# Análisis de las Prestaciones del Acondicionador de Tráfico CBM en un Dominio DiffServ

María-Dolores Cano, Fernando Cerdán, Joan García-Haro, Josemaría Malgosa-Sanahuja
Departamento de Tecnología de la Información y las Comunicaciones
Universidad Politécnica de Cartagena
Campus Muralla del Mar s/n
30202 Cartagena
Teléfono: 968 325953 Fax: 968 32 59 73
E-mail: {mdolores.cano, fernando.cerdan, joang.haro, josem.malgosa}@upct.es

***Abstract****. The Counters-Based Modified (CBM) traffic conditioner was introduced in a previous work as a feasible option to implement the Assured Forwarding (AF) service in DiffServ. In this paper we present an end-to-end performance analysis of TCP Reno sources that employ the CBM in a DiffServ domain. We present simulation results in a three-RIO-node topology under miscellaneous characteristics: different contract rates, heterogeneous RTT, co-existance of best-effort and AF sources, and eficience of CBM when some network node does not implement service differentiation. As shown in simulation results, it is possible to guarantee an AF service that ensures contracted target rates and performs a fair share of the excess bandwidth.*

## 1 Introducción

Los Servicios Diferenciados (*Differentiated Services*, DiffServ) han sido estandarizados como una de las soluciones más prometedoras a la hora de ofrecer Calidad de Servicio (*Quality of Service*, QoS) en las redes IP [1]. La arquitectura DiffServ hace uso de un esquema sencillo para proporcionar diferentes niveles de QoS en el que la complejidad permanece en los bordes de la red, intentando que los mecanismos empleados en el interior de la misma sean lo más sencillos posible.

La implementación de DiffServ se basa en el uso del byte DSCP (DiffServ *Code Point*) de la cabecera IP. En los nodos frontera o en la propia fuente de tráfico, los paquetes se marcarán, clasificarán y acondicionarán antes de entrar en la red con el fin de recibir un tratamiento particular en los nodos que atraviesen a lo largo de su camino. Este tratamiento que reciben los paquetes en los nodos interiores se conoce como *Per-Hop Behavior* (PHB). Actualmente existen dos PHB estandarizados por el IETF: el *Expedited Forwarding* (EF) PHB [2] y el *Assured Forwarding* (AF) PHB [3].

Los objetivos del servicio AF son asegurar un caudal (*throughput*) mínimo a cada fuente, que normalmente es la tasa contratada, también denominada CIR (*Committed Information Rate*); y además, permitir a las fuentes consumir más ancho de banda del contratado si la carga de la red es baja. El reparto del ancho de banda en exceso entre las diferentes fuentes se ha de realizar de modo justo, encontrándose dos definiciones para el término *justicia*. La primera define *justicia* como el reparto equitativo del ancho de banda en exceso entre todas las fuentes que componen el agregado. Mientras que la segunda definición, determina que un reparto justo del ancho de banda será aquel proporcional al CIR de cada fuente. En este trabajo, así como en la mayor parte de la literatura relacionada, se utiliza la primera definición, pues se asume que si se consigue un reparto equitativo del ancho de banda en exceso, pasar a un reparto proporcional dependerá únicamente del uso de un sistema de ponderación.

Para alcanzar los objetivos del servicio AF, los paquetes de cada flujo individual de tráfico se marcan como pertenecientes a una de las cuatro clases de tráfico AF. Como se detalla en [3], dentro de cada clase de tráfico AF un paquete puede pertenecer a tres niveles distintos de precedencia. En caso de congestión, el nivel de precedencia de un paquete determinará la importancia del mismo dentro de la clase AF a la que pertenece. Un nodo DiffServ que presente congestión descartará preferiblemente paquetes con un nivel de precedencia más alto, protegiendo así a los paquetes con un nivel de precedencia más bajo. A la hora de implementar un servicio AF dentro de la arquitectura DiffServ, se tendrá que definir por tanto qué tipo de funciones se van a utilizar para acondicionar el tráfico en los nodos frontera o en las fuentes de tráfico (marcar, clasificar, aplicar funciones policía, etc.) y cómo construir el AF PHB.

La introducción de RIO (RED (*Random Early Detection*) In y Out) [4] supuso un paso importante en el desarrollo de DiffServ. Este mecanismo que se usa para implementar el AF PHB, utiliza sólo dos niveles de precedencia dentro de cada clase AF. Los paquetes que se consideran dentro del perfil de tráfico de una fuente se marcarán como *in* y los que están fuera del perfil como *out*. Una vez un paquete queda marcado, el agregado de tráfico llega al *router* donde se aplica RIO. RIO es la combinación de dos algoritmos RED [5] con diferentes curvas de probabilidad de descarte, de tal manera que los paquetes *out* tienen más probabilidad de ser

eliminados. RIO utiliza una única cola FIFO (*First In First Out*) para servir ambos tipos de paquetes. La probabilidad de descartar un paquete *out* depende del número total de paquetes de la cola, mientras que la probabilidad de eliminar un paquete *in* depende exclusivamente de la ocupación de la cola con paquetes *in*.

Durante los últimos años se han presentado diferentes propuestas de acondicionadores de tráfico en la literatura especializada [4, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18]. No obstante, ha quedado demostrado que es difícil encontrar un acondicionador de tráfico cuya interacción con los mecanismos de gestión de colas para implementar los PHB permita lograr los dos objetivos del servicio AF. Algunos de los acondicionadores de tráfico propuestos no consiguen garantizar los CIR de modo estricto debido a la gran dependencia que existe con parámetros de la red como por ejemplo el tiempo de ida y vuelta (*Round Trip Time*, RTT). Otros, aún en condiciones favorables donde no hay diversidad en los parámetros de la red, presentan una configuración demasiado compleja que hace que cualquier pequeña variación en los valores de ésta no garantice los contratos. A su vez, existen propuestas que son capaces de asegurar los contratos de los usuarios pero que a la hora de distribuir el ancho de banda en exceso no lo hacen de modo justo (en ninguna de las dos definiciones contempladas para el término *justicia*). Las últimas tendencias en cuanto al desarrollo de acondicionadores de tráfico o bien requieren el uso de señalización, o necesitan una monitorización por flujos en el *router* con los consecuentes problemas de escalabilidad. Además, incluso en estas últimas propuestas que consiguen garantizar los CIR, existen claras deficiencias en cuanto al reparto del ancho de banda sobrante entre las distintas fuentes TCP que componen el agregado.

El acondicionador de tráfico *Counters-Based Modified* (CBM) introducido en [19] se presenta como un enfoque alternativo para conseguir un reparto *justo* del ancho de banda en exceso entre las diferentes fuentes TCP que componen un agregado dentro del servicio asegurado AF. Una vez quedan garantizados los CIR de cada una de las fuentes gracias al marcado de tráfico mediante el algoritmo CB [18], es posible lograr una distribución equitativa del ancho de banda en exceso utilizando una función policía que descarta de manera probabilística paquetes que están calificados como *out*. Empleando este mecanismo, conseguimos mantener la complejidad en los nodos frontera, utilizando exclusivamente RIO para implementar el PHB. La probabilidad de descarte de un paquete *out* se determina asumiendo que el acondicionador de tráfico conoce la cantidad de ancho de banda sobrante y una aproximación del RTT medio de las conexiones. Aunque este hecho implica que sea necesario algún tipo de señalización, ésta es más sencilla que la empleada en otras propuestas de acondicionadores de tráfico. Las primeras simulaciones realizadas en una topología sencilla de un solo nodo con características variadas (diferentes contratos, diferentes RTT y uso compartido de recursos con fuentes *best-effort*) mostraron que CBM consigue garantizar los CIR de cada fuente de manera estricta y repartir el ancho de banda no contratado de modo justo [19].

En este artículo se estudian las prestaciones de CBM cuando se utiliza en conjunción con RIO en un dominio DiffServ más realista formado por varios nodos. El análisis se centra en examinar el funcionamiento de TCP Reno extremo a extremo cuando nos encontramos en una red con características heterogéneas (diferentes contratos, diferentes RTT y coexistencia con tráfico *best-effort*); en concreto, en términos de qué garantías existen de asegurar los CIR de cada fuente TCP y de cómo se realiza el reparto del ancho de banda en exceso entre las distintas fuentes que componen el agregado. Nótese que en este estudio consideramos un reparto justo del ancho de banda en exceso como la distribución equitativa del mismo. Como se muestra en los resultados, es posible lograr justicia en la distribución del ancho de banda sobrante utilizando el acondicionador de tráfico CBM sin perder exactitud a la hora de garantizar los contratos de cada fuente en dominios DiffServ compuestos de tres nodos.

El resto del artículo queda organizado como sigue. En la sección 2 se describe el algoritmo CBM. En la sección 3 presentamos la herramienta de simulación, así como la topología y escenarios de simulación. A continuación, en la sección 4 se muestran y discuten los resultados obtenidos. Finalmente, la sección 5 resume los puntos más importantes de este trabajo.

# 2 El acondicionador de tráfico Counters-Based Modified

Partiendo de la suposición de que todos los paquetes que se inyectan en la red tienen un tamaño similar, se puede afirmar que si las fuentes introducen el mismo número de paquetes entonces cada fuente obtiene la misma porción de ancho de banda. Extendiendo este hecho a los paquetes fuera de perfil, podemos afirmar que si todas las fuentes introducen el mismo número de paquetes *out* entonces se consigue un reparto equitativo del ancho de banda sobrante.

Este comportamiento ideal se ve afectado por el diferente funcionamiento de cada fuente TCP, que se ve influenciada por el efecto de diferentes RTT o diferentes contratos de tráfico. Además, se ha de tener en cuenta la interacción con el mecanismo RIO empleado para la gestión de las colas en los *routers*. Con el objetivo de hacer frente a estos efectos de interacción, CBM penaliza a aquellas fuentes que envían paquetes fuera del perfil por encima del valor ideal: Una penalización basada en el descarte probabilístico de paquetes *out* en el propio acondicionador de tráfico.

En [19] se muestra que las conexiones con contratos pequeños y RTT reducidos generan más paquetes *out* entre paquetes *in* consecutivos, que las conexiones con mayores tasas contratadas y RTT más elevados. En consecuencia, obteniendo las primeras más recursos de la red. A partir de estas observaciones, el algoritmo CBM se desarrolla para funcionar como sigue (véase Fig. 1). Cada acondicionador de tráfico, situado junto a la fuente TCP, fuera del alcance del usuario final, dispone de una variable que cuenta el número de paquetes *out* entre dos paquetes *in* consecutivos. Cada vez que un paquete se marca como *out*, el acondicionador de tráfico CBM comprueba esta variable. Si la variable no sobrepasa un umbral mínimo al que llamaremos *min*, entonces el paquete *out* se inyecta en la red. Si la variable excede un umbral máximo al que denotaremos como *max*, entonces el paquete *out* se elimina. Por último, si la variable permanece entre estos dos umbrales *min* y *max*, el paquete se descarta con una probabilidad a la que llamaremos *p*.

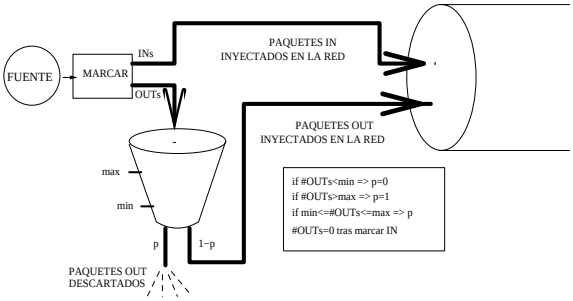


Figura 1. Descarte de paquetes out con CBM.

El uso de este algoritmo requiere por tanto la configuración de los límites *min* y *max* y el cálculo de la probabilidad de descarte *p* que pasamos a describir a continuación. Como se explica en [19], para obtener los valores de *min* y *max* se utilizan las ecuaciones (1) y (2), donde MSS es el acrónimo de tamaño máximo de segmento (*Maximum Segment Size*). El ancho de banda en exceso se puede imaginar como el correspondiente a otra fuente TCP cuya ventana máxima de transmisión fuese el producto ancho de banda en exceso por el RTT medio. De este modo, el umbral máximo *max* quedaría establecido a dicho valor. Una fuente que inyectase un número de paquetes *out* cercano a este límite consumiría casi por completo el ancho de banda sobrante. Además, en el caso en el que una conexión superase dicho límite significaría que no sólo consume todo el ancho de banda en exceso sino que además podría estar *robando* parte del ancho de banda correspondiente al contrato garantizado de otra conexión. En consecuencia, este umbral no debe sobrepasarse nunca. Es conocido el hecho de que en la arquitectura TCP/IP, un algoritmo de crecimiento aditivo y de disminución multiplicativo satisface las condiciones necesarias de convergencia para alcanzar un estado eficiente en la red, siendo utilizado para implementar mecanismos de prevención de la congestión. Por esta razón, se ha optado por un valor mínimo *min* como la mitad del umbral máximo.

$$max = \left\lceil \frac{Ancho\_de\_banda_{exceso} \cdot RTT_{medio}}{MSS} \right\rceil \quad (1)$$

$$min = \left\lceil \frac{max}{2} \right\rceil \quad (2)$$

La estimación del RTT se puede obtener a partir de una señalización enviada de manera periódica entre el *router* frontera y las fuentes TCP. El protocolo TCP implementa un algoritmo de estimación del RTT de la conexión actual. Esta estima puede ser enviada al *router*, el cual se encarga de calcular el RTT medio de las conexiones. Este valor es entonces devuelto a los acondicionadores de tráfico, donde los paquetes son marcados y descartados, si procede. Obsérvese que no se requiere una monitorización de los flujos de tráfico en el *router*, función conocida como *per-flow state monitoring*, en el sentido de que el *router* no mantiene información de cada flujo de paquetes activo. Sólo se encarga de determinar el RTT medio con la información que recibe de las fuentes TCP y una vez calculado, estos valores no se almacenan, a diferencia de otros acondicionadores de tráfico como [7, 8, 14, 15].

Finalmente, la probabilidad de descarte *p* se calcula mediante la ecuación (3). Cada fuente tiene un valor diferente de *p* entre 0 y 1, basándose en la tasa contratada. Por simplicidad denotaremos *x* al cociente tasa contratada entre capacidad del enlace. De las primeras observaciones, donde se advierte que fuentes de contrato pequeño y RTT reducidos generan más paquetes *out* entre dos paquetes *in* consecutivos que el resto de fuentes, sería intuitivo aplicar una ecuación de la forma *p=1-x* (véase Fig. 2). Así, las conexiones con contratos pequeños tendrían una mayor probabilidad de eliminar paquetes *out*. Sin embargo, no hay que olvidar que una vez establecido el umbral máximo, el acondicionador de tráfico elimina aquellos paquetes fuera del perfil que hacen que se supere dicho umbral. El hecho de eliminar paquetes se refleja en las fuentes TCP, que reducen su tasa de transmisión, dejando libres más recursos y permitiendo de este modo que el resto de fuentes introduzcan más tráfico en la red (más paquetes fuera del perfil).

En consecuencia, si utilizamos una ecuación para la probabilidad de descarte que penaliza en mayor medida a las fuentes con contratos pequeños, éstas se ven perjudicadas sobremanera. Así, cuando consiguen recuperarse de las sucesivas pérdidas de paquetes, los recursos están siendo utilizados por las fuentes con contratos mayores. Esta situación provocaría nuevas pérdidas haciendo que las fuentes de menor contrato volvieran a reducir sus tasas de transmisión y se originaría el efecto opuesto: Las conexiones con mayores contratos y RTT más elevados obtendrían más recursos de la red, lo que tampoco es deseable. En consecuencia, se debe utilizar una ecuación para la probabilidad de descarte *p* que tienda a favorecer ligeramente las fuentes con contratos pequeños.

$$p = 2 \cdot \frac{tasa\_contratada / capacidad\_enlace}{1 + tasa\_cot\ ratada / capacidad\_enlace} \quad (3)$$

En [19] se evaluó en principio una ecuación de la forma $p=x$ (véase Fig. 2). Las simulaciones mostraron que el mecanismo CBM realizaba un reparto más equitativo del ancho de banda sobrante que el original CB aunque todavía lejos del comportamiento ideal. Con el objetivo de observar el efecto que tendría sobre el reparto del ancho de banda sobrante una ecuación que, aún favoreciendo a las fuentes de menor tasa contratada lo hiciera de manera no lineal, se realizaron simulaciones con las ecuaciones $p=2 \cdot x/(1+x)$ y $p=x/(2-x)$. Estas dos ecuaciones se incluyen en la Fig. 2. Es importante resaltar que pequeñas variaciones en el valor de *p* pueden generar grandes diferencias a la hora de aplicar el algoritmo debido a la reacción de las fuentes TCP ante la pérdida de paquetes. De estos resultados, se concluyó que la ecuación (3) es la más apropiada para el mecanismo de descarte en CBM. Nótese que esta ecuación se aplica únicamente cuando el número de paquetes *out* entre dos paquetes *in* consecutivos está en el intervalo (*min, max*).

# 3 Escenarios de simulación

El acondicionador de tráfico CBM se evalúa mediante simulaciones en la topología de tres nodos de la Fig. 3 (los cuellos de botella son los propios nodos de la red; T≡Acondicionador de tráfico). Ocho fuentes generan tráfico TCP Reno, transmitiendo a la velocidad máxima del enlace, que ha sido establecida a 33 Mbps. Para comprobar el impacto de diferentes contratos y la influencia de RTT variables, se utilizarán distintos valores en las simulaciones.

La herramienta de simulación empleada para el protocolo de ventana deslizante TCP Reno fue desarrollada en [20] y ha sido ampliamente utilizada en [21, 22]. Además, se usó como herramienta de validación del estudio analítico desarrollado en [23]. Algunas de las características de esta herramienta son las siguientes: todas las fuentes TCP son codiciosas (*greedy sources*) con el fin de tener un peor caso en el que se consigue un estado de congestión de la red elevado; los destinos sólo envían reconocimientos que no presentan pérdidas ni retardos, y el tamaño máximo de la ventana es igual al producto ancho de banda por retardo como es habitual en redes de área amplia (*Wide Area Network*, WAN).

Para las simulaciones se utiliza un tamaño de paquete de 9.188 bytes que corresponde a IP sobre ATM (*Asynchronous Transfer Mode*) y puede representar DiffServ sobre MPLS (*Multi Protocol Label Switching*), donde se ha impuesto el uso de la tecnología ATM entre los fabricantes.

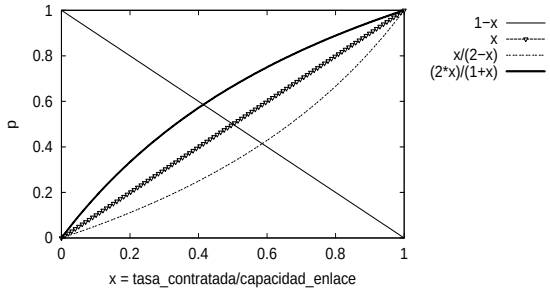


Figura 2. Funciones para la probabilidad de descarte p.

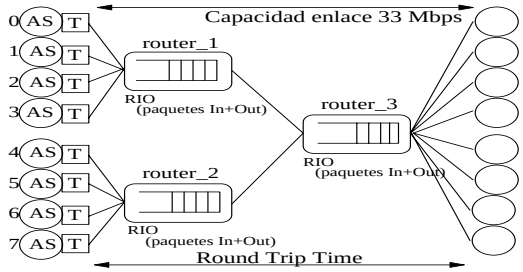


Figura 3. Topología general de tres nodos.

Se van a estudiar tres casos que quedan descritos en las Fig. 4, 5 y 7. Los *routers* almacenan y envían los paquetes de los agregados de tráfico. La gestión de las colas de estos dispositivos se realiza con el mecanismo RIO o con RED según quede indicado. El nodo etiquetado como *router_1* emplea el mecanismo RIO con parámetros ([minth, maxth, maxp]) [40/70/0.02] para los paquetes *in* y [10/40/0.2] para los paquetes *out*. El nodo etiquetado como *router_2* implementa el algoritmo RED con parámetros [10/40/0.2] o el RIO con los mismos valores que los utilizados en *router_1*. El último nodo, *router_3*, recibe el tráfico procedente de los dos nodos anteriores y ejecuta RIO con los mismos valores que los empleados en *router_1*. En cuanto a los valores de los parámetros de configuración que se usan en RED para calcular el tamaño medio de las colas, *weight_in* y *weight_out*, se han establecido a 0.002 siguiendo las recomendaciones de [5].

Obtendremos resultados para cinco escenarios diferentes. Trabajaremos en una situación donde la carga de la red está alrededor del 60%. Esta situación es más interesante para nuestro estudio ya que el ancho de banda en exceso representa una porción importante del ancho de banda total disponible. El intervalo de confianza de los resultados es del 95%, que ha sido calculado con una distribución normal y usando 30 muestras que proporcionan un valor aproximado de ±0.002 en los valores de justicia y de ±0.01 en los *throughputs* alcanzados. El término *throughput* hará referencia al *goodput*, es decir, no se tendrán en cuenta los paquetes retransmitidos. A continuación resumimos las características de los diferentes escenarios de simulación en la Tabla 1.

El escenario A es al que más se ha recurrido en estudios similares sobre prestaciones de acondicionadores de tráfico. Dadas sus características

es presumible obtener en él los mejores resultados. Con la introducción de contratos diferentes en el escenario B pretendemos acercarnos a un ambiente más realista con QoS [24]. En el caso del escenario C, opuesto al escenario B, podemos analizar el efecto producido en las prestaciones del mecanismo CBM por el hecho de tener fuentes con diferente RTT. El escenario D es el más complejo debido a que las conexiones TCP con contratos más bajos y RTT menores se ven claramente favorecidas como se demuestra en [25]. Finalmente, el escenario E por el hecho de asignar RTT mayores a las fuentes con contratos más pequeños evita parcialmente favoritismos en el reparto de los recursos de la red a diferencia de lo que ocurre en D.

Tabla 1. Escenarios de simulación.

| Escenario | Contrato (Mbps) | RTT (ms) |
|---|---|---|
| A | 2.5 | 50 |
| B | 1-1-2-2-3-3-4-4- | 50 |
| C | 2.5 | 10 to 80 a intervalos de 10 |
| D | 1-1-2-2-3-3-4-4 | 10 to 80 a intervalos de 10 |
| E | 4-4-3-3-2-2-1-1 | 10 to 80 a intervalos de 10 |

# 4 Resultados

En esta sección se presentan y discuten los resultados obtenidos para la topología y escenarios descritos anteriormente. Se evalúan las prestaciones de TCP extremo a extremo atravesando una red de tres *routers*. El estudio se realiza en términos de garantías de asegurar los contratos, reparto justo del ancho de banda sobrante y robustez del mecanismo cuando tráfico *best-effort* (BE) comparte recursos con el AF. Esta topología es notablemente más compleja y heterogénea que las empleadas normalmente en la literatura especializada. Los trabajos realizados en esta misma dirección concluyeron que no era asequible garantizar de modo estricto un servicio cuantificable al tráfico TCP [6, 27]. Aunque los últimos estudios presentan resultados más favorables [8], no parece del todo obvia una implementación factible. En todos los casos que se van a estudiar, los acondicionadores de tráfico CBM están situados junto a las fuentes TCP cuando se requiera un servicio asegurado AF. De otro modo, las fuentes pertenecen al servicio *best-effort* y sus paquetes son tratados como fuera del perfil (paquetes *out*).

Para evaluar la justicia utilizamos el índice *f* que se obtiene a partir de la ecuación (4). En esta ecuación, $x_i$ es el exceso en *throughput* de la fuente *i*, y *n* es el número de fuentes que componen el agregado [26]. Conforme más se aproxime a 1 el valor del índice *f* más justicia habrá en el sistema en el reparto del ancho de banda sobrante. Para calcular el índice de justicia *f* utilizamos el término *throughput* en el sentido de *goodput* comentado anteriormente.

$$f = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} x_i^2}; \; f \le 1 \qquad (4)$$

## 4.1 Servicio asegurado AF en una red de tres nodos

Este primer caso de estudio está compuesto por tres routers RIO y ocho fuentes TCP Reno con servicio asegurado como se ilustra en la Fig. 4. Las fuentes generan tráfico a la velocidad del enlace, establecida a 33 Mbps. Las características de los diferentes escenarios A, B, C, D y E fueron descritas en la sección 3. Los valores de los umbrales *min* y *max* utilizados en el mecanismo CBM se incluyen en la Tabla 2. Calculamos estos límites para los *routers router_1* y *router_2*, asumiendo el hecho de que *router_1* no sabe de la existencia de *router_2* y viceversa.

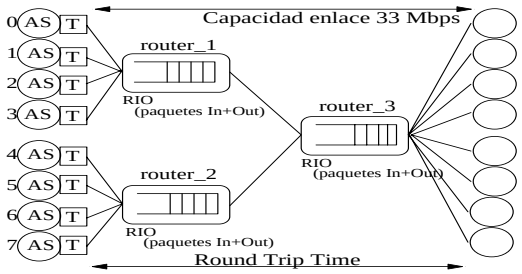


Figura 4. Topología con ocho fuentes TCP Reno con servicio AF y tres nodos RIO.

Tabla 2. Umbrales min y max de cada escenario de simulación con CBM en la topología de tres nodos.

| | Escenario A | Escenario B | Escenario C | Escenario D | Escenario E |
|---|---|---|---|---|---|
| Capacidad del enlace (Mbps) | 33 | 33 | 33 | 33 | 33 |
| **Valores de fuentes nº 0 a 3** | | | | | |
| Σ CIR (Mbps) | 10 | 6 | 10 | 6 | 14 |
| Ancho de banda $_{exceso}$ (Mbps) | 23 | 27 | 23 | 27 | 19 |
| $RTT_{medio}$ (ms) | 50 | 50 | 25 | 25 | 25 |
| *max router_1* (nº paquetes *out*) | 16 | 19 | 8 | 10 | 7 |
| *min router_1* (nº paquetes *out*) | 8 | 9 | 4 | 5 | 4 |
| **Valores de fuentes nº 4 a 7** | | | | | |
| Σ CIR (Mbps) | 10 | 14 | 10 | 14 | 6 |
| Ancho de banda $_{exceso}$ (Mbps) | 23 | 19 | 23 | 19 | 27 |
| $RTT_{medio}$ (ms) | 50 | 50 | 65 | 65 | 65 |
| *max router_2* (nº paquetes *out*) | 16 | 13 | 21 | 17 | 24 |
| *min router_2* (nº paquetes *out*) | 8 | 7 | 12 | 9 | 12 |

Tabla 3. Throughput (Mbps) de paquetes *in* de cada fuente obtenido en las simulaciones con la topología de la Fig. 4.

| Fuente | A | B | C | D | E |
|---|---|---|---|---|---|
| 0 | 2.50 | 0.99 | 2.50 | 0.99 | 3.85 |
| 1 | 2.49 | 1.00 | 2.50 | 0.99 | 3.99 |
| 2 | 2.49 | 1.99 | 2.49 | 2.00 | 3.00 |
| 3 | 2.49 | 1.99 | 2.49 | 2.00 | 2.99 |
| 4 | 2.49 | 2.99 | 2.50 | 2.99 | 1.99 |
| 5 | 2.49 | 2.99 | 2.50 | 2.99 | 1.99 |
| 6 | 2.49 | 3.99 | 2.49 | 3.95 | 1.00 |
| 7 | 2.50 | 3.99 | 2.48 | 3.70 | 1.00 |

Como se observa de los resultados de las simulaciones (véase Tabla 3), el uso combinado de CBM y RIO permite que los usuarios obtengan sus contratos a pesar de la heterogeneidad de la red. El descarte probabilístico de paquetes fuera del perfil es causante de la adaptación por parte de las fuentes TCP a las características de la red. Una vez se ha conseguido garantizar los contratos, cada fuente obtiene una fracción similar del ancho de banda en exceso como queda plasmado en la Fig. 8, en la que el índice de justicia está por encima de 0,8 para todos los escenarios excepto el escenario D.

La distribución poco pareja del ancho de banda sobrante en este escenario D se puede explicar como sigue. *Router_1* recibe el tráfico de las conexiones con contratos pequeños y RTT bajos, mientras que *router_2* se ocupa de los contratos más elevados y los RTT más altos. En una topología de un solo nodo con características misceláneas como éstas no supone ningún problema por la buena interacción entre CBM y RIO. No obstante, en este caso la tarea de distribución el ancho de banda en exceso recae sobre *router_3*. Un nodo que exclusivamente hace uso del gestor de colas RIO y en consecuencia apenas es capaz de proporcionar un reparto equitativo del ancho de banda no contratado ($f$=0,623).

## 4.2 Robustez de CBM frente a fallos en la red

En este segundo caso de estudio, *router_2* implementa RED en lugar de RIO (véase Fig. 5). Esta situación podría ser interesante para un proveedor de servicios de Internet (*Internet Service Provider, ISP*), principalmente porque sería una ventaja poder ofrecer un servicio asegurado con una implementación más sencilla, es decir utilizando RED que básicamente es una cola FIFO capaz de evitar el problema de la sincronización global. Incluso, podría resultarle interesante desde el punto de vista de la reconfiguración de los recursos de la red, siendo capaz de hacer frente a un posible fallo en algún nodo que deba ser reemplazado de modo temporal por otro que sólo sea capaz de implementar una gestión de colas sencilla como RED. De la misma forma que en el caso de estudio anterior, el tráfico se genera con ocho fuentes TCP Reno que contratan un servicio asegurado AF. Los escenarios de simulación A-B-C-D-E son los descritos en la sección 3 y quedan resumidos en la Tabla 2 junto con los valores de los límites *min* y *max*.

Los resultados muestran que se mantienen las garantías de asegurar a cada conexión la tasa contratada, cumpliéndose incluso para el peor caso, el escenario D, donde se alcanzan los contratos tras un intervalo transitorio (véase la Fig. 6 donde se incluyen los primeros 180 segundos de simulación). Nótese que el transitorio en el *throughput* no es relevante para las prestaciones finales y además, se puede considerar despreciable para el resto de escenarios. El índice de justicia se mantiene de nuevo por encima de 0,8 para todos los escenarios excepto para el D (véase Fig. 8). El escenario D es la peor situación en la que nos podemos encontrar, al igual que ocurría en el primer caso de estudio, con la añadidura de que en este segundo caso *router_2* recibe la mayor carga de paquetes *in* (recibe tráfico de las fuentes con mayores contratos) y no implementa diferenciación de servicios (sólo emplea RED). Aún así, el descarte de paquetes *out* en el acondicionador de tráfico CBM origina un equilibrio en el uso compartido del ancho de banda sobrante sin interferir en las garantías de asegurar los contratos de cada fuente.

## 4.3 Robustez de CBM ante el servicio *best-effort*

En este último caso de estudio, estamos interesados en conocer el efecto de la coexistencia de dos tipos de tráficos, servicio asegurado AF y *best-effort*, que además compiten en este caso por los recursos de la red. Normalmente, las implementaciones reales de DiffServ no mezclan diferentes tipos de tráfico como pueden ser el AF y el BE en una misma cola, sino que separan los paquetes correspondientes a uno y otro tipo y se almacenan en colas diferentes. Por este motivo, el objetivo de este tercer caso de estudio no es plantear la unión de tráfico AF y BE en una misma cola como configuración a emplear en redes reales, sino analizar si sería factible para un ISP reaccionar ante un fallo temporal en la red reconfigurando sus recursos de tal manera que ambos tipos de tráfico pudieran compartir la misma cola dentro del *router* sin afectar a las prestaciones de la red. Es decir, garantizando los contratos de las fuentes AF y haciendo un reparto justo del ancho de banda no contratado entre fuentes AF y BE.

Para las simulaciones se han utilizado doce fuentes TCP Reno transmitiendo a la velocidad del enlace (33 Mbps). Las fuentes numeradas del 0 a 3 y las numeradas del 6 al 9 contratan un servicio AF. Las fuentes restantes, 4 y 5 del *router_1* y 10 y 11 del *router_2* son *best-effort* (véase la Fig. 7). Las características de los diferentes escenarios de simulación, que han sido modificadas respecto a los casos anteriores, quedan resumidas en la Tabla 4. Los paquetes procedentes de fuentes *best-effort* se marcan como *out* y por ser *best-effort* estas fuentes no realizan contratos, por lo tanto, tratando de obtener tanto ancho de banda como les sea posible.

A pesar de no realizar diferenciación de servicios en *router_2*, nótese que sólo implementa RED, los resultados indican que los contratos de cada fuente con servicio AF están garantizados. Véase la Tabla 5 donde lógicamente las fuentes *best-effort* no están incluidas. Nuevamente, experimentamos algunos problemas en el escenario D, ya que las fuentes 8 y 9 quedan por debajo de la tasa garantizada. En el escenario D, estas dos conexiones presentan unos RTT de 90 y 100 ms respectivamente, junto con los contratos más elevados (4 Mbps), y ambas confluyen en el router RED (*router_2*). Este hecho provoca que sea *router_3* quien deba dar precedencia a los paquetes dentro del perfil procedentes de las conexiones con mayores contratos, que junto con la presencia de tráfico *best-effort* hace que las fuentes 8 y 9 no alcancen al cien por cien sus contratos.

Debido a las diferencias substanciales que existen entre las distintas conexiones en cuanto a retardos y contratos, no es posible garantizar de modo estricto las tasas contratadas. Sin embargo, no hay que olvidar que el objetivo de mezclar en una misma cola tráfico asegurado AF y tráfico *best-effort* tiene como única finalidad hacer frente a fallos en la red en los que el ISP se vea obligado a utilizar este tipo de configuración. En consecuencia, el hecho de asegurar los contratos prácticamente en su totalidad, donde en el caso peor quedan garantizados al 70%, puede entenderse como un avance a la hora de ofrecer diferenciación de servicios con el servicio asegurado AF cuando coexiste con tráfico BE, incluso cuando no es posible implementar diferenciación de paquetes en alguno de los *routers* de la red.

En cuanto al reparto del ancho de banda en exceso, nuevamente el algoritmo CBM controla el número de paquetes *out* que entran en la red, obligando a las fuentes *best-effort* a adaptarse a las condiciones de la red y generar menos paquetes *out*. Por lo que el índice de justicia *f* se mantiene por encima de 0,75 en todos los escenarios excepto el D (véase la Fig. 8).

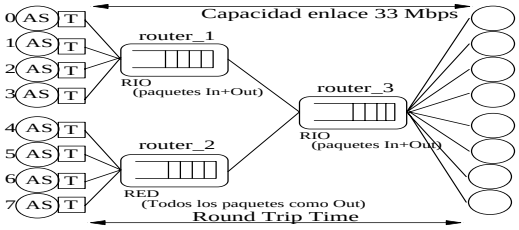


Figura 5. Topología con ocho fuentes TCP Reno con servicio AF y un nodo RED.

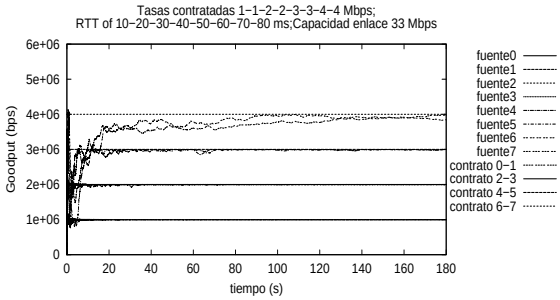


Figura 6. Los contratos de todas las fuentes quedan garantizados con CBM en el peor escenario (D).

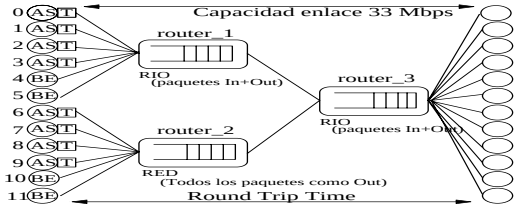


Figura 7. Topología con ocho fuentes TCP Reno con servicio AF, cuatro fuentes TCP Reno con servicio BE y un nodo RED.

Tabla 5. Throughput (Mbps) de paquetes in de cada fuente obtenido en las simulaciones con la topología de la Fig. 7.

| Fuente | A | B | C | D | E |
|---|---|---|---|---|---|
| 0 | 2.50 | 1.00 | 2.49 | 1.00 | 3.30 |
| 1 | 2.49 | 0.99 | 2.50 | 1.00 | 3.99 |
| 2 | 2.49 | 1.99 | 2.49 | 1.99 | 2.99 |
| 3 | 2.49 | 1.99 | 2.50 | 2.00 | 2.99 |
| 6 | 2.49 | 2.99 | 2.49 | 2.90 | 1.99 |
| 7 | 2.49 | 2.99 | 2.49 | 2.70 | 1.99 |
| 8 | 2.50 | 3.97 | 2.50 | 2.70 | 1.00 |
| 9 | 2.49 | 3.99 | 2.49 | 2.60 | 1.00 |

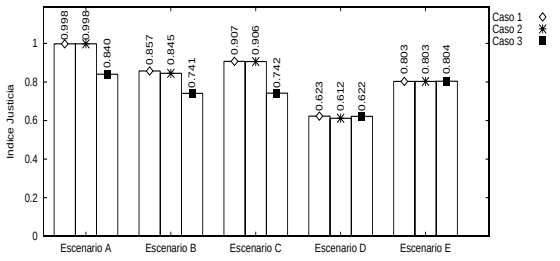


Figura 8. Índice de justicia obtenido en los tres casos de estudio.

Tabla 4. Contratos, RTT y umbrales min y max de las fuentes TCP Reno en la topología de la Fig. 7.

| | CIR (Mbps) | RTT (ms) | *Fuentes 0 a 3* | | *Fuentes 6 a 9* | |
|---|---|---|---|---|---|---|
| | | | *max* | *min* | *max* | *Min* |
| Escenario A | 2.5-2.5-2.5-2.5-0-0-2.5-2.5-2.5-2.5-0-0 | 50 | 16 | 8 | 16 | 8 |
| Escenario B | 1-1-2-2-0-0-3-3-4-4-0-0 | 50 | 19 | 9 | 13 | 7 |
| Escenario C | 2.5-2.5-2.5-2.5-0-0-2.5-2.5-2.5-2.5-0-0 | 10 a120 a intervalos de 10 | 11 | 6 | 30 | 15 |
| Escenario D | 1-1-2-2-0-0-3-3-4-4-0-0 | 10 a120 a intervalos de 10 | 13 | 7 | 25 | 13 |
| Escenario E | 4-4-3-3-0-0-2-2-1-1-0-0 | 10 a 120 a intervalos de 10 | 10 | 5 | 35 | 18 |

# 5 Conclusiones

En este artículo hemos desarrollado un análisis de las prestaciones del acondicionador de tráfico *Counters-Based Modified* (CBM), una nueva propuesta que consigue distribuir de manera justa el ancho de banda en exceso de la red entre las fuentes TCP que componen el agregado, garantizando además de modo estricto los contratos de las fuentes. CBM logra este objetivo gracias al descarte probabilístico de paquetes fuera de perfil que realiza en función del contrato de cada fuente, del ancho de banda en exceso y de una estimación del RTT medio.

Cuando se utiliza CBM en combinación con RIO, se mitiga parcialmente el efecto que tiene sobre las fuentes TCP la diversidad en contratos, en RTT o la coexistencia con tráfico *best-effort*. Así se ha comprobado mediante simulaciones en una topología de tres nodos con características heterogéneas: contratos variables, RTT variables, reacción ante nodos que no implementan diferenciación de paquetes (RED) y coexistencia con tráfico BE. Concluyendo de los resultados de las simulaciones que los contratos quedan garantizados prácticamente en todos los casos independientemente de las particularidades del escenario de simulación. Además el ancho de banda en exceso se distribuye con un índice de justicia que en la mayoría de los casos está por encima de 0,8. En consecuencia, alcanzando los objetivos del servicio Assured Forwarding.

# Agradecimientos

Este trabajo se enmarca dentro del proyecto CICYT FAR-IP (TIC2000-1734-C03-03).

# Referencias

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services”, RFC 2475, December 1998.

[2] B. Davie, A.Charny, J. C. R. Bennet, K. Benson, J. Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, “An expedited forwarding PHB (Per-Hop Behavior)”, RFC 3246, March 2002.

[3] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, “Assured Forwarding PHB Group”, RFC 2597, June 1999.

[4] D. Clark y W. Fang, “Explicit Allocation of Best-Effort Packet Delivery Service”, IEEE/ACM Transactions on Networking, Vol. 6, No. 4, pp. 362-373, August 1998.

[5] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, Vol. 1, No.4, pp. 397-413, August 1993.

[6] J. Ibañez, K. Nichols, “Preliminary simulation evaluation of an assured service”, Internet draft, work in progress, draft-ibanez-diffserv-assured-eval-00.txt, August 1998.

[7] W. Lin, R. Zheng, J. Hou, “How to make assured service more assured”, Proceedings of the 7th International Conference on Network Protocols (ICNP'99), pp. 182-191, Toronto, Canada, October 1999.

[8] B. Nandy, N. Seddigh, P. Pieda, J. Ethridge, “Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks”, Proceedings of Networking 2000, LNCS 1815, Paris, France, pp.540-554, May 2000.

[9] Elloumi O, De Cnodder S, Pauwels K, “Usefulness of the three drop precedences in Assured Forwarding Service”, Internet draft, work in progress, July 1999.

[10] J. Heinanen, R. Guerin, “A single rate three color marker”, RFC 2698, septiembre 1999.

[11] J. Heinanen, R. Guerin, “A two rate three color marker”, RFC 2698, septiembre 1999.

[12] M. Goyal, A. Durresi, P. Misra, C. Liu, R. Jain, “Effect of number of drop precedences in assured forwarding”, Proceedings of Globecom 1999, Rio de Janeiro, Brazil, Vol. 1(A), pp. 188-193, December 1999.

[13] H. Kim, “A Fair Marker”, Internet draft, work in progress, April 1999.

[14] I. Alves, J. De Rezende, L. De Moraes, “Evaluating Fairness in Aggregated Traffic Marking”, Proceedings of IEEE Globecom’2000, San Francisco, USA, pp. 445-449, November 2000.

[15] I. Andrikopoulos, L. Wood, G. Pavlou, “A fair traffic conditioner for the assured service in a differentiated services internet”, Proceedings of IEEE International Conference on Communications ICC2000, New Orleans, LA, Vol. 2, pp. 806-810, June 2000.

[16] Mohamed A. El-Gendy, Kang G. Shin, “Assured forwarding fairness using equation-based packet marking and packet separation”, Computer Networks, Vol. 41 Issue 4, pp. 435-450, 2003.

[17] S. Tartarelli, A. Banchs, “Random Early Marking: Improving TCP Performance in DiffServ Assured Forwarding”, Proceedings of ICC 2002, New York, USA, May 2002.

[18] Maria-Dolores Cano, Fernando Cerdan, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja, “Performance Evaluation of Traffic conditioner Mechanisms for the Internet Assured Service”, in Quality of Service over Next-Generation Data Networks, Proceedings of SPIE Vol. 4524, pp. 182-193, 2001.

[19] Maria-Dolores Cano, Fernando Cerdan, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja, “Counters-Based Modified Traffic Conditioner”, Lecture Notes in Computer Science (QofIS 2002), Vol. 2511, pp. 57-67, Springer-Verlag, 2002.

[20] F. Cerdan, O.Casals, “Performance of Different TCP Implementations over the GFR Service Category”, ICON Journal, Special Issue on QoS Management in Wired &Wireless Multimedia Communications Network, Vol.2, pp.273-286, Baltzer Science, January 2000.

[21] F. Cerdan, O.Casals, “Mapping an Internet Assured Service on the GFR ATM Service”, Lecture Notes in Computer Science (Networking 2000), Vol. 1815, pp. 398-409, Springer-Verlag, 2000.

[22] V. Bonin, F. Cerdan, O.Casals, “A simulation study of Differential Buffer Allocation”, Proceedings of 3rd International Conference on ATM, ICATM’2000, pp. 365-372, Germany, June 2000.

[23] V. Bonin, O.Casals, B. Van Houdt, C. Blondia, “Performance Modeling of Differentiated Fair Buffer Allocation”, Proceedings of the 9th International Conference on Telecommunications Systems, Dallas, USA, 2001.

[24] F. Cerdan, J. Malgosa-Sanahuja, J. Garcia-Haro, F. Burrull, F. Monzo-Sanchez, “Quality of Service for TCP/IP Traffic: An overview”, Proceedings of PROMS’00, pp.91-99, Cracow 2000.

[25] N. Seddigh, B. Nandy, P. Pieda, “Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network”, Proceedings of IEEE Globecom’99, Rio de Janeiro, Brazil, Vol. 3, pp. 1792-1798, December 1999.

[26] R. Jain, “The Art of Computer Systems Performance Analysis”, John Wiley and Sons Inc., 1991.

[27] J. F. De Rezende, “Assured Service Evaluation”, Proceedings of IEEE Globecom’99, Rio de Janerio, Brazil, December 1999.