

Una vez negociadas las modulaciones y fijadas las tasas de transmisión queda impuesta una limitación física que impide enviar datos a una tasa superior a la máxima a diferencia de lo que pasaba en las líneas limitadas mediante Policing o Shaping. Esta limitación física debe ser tomada en cuenta a la hora de realizar medidas de ancho de banda ya que simplifica el proceso de medición y no impone restricciones como las existentes en los algoritmos basados en Token Bucket.

Aparte de las limitaciones físicas se pueden encontrar limitaciones lógicas como las anteriormente mencionadas entre los dos extremos de un camino usado para medir el ancho de banda.

2.5. Problemas derivados de las limitaciones de ancho de banda

Las causas por las cuales un cliente no está obteniendo la tasa de transferencia especificada en un contrato o acuerdo de nivel de servicio son debidas siempre a un punto en la red que no funciona como debería. Este punto puede encontrarse tanto en el último salto, lo cual suele ser habitual, como en algún salto del camino entre el cliente y el recurso al que desea acceder, por ejemplo un servidor HTTP.

Generalmente estos problemas en la red se producen por una mala planificación del uso de los recursos que comprenden lo cual es producido en gran medida por la sobre-subscripción. Basado en el principio de que no todos los clientes están activos al mismo tiempo, se realiza una multiplexación estadística. Al realizar el dimensionado de las redes y equipos no se tiene en cuenta el número máximo de usuarios si no más bien el número medio de usuarios activos en cada periodo de tiempo. De esta manera se produce el problema de

la sobre-subscripción en la que un operador puede ofrecer servicio a más clientes de los que puede atender en un tiempo dado. Haciendo una analogía con el sector de los transportes, la sobre-subscripción es vender más billetes que plazas hay en el transporte en cuestión.

Por ejemplo en el caso de las líneas ADSL la sobre-subscripción se produce generalmente en la salida del DSLAM. El DSLAM puede tener muchas tarjetas a través de las cuales da acceso a Internet a sus clientes. El número de tarjetas que puede tener un DSLAM puede variar entre 4 y 32 aproximadamente. Cada una de estas tarjetas puede proveer acceso a entre 20 y 25 usuarios lo cual supone un número máximo aproximado de 800 usuarios por DSLAM. Si suponemos que en media estos usuarios pueden tener contratados servicios de 2Mbps, a la salida del DSLAM debería existir una línea tal que al menos proveyese 1600Mbps para soportar el tráfico de todos los clientes. Generalmente la salida del DSLAM es de menor velocidad ya que no se ha dimensionado para que todos los usuarios hagan uso de sus líneas a la vez. Es por ello que cuando un número alto de usuarios hacen uso de sus líneas se produce un cuello de botella al no poder dar a todo el mundo la velocidad que demanda. Esta situación es más grave teniendo en cuenta que las velocidades de los accesos están aumentando mucho y el número de usuarios también. En algunos casos la tasa de sobre-subscripción puede llegar a 36 a 1 [20].

Para el caso de accesos comerciales la situación es la misma. Cuando un cliente pide una línea dedicada, el operador suele instalar una línea Metro Ethernet que limita lógicamente a través de los mecanismos comentados en las secciones anteriores. La limitación viene cuando el operador agrupa varias de estas líneas en un switch y la línea de salida es de menor capacidad que la suma de las velocidades contratadas por los clientes en la entrada. Típicamente un switch de este tipo suele tener entre 24 y 240 puertos para líneas Metro Ethernet. En este caso las tasas de sobre-subscripción suelen ser menores 2 a 1, 4 a 1 u 8 a 1 [26];

Debido a lo anteriormente mencionado la medida del ancho de banda tanto de líneas ADSL como de líneas comerciales Metro Ethernet tiene especial importancia para verificar los términos de los contratos y SLA's y poder realizar reclamaciones formales a los operadores u organismos pertinentes.

2.6. Métodos de medida de ancho de banda basados en descarga de fichero

Este método que ha sido definido formalmente en la recomendación ETSI EG 202 057-4 [9] y posteriormente adoptado en la orden ITC/912/2006 del Ministerio de Industria Turismo y Comercio de España, se basa en la descarga de un fichero mediante el protocolo HTTP para el cálculo del ancho de banda de un enlace. En este caso el término ancho de banda de un enlace se refiere al máximo throughput TCP que se puede desarrollar realizando una descarga mediante HTTP.

Para realizar el cálculo se procede a la descarga de un fichero que sea en tamaño 8 veces el valor nominal del enlace que se desea medir. Por ejemplo para un enlace de 1Mbps el fichero a descargar debe ser de tamaño 8 Mbits (10^6 bytes) lo cual es aproximadamente 1Mbyte. Este fichero debe ser binario y su contenido debe ser generado de manera aleatoria para evitar que los servidores Web realicen cualquier tipo de optimización en la transferencia del fichero. Además estos servidores deben estar dedicados única y exclusivamente a esta tarea para evitar contaminación en el resultado obtenido.

Según la norma ETSI, el fichero de tamaño 8 veces superior a la velocidad nominal del enlace se debe descargar en aproximadamente 8 segundos para que se verifique que el ancho de banda medido es aproximadamente el 100% del ancho de banda nominal. En el caso de la orden ITC/912/2006 si el tiempo de descarga del fichero es superior a 60 s. se considera que el enlace está incumpliendo el mínimo requerido por contrato para una conexión (10%).

Para calcular el ancho de banda mediante la descarga de un fichero basta con aplicar la siguiente fórmula:

$$BW = \frac{8N}{\tau_f - \tau_i} (Mbps) \quad (6)$$

donde N es el tamaño en bytes del fichero descargado y τ_f y τ_i representan el tiempo, expresado en μs , de finalización e inicio de la descarga del fichero respectivamente.

Este método tiene como ventaja que la medida que se realiza está a nivel de usuario de manera que se da una idea cercana de la experiencia del usuario.

La principal desventaja de este método aparte de la duración (como mínimo 8 s) es que se ve afectado en gran medida por el tráfico concurrente que hay en un equipo. Esto es debido a los mecanismos de control de congestión que usa TCP.

Cuando se inicia una conexión TCP, por ejemplo una descarga HTTP, se usa un mecanismo basado en una ventana de congestión para acotar la cantidad de datos que pueden ser enviados sin recibir un asentimiento. Debido al método de arranque lento de TCP esta ventana de congestión se va ampliando hasta que se detectan pérdidas o expiran los contadores de recepción de las tramas ACK. Llegado este punto el mecanismo va reduciendo la ventana de congestión hasta que se vuelve a una situación sin pérdidas o se llega al mínimo de la ventana de congestión. Cuando se llega a un estado en el que no hay pérdidas se vuelve a aumentar la ventana de congestión y se repite el proceso anterior. Este mecanismo produce un funcionamiento en forma de dientes de sierra como muestra la figura 5 para el caso de TCP Reno ($W/2$).

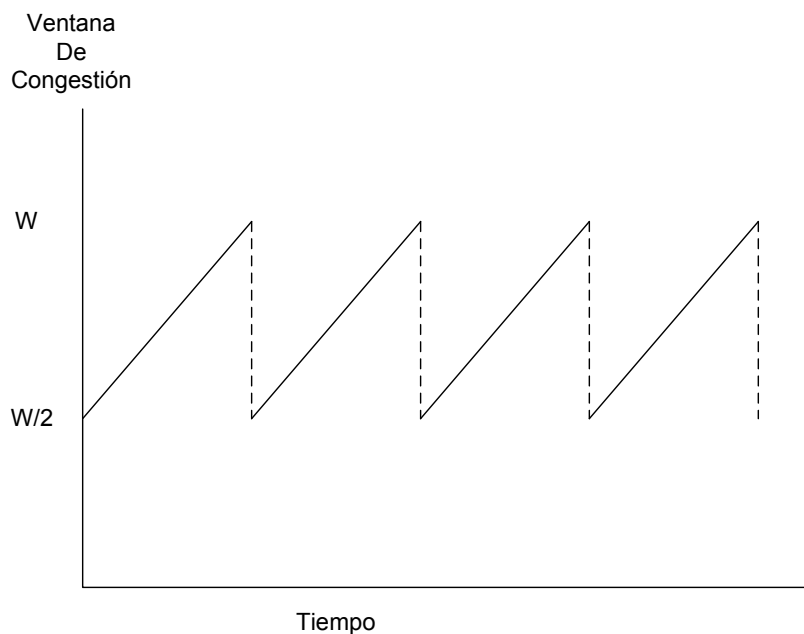


Figura 5: Dientes de sierra producidos por el mecanismo de congestión TCP

Si tomamos en cuenta que en un equipo de medida que realiza una descarga de fichero puede haber varias conexiones TCP y UDP concurrentes y las ecuaciones 7 y 8 de [27].

$$BW \leq \frac{MSS}{RTT} \frac{C}{\sqrt{p}} \quad (7)$$

$$BW_{agg} \leq \frac{MSS}{RTT} \left[\frac{1}{\sqrt{p_1}} + \frac{1}{\sqrt{p_2}} + \dots + \frac{1}{\sqrt{p_n}} \right] \quad (8)$$

se observa que tanto el ancho de banda individual de una conexión TCP como el ancho de banda agregado de un conjunto de conexiones TCP dependen de las probabilidades p_i de pérdida de cada flujo. Si mientras se realiza una medida hay tráfico concurrente (TCP o UDP), que sature el enlace a medir, se pueden registrar pérdidas y el ancho de banda máximo disminuye en consecuencia.

Adicionalmente se debe tener en cuenta que además, por el mecanismo de arranque lento, si se inician dos conexiones TCP secuencialmente se verá favorecida la primera y para obtener un reparto equitativo del ancho de banda de un enlace deberá pasar un tiempo que en muchos casos supera el tiempo necesario para una prueba de descarga de fichero.

Aparte, este método se ve afectado a velocidades altas por la carga de CPU y memoria de un sistema de manera negativa proporcionando resultados incorrectos en las medidas de ancho de banda como se muestra en la sección 4.

Este tipo de medidas son las realizadas por algunos medidores de velocidad que hay en Internet [25]. Promediando los resultados de descargas más pequeñas se pueden obtener resultados más precisos.

2.7. Métodos de medida de ancho de banda basados en pares de paquetes

2.7.1. Packet-Pair

Esta técnica [5, 6, 14, 16, 22] se basa en el envío de múltiples pares de paquetes desde un emisor a un receptor para calcular la capacidad de un camino. Cada par está formado por dos paquetes del mismo tamaño enviados back-to-back, es decir, a la máxima velocidad permitida por el equipamiento y los enlaces de un camino. Se define dispersión de un par en un enlace específico como el tiempo entre el último bit del primer paquete y el último bit del siguiente. Suponiendo que no hay tráfico interferente en el enlace al realizar la prueba, la

dispersión en el receptor viene dada por:

$$\Delta_R = \frac{L}{\min_{i=0,\dots,H} C_i} = \frac{L}{C} \quad (9)$$
$$H = \# \text{ saltos}$$

y la capacidad del camino es por tanto:

$$C = \frac{L}{\Delta_R} \quad (10)$$

donde L es el tamaño del paquete expresado en bits, C es la capacidad del enlace expresada en Mbps y Δ_R es la dispersión o interarrival expresado en μs .

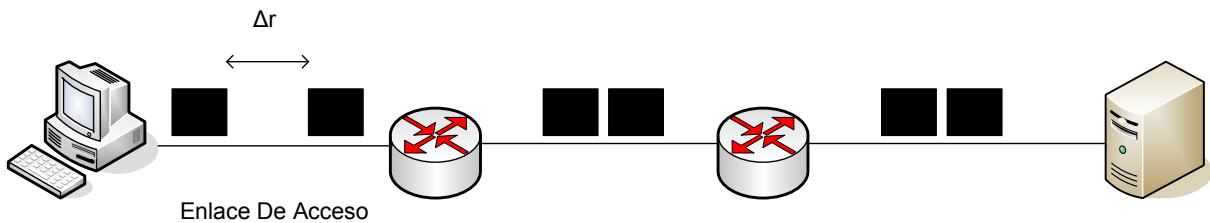


Figura 6: Dispersión producida en el cuello de botella en un enlace residencial

El principal problema de este método es la suposición de que no existe tráfico interferente durante la medida. En condiciones normales, en una red hay tráfico interferente y éste puede influir provocando sobre-estimación o subestimación de la capacidad del camino.

El tráfico interferente puede provocar expansiones o contracciones en el tiempo entre paquetes. Por ejemplo si se introduce uno o más paquetes entre un par de paquetes de prueba se produce una expansión en el interarrival de los paquetes debido al encolamiento del segundo paquete del par y por tanto se obtiene una medida que subestima la capacidad del enlace como se muestra en la figura 7.

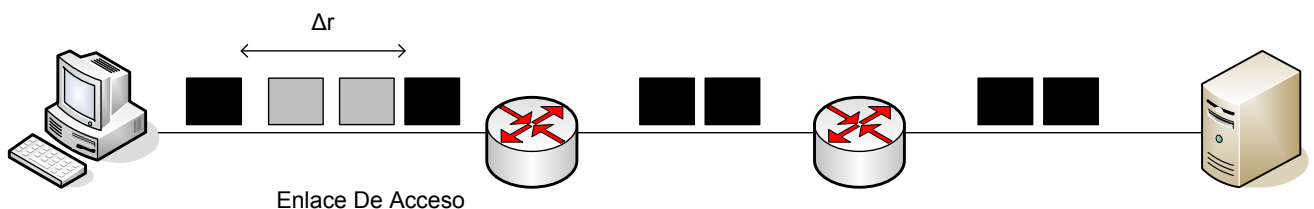


Figura 7: Expansión de la dispersión debida a trafico interferente

Por el contrario si el tráfico interferente hace que el primer paquete del par quede encolado se puede producir una compresión del interarriual al desencolar los dos paquetes del par muy próximos en el tiempo y producir una medida que sobrestima la capacidad del camino como se muestra en la figura 8.

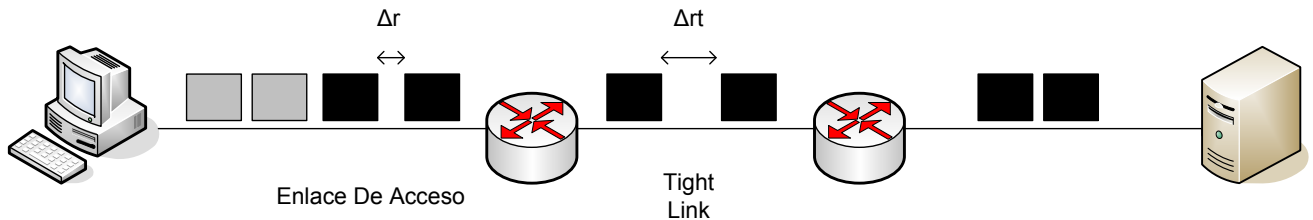


Figura 8: Contracción de la dispersión debida a trafico interferente

Aparte, la medida basada en trenes de paquetes puede verse alterada produciendo sobre-estimación debido al efecto de *Interrupt Coalescence*. Este efecto se explicará en profundidad en el capítulo 4.

2.7.2. Packet-Pair Train

Una variante del método de pares de paquetes es el método de trenes de pares de paquetes [5, 15, 18, 19]. En este caso se envían N paquetes back-to-back y se calcula la dispersión $\Delta_R(N)$ en el receptor como el tiempo transcurrido entre el último bit del primer paquete y el último bit del último paquete del tren. Se define el factor de dispersión como:

$$D = \frac{(N - 1)L}{\Delta_R(N)} \quad (11)$$

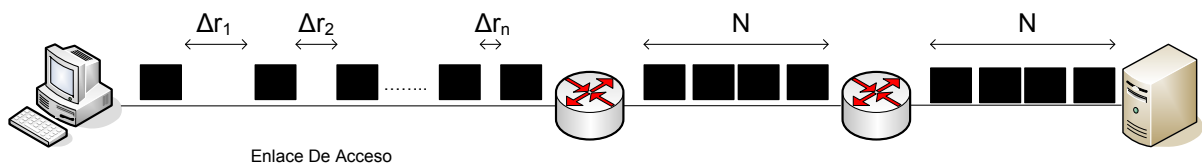


Figura 9: Dispersión producida en el cuello de botella en un enlace residencial para un tren de paquetes

En caso de que no haya tráfico interferente el factor de dispersión es igual a la capacidad del camino. En caso de tráfico interferente, éste influye en la medida

y sólo se puede obtener una estimación media de la capacidad del camino. A medida que el parámetro N se incrementa se logra reducir la varianza de la tasa de dispersión pero no se logra aproximarla a la capacidad real del camino.

3. Evaluación de entornos de medida

Para poder acotar el problema de la medida del ancho de banda se debe definir un conjunto de escenarios en los que los métodos anteriormente descritos apliquen. Para abarcar la diversidad de accesos a Internet tanto residenciales como empresariales existentes se definen dos escenarios: accesos limitados físicamente (ADSL) y accesos limitados lógicamente. Estos últimos aparecen con más frecuencia en los entornos empresariales mientras que los primeros aparecen con más frecuencia en entornos residenciales y de pequeñas empresas. A continuación se tratarán cada uno de estos escenarios con más detalle comentando sus características y retos a la hora de realizar medidas de ancho de banda.

3.1. Accesos limitados físicamente: el caso de ADSL

En España aproximadamente el 44% de la población accede a Internet habitualmente de los cuales el 75% lo hace desde su hogar. Es por ello que los accesos domésticos tienen gran importancia. Dentro de los accesos domésticos alrededor del 78% son accesos ADSL [8]. Esta línea general en la que el ADSL prevalece sobre otras tecnologías de acceso es extensible al resto de países europeos en la actualidad. Es por ello que el caso de estudio de las medidas de ancho de banda sobre líneas ADSL tiene gran importancia.

Las velocidades de las líneas ADSL varían entre 320 Kbps y 24 Mbps. Por tanto este escenario puede ser clasificado como de “baja velocidad”. Por lo general se considera que el cuello de botella de estos accesos a Internet se localiza en el último salto entre el DSLAM y el router/módem del usuario. Este cuello de botella viene dado por las limitaciones físicas de la línea (distancia, ruido, etc) como se comentó en la sección 2.4. Debido a estas limitaciones y a la ausencia, por lo general, de técnicas de limitación lógica de la velocidad, los métodos tanto de descarga de fichero como de trenes de paquetes son adecuados para este tipo de entornos con velocidades bajas.

El problema se produce cuando el equipo desde el que se mide tiene un entorno incontrolado. Por entorno incontrolado se entiende un entorno en el que el propio usuario no tiene conciencia clara de las aplicaciones que corren en la máquina ni de las conexiones de red que tiene abiertas. Por ejemplo existen varios programas y sitios en Internet que instalan elementos de publicidad o spyware sin el consentimiento del usuario y que aparte de consumir recursos de procesamiento de la máquina en que se instalan también consumen recursos de red para enviar o recibir información, publicidad u otro tipo de datos. En la figura 10 se muestra un ejemplo de este comportamiento y cómo interfiere en el correcto uso de los recursos.

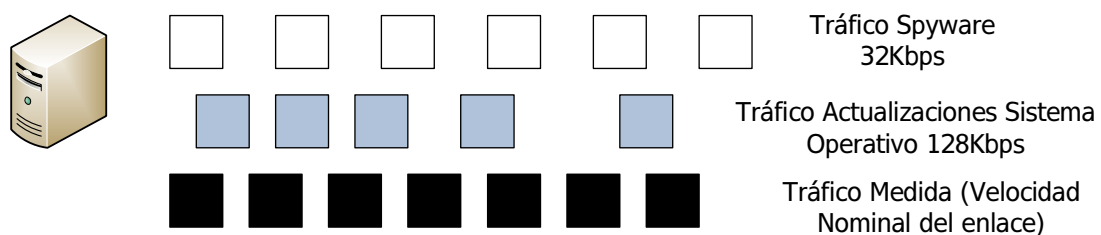


Figura 10: Tráfico concurrente de spyware y actualizaciones automáticas

Este escenario incontrolado aparece con más facilidad en entornos domésticos ya que en los entornos empresariales la instalación de aplicaciones y el tráfico de red es controlado por un administrador de sistemas. Para estos casos, se debe intentar prescindir de los métodos basados en descarga de fichero ya que como se mostró en la sección 2.6 no funcionan correctamente en presencia de tráfico simultáneo debido a la naturaleza del protocolo TCP. Por el contrario los métodos basados en trenes de paquetes proveen buenos resultados incluso con alta carga de procesamiento y red.

3.2. Accesos limitados lógicamente

En España el 94% de las empresas con más de 10 empleados tienen acceso a Internet [8]. En el entorno empresarial también prevalece el acceso a Internet mediante líneas ADSL aunque existe un sector con cierta relevancia formado por grandes corporaciones, bancos y empresas de comercio electrónico que demandan acceso a Internet a través de líneas dedicadas. En la actualidad cuando una empresa solicita una línea de este tipo, el operador instala una línea Metro Ethernet que limita lógicamente mediante los mecanismos anteriormente explicados. Además, las líneas Metro Ethernet conviven con las anteriores tecnologías usadas para este fin como líneas Frame Relay o líneas de cliente ATM también limitadas o controladas lógicamente. En este entorno la velocidad

de las líneas contratadas puede variar desde 2 Mbps hasta 100 Mbps. En este entorno los equipos en los que se realiza una medida suelen estar controlados. El problema surge cuando se pretenden medir velocidades altas (mayores de 50 Mbps). En estos casos la medida es sensible a perturbaciones y además dependiendo del método usado se puede mantener el enlace con una alta carga durante un periodo de tiempo relativamente grande lo cual en un entorno empresarial no es aceptable. Por lo tanto es necesario usar en este entorno métodos no intrusivos y que permitan medir rápida y precisamente el ancho de banda del enlace y que además se mantengan relativamente inmunes al tráfico interferente ya que en un entorno empresarial no es controlable. En la figura 11 se puede observar un diagrama típico de accesos empresariales. En el diagrama se muestra un anillo central que interconecta todos los accesos y varias ramas que se unen a él. Cada una de estas ramas agrupa varios accesos y tiene un switch que realiza policing o shaping sobre cada línea Gigabit Ethernet de cada cliente en función del SLA especificado. En este escenario se observa que la sobre-subscripción se puede producir en estos conmutadores de entrada al anillo en los que se pueden conectar más usuarios de los que soporta el equipo produciendo así degradaciones en la calidad de las líneas.

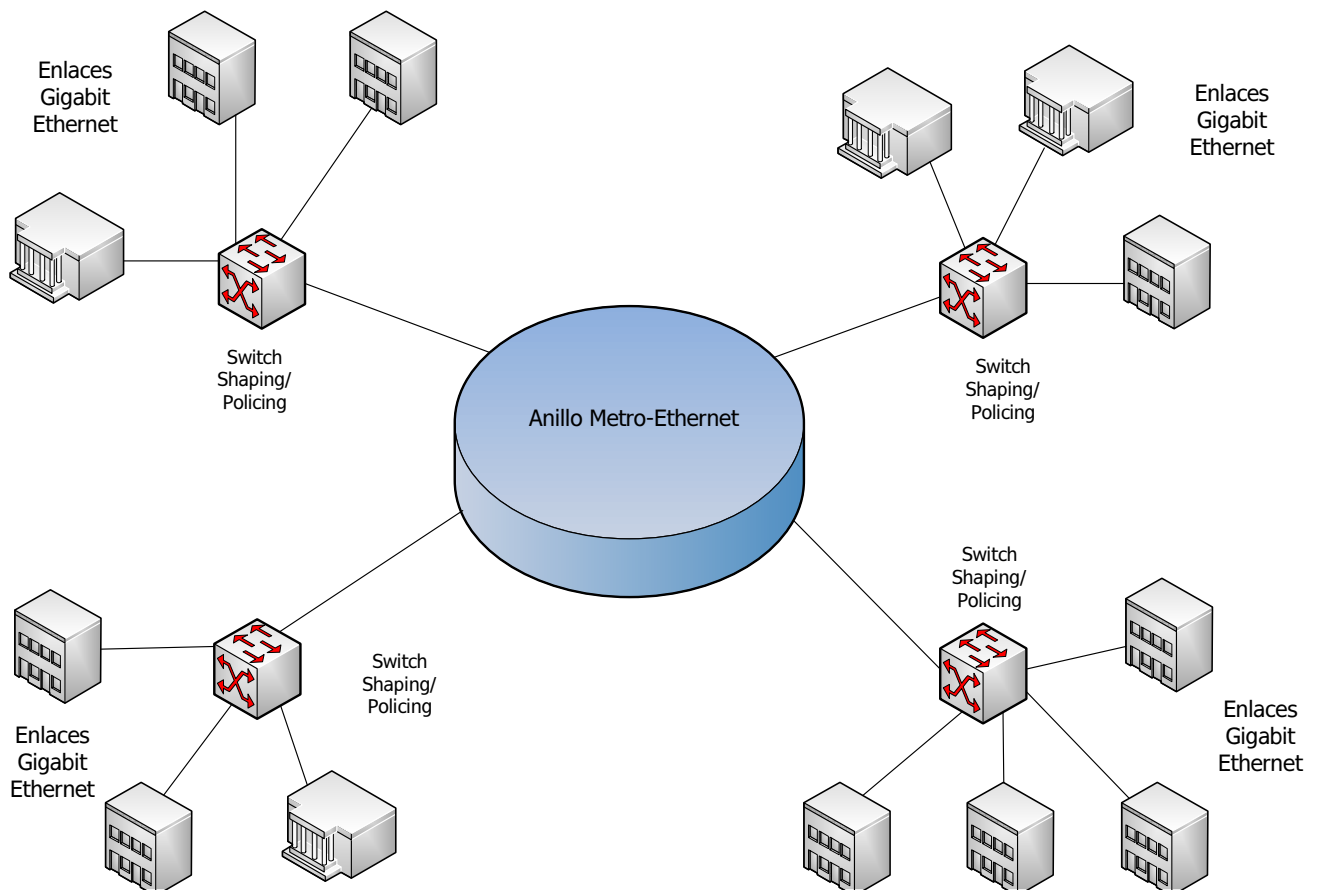


Figura 11: Diagrama de accesos empresariales

4. Factores que influyen en la medida del ancho de banda

En esta sección se describirán con detalle los factores relevantes a la hora de realizar una medida de ancho de banda así como sus métodos de evaluación y la justificación formal de por qué éstos son importantes. La metodología seguida se basa en cuatro pasos fundamentales:

1. Selección de un conjunto de parámetros más o menos generales relacionados con un equipo.
2. Definición de métodos y fórmulas para evaluar los parámetros en un equipo.

3. Realización de múltiples medidas de ancho de banda usando varios métodos y forzando valores determinados definidos en un rango para los parámetros anteriormente seleccionados.
4. Análisis y discriminación de parámetros relevantes.

Para acceder a los valores necesarios para calcular los parámetros de manera unificada, se ha hecho uso de una API JNI de Java que encapsula las llamadas a bajo nivel que permiten obtener estos valores. La API SIGAR (System Information Gatherer And Reporter) [24] permite acceder a datos de bajo nivel de diferentes sistemas operativos (Windows, Linux, Mac OS X) de una manera única a través de un programa Java y un conjunto de librerías en C dependientes de cada sistema operativo.

4.1. Descripción del entorno de laboratorio

Para realizar todas las pruebas y análisis que se comentan a continuación se ha usado una topología controlada en el laboratorio de manera que se puedan obtener resultados de manera fiable. La figura 12

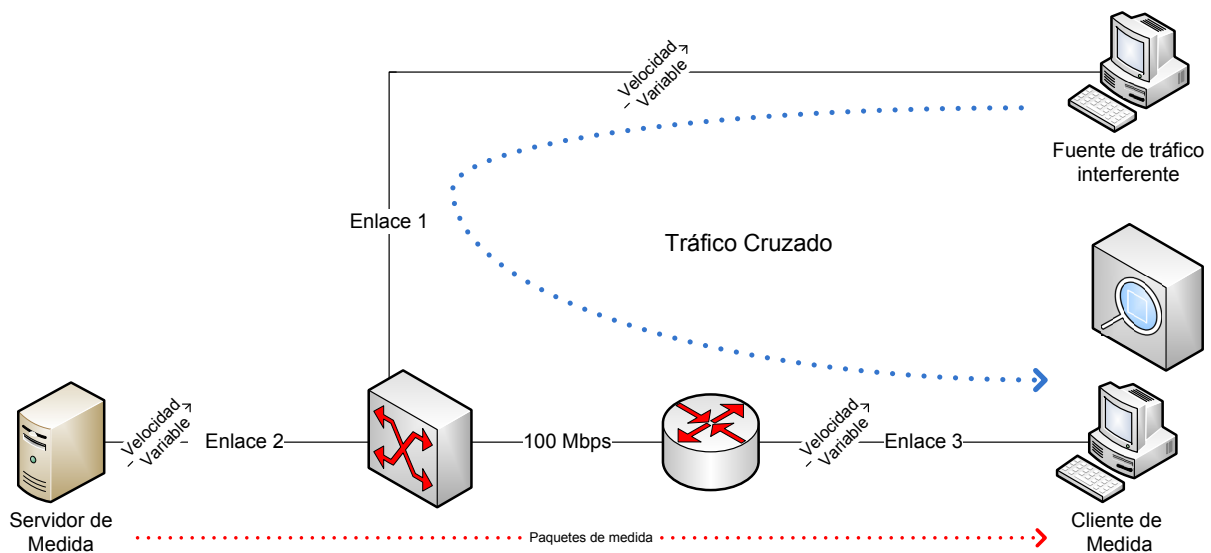


Figura 12: Topología usada para realizar las pruebas

Para simular los anchos de banda de los enlaces de velocidad variable se ha utilizado la herramienta de Linux *tc* [1]. En concreto para realizar las pruebas se han tomado dos velocidades 6 Mbps y 100 Mbps de manera que se cubran los escenarios tanto comerciales como domésticos.

Para las pruebas de tráfico cruzado se ha inyectado tráfico con `hping2`¹ para el caso de tráfico UDP y con `wget`² para el caso de TCP/HTTP.

Para la generación de tráfico cruzado con retardos aleatorios se ha usado la herramienta `tc` con `Netem`³ para generar salidas de paquetes con tiempos basados en una distribución Pareto-Normal [7, 10, 28].

El equipamiento usado en el entorno de laboratorio incluye:

- PC cliente: Pentium Dual Core 1.80GHz 32 bits con Linux (Ubuntu 8.10) y Windows (XP/Vista/7).
- PC generador de tráfico interferente: SOM-PC Intel Atom N470 (1,83GHz) y Linux (Ubuntu mobile 8.10).
- Servidor de medida: Intel Xeon Quad Core 2.33GHz 64 bits con Linux (Ubuntu 8.10).

4.2. CPU

El parámetro CPU indica el porcentaje de tiempo de CPU dedicado al proceso de medida. De esta manera se puede obtener una estimación fiable de la atención que ha recibido el proceso durante su ejecución. Para calcular este parámetro se define la fórmula 12.

$$cpu_frac = \frac{t_{proc_end} - t_{proc_begin}}{t_{total_end} - t_{total_begin}} \quad (12)$$

donde t_{proc_end} y t_{proc_begin} se corresponden con los tiempos acumulativos dedicados al proceso de medida al final y al principio del proceso de medida respectivamente. Por otro lado, t_{total_end} y t_{total_begin} se corresponden con los tiempos acumulativos dedicados a todos los procesos del sistema al final y al principio del proceso de medida respectivamente.

Para tomar estos tiempos se ha hecho uso de los contadores que poseen todos los sistemas operativos para contabilizar los tiempos de CPU. Los contadores están basados en ticks dependientes del reloj del sistema. Cada cierto tiempo el reloj del sistema genera interrupciones y se actualizan el número de ticks de los procesos y totales en el sistema. Estos contadores son más o menos comunes a todos los sistemas operativos y se han obtenido haciendo uso de la API SIGAR

¹<http://linux.die.net/man/8/hping2>

²<http://linux.die.net/man/1/wget>

³www.linuxfoundation.org/en/Net:Netem

anteriormente comentada.

Es importante especificar que el tiempo total de CPU usado es la unión de varios tiempos parciales que pueden ser desglosados básicamente en los siguientes tiempos:

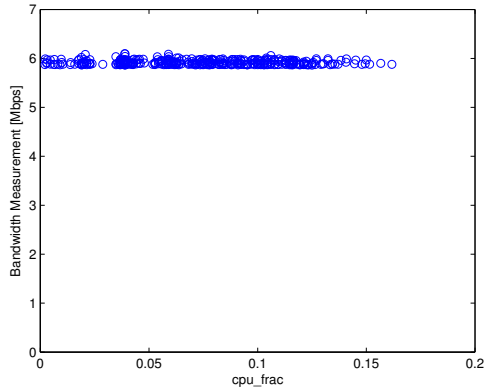
- **Tiempo de Usuario:** especifica el número de ticks de procesador dedicados a procesos en el espacio de usuario
- **Tiempo de Kernel:** especifica el número de ticks de procesador dedicados a procesos del núcleo del sistema operativo y los cambios de contextos realizados.
- **Tiempo de Interrupción:** especifica el número de ticks de procesador dedicados a detectar y atender interrupciones de sistema ya sean software o hardware.
- **Tiempo de Idle:** especifica el número de ticks de procesador dedicados al proceso Idle. El proceso Idle es un proceso con un único hilo de ejecución que se ejecuta constantemente y que ayuda a calcular el porcentaje de CPU libre. Cuando la CPU se encuentra cargada este proceso tiene la menor prioridad y se ejecuta sólo en los espacios de tiempo que dejan libres el resto e procesos.

El tiempo total de CPU usado a lo largo de este trabajo se ha definido como la suma de todos los tiempos parciales de CPU usados por el proceso de medida.

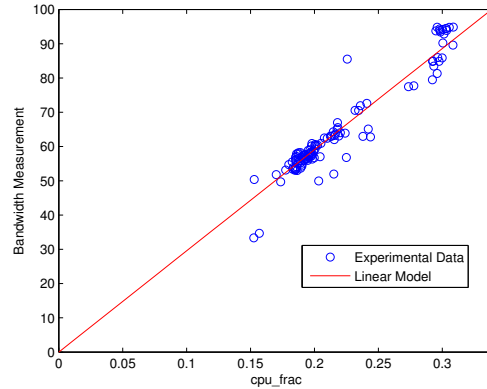
Este parámetro tiene impacto en las medidas ya que si durante una medida no se dedica tiempo suficiente al proceso de medición, los paquetes usados tanto en métodos de pares de paquetes como en métodos de descarga de fichero pueden retrasarse al procesarlos y producir resultados degradados que se alejan de la realidad.

Para probar el impacto de este parámetro sobre la calidad de una medida se han realizado varias medidas mientras se incrementaba el uso de CPU. Para incrementar este uso de CPU se ha generado un programa en C que realiza operaciones en punto flotante y operaciones sobre ficheros. Adicionalmente se ha generado una versión de Java de este programa para probar diferentes tipos de cargas y simular el impacto de varias aplicaciones concurrentes. Estas pruebas se han realizado sobre los dos escenarios anteriormente comentados de 6 Mbps y 100 Mbps.

En la figura 13 se presentan unas gráficas en las que se muestra el resultado de las medidas usando la técnica de descarga de fichero mientras se varía la carga de CPU en Windows XP.



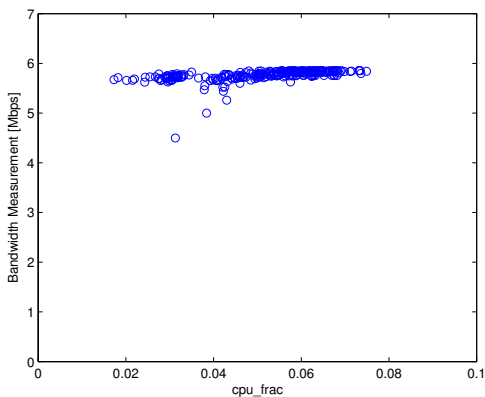
(a) Velocidad Teórica 6 Mbps



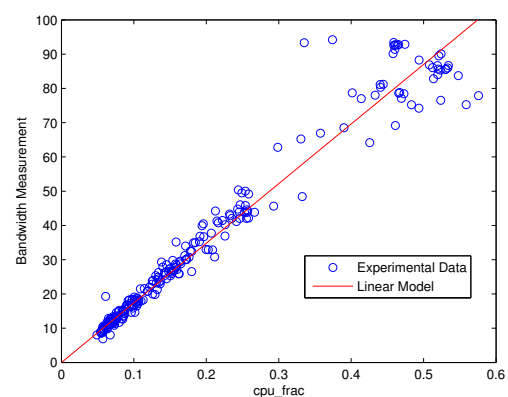
(b) Velocidad Teórica 100 Mbps

Figura 13: Variación de la medida del ancho de banda en función de la carga de CPU Windows

En la figura 14 se presentan unas gráficas en las que se muestra el resultado de las medidas usando la técnica de descarga de fichero mientras se varía la carga de CPU en Linux.



(a) Velocidad Teórica 6 Mbps



(b) Velocidad Teórica 100 Mbps

Figura 14: Variación de la medida del ancho de banda en función de la carga de CPU Linux

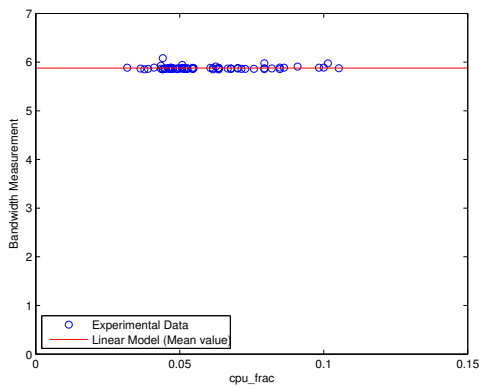
Como se puede observar, tanto en Windows como en Linux, el impacto de la carga de CPU en las medidas de ancho de banda en los enlaces de baja velocidad es casi inexistente. Sin embargo, en los enlaces de alta velocidad la medida

se ve degradada de tal manera que alcanza incluso 1/10 del valor nominal del enlace. Este comportamiento se produce al no poder atender la CPU los paquetes que llegan ya que se encuentra saturada.

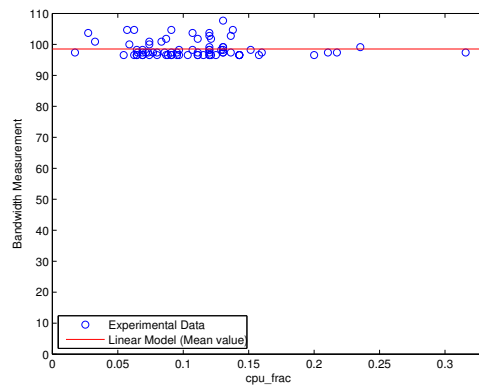
El resultado es similar en el sistema operativo Mac OS X de lo cual se puede deducir que para enlaces de baja velocidad la CPU no es un factor relevante pero para enlaces de alta velocidad si. Una alta carga de CPU degrada las medidas haciéndolas inservibles. Se observa además un comportamiento lineal en el decremento de la velocidad medida en función de la carga de CPU.

Debe notarse que en las figuras existen regiones en las que no hay resultados debido a la dificultad de realizar un barrido sistemático de los diferentes valores de carga de CPU. En este sentido Windows y Linux si se diferencian permitiendo este último alcanzar cargas de CPU más altas y mayor número de procesos concurrentes.

Para el caso de los métodos de medida basados en pares de paquetes se obtienen resultados diferentes. En la figura 15 se presentan unas gráficas en las que se muestra el resultado de las medidas usando la técnica de pares de paquetes mientras se varía la carga de CPU en Windows.



(a) Velocidad Teórica 6 Mbps



(b) Velocidad Teórica 100 Mbps

Figura 15: Variación de la medida del ancho de banda en función de la carga de CPU Windows

Para Linux se han realizado los mismos experimentos. Los resultados se muestran en la figura 16.

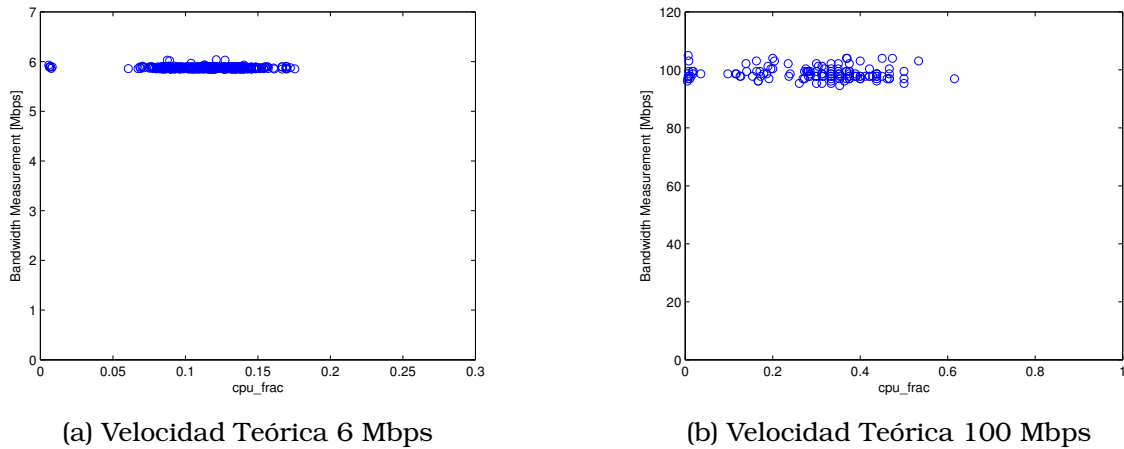


Figura 16: Variación de la medida del ancho de banda en función de la carga de CPU Linux

Al diferencia de los métodos de descarga de fichero, los métodos de pares de paquetes aseguran buenos resultados cuando la carga de CPU es muy alta incluso cuando está cerca del 100%. Se puede observar que los resultados tanto en Windows como Linux son muy similares.

4.3. Memoria

El parámetro memoria indica el porcentaje de memoria RAM libre en el sistema en el momento de la medida. Este parámetro adquiere relevancia debido a que para procesar los paquetes de ambos métodos de medida se debe contar con memoria suficiente ya que la falta de la misma puede producir pérdidas de paquetes. Además, por norma general, cuando un sistema operativo sufre una alta carga en la memoria se activan mecanismos que permiten gestionar de manera óptima la memoria y el espacio de swap o memoria virtual lo cual genera una sobrecarga de CPU y puede producir retardos en los procesos no prioritarios.

Para calcular este parámetro se define la fórmula 13.

$$mem = \frac{free_mem}{total_mem} \quad (13)$$

free_mem se corresponde con la cantidad de memoria libre del sistema expresada en MB en el momento de la medida y total_mem se corresponde con el

total de memoria del sistema.

Al igual que en el caso de la CPU se ha hecho uso de los indicadores de uso de memoria que contienen todos los sistemas operativos, accediendo a ellos de manera independiente a través de la API SIGAR hecha en Java.

Para probar el impacto de este parámetro sobre la calidad de una medida se han realizado varias medidas mientras se incrementaba el uso de memoria. Para incrementar este uso de memoria se ha generado un programa en C que hace reservas de memoria y las mantiene durante un tiempo lo suficientemente grande. Estas pruebas se han realizado sobre los escenarios anteriormente comentados de 6 Mbps y 100 Mbps.

En la figura 17 se presentan unas gráficas en las que se muestra el resultado de las medidas usando la técnica de descarga de fichero mientras se varía la carga de memoria en Windows XP y Windows Vista.

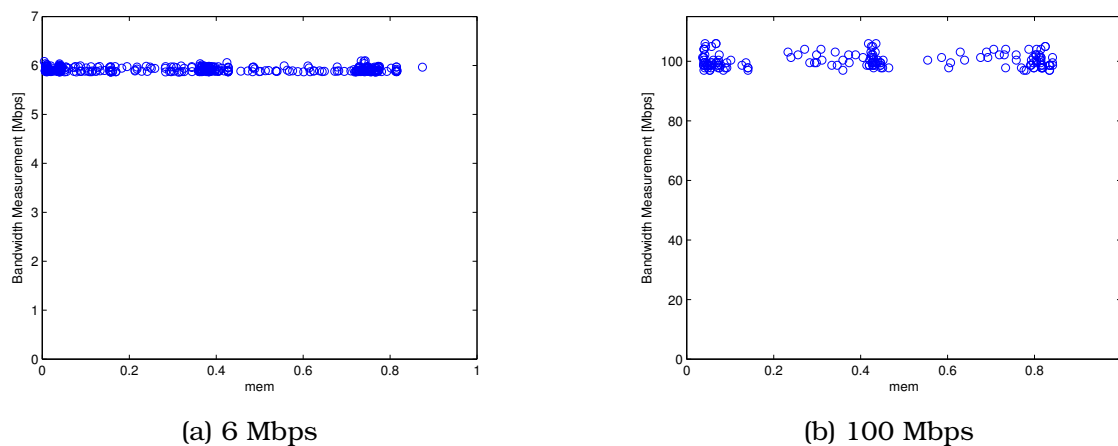
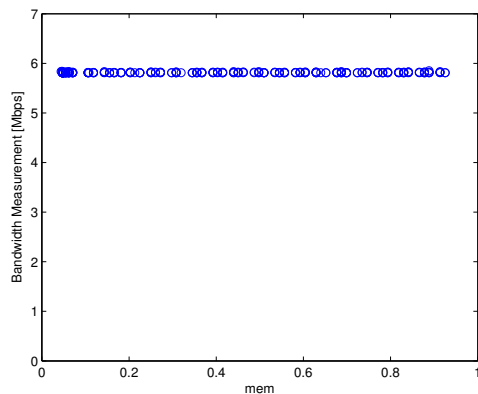
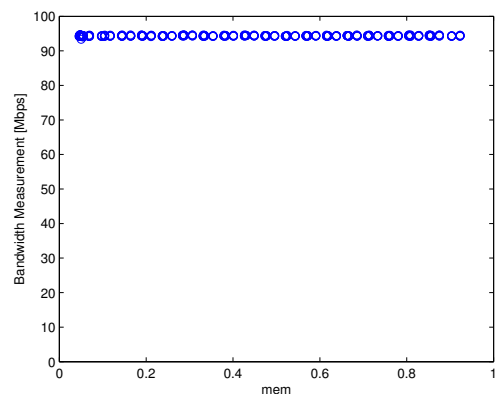


Figura 17: Variación de la medida del ancho de banda en función de la carga de memoria Windows

En la figura 18 se presentan unas gráficas en las que se muestra el resultado de las medidas usando la técnica de descarga de fichero mientras se varía la carga de memoria en Linux.



(a) 6 Mbps

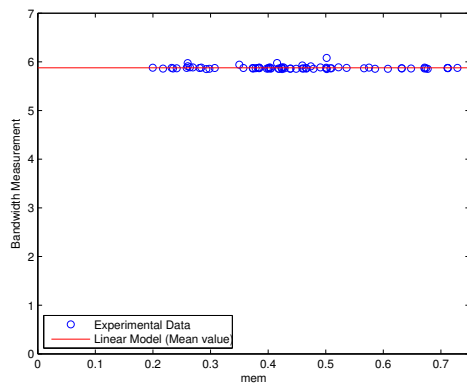


(b) 100 Mbps

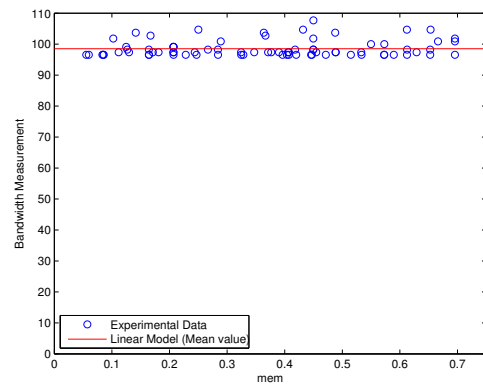
Figura 18: Variación de la medida del ancho de banda en función de la carga de memoria Linux

Como se puede observar en las gráficas este parámetro no tiene impacto apenas sobre las medidas en ninguno de los dos escenarios planteados. Se observa un ligero decremento en la calidad de la medida en el el escenario de 100 Mbps cuando la carga de memoria es extremadamente alta pero no es significativo. Al igual que en el caso anterior se observa que los comportamientos son similares en todos los sistemas operativos probados. En este caso se observa que las gráficas barren con mayor facilidad el rango de memoria disponible. Aun así debe hacerse notar que también existen áreas sin barrer debido a la dificultad de controlar el uso exacto de memoria de un sistema.

Para los métodos de pares de paquetes los resultados obtenidos son muy similares. En la figura 19 pueden observarse unas gráficas en las que se muestra el resultado de las medidas usando la técnica de pares de paquetes mientras se varía la carga de memoria en Windows.



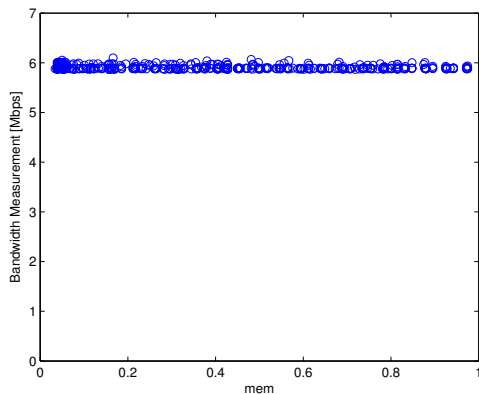
(a) 6 Mbps



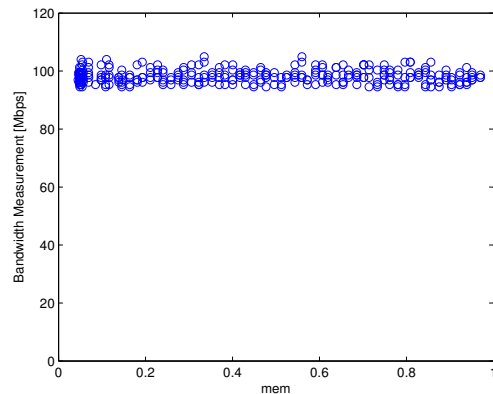
(b) 100 Mbps

Figura 19: Variación de la medida del ancho de banda en función de la carga de memoria Windows

El mismo experimento repetido para Linux se muestra en la figura 20



(a) 6 Mbps



(b) 100 Mbps

Figura 20: Variación de la medida del ancho de banda en función de la carga de memoria Linux

Los resultados obtenidos variando la memoria y usando las dos técnicas de medida de ancho de banda muestran el poco impacto que tiene el factor memoria sobre el resultado de las medidas.

4.4. Sistema Operativo y Hardware

Otros parámetros que aparentemente pueden tener influencia en un sistema de medida multiplataforma como el que se plantea en este trabajo pueden ser el sistema operativo usado y el hardware del equipo que realiza la medida. Este hardware comprende tanto la velocidad de procesador como la arquitectura del mismo (32 o 64 bits) y diferentes tipos de componentes (tarjetas de red, memoria, etc).

Para comprobar la influencia de estos factores se han realizado varios experimentos en un entorno controlado de carga de CPU y memoria de manera que si existe alguna diferencia debido al sistema operativo usado o el hardware de los equipos quede de manifiesto. Para ello se han utilizado los dos métodos de medida propuestos.

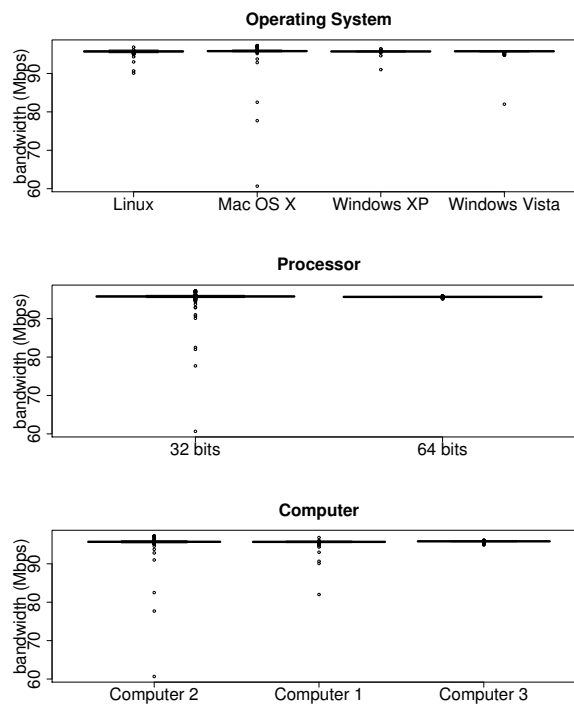


Figura 21: Influencia del sistema operativo y factores hardware sobre la medida del ancho de banda

En la figura 21 se representa el ancho de banda medido frente a los diferentes sistemas operativos, las diferentes arquitecturas y los diferentes equipos usados en las medidas. La figura muestra los resultados obtenidos marcando la

muestra más pequeña, la más grande, el primer cuartil, la mediana y el tercer cuartil. En los tres experimentos se observa que tanto la mediana como el primer y tercer cuartil están muy cercanos entre sí y muy próximos al valor de capacidad teórica del enlace utilizado en la prueba (100 Mbps). Observando estos resultados y los mostrados en las anteriores secciones se puede concluir que no existe una influencia clara de estos factores a la hora de realizar la medida con lo que serán descartados a la hora de obtener el modelo teórico que explique los resultados obtenidos. Los resultados obtenidos son producto de diferentes pruebas en las que los factores se han variado de manera independiente para no provocar interacciones entre ellos. Además en cada prueba se ha comprobado que los valores de carga de CPU, memoria y tráfico interferente no interferían en la medida para aportar más independencia.

4.5. Tráfico interferente

Uno de los parámetros que sin duda influyen la hora de realizar una medida de ancho de banda es el tráfico interferente autogenerado por un equipo. Para comprobar el impacto y evaluar su magnitud se han realizado diferentes experimentos en función del método de medida usado ya que la naturaleza de los mismos es diferente. Además de los métodos implementados por el sistema que se propone en el trabajo se realiza la comparativa frente a un medidor de ancho de banda de gran uso como es Speedtest [25].

Para realizar los experimentos sobre la influencia del tráfico interferente en los métodos de descarga de fichero se debe tomar en cuenta que estos métodos usan como base de la medida la velocidad alcanzada en una descarga HTTP que funciona sobre TCP. Debido a las limitaciones propias de TCP explicadas en secciones anteriores, la existencia de flujos concurrentes tanto TCP como UDP, limitan la velocidad máxima que un flujo TCP puede alcanzar. Por el contrario los métodos de pares de paquetes que usan flujos UDP de manera que se ven poco afectados por los flujos TCP y compiten en igualdad de condiciones con el resto del flujos UDP concurrentes en un sistema.

El primer experimento realizado consiste en la descarga paralela de uno o varios archivos de gran tamaño usando `wget`⁴ mientras se realiza una medida en un enlace de 6 Mbps. Las descargas mediante HTTP introducen tráfico interferente de tipo TCP con lo que a priori tendrán impacto relevante frente a los métodos de descarga e fichero. En la figura 22 puede observarse como el método basado

⁴<http://linux.die.net/man/1/wget>

en descarga de fichero y el usado en Speedtest se ven claramente afectados reduciendo la medida final hasta 1/6 del valor nominal del enlace mientras que los resultados del método de pares de paquetes se mantiene inalterados hasta con tres descargas TCP simultáneas. Este tipo de tráfico supone la situación real en la que un usuario que mide tiene varias descargas tanto controladas (tráfico Web, streaming de vídeo/audio) como incontroladas (descargas en segundo plano del antivirus, rootkits, spyware, etc).

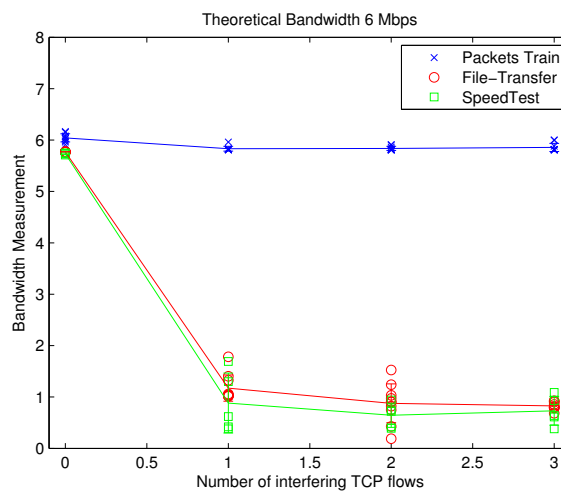


Figura 22: Ancho de banda medido frente al número de flujos TCP interferentes

El segundo experimento realizado para comprobar el impacto del tráfico interferente consiste en la generación de flujos CBR UDP de velocidades incrementales mientras se realiza una medida. Este tipo de tráfico se entremezcla mejor con los flujos TCP de los métodos de medida de descarga de fichero haciendo que las medidas no se vean tan degradadas. Este tipo de tráfico es realista ya que existen múltiples aplicaciones hoy en día que usan flujos UDP CBR como por ejemplo Skype, algunos servicios de streaming y programas P2P en general.

En la figura 23 se puede observar cómo la velocidad medida tanto usando el método de descarga de fichero como usando Speedtest sigue un decremento casi lineal a medida que la velocidad del tráfico interferente aumenta. Esto es debido a que TCP detecta pérdidas a medida que hay más tráfico concurrente y decremента la ventana para adaptarse al ancho de banda disponible del enlace hasta que prácticamente es 0 cuando se alcanza un flujo interferente de velocidad 6 Mbps. Por el contrario se observa que el ancho de banda medido por el método de pares de paquetes se mantiene constante obteniendo buenos

resultados. Esto es debido a que mientras al menos un par de paquetes de los enviados en la prueba lleguen de manera correcta y sin tráfico entre medias se obtendrá una buena medida. Esta situación tiene una probabilidad no despreciable con lo cual este método puede aguantar bien tasas altas de tráfico interferente. En la sección 5.3 se explica cómo calcular esta probabilidad y qué valores típicos presenta.

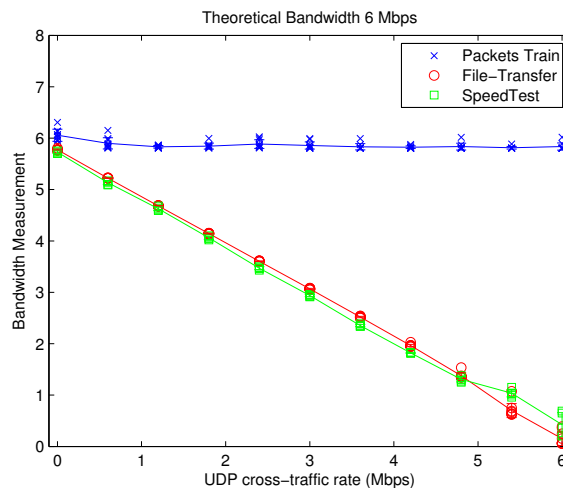


Figura 23: Ancho de banda medido frente a la velocidad el flujo UDP CBR interferente

Para observar la calidad de las medidas obtenidas mediante el método de pares de paquetes y para el posterior análisis y verificación del método de estimación de calidad de la medida se ha realizado un tercer experimento con tráfico interferente. En este experimento se realizan más de 3000 medidas con enlaces a diferentes velocidades y tráfico interferente UDP con interarrivals aleatorios y una tasa media 3 veces superior a la velocidad del enlace de manera que por cada tren de 100 paquetes, de medida hay 300 paquetes de tráfico interferente. La tabla 1 resume los resultados obtenidos para las pruebas realizadas donde \bar{N} es el número medio de paquetes recibidos del tren de prueba y \bar{m} es el número medio de paquetes interferentes durante el periodo de prueba. El error estimado se calcula como $\frac{\#medidas\ incorrectas}{\#medidas\ totales}$ y proporciona una estimación de la calidad del método de medida de pares de paquetes. Como se puede observar se obtienen tasas de fallos muy bajas de fallo incluso en presencia de tráfico interferente.

Ancho de banda	\bar{m}	\bar{N}	# Medidas	Error Estimado
900 Kbps	282	96	3601	0.0061
6 Mbps	281	96	5021	0.0030
12 Mbps	289	96	4332	0.0115

Tabla 1: Resultados del método de pares de paquetes con tráfico UDP interferente

A la luz de los resultados anteriores puede verse que el tráfico interferente es un parámetro muy importante y por tanto debe ser tenido en cuenta a la hora de realizar un modelo preciso que permita obtener una estimación de la calidad de la medida. En la sección 5.3 se comentará como realizar esta estimación y el modelo usado para ello.

4.6. Interrupt Coalescence

Cuando se realizan medidas en redes de alta velocidad (100 Mbps o 1 Gbps) éstas pueden verse alteradas debido a un efecto colateral del uso de la técnica Interrupt Coalescence [23], [2]. Interrupt Coalescence es una técnica que pretende reducir la sobrecarga producida por interrupciones producidas por las tarjetas de red en redes de alta velocidad. Para ello, a diferencia del método normal de interrupciones, cuando llega un paquete a la tarjeta de red ésta lo almacena y espera un tiempo antes de generar una interrupción. De esta manera si llegan más paquetes mientras la tarjeta está esperando, se enviarán varios a la CPU usando una única interrupción. Este mecanismo logra que la CPU esté libre durante más tiempo y se generen menos interrupciones. El efecto negativo a la hora de realizar medidas de red viene dado por el tiempo de espera antes de la generación de interrupción.

La figura 24 muestra el funcionamiento de esta técnica. Cuando los paquetes de medida de los métodos basados en pares de paquetes llegan a la tarjeta vienen equiespaciados por un tiempo t . Cuando el tiempo de espera de la tarjeta para la generación de una interrupción termina se genera una interrupción y se pasan los paquetes a la CPU con un espaciado T_{cpu_i} donde i es la interrupción i -ésima. Este tiempo T_{cpu_i} es mucho menor generalmente que el interespaciado original t lo cual provoca que al calcular el mínimo interarriual de dos paquetes se tomen valores muy pequeños y por tanto se obtengan estimaciones de ancho de banda muy grandes. El tiempo entre interrupciones es generalmente mayor que t y provoca que haya un funcionamiento a ráfagas

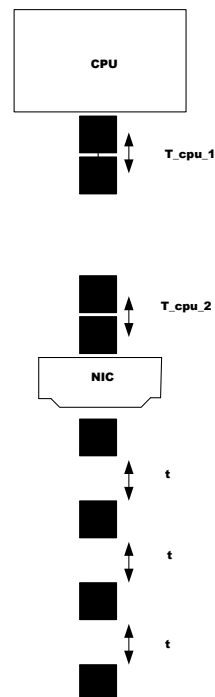


Figura 24: Funcionamiento Interrupt Coalescence

como muestra la figura. Los métodos basados en promedio de los trenes de paquetes y en descarga de fichero no se ven tan afectados por esta técnica ya que en media el tiempo total entre el primer paquete y el último de un flujo es similar al tiempo original con que se recibieron en la tarjeta de red.

A la luz de lo anteriormente explicado se debe tener en cuenta este parámetro a la hora de medir y debe ser reducido el tiempo de generación de interrupciones de manera que no se produzca este comportamiento a ráfagas. Este efecto puede reducirse en gran medida si no desaparece realizando una serie de ajustes en la configuración de la tarjeta de red. Estos ajustes pueden realizarse tanto en Linux como en Windows. Para el caso de Linux basta con ejecutar el siguiente comando:

```
ethtool -C interfaz rx-usecs 0
```

Para el caso de Windows se debe acceder a las propiedades del adaptador de red y desactivar la opción Moderación de interrupciones. Estos ajustes no aseguran aun así que el problema de la coalescencia quede resuelto completamente ya

que depende en gran parte del adaptador de red y de la tecnología que existe por debajo.

5. Implementación de un sistema real: QoS_{Poll}

Una vez realizado el estudio y análisis de los resultados y pruebas anteriores el siguiente paso consiste en la implementación de estos algoritmos de medida en una plataforma sencilla, portable y potente. Para ello se ha realizado una implementación de los métodos de medida de descarga de fichero y de pares de paquetes sobre Java. La elección de Java como lenguaje usado responde a las necesidades de simplicidad y portabilidad anteriormente comentadas. Además debido al uso extendido de los applets, podemos encontrar una máquina virtual de Java casi en cualquier equipo de usuario estándar lo cual facilita la labor de diseminación e instalación. Una aproximación similar a este sistema puede encontrarse descrita en [3].

A continuación se comentarán los diversos aspectos que engloban desarrollo de la aplicación de medida QoS_{Poll} así como las técnicas usadas para descartar una medida realizada.

5.1. Arquitectura

La arquitectura de QoS_{Poll} se basa en el modelo cliente servidor con algunas especificaciones extra. La figura 25 muestra la arquitectura típica de QoS_{Poll}.

Cuando un cliente desea realizar una medida usando el sistema QoS_{Poll} realiza una petición al servidor de medida QoS_{Poll}. Es importante hacer notar que las peticiones son serializadas en orden de llegada de manera que no se puedan estar realizando dos o más medidas simultáneas lo cual introduciría ruido en la prueba realizada. Una vez que el servidor QoS_{Poll} puede atender la petición de medida, ésta se inicia y cuando finaliza los resultados son almacenados en una base de datos MySQL para su posterior análisis y estudio. Esta base de datos posee un front-end gráfico que hace uso de páginas HTML, JSP's y servlets de Java para mostrar la información de las medidas agrupadas por diversos criterios y clasificadas en función de su calidad siguiendo un criterio tricolor. El color verde representa una medida correcta el amarillo una medida con baja probabilidad de ser incorrecta y el rojo una medida con una probabilidad alta-media de ser incorrecta.

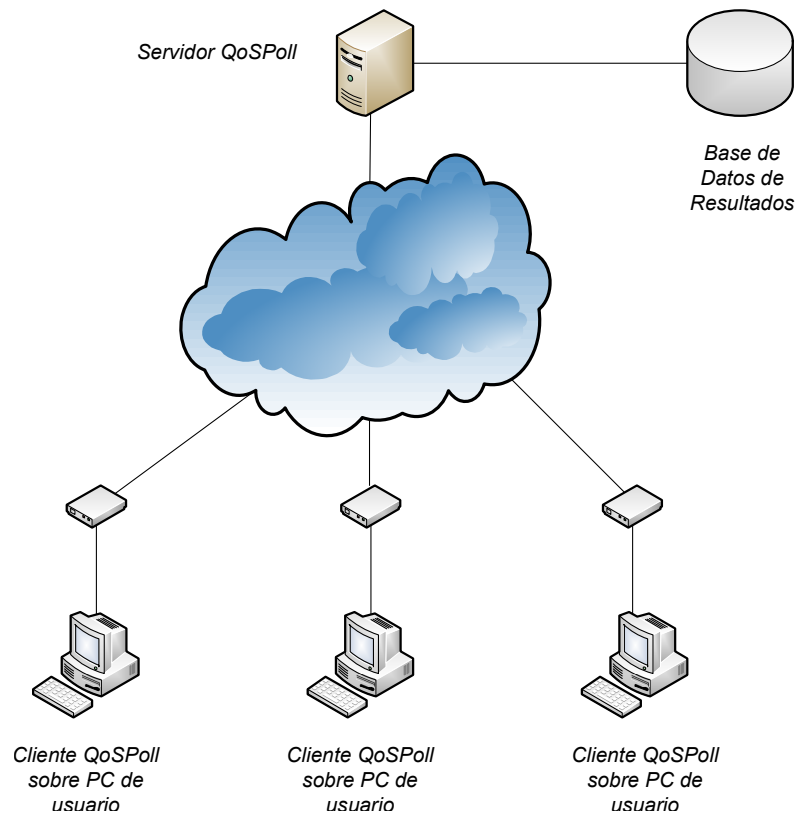


Figura 25: Arquitectura QoS-Poll

5.2. Diseño

El sistema QoS-Poll se basa en varios módulos que interactúan para realizar medidas de ancho de banda. En la Figura 26 se muestra un diagrama de los principales módulos:

El cliente QoS-Poll se compone de los siguiente módulos:

- **Interfaz Gráfica:** Este módulo permite introducir los datos y opciones necesarias para realizar una medida y mostrar los resultados reducidos de cada medida realizada. La figura 27 muestra una captura de pantalla de la interfaz que presenta el cliente QoS-Poll.
- **Módulo de Medidas:** La función principal de este módulo es realizar una medida usando uno de los dos métodos propuestos y compactar los resultados para introducirlos en el módulo generador de medidas. Adicionalmente el módulo realiza medidas de carga de CPU, memoria y tráfico interferente para su posterior proceso y detección de impacto sobre las medidas.

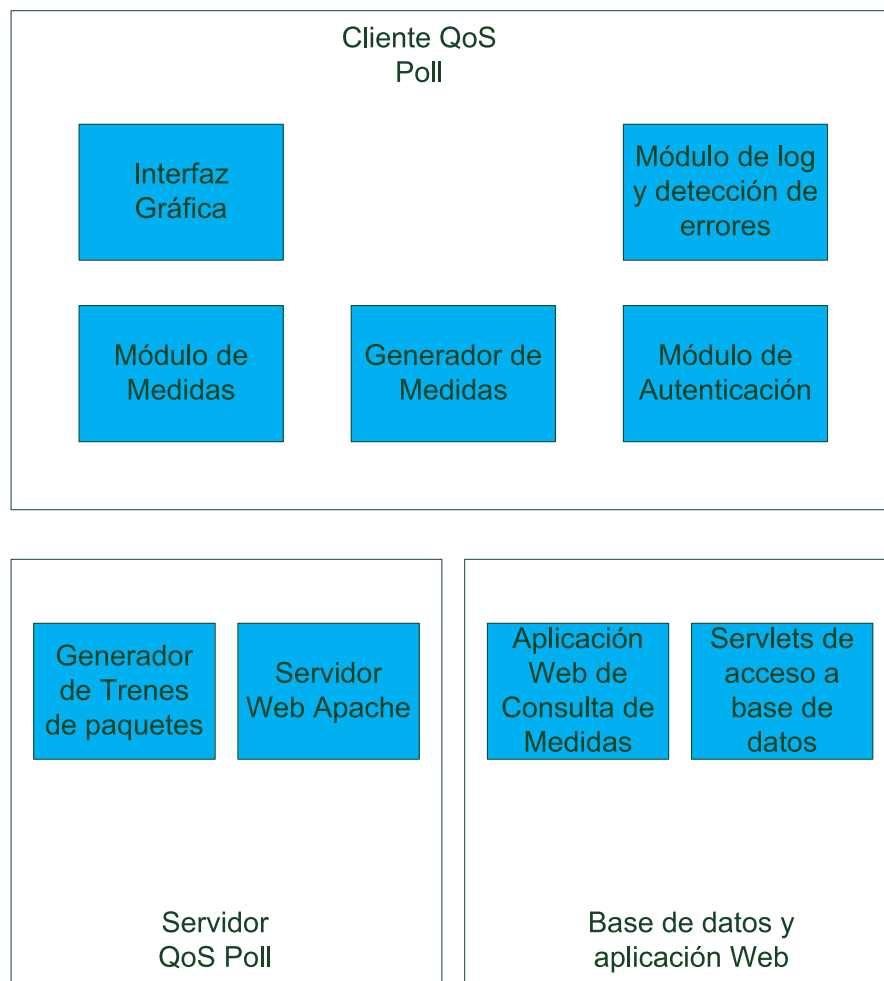


Figura 26: Diseño del Sistema QoS Poll

- **Generador de Medidas:** Este módulo es el encargado de comunicarse con los servlets de acceso a la base de datos en el lado del servidor. Cuando se inicia una prueba se genera un mensaje de creación de una nueva prueba que es enviado a los servlets de acceso a la base de datos en el lado del servidor. Este mensaje contiene información sobre el operador de la red que se está midiendo, las velocidades teóricas de subida y bajada, el usuario que está realizando la medida y la fecha y duración de la prueba. Adicionalmente este módulo agrega las medidas correspondientes a una prueba y genera un mensaje de creación de una nueva medida. Este mensaje contiene entre otros campos el identificador de la prueba a la que pertenece, el sentido (ascendente o descendente) de la misma, los resultados obtenidos, la carga de memoria y CPU, el tráfico interferente, y la fecha. Estos mensajes son enviados también a los servlets de acceso a la base de datos.

- **Módulo de Autenticación:** El cometido de este módulo es la recopilación del nombre de usuario y contraseña de un usuario del sistema y su posterior verificación frente a la base de datos. Para ello se genera un mensaje que es enviado a los servlets de acceso a la base de datos en el servidor y tras la verificación se recibe un mensaje en el que se indica si el usuario se ha autenticado correctamente y el número de pruebas que puede ejecutar. Esta opción permite establecer roles para los usuarios de manera que un usuario no haga un uso abusivo del servicio y no permita realizar pruebas a otros usuarios.
- **Módulo de Log y detección de errores:** Este módulo monitoriza el estado del programa y recoge los errores que pueden producirse y los envía a la base de datos a través de los servlets de acceso a la base de datos. Esta información permite depurar los posibles errores existentes en el cliente y localizar los casos de uso conflictivos.

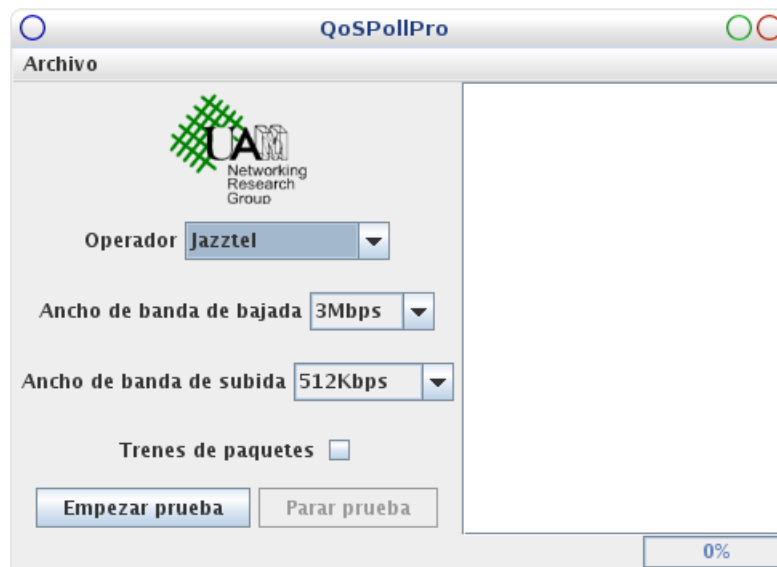


Figura 27: Interfaz de QoS Poll

El servidor QoS Poll se compone de los siguientes módulos:

- **Generador de Trenes de Paquetes:** Este módulo se encarga de escuchar peticiones de generación de trenes de paquetes en un puerto determinado. Cuando un cliente quiere hacer una medida de paquetes envía un mensaje a través de la red a este módulo con información sobre la cantidad y el tamaño de los paquetes que deben ser generados. Con esta información el módulo envía al cliente un tren de paquetes a la máxima velocidad de

la red. Para el sentido ascendente de la medida se realiza un proceso de intercambio de mensajes similar solo que el cliente realiza el envío del tren de paquetes al servidor y éste le comunica al cliente los resultados obtenidos.

- **Servidor Web Apache:** El cometido de este módulo es aceptar y atender las peticiones HTTP de descarga de fichero. Cuando un cliente desea medir con el método de descarga de fichero manda una petición al servidor Apache que contesta enviando al cliente el fichero del tamaño solicitado. Este tamaño dependerá del ancho de banda teórico del enlace que se desea medir. En el sentido ascendente el servidor Apache es el encargado de recibir el fichero patrón a través de una página PHP.

La base de datos y la aplicación web se compone de los siguientes módulos:

- **Servlets de acceso a la base de datos:** Este módulo encapsula el acceso a la base de datos mediante el uso de Servlets de Java. De esta manera se proporciona una capa de seguridad y control que no se logra mediante el acceso directo a la base de datos MySQL. Entre las funciones que realiza este módulo se encuentran la inserción de registros de medidas y pruebas y la consulta y autenticación de usuarios. El cliente intercambia mensajes HTTP con estos servlets especificando los datos necesarios para realizar consultas o inserciones en la base de datos. Estos servlets devuelven al cliente una serie de códigos numéricos encapsulados en mensajes HTTP que representan el resultado de las operaciones realizadas sobre la base de datos.
- **Aplicación Web de Consulta de Medidas:** La función principal de este módulo es proveer una interfaz Web que permita acceder a los resultados de las pruebas de una manera ordenada y selectiva. El módulo se compone básicamente de una serie de páginas JSP que acceden mediante Servlets y JavaBeans a la información contenida en la base de datos. Una vez extraída la información y a la luz de los resultados obtenidos y los parámetros de CPU, memoria y tráfico interferente se aplican una serie de algoritmos para determinar si la medida está contaminada por factores externos. Estos algoritmos se basan en las técnicas explicadas en la sección 5.3. En caso de que se detecte un comportamiento anómalo en la medida, ésta se marca con el color amarillo. Para el caso en que no se detecta un comportamiento anómalo la medida se marca en color verde. En la figura 28 se muestra un ejemplo de la interfaz desarrollada.

Consulta de medidas
High Performance Computing and Networking

Consulta de medidas

Id de prueba: 120
Mostrar: Todas
Consultar

Id Prueba	Usuario	Operador	Fecha	Tipo de Medida	Tiempo de Medida	Velocidad teorica	Velocidad Medida	Download/Upload	CPU	Memoria	S.O	Tamaño Paquete	Numero de Paquetes	Numero de Paquetes Recibidos
120	lavi	Ono	2009-04-27 11:52:19.0	Descarga de Fichero	7897000.0	50.0	53.1126	D	0.20815	0.530572	Windows XP	0	0	0
120	lavi	Ono	2009-04-27 11:52:19.0	Descarga de Fichero	270962.0	50.0	88.5733	U	0.20815	0.530572	Windows XP	0	0	0
120	lavi	Ono	2009-04-27 11:52:59.0	Descarga de Fichero	7813990.0	50.0	53.6768	D	0.205287	0.521795	Windows XP	0	0	0
120	lavi	Ono	2009-04-27 11:52:59.0	Descarga de Fichero	273061.0	50.0	87.8924	U	0.205287	0.521795	Windows XP	0	0	0
120	lavi	Ono	2009-04-27 11:53:38.0	Descarga de Fichero	7667210.0	50.0	54.7044	D	0.2086	0.545197	Windows XP	0	0	0
120	lavi	Ono	2009-04-27 11:53:38.0	Descarga de Fichero	339415.0	50.0	70.7099	U	0.2086	0.545197	Windows XP	0	0	0
120	lavi	Ono	2009-04-27 11:54:17.0	Descarga de Fichero	7816130.0	50.0	53.6622	D	0.206117	0.51798	Windows XP	0	0	0
120	lavi	Ono	2009-04-27 11:54:17.0	Descarga de Fichero	269276.0	50.0	89.1279	U	0.206117	0.51798	Windows XP	0	0	0

Volver

Figura 28: Página de consulta de QoS Poll

5.3. Cálculo de la calidad de la medida

Una de las partes importantes del diseño es la selección de algoritmos y técnicas para la detección de situaciones en las que las medidas están contaminadas por factores externos. Para ello, tras el análisis realizado en las secciones anteriores, se seleccionan tres parámetros como relevantes: carga de CPU, carga de memoria y cantidad de tráfico interferente.

Para el caso de la carga de CPU y memoria la técnica de detección se basa en la comparación con valores umbrales. Al realizar una medida, el módulo de medida obtiene la carga de CPU y memoria y se inserta en la base de datos junto con los resultados de la misma. Cuando se desea comprobar si la medida está contaminada o no, se comparan los valores almacenados con los umbrales de funcionamiento obtenidos a partir de las gráficas empíricas mostradas en la sección 4. En la tabla 2 se muestran los umbrales de funcionamiento para los métodos de descarga de fichero y pares de paquetes. Si el valor de carga de CPU de una medida es menor que el valor umbral, la medida se puede considerar como contaminada. Si el valor de carga de memoria de una medida es menor que el umbral definido, la medida se puede considerar como contaminada.

Método de Medida	Velocidad Teórica	Umbral de Carga de CPU	Umbral de Carga de Memoria
	$x \leq 25Mbps$	0.05	0.1
Descarga de fichero	$25Mbps \leq x \leq 100Mbps$	0.47	0.1
Pares de paquetes	$x \leq 100Mbps$	0.002	0.1

Tabla 2: Umbrales de funcionamiento para carga de CPU y memoria

Estos umbrales son muy restrictivos de manera que se asegure la calidad de una medida. Es posible por tanto, que existan medidas correctas aun no cumpliéndose los requisitos recomendados para estos parámetros.

Para el caso del tráfico interferente se realiza un análisis de los paquetes y bytes interferentes durante el intervalo de tiempo en que se ha realizado la medida dependiendo del método utilizado. Para el caso de medidas con pares de paquetes se utiliza una solución basada en el cálculo de la probabilidad del valor mínimo de una distribución multinomial. El método de cálculo está descrito en detalle en [4]. La explicación y análisis de estos métodos está fuera del ámbito de este trabajo. Para aplicar esta técnica se parte de los valores generados por la figura 29. En la figura se puede observar que para el caso típico de tamaño de tren de paquetes $n=100$ si hay menos de 350 paquetes interferentes la probabilidad de error en la medida es de un 5%. Este criterio tan restrictivo asegura, al igual que en los casos anteriores, que la medida que se marca como correcta realmente lo sea con una probabilidad muy alta. Esto no implica que haya medidas correctas con valores de tráfico interferentes mayores.

Para el caso de descarga de fichero se realiza una aproximación basada en número de paquetes. Para ello se toma como caso peor paquetes de 60 bytes. A partir de los bytes interferentes medidos durante una descarga de fichero es fácil obtener una estimación del número de paquetes. Para ello se aplica la siguiente fórmula:

$$n_{paquetes_interferentes} = \frac{n_{bytes_interferentes}}{60} \quad (14)$$

Una vez conocidos los paquetes interferentes se aplica el mismo proceso anteriormente explicado y se obtiene la probabilidad de que la medida sea errónea.

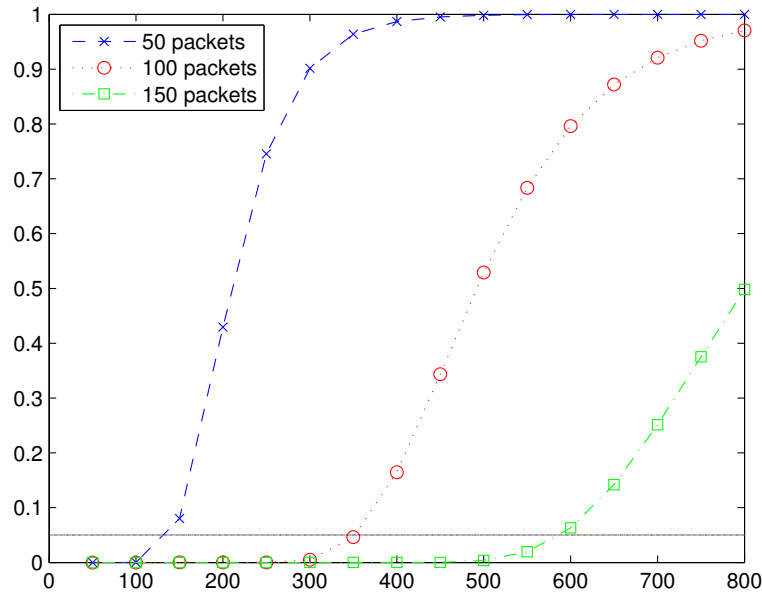


Figura 29: Probabilidad de error en la medida para diferentes longitudes de trenes de paquetes y diferente número de paquetes interferentes.

Estas aproximaciones son de caso peor y sus parámetros pueden variar a un tamaño medio de paquete de aproximadamente 470 bytes lo cual se corresponde con un entorno más realista. Este umbral al igual que el de la probabilidad del 5% son configurables de manera que se aumente o disminuya la tolerancia del sistema. Adicionalmente se puede aplicar una aproximación en la que se calculen el número de conexiones TCP simultáneas en conjunto con el número de bytes interferentes para obtener una estimación de la calidad de la medida. En este escenario aplicando una política restrictiva, cualquier número de conexiones simultáneas superior a 1 provocaría una posible degradación en la medida. Esta y otras opciones se comentan más en profundidad en la sección 8.

6. Validación del sistema

Tras el diseño y la implementación del sistema de medida de ancho de banda QoS Poll se han realizado una serie de pruebas que han permitido validar el sistema en diferentes entornos. Para ello, se ha generado dos escenarios bien diferenciados: un entorno controlado de laboratorio y un entorno real.

6.1. Entorno de laboratorio

El entorno de laboratorio se compone de ordenadores de sobremesa y servidores conectados con el servidor de medida a través de un enlace de 100 Mbps Ethernet. En este escenario se pueden encontrar equipos con diferentes sistemas operativos (Windows, Linux, Mac OS X) y prestaciones distintas. Las características de los equipos usados para estas pruebas pueden encontrarse en la tabla 3.

Equipo	Memoria	Procesador	Sistema Operativo
Equipo 1 (PC)	2GB	Pentium Dual Core 1.80GHz	Linux Windows Vista
Equipo2 (PC)	2GB	Pentium Dual Core 1.80GHz	Linux Windows XP Mac Os X
Equipo 3 (Servidor)	8GB	Intel Xeon Quad Core 2.33GHz	Linux

Tabla 3: Equipos usados en el entorno de laboratorio

Las pruebas de validación consisten en la realización de varias medidas en diversos momentos del día y con diferentes usuarios de manera que se asegure un uso estándar. El conjunto de muestras de validación consiste en 376 medidas realizadas en los entornos anteriormente descritos.

En la figura 30 se muestra el *scatterplot* del ancho de banda de las medidas (normalizado al ancho de banda teórico) y la carga de CPU y memoria. Se puede observar que las medidas no están condicionadas por la carga de memoria o CPU y que la varianza no cambia al cambiar los valores de *cpu_frac* y *mem*.

En la figura 31 se pueden observar los histogramas para la carga de CPU y memoria en las medidas recolectadas. Se puede observar que aproximadamente un 40% de las medidas se han realizado con un valor de CPU menor o igual al 0.05 lo cual excede los límites impuestos por el algoritmo. Aun así se observa que el método de medida es inmune a este parámetro consiguiendo una buena precisión en el 93% de las medidas.

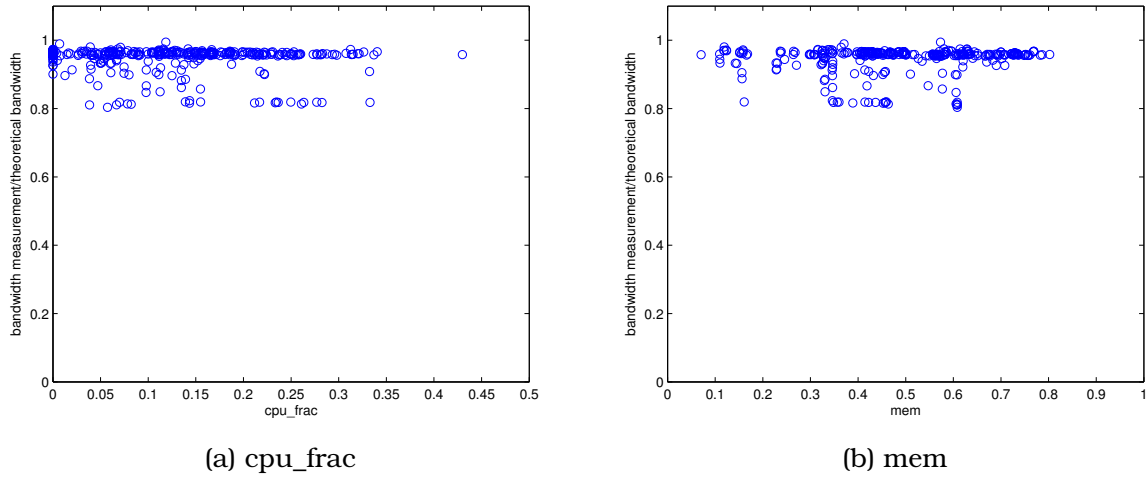


Figura 30: Medidas de ancho de banda en el entorno de laboratorio

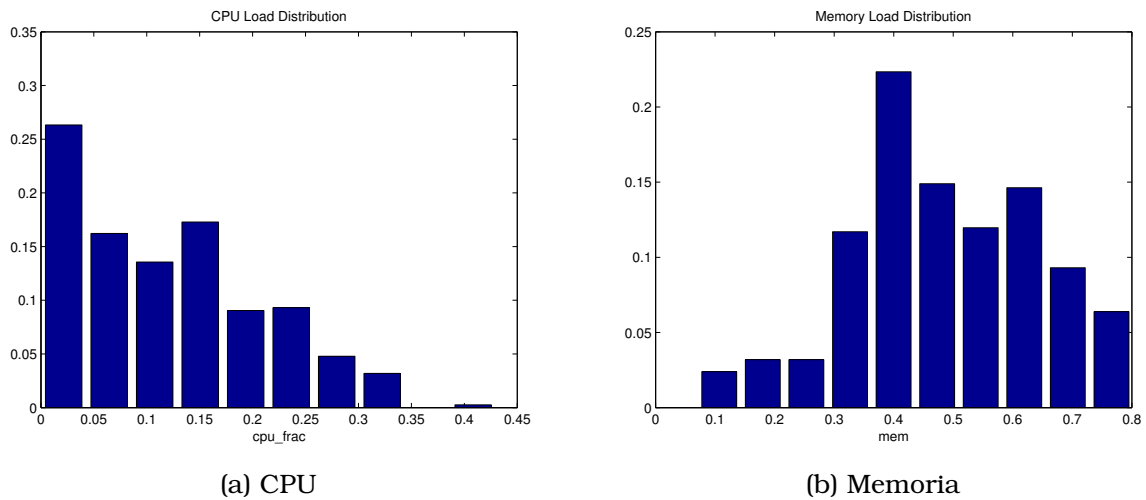


Figura 31: Histograma de carga de CPU y memoria en medidas en entorno de laboratorio

Para concluir con la validación en entorno de laboratorio se ha realizado un estudio de la influencia del tráfico cruzado en las medidas. La figura 32 muestra la cantidad de paquetes interferentes en cada medida. A partir de los datos anteriores se puede ver que alrededor de un 55% de las medidas están realizadas bajo la influencia de tráfico interferente de unos 200 paquetes. Sin embargo se puede observar que la mayoría de las medidas realizadas, alrededor de 93%, obtienen un valor preciso. Los outliers que se observan pueden explicarse por

las medidas que tienen un tráfico interferente mayor de 300 paquetes lo cual representa aproximadamente un 15% de las mismas.

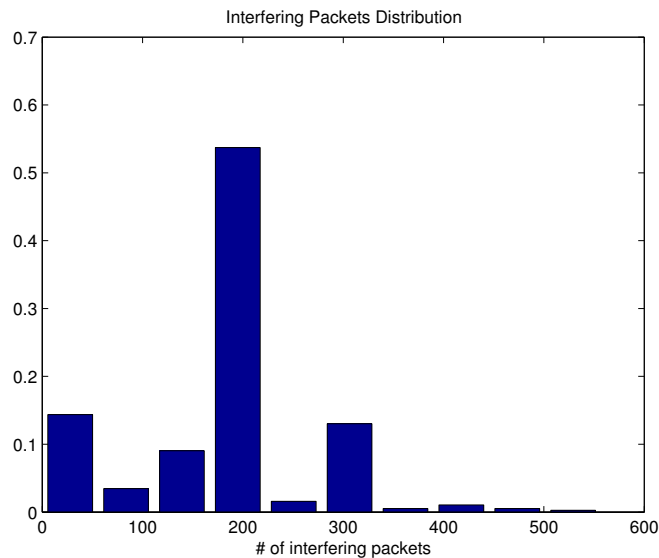


Figura 32: Distribución de paquetes interferentes en las medidas en entorno de laboratorio

6.2. Entorno Real

Para la validación de sistema de medida en un entorno real se ha realizado una campaña de medidas entre varios alumnos y miembros de la Escuela Politécnica Superior de la Universidad Autónoma así como personas de diferentes universidades españolas y personas ajenas al ámbito universitario. Para realizar esta campaña la aplicación de medida QoS Poll se ha distribuido a través de una página Web usando el sistema Java Web Start. Este sistema permite descargar y ejecutar automáticamente una aplicación basada en Java de manera que se facilite su uso y difusión.

Tras realizar una limpieza de las medidas incorrectamente realizadas, por ejemplo aquellas cuyo ancho de banda teórico es menor al real, se han elegido 79 medidas de 4 operadoras españolas diferentes tomadas en diferentes localizaciones y con dos modalidades de velocidad 6 Mbps y 3 Mbps. Estas modalidades han sido elegidas para el análisis debido a que eran las más numerosas en la base de datos. La tabla 4 resume los datos de la campaña de medidas.

	Operador 1	Operador 2	Operador 3	Operador 4	Total
Ancho de Banda Teórico = 3 Mbps	16	0	0	0	16
Ancho de Banda Teórico = 6 Mbps	0	58	4	1	63
Total	16	58	4	1	79

Tabla 4: Datos de la campaña de medidas realizada en el entorno real

La figura 33 muestra el *scatterplot* del ancho de banda de las medidas (normalizado al ancho de banda teórico indicado por el usuario) y la carga de CPU y memoria. Tal y como se esperaba, se puede observar que las medidas no están condicionadas por la carga de memoria o CPU y que la varianza no cambia al cambiar los valores de *cpu_frac* y *mem*. Es interesante observar que el valor del cociente $\frac{\text{ancho de banda medido}}{\text{ancho de banda}}$ es menor que el obtenido en los experimentos realizados en el entorno de laboratorio. Este hecho puede ser debido a degradaciones reales del servicio ofrecido por el operador. Este dato es muy interesante ya que de manera implícita se está validando el sistema al ser capaz de detectar una disminución del ancho de banda contratado.

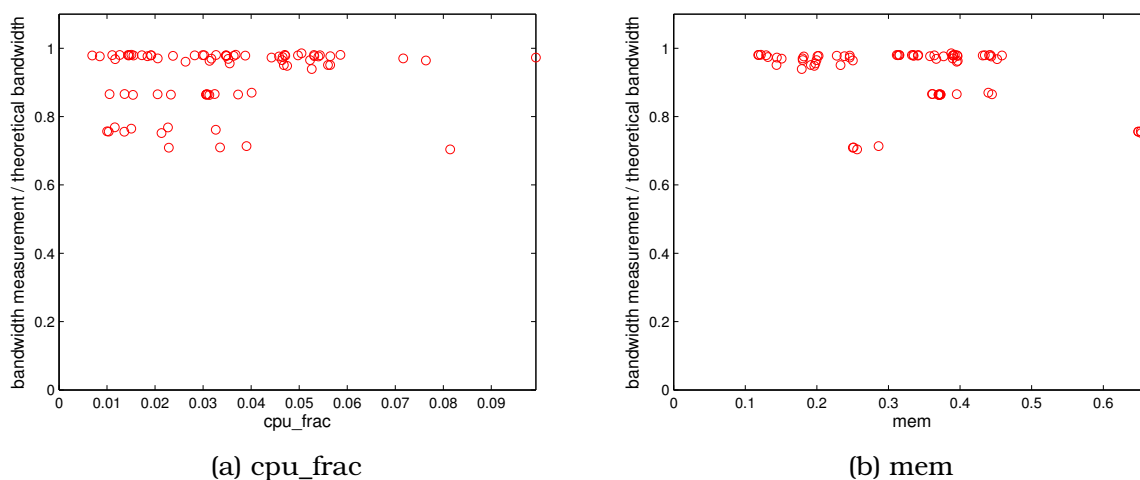


Figura 33: Medidas de ancho de banda en un entorno real

Analizando las medidas de carga de CPU y memoria recolectadas se puede observar que la mayoría de los casos están dentro de los umbrales permitidos para la correcta medida del ancho de banda. En la figura 34 se pueden observar los histogramas para la carga de CPU y memoria en las 79 medidas recolectadas.

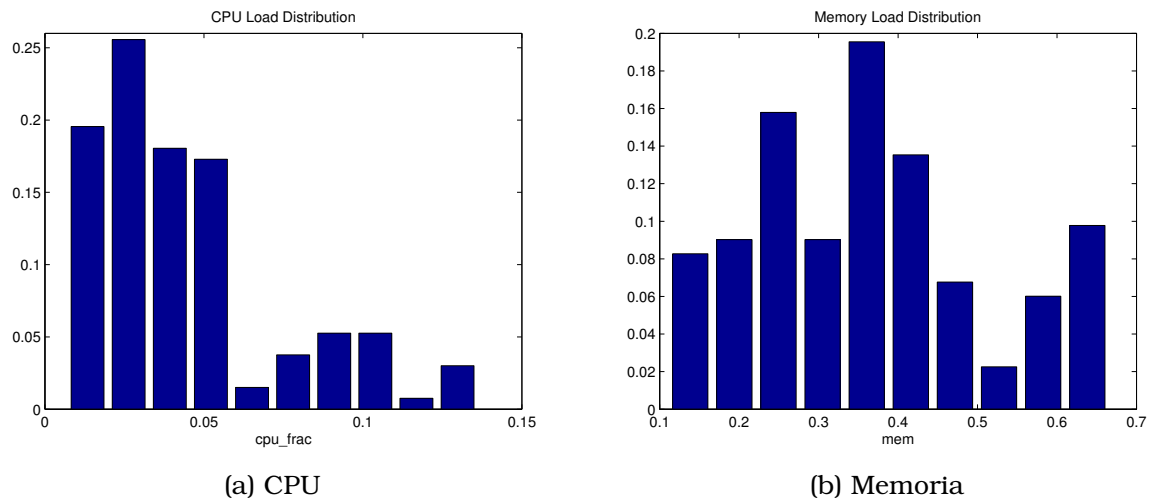


Figura 34: Histograma de carga de CPU y memoria en medidas en entorno real

Finalmente se ha realizado un estudio de la influencia del tráfico cruzado en las medidas en entorno real. La figura 35 muestra la cantidad de paquetes interferentes en cada medida. Se puede observar que el número de paquetes interferentes no es suficientemente grande como para degradar una medida según el modelo multinomial anteriormente comentado.

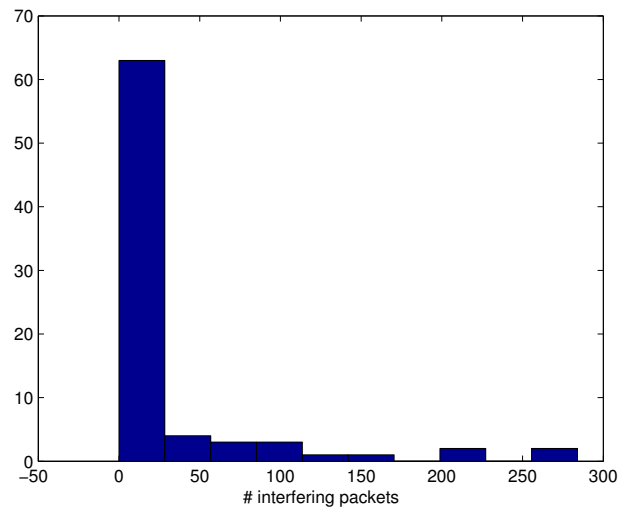


Figura 35: Distribución de paquetes interferentes en las medidas en entorno real

7. Conclusiones

En este trabajo se ha propuesto un acercamiento diferente a los métodos de medida de ancho de banda analizando la influencia de factores externos tales como la carga de CPU, memoria y tráfico interferente que pueden darse en un escenario de medida real. Para realizar esta aproximación primero se ha realizado un estudio exhaustivo de los métodos de medida existentes en la literatura y se ha realizado una selección de los métodos más relevantes. Los métodos elegidos han sido descarga de fichero siguiendo la guía ETSI y trenes de pares de paquetes. Una vez seleccionados los métodos de medida se ha realizado un estudio de los diferentes escenarios que se pueden presentar a la hora de realizar una medida diferenciando dos tipos: los accesos limitados lógicamente y los limitados físicamente. Para los primeros se ha realizado un estudio detallado de las técnicas de limitación del ancho de banda y el impacto que éstas presentan sobre las medidas en función del método de medida usado. Para los segundos se ha realizado un estudio de la tecnología que impone la limitación física, más concretamente de las diferentes modulaciones que proveen las líneas ADSL y la manera en que esto afecta a las medidas de ancho de banda.

En este trabajo además, se ha realizado una selección de factores externos a los métodos de medida que pueden influir en el resultado de la misma. Los

factores elegidos para el estudio han sido carga de CPU y memoria, sistema operativo y tipo de hardware empleado en la medida y tráfico interferente. Para analizar el impacto de cada uno de estos factores se ha definido un entorno de laboratorio controlado sobre el que se han hecho pruebas. Los resultados obtenidos muestran que en el caso de descarga de fichero para velocidades cercanas a 100 Mbps la carga de CPU y la cantidad de tráfico interferente son factores muy influyentes llegando a producir degradaciones importantes en la medida realizada. Para el caso de las medidas con trenes de pares de paquetes se ha observado que el único factor realmente influyente es la cantidad de tráfico interferente durante la medida lo cual lo convierte en una buena elección a la hora de realizar medidas.

Una vez analizados los métodos de medida y los parámetros externos, se ha descrito la arquitectura y e implementación del sistema de medida QoS-Poll desarrollado en el ámbito de este trabajo. En este punto se han descrito los diferentes módulos en que está dividido QoS-Poll y se ha explicado de manera superficial la aproximación matemática usada para el cálculo de la validez de una medida. Para ello se han definido valores umbral de los diferentes factores externos de manera que se obtenga una región en la que las medidas sean confiables. Para el caso del tráfico interferente se ha usado un modelo matemático basado en el cálculo del mínimo de una distribución multinomial para obtener los umbrales de tráfico interferente que no se deben superar para obtener una medida que no esté degradada.

Por último se ha realizado una validación del sistema QoS-Poll tanto en un entorno controlado de laboratorio como en un entorno real realizando una campaña de medidas. Los resultados de estos experimentos han permitido obtener los valores típicos de los factores externos y observar las degradaciones reales en servicios de ADSL en España.

A la luz del trabajo realizado se puede determinar que a la hora de realizar medidas los factores externos tienen cierta importancia y que los métodos basados en pares de paquetes proveen una solución eficaz, poco intrusiva e independiente de los factores externos para el problema de las medidas de ancho de banda extremo a extremo.

8. Trabajo Futuro

Como trabajo futuro queda comprobar la influencia de otro tipo de factores externos como número de flujos TCP/UDP concurrentes o número de procesos concurrentes. Además se debe estudiar el comportamiento de del sistema QoS-

Poll cambiando los diferentes umbrales a valores menos restrictivos calculando además la tasa de falsos positivos y falsos negativos para las medidas realizadas. También se propone como trabajo futuro el análisis modelos matemáticos alternativos que permitan analizar los factores externos con más precisión así como la influencia que estos ejercen sobre una medida de ancho de banda en función del método utilizado. Como último punto del trabajo futuro, se propone realizar optimizaciones y generalizar el sistema de medida QoS-Poll para que trabaje de manera distribuida y embebido en una página Web o algún mecanismo estándar de manera que se logre la máxima facilidad de uso y difusión.

Este trabajo de fin de máster ha dado lugar a la presentación de una solicitud de patente nacional de invención titulada “Método para estimar los parámetros de un elemento de control del tipo Token Bucket” con número de registro P201030516. Además, basado en el trabajo realizado se ha enviado un artículo titulado “On the effect of concurrent applications in bandwidth measurement speedometers” a la revista Computer Networks el cual está pendiente de aceptación.

Apéndices

Índice de figuras

1. Traffic Policing	8
2. Traffic Shaping	9
3. Diagrama del método Token Bucket	10
4. Esquema básico ADSL	17
5. Dientes de sierra producidos por el mecanismo de congestión TCP	22
6. Dispersión producida en el cuello de botella en un enlace residencial	24
7. Expansión de la dispersión debida a trafico interferente	24
8. Contracción de la dispersión debida a trafico interferente	25
9. Dispersión producida en el cuello de botella en un enlace residen- cial para un tren de paquetes	25
10. Tráfico concurrente de spyware y actualizaciones automáticas . . .	27
11. Diagrama de accesos empresariales	29
12. Topología usada para realizar las pruebas	30
13. Variación de la medida del ancho de banda en función de la carga de CPU Windows	33
14. Variación de la medida del ancho de banda en función de la carga de CPU Linux	33
15. Variación de la medida del ancho de banda en función de la carga de CPU Windows	34
16. Variación de la medida del ancho de banda en función de la carga de CPU Linux	35
17. Variación de la medida del ancho de banda en función de la carga de memoria Windows	36

18.	Variación de la medida del ancho de banda en función de la carga de memoria Linux	37
19.	Variación de la medida del ancho de banda en función de la carga de memoria Windows	38
20.	Variación de la medida del ancho de banda en función de la carga de memoria Linux	38
21.	Influencia del sistema operativo y factores hardware sobre la medida del ancho de banda	39
22.	Ancho de banda medido frente al número de flujos TCP interferentes	41
23.	Ancho de banda medido frente a la velocidad el flujo UDP CBR interferente	42
24.	Funcionamiento Interrupt Coalescence	44
25.	Arquitectura QoS Poll	46
26.	Diseño del Sistema QoS Poll	47
27.	Interfaz de QoS Poll	48
28.	Página de consulta de QoS Poll	50
29.	Probabilidad de error en la medida para diferentes longitudes de trenes de paquetes y diferente número de paquetes interferentes. .	52
30.	Medidas de ancho de banda en el entorno de laboratorio	54
31.	Histograma de carga de CPU y memoria en medidas en entorno de laboratorio	54
32.	Distribución de paquetes interferentes en las medidas en entorno de laboratorio	55
33.	Medidas de ancho de banda en un entorno real	56
34.	Histograma de carga de CPU y memoria en medidas en entorno real	57
35.	Distribución de paquetes interferentes en las medidas en entorno real	58

Índice de tablas

1. Resultados del método de pares de paquetes con tráfico UDP in- terferente	43
2. Umbrales de funcionamiento para carga de CPU y memoria	51
3. Equipos usados en el entorno de laboratorio	53
4. Datos de la campaña de medidas realizada en el entorno real	56

Referencias

- [1] Traffic control. <http://linux.die.net/man/8/tc>.
- [2] Go Hasegawa Cao Le Thanh Man and Masayuki Murata. Icim: An inline network measurement mechanism for highspeed networks. *Workshop on End-to-End Monitoring Techniques and Services (E2EMON 2006)*, 2006.
- [3] L. Cheng and I. Marsic. Java-based tools for accurate bandwidth measurement of digital subscriber line networks. *Integrated Computer-Aided Engineering*, 9:333–344, 2002.
- [4] C. J. Corrado. The exact joint distribution for the multinomial maximum and minimum and the exact distribution for the multinomial range. *Available at SSRN: http://ssrn.com/abstract=989082*, 2007.
- [5] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? in *Proc. IEEE INFOCOM*, pages 905–914, 2001.
- [6] C. Dovrolis, P. Ramanathan, and D. Moore. Packet-dispersion techniques and a capacity-estimation methodology. *IEEE/ACM Transactions on Networking (TON)*, 12:963–977, 2004.
- [7] K. Fujimoto, S. Ata, and M. Murata. Statistical analysis of packet delays in the internet and its application to playout control for streaming applications. *IEICE Trans. Commun*, E84(6):1504–1512, 2001.
- [8] Telefonica I+D. Informe sobre la sociedad de la información en españa 2008. http://e-libros.fundacion.telefonica.com/sie08/aplicacion_sie.html, 2008.
- [9] European Telecommunications Standards Institute. *Speech Processing, Transmission and Quality Aspects (STQ); User related QoS parameter definitions and measurements; Part 4: Internet access*. 2008.
- [10] Y. Ito, S. Tasaka, and Y. Fukuta. Psychometric analysis of the effect of end-to-end delay on user-level qos in live audio-video transmission. *IEEE International Conference on Communications*, 4:2214–2220, 2004.
- [11] ITU-T. *Recommendation G.994.1: Handshake procedures for digital subscriber line (DSL) transceivers*. 2001.
- [12] R. Guerin J. Heinanen, Teli Finland. *RFC2697-A Single Rate Three Color Marker*. Networking Group University of Pennsylvania, 1999.

- [13] R. Guerin J. Heinanen, Teli Finland. *RFC2698- A Two Rate Three Color Marker*. Networking Group University of Pennsylvania, 1999.
- [14] V. Jacobson. Congestion avoidance and control. *in Proc. SIGCOMM'88*, 1988.
- [15] Andreas Johnsson. On the comparison of packet-pair and packet-train measurements. *Poster proposal for SNCNW2003*, 2003.
- [16] S. Keshav. A control-theoretic approach to flow control. *in Proc. SIGCOMM'91*, 1991.
- [17] M. Allman M. Mathis. A framework for defining empirical bulk transfer capacity metrics. *RFC3148*, 2001.
- [18] B. Melander, M. Bjorkman, and P. Gunningberg. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. *GLOBECOM*, 2000.
- [19] B. Melander, M. Bjorkman, and P. Gunningberg. Regression-based available bandwidth measurements. *International Symposium on Performance Evaluation of Computer and Telecommunications Systems*, 2002.
- [20] Tom Mitchell. Avoiding the pitfalls of oversubscription in dsl networks. <http://www.xchangemag.com/articles/041feat1.html>, 2000.
- [21] A.K. Parekh. A generalized processor sharing approach to flow control in integrated services networks. *Ph.D. dissertation, Dep. Elec. Eng. Comput. Sci., M.I.T.*, 1992.
- [22] Vern Paxson. *Measurements and Analysis of End-to-End Internet Dynamics PhD. Thesis*. U.C. Berkeley, 1996.
- [23] Manish Jain Ravi Prasad and Constantinos Dovrolis. Effects of interrupt coalescence on network measurements. *Lecture Notes in Computer Science Volume 3015/2004,Pages 247-256*, 2004, 2004.
- [24] SIGAR. System information gatherer and reporter. www.hyperic.com/products/sigar.html.
- [25] Speedtest. www.speedtest.net.
- [26] Cisco Systems. Quality of service on cisco catalyst 4500 series supervisor engines. http://www.ciscosystems.org.ro/en/US/prod/collateral/switches/ps5718/ps4324/prod_white_paper0900aecd8041691c.pdf.

- [27] Brian D. Athey Thomas J. Hacker. *The End-to-End Performance Effects of Parallel TCP Sockets on a Lossy Wide-Area Network*. 2002.
- [28] W Zhang and J. He. Modeling end-to-end delay using pareto distribution. *Proceedings of the Second International Conference on Internet Monitoring and Protection*, 2007.

UAM

UNIVERSIDAD AUTONOMA
DE MADRID