

# Control Difuso de la Tasa de Transferencia de Extremo a Extremo en Protocolos de Transporte de Internet

Federico Montesino<sup>1</sup>

Diego R. Lopez<sup>2</sup>

Ángel Barriga<sup>1</sup>

Santiago Sánchez-Solano<sup>1</sup>

<sup>1</sup>Instituto de Microelectrónica de Sevilla. Centro Nacional de Microelectrónica – CSIC  
Avda. Reina Mercedes, s/n. Edif. CICA. E-41012 Sevilla  
{Federico.Montesino,Angel.Barriga,Santiago.Sanchez}@imse.cnm.es

<sup>2</sup>RedIRIS, Red Española de I+D  
Edif. Bronce, Plaza Manuel Gómez Moreno s/n, Planta 2. E-41012 Madrid  
Diego.Lopez@rediris.es

## Resumen

La dinámica del tráfico de extremo a extremo en Internet es un problema complejo para el cual los modelos disponibles son, en el mejor de los casos incompletos. Esta comunicación describe nuevos mecanismos para regulación de la tasa de transferencia de extremo a extremo en la capa de transporte por medio de sistemas difusos. Se describen una generalización basada en lógica difusa de los mecanismos de control de flujo y congestión de TCP (Transport Control Protocol), el diseño de un regulador difuso basado en mecanismo de ventana para TCP, así como la metodología de diseño empleada para simular e implementar de manera experimental el sistema. Se resume un estudio comparativo del regulador difuso presentado frente a los mecanismos tradicionales. El regulador difuso resulta útil como enfoque de modelado y proporciona significativas mejoras de prestaciones respecto a un conjunto de criterios.

**Palabras Clave:** TCP, Control difuso, Inferencia difusa, Control de flujo, Control de congestión, Protocolo de transporte, Ventana de congestión.

## 1. Introduction

La dinámica del tráfico de paquetes de extremo a extremo en Internet es un problema complejo para el cual los modelos disponibles son, en el mejor de los casos, incompletos [7]. Dos áreas de investigación en este ámbito reciben amplia atención actualmente: protocolos de transporte para redes de altas prestaciones, y soluciones para nuevos servicios y aplicaciones.

---

Este trabajo ha sido financiado parcialmente por los proyectos TEC2005-04359/MIC del Ministerio de Educación y Ciencia y TIC2006-635 de la Junta de Andalucía.

Los esquemas de regulación de tráfico desplegados actualmente en Internet (así como las alternativas propuestas) siguen uno de los dos enfoques siguientes [6]: a) control distribuido entre los nodos finales, y b) planificadores de colas en routers o nodos intermedios. Por ello, la regulación de flujos de paquetes entre fuente y destino puede involucrar a todos los nodos en el camino de extremo a extremo y se realiza tanto en cada uno de los saltos entre nodos como entre los extremos de las conexiones. Así, se tiene un sistema complejo formado por múltiples bucles realimentados cuyas interacciones son difíciles de modelar.

Ambos enfoques de regulación de tráfico se pueden redefinir en términos de sistemas difusos [12]. Mientras que existen diversas propuestas de reguladores difusos para colas de paquetes en routers de Internet [11], el único trabajo publicado sobre la aplicación de sistemas difusos a la regulación de tráfico de extremo a extremo que conocemos [3] plantea un estudio preliminar mediante simulación off-line basado en un controlador Takagi-Sugeno aplicado a algunos de los mecanismos básicos de TCP. Asimismo, se han propuesto sistemas difusos para mejorar los esquemas de control de congestión actuales en redes inalámbricas [14, 4].

En esta comunicación estudiamos el control de la tasa de transferencia de extremo a extremo. Más específicamente, describimos un regulador difuso que generaliza los mecanismos de control de flujo y congestión basados en ventana de TCP. Así, pretendemos proporcionar una reinterpretación de los mecanismos actuales, que aporta diversas mejoras en cuanto a prestaciones e interpretabilidad.

## 2. Generalización del Control de Tráfico de Extremo a Extremo Basado en Ventana

Existen diversos esquemas de control de la tasa de transferencia en protocolos de transporte de Inter-

net [7], de entre los cuales destacan los basados en ventana y los basados en ecuaciones. El protocolo de transporte predominante en la Internet actual, TCP, define mecanismos de regulación basados en ventana para implementar funciones de control de flujo y congestión.

Los mecanismos de control de la tasa de transferencia de TCP comprenden cuatro algoritmos fuertemente acoplados entre sí: slow start, congestion avoidance, fast retransmit y fast recovery, definidos y ampliados de manera progresiva durante las últimas dos décadas [1, 9]. Por simplicidad, consideraremos en primer lugar la versión simplificada de los mecanismos de control de flujo y congestión de TCP que se muestra en el algoritmo 1.

---

**Algoritmo 1** Esquema AIMD (Additive Increase Multiplicative Decrease) Básico

---

```

if congestion then
     $w_i = d w_{i-1}, \quad (d < 1)$ 
else
     $w_i = w_{i-1} + u, \quad (u \ll w_{max})$ 
end if

```

---

Donde  $w_i$  son los valores que toma consecutivamente la ventana de congestión, definida como un valor de referencia que limita la cantidad de datos sin recibo acusado que una fuente de datos de una conexión TCP puede transmitir.  $w_{max}$  es el producto retardo - ancho de banda del camino de extremo a extremo. El algoritmo trata de reaccionar con rapidez ante situaciones de congestión. Cuando la red está congestionada, se adopta una ley de actualización de la ventana de congestión de tipo multiplicativo, lo que bajo condiciones de congestión persistente lleva a una reducción exponencial de la ventana de congestión.

Si no se detecta congestión, se adopta una ley de actualización de la ventana de congestión de tipo aditivo. Pese a la efectividad de este esquema bajo un gran rango de condiciones [9], bajo determinadas condiciones puede llevar a un pobre aprovechamiento de los recursos de la red, especialmente en redes de altas prestaciones, para las que en la actualidad existe una gran actividad de investigación en protocolos de transferencia eficientes, muchos de los cuales se han definido como variaciones o ampliaciones de TCP [15, 6].

En este contexto, las interacciones entre los diferentes componentes de la red conforman un sistema complejo caracterizado por una dinámica altamente no lineal que dificulta el desarrollo, simulación y prueba de esquemas de regulación de tráfico.

El algoritmo AIMD básico distingue dos etapas en la evolución de la ventana de congestión. Cada etapa

corresponde a un estado de red claramente identificado que lleva a la aplicación de una ley de actualización de la ventana de congestión específica [9]. No obstante, es común que el estado real de la red no se pueda definir ni conocer de manera unívoca. Es decir, la lógica binaria descrita no describe el sistema de manera correcta en aquellas situaciones en que el estado de la red no se corresponde con ninguna de las dos etapas consideradas. En casos intermedios, la respuesta del sistema es demasiado agresiva o demasiado conservadora. Podemos formular una primera generalización del algoritmo AIMD del siguiente modo [3]:

$$w_{i+1} = w_i + \alpha_D f_D(w_i) + \alpha_I f_I(w_i) \quad (1)$$

Donde  $f_D$  y  $f_I$  son funciones que establecen las leyes respectivas de incremento y decremento de la ventana de congestión, es decir,  $f_I$  es una generalización de la ley aditiva, mientras que  $f_D$  generaliza la ley multiplicativa.  $\alpha_D$  y  $\alpha_I$  se pueden considerar valores de certeza que representan en qué grado se encuentra el sistema en estado de congestión (decremento de la ventana) o sin congestión (incremento de la ventana). Ambos valores se pueden definir de manera mutuamente excluyente,  $\alpha_D, \alpha_I \in [0, 1]; \alpha_D + \alpha_I = 1$ .

### 3. Diseño de un Regulador Difuso de Tráfico de Extremo a Extremo Basado en Ventana

De la ecuación 1 se desprende la posibilidad de gestionar el proceso de actualización de la ventana de congestión mediante sistemas de inferencia difusa. En lugar de considerar que la red se encuentra en un estado de entre un conjunto de estados mutuamente excluyentes, consideraremos que la red puede encontrarse simultáneamente en varios estados con un grado de certeza variable. El grado de certeza se inferirá mediante bases de reglas difusas.

La definición completa del esquema de control de flujo y congestión de TCP [1] comprende cuatro algoritmos (slow start, congestion avoidance, fast recovery y fast retransmit) para los cuales se define un conjunto de aproximadamente 20 leyes o reglas de actualización de la ventana de congestión. La definición detallada de estas reglas y su redefinición en términos de sistemas difusos se puede consultar en [10].

En el esquema AIMD estándar actual de TCP se define el parámetro *ssthresh*, que se emplea como valor umbral para seleccionar las etapas slow start y congestion avoidance según se muestra en el algoritmo 2. Por otra parte, las etapas fast retransmit y fast recovery se activan cuando se detectan acuses de recibo repetidos que indican posibles pérdidas de paquetes.

---

**Algoritmo 2** Esquema AIMD estándar
 

---

```

if  $w < ssthresh$  then
  Realizar slow start
else if  $w \geq ssthresh$  then
  Realizar congestion avoidance
end if
  
```

---

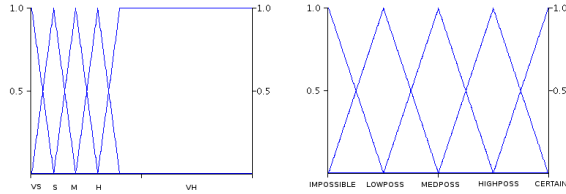


Figura 1: Funciones de pertenencia de  $Tloss$  y  $Tconfidence$

En el esquema ampliado mediante sistemas difusos que proponemos, se definen tres fases de control de la ventana de congestión (para slow start, congestion avoidance y fast retransmit/fast recovery). Puesto que el estado real de la red se conoce con cierto grado de incertidumbre, se considera que todos los estados se dan al mismo tiempo con diversos grados de certeza. Tres sistemas de inferencia difusa proporcionan como salidas estimaciones del grado en que se tiene la certeza de que la red se encuentra en cada uno de los tres estados definidos.

Las entradas consideradas son las siguientes:  $w$ , valor actual de la ventana de congestión;  $ssthresh$ , valor umbral de entrada en la etapa slow start;  $loss$ , porcentaje de pérdidas acumulado durante la conexión;  $rtt$ , tiempo de ida y vuelta de paquetes de extremo a extremo; y  $timeout$ , activa si ha expirado algún temporizador que indique que se ha de retransmitir un paquete por no haberse recibido acuse de recibo.

La figura 1 muestra la definición de etiquetas lingüísticas para la entrada  $loss$  y la salida de los sistemas,  $confidence$ , para la que se definen 7 etiquetas para grados de certeza crecientes entre imposible (IM) y cierto (CE). Se han definido etiquetas para los rangos de valores considerados relevantes en los análisis de prestaciones de TCP y otros protocolos relacionados [15, 4].

Las bases de reglas de los sistemas de inferencia del estado slow start (SlowStartConfidence) y fast retransmit/fast recovery (FRFRConfidence) se muestran en las tablas 1 y 2, respectivamente. La base de reglas del sistema de inferencia del estado congestion avoidance (CongestionAvoidance) emplea las mismas entradas que SlowStartConfidence. Véase [10] para detalles adicionales que no mostramos por restricciones de

Cuadro 1: Base de reglas SlowStartConfidence

$timeout$	$w$	$ssthresh$	$ss - w$	$loss$	$rtt$	$conf.$
VS	x	x	x	x	x	HI
S	x	x	x	x	x	HI
M	x	x	x	x	x	ME
H	x	x	x	x	x	ME
VH	x	x	x	x	x	IM
x	IW	x	x	x	x	CE
x	S	x	x	x	x	LO
x	M	x	x	x	x	LO
x	H	x	x	x	x	LO
x	VH	x	x	x	x	IM
x	x	VS	x	x	x	IM
x	x	S	x	x	x	LO
x	x	M	x	x	x	LO
x	x	H	x	x	x	LO
x	x	VH	x	x	x	ME
x	x	x	VS	x	x	IM
x	x	x	S	x	x	LO
x	x	x	M	x	x	ME
x	x	x	H	x	x	HI
x	x	x	VH	x	x	CE
x	x	x	x	VH	VH	IM
x	x	x	x	VH	H	LO
x	x	x	x	H	VH	LO
x	x	x	x	VS	VS	ME

Cuadro 2: Base de reglas FRFRConfidence

$timeout$	$ssthresh - w$	$loss$	$rtt$	$conf.$
VS	x	x	x	IM
S	x	x	x	IM
M	x	x	x	IM
H	x	x	x	IM
VH	x	x	x	CE
x	VS	x	x	IM
x	S	x	x	LO
x	M	x	x	ME
x	H	x	x	HI
x	VH	x	x	CE
x	x	VH	VH	CE
x	x	VH	H	HI
x	x	H	VH	LO
x	x	M	H	HI

espacio.

Las reglas de actualización de  $ssthresh$  y  $w$  empleadas en las implementaciones comunes de TCP se mantienen. Sin embargo, dado que el estado de la red es en general incierto, todas las reglas se evalúan simultáneamente y se aplican en un grado variable dado por los tres sistemas de inferencia difusa. Las reglas asociadas a las fases slow start y congestion avoidance son evaluadas en todo momento, mientras que las asociadas a la fase fast retransmit/fast recovery son evaluadas cuando se detectan paquetes de acuse de recibo duplicados. Así, mediante estos tres sistemas se puede describir, utilizando reglas lingüísticas, estados de la red cuya definición formal puede alcanzar una gran complejidad y no ser viable.

En el caso más sencillo, los tres conjuntos de reglas se combinan como sigue. Si denotamos por  $\mu_{ss}$ ,  $\mu_{ca}$  y  $\mu_{FRFR}$  las salidas de los tres sistemas de inferencia difusa, y por  $f_{ss}$ ,  $f_{ca}$  y  $f_{FRFR}$  los valores dados por las reglas de actualización estándar (que podemos ge-

neralizar como  $\mu_i$  y  $f_i$  ( $1 \leq i \leq n$ ) para un número variable  $n$  de estados de red (o etapas del algoritmo de regulación de transferencia, o conjuntos de reglas de actualización)), definimos el grado de certeza de un estado de la red o etapa de control como:

$$c_i = \frac{\mu_i}{\sum_{j=1}^n \mu_j}$$

Donde las bases de reglas deben verificar  $\sum_{j=1}^n \mu_j > 0$ . La regla final de actualización de  $w$  se calcula como la media ponderada de las reglas asociadas a todas las posibles etapas de control (de manera análoga a la ecuación 1):

$$w_{i+1} = w_i + \sum_{i=1}^n c_i f_i(w_i, ssthresh)$$

En el caso que describimos,  $n = 3$ . De este modo, se combinan, empleando reglas lingüísticas, leyes de actualización de la ventana de congestión que han mostrado su efectividad bajo ciertas condiciones. Los conjuntos de reglas lingüísticas pueden llevar a soluciones de compromiso entre condiciones complejas. Por ejemplo, algunas de las reglas asociadas a la etapa fast retransmit normalmente implican una reducción drástica de la ventana de congestión. Cuando algunos parámetros tomados como entradas de los sistemas de inferencia indican que la red está congestionada, tales reglas de actualización son adecuadas. Sin embargo, cuando la red no parece estar congestionada, un comportamiento más agresivo del regulador puede mejorar las prestaciones en términos de tasa de transferencia y velocidad de respuesta a problemas transitorios.

#### 4. Metodología de desarrollo y cadena de herramientas

Como resultado de más de una década de investigación sobre la implementación digital de sistemas difusos, el IMSE ha desarrollado y consolidado metodologías de diseño [2] y una cadena de herramientas de CAD [13] que cubren el flujo de diseño de sistemas difusos. Con el entorno Xfuzzy como base, se han articulado una metodología y una cadena de herramientas adaptada al desarrollo de módulos de inferencia difusa para regulación de tráfico de Internet, según se esquematiza en la figura 2. Así, queda cubierto todo el proceso de desarrollo, desde la etapa inicial de descripción mediante un lenguaje de alto nivel hasta su implementación como componentes software.

Las etapas posteriores a la especificación se han adaptado al diseño de reguladores de tráfico del siguiente

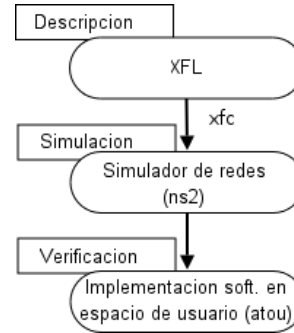


Figura 2: Flujo de diseño de sistemas difusos

modo. Para simular el procesamiento y control de tráfico de red se emplea el simulador ns-2 [8], un simulador discreto dirigido por eventos que soporta una extensa variedad de protocolos y sistemas de colas, así como esquemas de rutado y de acceso al medio, y reproduce el comportamiento de la pila de protocolos completa de cada nodo de una red simulada. Los reguladores difusos se integran dentro de ns-2 como componentes implementados en C. La verificación se realiza mediante implementaciones del protocolo TCP en espacio de usuario, en concreto se ha empleado la herramienta atou [5], ampliada según se describe en el apartado 6.

#### 5. Resultados de Simulación

La simulación de los sistemas difusos se ha realizado mediante el simulador ns-2 [8], el estándar de facto dentro la comunidad de investigadores en protocolos Internet. Mostramos resultados de una comparación entre TCP Reno [1] (la versión de TCP más extendida actualmente en Internet), HighSpeed TCP [6] y el sistema difuso descrito en esta comunicación, que denominaremos Fuzzy TCP.

En la figura 3 se compara la evolución de la ventana de congestión para los tres sistemas, mientras que la figura 4 contrasta la evolución de la tasa de transferencia. En el escenario de red simulado se establecen conexiones TCP durante 10 segundos a través de caminos de extremo a extremo con 10 nodos intermedios, cuello de botella de 100 Mbps de ancho de banda en los bordes y 400 ms de tiempo de ida y vuelta medio. Se simulan asimismo pérdidas aleatorias en uno de los nodos intermedios, y se introduce tráfico cruzado con diversas fuentes, caminos y tiempos de ida y vuelta, consistente en un 10 % de tráfico UDP y 90 % de tráfico TCP Reno.

Como conclusión general, la versión difusa de TCP muestra mayor robustez ante pérdidas. Por ello, la tasa de transferencia final se incrementa (aproximadamente en un 12 % y 11 % frente a TCP Reno y HighSpeed

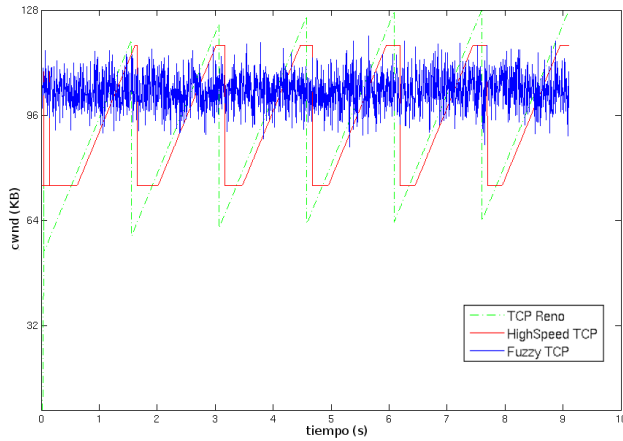


Figura 3: Evolución de la ventana de congestión (simulación)

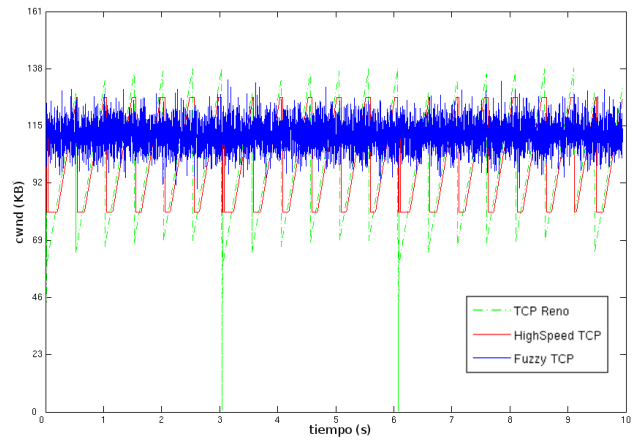


Figura 5: Evolución de la ventana de congestión (implementación)

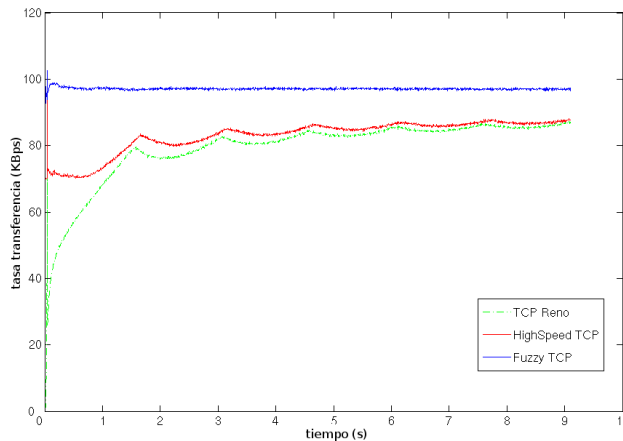


Figura 4: Evolución de la tasa de transferencia (simulación)

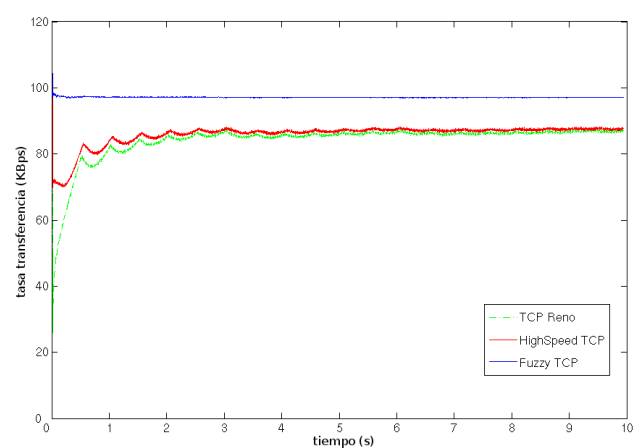


Figura 6: Evolución de la tasa de transferencia (implementación)

TCP, respectivamente). Otras simulaciones realizadas en un conjunto de escenarios representativo y con condiciones de congestión elevada confirman que el regulador difuso reacciona de manera rápida y adecuada. Asimismo, se satisfacen los principios de equidad [15] considerados comúnmente en el diseño de esquemas de control de congestión de extremo a extremo en Internet.

## 6. Resultados de Implementación

Se ha desarrollado una implementación experimental de TCP con ampliaciones difusas como implementación en espacio de usuario (en lugar de en el núcleo del sistema operativo). De este modo, la flexibilidad de experimentación es mayor. La herramienta realizada está basada en la utilidad «Almost TCP over UDP» (atou) [5]. Se han efectuado los cambios necesarios en atou para incorporar el esquema de inferencia difusa

presentado anteriormente. Estos sistemas se generaron como código C empleando el entorno Xfuzzy. La herramienta atou modificada genera registros detallados de eventos y trazas de paquetes con nivel de detalle configurable.

Se han introducido además modificaciones para activar y desactivar opciones de control y algoritmos complementarios de TCP. De este modo, se dispone de un entorno flexible para experimentar con implementaciones de TCP significativamente modificadas entre dos nodos cualquiera de Internet.

Mostramos una comparación de las tres variantes de reguladores de TCP comparadas anteriormente mediante simulación. Las figuras 5 y 6 comparan la evolución de la ventana de congestión y la tasa de transferencia. El escenario de red es el siguiente: dos nodos transfieren datos durante 10 segundos; el camino de la conexión TCP contiene 5 nodos intermedios. El cue-

llo de botella del camino es 100 *Mbps* y el tiempo de ida y vuelta medio es 110 *ms* aproximadamente. Así, se confirman los resultados de simulación en cuanto a las mejoras en robustez y volumen de transferencia. La mejora global de la tasa de transferencia es de aproximadamente un 15 % y un 13 % frente a TCP Reno y HighSpeed TCP, respectivamente.

## 7. Conclusiones

Se ha presentado una reinterpretación de los mecanismos de control de flujo y congestión de TCP en términos de lógica difusa. El modelo difuso descrito da una perspectiva basada en reglas de los esquemas de regulación de tráfico actuales, constituyendo una primera aportación en la aplicación de sistemas difusos a la inferencia de estados de la red en nodos finales.

Los resultados de simulación e implementación muestran que el sistema presentado mejora las prestaciones respecto a un conjunto de criterios que incluye velocidad de convergencia a la tasa de transferencia alcanzable, volumen final de transferencia y magnitud de las oscilaciones en torno a la tasa estabilizada para transferencias prolongadas. El enfoque seguido facilita asimismo la definición de soluciones de compromiso aptas para diferentes requisitos de usuario.

Como continuación de la línea de desarrollo iniciada aquí consideramos la incorporación de entradas adicionales y el empleo de técnicas de aprendizaje supervisado con objeto de refinar las reglas a priori definidas.

## Referencias

- [1] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581, Internet Engineering Task Force, Network Working Group, 1999. status: Proposed Standard.
- [2] A. Cabrera, S. Sánchez-Solano, P. Brox, A. Barriga, and R. Senhadji. Hardware/Software Code-sign of Configurable Fuzzy Control Systems. *Applied Soft Computing*, 4(3):271–285, Dic. 2004.
- [3] P. Carbonell, J. Zhong-Ping, and S. Panwar. Fuzzy TCP: A Preliminary Study. In *15th IFAC World Congress*, pages 21–26, Barcelona, Spain, Jul. 2002.
- [4] L. Chang and I. Marsic. Fuzzy Reasoning for Wireless Awareness. *International Journal of Wireless Information Networks*, 8 (1):15–26, Ene. 2001.
- [5] T. Dunigan, F. Fowler, et al. Almost TCP over UDP (atou). Oak Ridge National Laboratory, U.S Department of Energy, Feb. 2006. <http://www.csm.ornl.gov/~dunigan/net100/atou.html>.
- [6] S. Floyd. HighSpeed TCP for Large Congestion Windows. RFC 3649, Internet Engineering Task Force, Network Working Group, Dic. 2003. Category: Experimental.
- [7] S. Floyd and E. Kohler. Internet Research Needs Better Models. In *ACM SIGCOMM First Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, New Jersey, USA, Oct. 2002.
- [8] Information Sciences Institute. University of Southern California, Viterbi School of Engineering. The Network Simulator – ns-2. <http://www.isi.edu/nsnam/ns/>, Abr. 2006.
- [9] V. Jacobson and M. J. Karels. Congestion Avoidance and Control. *ACM Computer Communication Review SIGCOMM '88 Symposium: Communications Architectures and Protocols*, 18(4):314–329, Ago. 1988.
- [10] F. Montesino, D. R. Lopez, A. Barriga, and S. Sánchez-Solano. Fuzzy End-to-End Rate Control for Internet Transport Protocols. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Vancouver, Canada, Jul. 2006.
- [11] F. Montesino, D. R. Lopez, A. Barriga, and S. Sánchez-Solano. Intelligent Scheduling of Aggregate Traffic in Internet Routers by Means of Fuzzy Systems. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, Paris, France, Jul. 2006.
- [12] F. Montesino, D. R. Lopez, A. Barriga, and S. Sánchez-Solano. Sistemas Difusos para Control de Congestión y Calidad de Servicio en Internet. In *XII Congreso Español Sobre Tecnologías y Lógica Fuzzy (ESTYLF '2004)*, Jaén, España, Sep. 2004.
- [13] F. Moreno-Velo, I. Baturone, S. Sánchez-Solano, and A. Barriga. Rapid Design of Fuzzy Systems With Xfuzzy. In *FUZZ-IEEE'03. The 12th IEEE International Conference on Fuzzy Systems*, pages 342–347, Mayo 2003.
- [14] Ruy de Oliveira and Torsten Braun. A Delay-based Approach Using Fuzzy Logic to Improve TCP Error Detection in Ad Hoc Networks. In *IEEE Wireless Communications and Networking Conference*, Atlanta, USA, Mar. 2004.
- [15] J. Wang, D. X. Wei, and S. H. Low. Modelling and Stability of FAST TCP. In *24th IEEE INFOCOM*, pages 938–948, Miami, FL, USA, Mar. 2005.