

ESTRATEGIA PARA EL PRONÓSTICO DEL RTO DE FAST TCP, MEDIANTE LÓGICA DIFUSA

Carlos Peña

Ingeniero de Sistemas
Universidad de la Amazonia
capeñam@uniamazonia.edu.co
Florencia, Colombia

Octavio Salcedo Parra

Magister en Ciencias de la Información y las Comunicaciones
Docente planta de la Universidad Distrital Francisco José de Caldas
ojsalcedop@udistrital.edu.co
Bogotá, Colombia

Sergio Enrique Hernández Rubio

Especialista Matemáticas Avanzadas
Universidad de la Amazonia
sergioehr@gmail.com
Florencia, Colombia

Tipo: Artículo de investigación

Fecha de Recepción: Febrero 11 de 2013

Fecha de Aceptación: Mayo 2 de 2013

STRATEGY FOR RTO FORECAST OF FAST TCP, USING FUZZY LOGIC

ABSTRACT

This paper presents the design and simulation of a strategy for the retransmission timer forecast control algorithm of FAST TCP congestion. The strategy uses computational intelligence technique called fuzzy logic. The simulation results show that through this new form of RTO forecast, FAST TCP makes better track of the actual situation of network congestion, in the context of a connection between two ends thereof.

Key words: mechanical device, micro-controller, speech recognition.

RESUMEN

Este artículo presenta el diseño y la simulación de una estrategia para el pronóstico del temporizador de retransmisión del algoritmo de control de la congestión FAST TCP. La estrategia utiliza la técnica de inteligencia computacional denominada lógica difusa. Los resultados de la simulación demuestran que a través de esta nueva forma de pronóstico del RTO, FAST TCP hace un mejor seguimiento a la situación real de congestión de la red, en el marco de una conexión entre dos extremos de la misma.

Palabras claves: control de congestión, FAST TCP, reconocimientos, retransmisión.

1. INTRODUCCIÓN

A comienzos del primer decenio del siglo XXI nace el algoritmo de control de la congestión FAST TCP (Fast Active queue management Scalable Transmission Control Protocol), una variante de TCP (Transmission Control Protocol), que soporta tasas de transferencia del orden de los Gigabits por segundo sobre enlaces de gran distancia [1] y surge como una evolución de versiones anteriores de TCP que controlan la congestión en entornos de redes con un elevado producto ancho de banda-retardo, o lo que es similar, redes de altas prestaciones.

FAST TCP, está dividido en cuatro componentes principales: control de datos, control de ventana, control de explosividad y componente de estimación. Estos pueden ser diseñados por separado y actualizados de forma asíncrona, ya que son funcionalmente independientes [2].

“El componente de control de datos determina cuales paquetes transmitir, el control de ventana determina cuantos paquetes transmitir, y el control de explosividad determina cuando transmitir esos paquetes” [2]. Estas decisiones se toman sobre la base de la información proporcionada por el componente de estimación, el cual calcula para cada segmento de datos enviado el retardo en cola (dato que proporciona varios bits de información) y una indicación de pérdida o no pérdida de segmentos (dato que proporciona un solo bit de información). Esta información es usada por los otros tres componentes.

FAST reacciona ante la recepción de reconocimientos positivos o negativos. El reconocimiento de un segmento es identificado como positivo, cuando su recepción en la fuente TCP se registra antes de la expiración de un temporizador de retransmisión (RTO). Este mismo reconocimiento se identifica como negativo cuando se registran tres reconocimientos duplicados o cuando expira el RTO.

En el caso de reconocimientos negativos, se genera una indicación de pérdida del segmento de datos para el componente de control de da-

tos, el que a su vez lo retransmite.

El algoritmo FAST TCP delega en TCP la función del pronóstico del RTO. Esta variable representa el tiempo máximo que una fuente TCP debe esperar por la llegada del reconocimiento de un segmento.

Desde el año 2000 hasta la actualidad, el enfoque adoptado por TCP para la actualización del RTO se basa en una estrategia de pronóstico estadística [4] que tiene en cuenta tanto el valor del tiempo de ida y vuelta, RTT (Round Trip Time), como la variación de este valor en el tiempo. No obstante, la complejidad del tráfico TCP, genera un amplio rango de posibles RTT's, entre cualquier par de extremos de la red y el enfoque mencionado para el pronóstico del RTO no hace un seguimiento ajustado a estas variaciones, es decir, que la diferencia entre el RTO pronosticado para el siguiente segmento de datos a enviar y el RTT registrado para ese mismo segmento (posteriormente denominada Δ_{RTO_RTT}), tiene una alta variación.

Cuando el flujo de datos circulando a través de un enlace se acerca a la capacidad del mismo, un pronóstico adecuado del RTO que siga el comportamiento del RTT es trascendental para garantizar que FAST TCP mantenga tasas de transferencia estables. De esta manera se evita estimular la sobrecarga en cola de los dispositivos intermedios, y por lo tanto la retransmisión de paquetes.

En este artículo se plantea una nueva estrategia de pronóstico del RTO basada en lógica difusa, la cuál se adapta a las variaciones del RTT en redes con un elevado producto ancho de banda-retardo, que utilizan el algoritmo FAST TCP para controlar la congestión.

Ya en 2005 se desarrolló una estrategia mediante lógica difusa para el pronóstico del temporizador de retransmisión de la versión TCP-Reno de TCP [5]. Su objetivo es hacer un mejor seguimiento a las variaciones de los tiempos de ida y vuelta de paquetes, en esta ocasión, en redes con un bajo producto ancho de banda-retardo y con alto grado de congestión.

En marzo de 2011 se patentó un nuevo método para transmisión en redes [6]. La novedad que presenta esta patente respecto al método convencional de transmisión en redes, es que ajusta el temporizador de retransmisión de TCP por medio de un sistema difuso. El nuevo método minimiza la retransmisión y pérdida de paquetes en redes de bajas prestaciones, logrando una mejor utilización del canal de comunicación.

Aunque los últimos dos trabajos mencionados incluyen sistemas difusos al sistema de reconocimientos y retransmisiones de TCP, ninguno involucra a FAST TCP.

2. METODOLOGÍA

En esta primera etapa se realizó el análisis y diseño del sistema de inferencia difusa, la cual incluye la descripción del modelo matemático, la evolución del algoritmo para el pronóstico del temporizador de retransmisión y el diseño del sistema de inferencia difusa para el pronóstico del mismo, así como su validación en un software de computación técnica.

2.1. Algoritmo para el pronóstico del RTO

TCP se adapta a la variación de los retardos en red usando un algoritmo de retransmisión adaptativo. El protocolo monitorea el comportamiento de cada conexión y deduce valores razonables para el RTO. Los datos necesarios para el algoritmo de retransmisión adaptativo se consiguen registrando el tiempo en el cual cada segmento es enviado y el tiempo de llegada de su respectivo reconocimiento, con estos dos registros TCP calcula un tiempo transcurrido, conocido como muestra de tiempo de ida y vuelta (*Sample_RTT*). Una vez obtenido un nuevo *Sample_RTT*, TCP ajusta su noción de un tiempo de ida y vuelta promedio (*Smotherd_RTT*) para la conexión. Inicialmente, la estimación del tiempo de ida y vuelta medio se hacía mediante la ecuación (1).

$$Smotherd_{RTT} = \alpha * Smotherd_{RTTold} + (1 - \alpha) * Sample_RTT \quad (1)$$

Donde $0 \leq \alpha \leq 1$.

Cuando un paquete es enviado, TCP calcula el valor del RTO como una función de la actual estimación del RTT (ecuación (2)).

$$RTO = \beta * Smotherd_{RTT} \quad (2)$$

Donde $1,3 \leq \beta \leq 2$, según [7].

Debido a que TCP usa un esquema de reconocimientos acumulativo en el cual un reconocimiento se refiere a datos recibidos, y no a la instancia de un datagrama específico portador de los datos, la medición de un *Sample_RTT* es más compleja en presencia de retransmisiones, situación en donde se presenta un fenómeno denominado *acknowledgement ambiguity* [8].

La idea conocida como el algoritmo de Karn (introducido en 1987) [9], evita el problema de reconocimientos ambiguos tan solo ajustando el RTT estimado para reconocimientos de segmentos que solo han sido transmitidos una vez. Sin embargo, cuando TCP envía un segmento después de un incremento sostenido en el retardo, este calcula un RTO usando el actual RTT promedio, el cual será muy pequeño para el nuevo retardo y forzará una retransmisión. Si TCP ignora reconocimientos de segmentos retransmitidos, nunca actualizará el RTT promedio y el ciclo continuará.

Para evitar este problema, la mayoría de implementaciones de TCP incorporan el algoritmo *exponential backoff* [9]. El algoritmo calcula un RTO inicial mediante la ecuación (2). No obstante, si el temporizador expira y causa una retransmisión, TCP incrementa el RTO (ecuación (3)).

$$RTO = \gamma * RTO_{old} \quad (3)$$

Donde $\gamma=2$, normalmente. La estrategia *exponential backoff* es utilizada hasta que llega el reconocimiento de un segmento que no ha sido retransmitido.

Los mecanismos utilizados en la estimación del RTT y el RTO anteriormente descritos, no se adaptaron al amplio rango de la variación en el retardo. En 1988 surge un nuevo mecanismo,

conocido como el algoritmo de Jacobson [10], para estimar tanto el tiempo de ida y vuelta medio como la varianza en el valor del RTT, en el cual se usa esta varianza estimada en lugar de la constante β , mediante la ecuación (4), (5), (6), (7).

$$\Delta RTT = Sample_RTT - Smothed_{RTT_{old}} \quad (4)$$

$$Smothed_RTT = Smothed_RTT_{old} + \delta * \Delta RTT \quad (5)$$

$$Deviation = Deviation_{old} + \rho * (|\Delta RTT| - Deviation_{old}) \quad (6)$$

$$RTO_{Next} = Smothed_RTT + \eta * Deviation \quad (7)$$

Donde *Deviation* es la desviación estimada, δ es una fracción entre 0 y 1 que controla qué tan rápidamente la nueva muestra afecta el RTT promedio, ρ es una fracción entre 0 y 1 que controla qué tan rápidamente la nueva muestra afecta la desviación media, y η es un factor que controla qué tanto la desviación afecta el RTO. Para hacer el cálculo eficiente, TCP elige δ y ρ como el inverso de una potencia de 2, así, la operación producto que involucre estas variables se puede realizar a nivel de bits. La especificación RFC2988 [4] sugiere los siguientes valores $\delta=1/2^3$, $\rho=1/2^2$, el valor de η es 4.

Las reglas que gobiernan el cálculo de ΔRTT , *Smothed_RTT*, *Deviation*, y *RTO*, según esta misma especificación, son:

- Hasta que se registre el muestreo de un RTT (*Sample_RTT*) debido al envío de un segmento entre un emisor y un receptor, el emisor debe mantener a RTO en 3 s [11], aunque se esté aplicando la estrategia *exponential backoff* para retransmisiones consecutivas.
- Cuando se registre la primera muestra de un RTT, el host debe actualizar:
 - Smothed_RTT*, igual al valor de *Sample_RTT*
 - Deviation*, igual a la mitad del valor de *Sample_RTT*
 - RTO, de acuerdo a la ecuación 2.12
- Cuando se registren muestras RTT's de forma consecutiva el host debe actualizar:
 - ΔRTT , de acuerdo a la ecuación (3)
 - Deviation*, de acuerdo a la ecuación (5)

- Smothed_RTT*, de acuerdo a la ecuación (4)
 - RTO, de acuerdo a la ecuación (6)
- Una vez calculado RTO, se debe verificar que su valor no sea inferior a 1 s.
 - El valor máximo de RTO puede ser fijado en al menos 60 s.

Los RFCs 1122 y 2988 que especifican los algoritmos de Karn y Jacobson, así como la estrategia *exponential backoff* para el pronóstico del RTO, fueron publicados en la década de los 80's, desde ese entonces, las tecnologías de red han evolucionado de tal forma, que hoy en día los tiempos de ida y vuelta, incluso en redes con enlaces inalámbricos son del orden de ms.

De acuerdo a esta evolución, las versiones actuales de TCP, como es el caso de FAST TCP para redes de altas prestaciones, fijan el valor mínimo del RTO en una fracción del valor especificado en los RFC1122 y RFC2988. El estudio realizado en [12] confirma la necesidad de reducir el RTO mínimo.

2.2. Diseño del sistema de inferencia difusa para el pronóstico del RTO

El modelo matemático que describe la actualización del temporizador de retransmisión mediante la utilización del sistema difuso se describe en la ecuación (8), (9), (10), (11), (12).

$$\Delta RTT = Sample_RTT - Smothed_{RTT_{old}} \quad (8)$$

$$Smothed_RTT = Smothed_RTT_{old} + \delta * \Delta RTT \quad (9)$$

$$\Delta RTO_RTT = RTO_{old} - Sample_RTT \quad (10)$$

$$\Delta RTO = f_{Fuzzy}(\Delta RTO_{RTT}, \Delta RTT) \quad (11)$$

$$RTO_{Next} = RTO_{old} + \Delta RTO \quad (12)$$

Las ecuaciones (8) y (9) son idénticas a las ecuaciones (3) y (4) utilizadas por la estrategia estadística, mientras que las ecuaciones (10), (11) y (12) reflejan el trabajo realizado en [5].

Las ecuaciones (7) y (9) actualizan las variables de entrada del sistema difuso ΔRTT y ΔRTO

RTT, mientras que la ecuación (10) actualiza la variable de salida del mismo, ΔRTO mediante inferencia difusa.

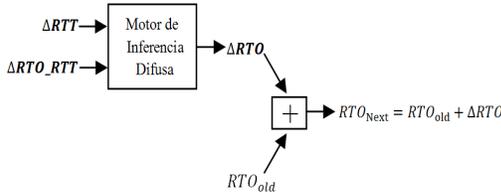


Figura 1. Esquema de actualización del RTO mediante el sistema de inferencia difusa.

A diferencia de la estrategia estadística en donde el pronóstico del RTO depende del RTT promedio y de la desviación estimada, la estrategia difusa calcula este valor en función del RTO anterior y de la variación pronosticada por el sistema de inferencia difusa, ver ecuación (11). La figura 1 muestra el esquema del sistema de inferencia difusa junto con la fórmula de actualización del RTO.

La variable de entrada ΔRTT , ecuación (7), mide el impredecible comportamiento de los tiempos de transmisión debido la complejidad del sistema de redes. Un valor cercano a cero, indica estabilidad en el tiempo de ida y vuelta respecto a su promedio ponderado, un valor muy positivo, evidencia una posible congestión (en caso de que se mantenga), mientras que un valor muy negativo indica una disminución significativa en relación al grado de congestión en la red.

La variable de entrada ΔRTO_RTT , ecuación (9), debe ser positiva, puesto que el tiempo de ida y vuelta siempre debe ser menor que el actual RTO, para que la muestra RTT sea válida. En otras palabras, no se puede muestrear un RTT de segmentos cuyo temporizador de retransmisión ha expirado. Cuando esta variable toma un valor cercano a cero, se entiende que el temporizador estuvo cerca de expirar, por lo tanto la situación de congestión en la red es crítica. Si por el contrario, el valor de ΔRTO_RTT alcanza un valor cercano a al valor del RTO, se deduce que la congestión en la red ha disminuido.

La variable de salida ΔRTO es el resultado de la inferencia difusa. En este valor incrementa o

decrementa la RTO actual, descrita por ΔRTT y ΔRTO_RTT .

Se definen siete funciones de pertenencia por cada una de las variables de entrada y de salida, cuyas etiquetas lingüísticas atienden la sugerencia del trabajo realizado en [5]. Las tablas 1, 2 y 3 muestran los conjuntos difusos que definen las funciones de pertenencia para ΔRTT , ΔRTO_RTT y ΔRTO , respectivamente.

Tabla 1. Conjuntos difusos de la variable de entrada ΔRTT .

Conjunto difuso	Significado
d_rtt_neg_b	delta RTT negativo grande
d_rtt_neg_m	delta RTT negativo medio
d_rtt_neg_s	delta RTT negativo pequeño
d_rtt_z	delta RTT cero
d_rtt_pos_s	delta RTT positivo pequeño
d_rtt_pos_m	delta RTT positivo medio
d_rtt_pos_b	delta RTT positivo grande

Tabla 2. Conjuntos difusos de la variable de entrada ΔRTO_RTT .

Conjunto difuso	Significado
d_rto_rtt_vl	delta RTO_RTT muy bajo
d_rto_rtt_l	delta RTO_RTT bajo
d_rto_rtt_ml	delta RTO_RTT medio bajo
d_rto_rtt_m	delta RTO_RTT medio
d_rto_rtt_mh	delta RTO_RTT medio alto
d_rto_rtt_h	delta RTO_RTT alto
d_rto_rtt_vh	delta RTO_RTT muy alto

Tabla 3. Conjuntos difusos de la variable de entrada ΔRTO .

Conjunto difuso	Significado
d_rto_neg_b	delta RTO negativo grande
d_rto_neg_m	delta RTO negativo medio
d_rto_neg_s	delta RTO negativo pequeño
d_rto_z	delta RTO cero
d_rto_pos_s	delta RTO positivo pequeño
d_rto_pos_m	delta RTO positivo medio
d_rto_pos_b	delta RTO positivo grande

Las figuras 2, 3 y 4 muestran la distribución gráfica de las funciones de pertenencia de las variables lingüísticas ΔRTT , ΔRTO_RTT y ΔRTO , respectivamente. El diseño propuesto en este trabajo para las funciones de pertenencia de ΔRTO_RTT , implica que esta diferencia se mantenga alrededor de los 0,04s.

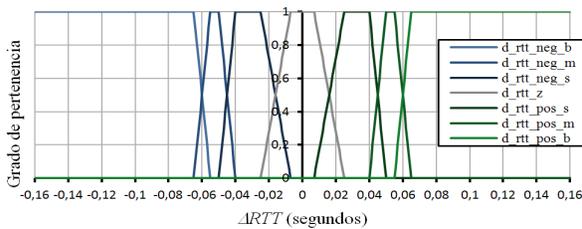


Figura 2. Distribución gráfica de las funciones de pertenencia de ΔRTT .

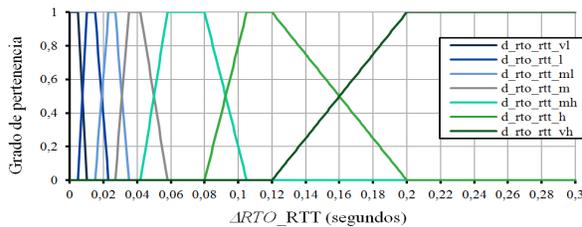


Figura 3. Distribución gráfica de las funciones de pertenencia de ΔRTO_RTT .

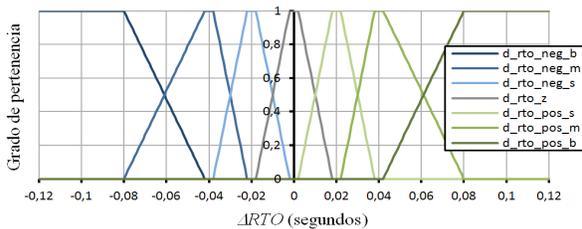


Figura 4. Distribución gráfica de las funciones de pertenencia de ΔRTO .

Definidos los conjuntos difusos y sus funciones de pertenencia para cada una de las variables lingüísticas de entrada y salida, el planteamiento del conjunto de reglas difusas que las relaciona tiene en cuenta las siguientes consideraciones:

Si ΔRTT es muy negativo, el tiempo de ida y vuelta disminuyó considerablemente, por lo tanto, el temporizador de retransmisión estuvo muy lejos de expirar, es decir ΔRTO_RTT es muy grande. Como regla general, un ΔRTT muy negativo implica un ΔRTO_RTT muy grande y viceversa. En este escenario extremo de mejoramiento significativo en las condiciones de la red, el siguiente RTO debe disminuir en un gran ΔRTO .

De otro lado, Si ΔRTT es muy positivo, el tiempo de ida y vuelta aumentó considerablemente, por lo tanto, el temporizador de retransmisión estuvo cerca de expirar, es decir ΔRTO_RTT es muy pequeño. Como regla general, un ΔRTT muy positivo implica un ΔRTO_RTT muy pequeño y viceversa. En este escenario extremo de aumento significativo de la congestión en la red, el siguiente RTO debe aumentar en un gran ΔRTO .

Así, el grado de aumento o disminución de la congestión en la red se puede aproximar de acuerdo a los valores de las variables ΔRTT y ΔRTO_RTT . La base de reglas que describe las consideraciones mencionadas y cuya estructura sigue el trabajo realizado en [5], se muestra en la tabla 4.

Tabla 4. Base de reglas del sistema difuso.

Regla número	SI	Valor lingüístico ΔRTT	O	Valor lingüístico ΔRTO_RTT	ENTONCES	Valor lingüístico ΔRTO
1	si	d_rtt_neg_b	o	d_rto_rtt_vh	entonces	d_rto_neg_b
2	si	d_rtt_neg_m	o	d_rto_rtt_h	entonces	d_rto_neg_m
3	si	d_rtt_neg_s	o	d_rto_rtt_mh	entonces	d_rto_neg_s
4	si	d_rtt_z	o	d_rto_rtt_m	entonces	d_rto_z
5	si	d_rtt_pos_s	o	d_rto_rtt_ml	entonces	d_rto_pos_s
6	si	d_rtt_pos_m	o	d_rto_rtt_l	entonces	d_rto_pos_m
7	si	d_rtt_pos_b	o	d_rto_rtt_vl	entonces	d_rto_pos_b

El principal criterio de selección de operadores para el fusificador, motor de inferencia y defusificador, es la simplicidad algorítmica. En un sistema de red actual, el intervalo de tiempo de muestreo de RTT's válidos para la actualización del RTO, es del orden de ms. En una fracción de este intervalo, el protocolo TCP de un dispositivo emisor debe realizar el pronóstico del RTO para el siguiente segmento de datos a enviar. En consecuencia, es deseable que el sistema difuso tenga una complejidad algorítmica mínima. Las características de este sistema se muestran en la figura 5.

Aunque los esquemas de defusificación "promedio de centros" y "máximo" presentan menor complejidad algorítmica que el esquema "centro de gravedad", este último se eligió teniendo en cuenta un segundo criterio de selección, la precisión. "...la ventaja del defusificador centro de gravedad se encuentra en su intuitiva plausibilidad" [13]. En este contexto, plausibilidad se entiende como la propiedad de representar (desde un punto de vista intuitivo) al conjunto difuso producto de la inferencia difusa, por medio de un solo valor

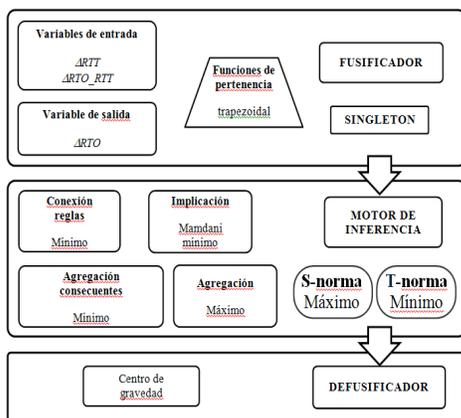


Figura 5. Características del sistema de inferencia difusa.

2.3. Validación del diseño del sistema de inferencia difusa para el pronóstico del RTO

El diseño del sistema difuso para el pronóstico del RTO fue simulado en un software de computación técnica. Para realizar la validación del sistema diseñada

do fue necesario contar con una colección de muestras RTT válidas para el pronóstico del RTO, resultado de la conexión entre los nodos 0 y 5 de la simulación del escenario de red en NS2, caracterizado a continuación. Cabe aclarar que el pronóstico del RTO, ya sea mediante la estrategia estadística o la estrategia difusa, se simuló utilizando los mismos datos mencionados.

La topología usada en la simulación 1 se muestra en la figura 6. Los nodos 0, 1 y 2 generan tráfico hacia el nodo 5 a través de conexiones entre los nodos 0-5, 1-5 y 2-5, las cuales están activas durante los 20 s de duración de la simulación. El tráfico generado por cada nodo es de tipo exponencial con diferentes intervalos de tiempo de actividad e inactividad, a una tasa de transferencia media de 70000 Kb/s. Cada conexión utiliza un agente FAST TCP para controlar la congestión. Todos los enlaces cuentan con un ancho de banda de 400 Mb/s, un límite de cola de 1×10^6 paquetes y un retardo de propagación de 2 ms, a excepción del enlace de embotellamiento entre los nodos 4 y 5 que tiene un ancho de banda de 200 Mb/s, un límite de cola de 1300 paquetes y un retardo de 50 ms.

El tráfico proveniente de los nodos 0, 1 y 2 garantiza la saturación del enlace de embotellamiento, situación que asegura un amplio rango de valores en los tiempos de ida y vuelta de paquetes en la red.

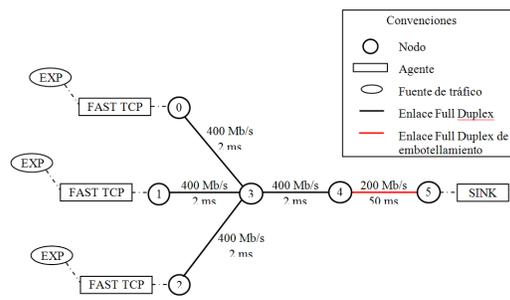


Figura 6. Topología de la simulación 1.

Como resultado de la simulación, el seguimiento de las variables de entrada y salida del sistema difuso se muestra en la figura 7. Al comparar la tendencia de los ΔRTT 's con la de los ΔRTO_RTT 's, se comprueba lo expresado en la

base de reglas difusa; un ΔRTT muy negativo implica un ΔRTO_RTT muy alto y viceversa, a excepción de lo que ocurre en el intervalo de tiempo [0,1]. Durante este periodo, la variable ΔRTO_RTT tarda en converger a los valores reales que reflejan la situación de congestión en la red, debido a que el valor inicial del RTO es 0,3 s y a que las primeras muestras RTT son muy pequeñas comparadas con este valor. El tiempo de convergencia no es menor porque ΔRTO no es lo suficientemente negativo para compensar el RTO elevado. Por ejemplo, se observa que el ΔRTO_RTT máximo en el intervalo [0,1] es de aproximadamente 0,2 s, mientras que el ΔRTO en el mismo instante de muestreo es aproximadamente -0,07 s.

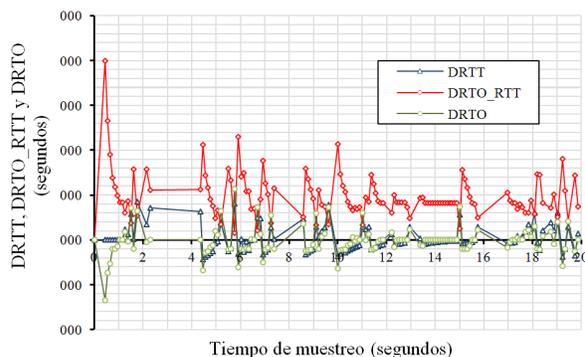


Figura 7. Seguimiento de las variables de entrada y salida del sistema difuso.

Aunque en [11] se sugiere un RTO inicial de 3 s, los entornos modernos de redes manejan unos tiempos de latencia muy inferiores a este valor. Se eligió un RTO inicial a los 0,3 s, teniendo en cuenta que respecto a los valores iniciales del RTT y RTO, el mismo documento enuncia lo siguiente: “La experiencia ha mostrado que estos valores iniciales son razonables, y que en cualquier caso los algoritmos de Karn y Jacobson hacen que el comportamiento de TCP sea razonablemente insensible a la elección de los mismos”.

Al margen de la estrategia utilizada para el pronóstico del RTO, el tiempo de convergencia, no solo de ΔRTO_RTT sino también del RTO, disminuye de forma considerable al reducir el RTO inicial de 3 a 0,3 s.

Continuando con el análisis de la figura 7, se observa que ΔRTO es muy negativo cuando ΔRTT también lo es o cuando ΔRTO_RTT es muy alto. En el caso contrario, ΔRTO es muy positivo cuando ΔRTT también lo es o cuando ΔRTO_RTT es muy bajo. Nuevamente se valida lo expresado en la base de reglas difusa.

La figura 8 presenta las muestras RTT recolectadas durante el tiempo de simulación, el valor suavizado de las mismas y el RTO pronosticado a través de la estrategia difusa y estadística. Al comparar estos últimos, se aprecia que el RTO difuso hace un seguimiento más ajustado a las variaciones del RTT que el RTO estadístico.

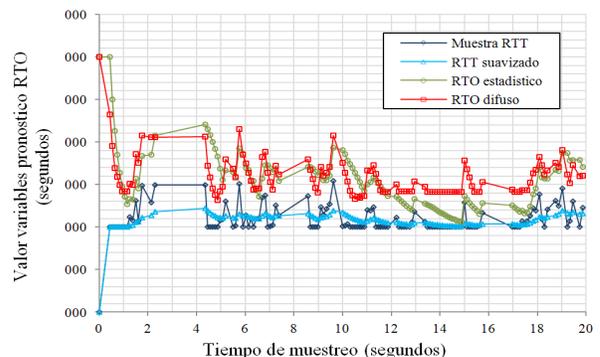


Figura 8. Comportamiento del RTO estadístico y del RTO difuso.

Cabe aclarar que pronosticar el RTO por medio de dos estrategias, utilizando las mismas muestras RTT, se hace solo para efectos de validación del diseño propuesto, debido a que RTO's diferentes derivan en comportamientos de red diferentes. Por ejemplo, es posible que el RTO producto de una estrategia ocasione una expiración, mientras que el otro RTO permita que se registre la muestra RTT, a partir de esta diferencia las muestras RTT comienzan a diferir.

3. EXPERIMENTACIÓN

En esta etapa se implementó el sistema difuso diseñado en la herramienta de simulación NS2. Además, se realizó la caracterización de un escenario de red simulado en la misma herramienta, sobre el cual se evaluó el comportamiento de la estrategia difusa para el pronóstico del RTO, en comparación a la estrategia

estadística tradicional.

La implementación del sistema difuso diseñado permite que la simulación de cualquier escenario red se pueda ejecutar de dos formas. La primera, con el uso de la estrategia estadística para pronosticar el RTO, y la segunda, con el uso de la estrategia difusa para el pronóstico del mismo.

El comportamiento de las dos estrategias se evaluó mediante la simulación del escenario de red, cuya topología se muestra en la figura 9. Los nodos 0, 1 y 2 generan tráfico a través de las conexiones especificadas en la tabla 5. El tráfico generado por cada nodo es de tipo exponencial con diferentes intervalos de tiempo de actividad e inactividad. Cada conexión utiliza un agente FAST TCP para controlar la congestión. Todos los enlaces cuentan con un ancho de banda de 400Mbps, un límite de cola de 1×10^6

paquetes y un retardo de propagación de 5ms, a excepción de los enlaces de embotellamiento entre los nodos 6-8 y 7-9 que tienen un ancho de banda de 175Mbps y 225Mbps respectivamente, un límite de cola de 1200 paquetes y un retardo de 10ms.

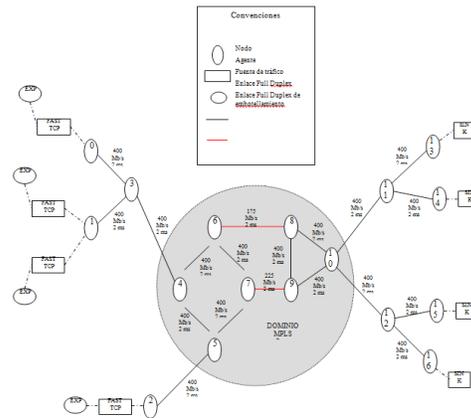


Figura 9. Topología de la simulación 2.

Tabla 5. Características de las conexiones en la simulación 2.

Id conexión	Nodo fuente	Nodo destino	Tiempo inicio (s)	Tiempo	Tráfico exponencial		
					Tasa de envío (Kb/s)	Intervalo de actividad (ms)	Intervalo de inactividad (ms)
0	0	13	2	40	42000	500	600
1	1	14	4	38	38000	650	300
2	1	16	6	36	44000	750	250
3	2	15	8	34	40000	800	450

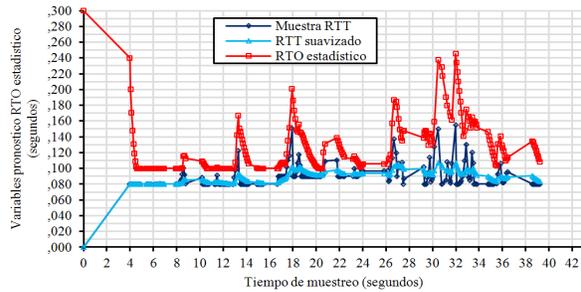
Los nodos 0, 1, 2, 3, 11, 12, 13, 14, 15, y 16 son convencionales en NS2, mientras que los nodos 4, 5, 6, 7, 8, 9, y 10 son enrutadores de conmutación de etiquetas que forman un dominio MPLS.

El tráfico proveniente de los nodos 0, 1 y 2 garantiza la saturación de los enlaces de embotellamiento. Además, para complicar aún más la congestión en la red, en el instante 16 s se simula la caída del enlace 6-8, el cual se restablece en el instante 24 s, al mismo tiempo se cae el enlace 7-9 hasta su restablecimiento a los 40 s del tiempo total de simulación (43 s). Esta situación asegura que todo el tráfico circule por un solo enlace de embotellamiento desde

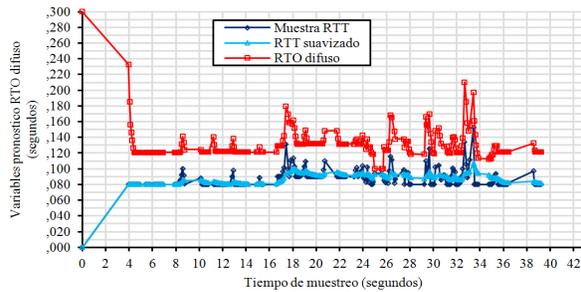
los 16 hasta los 40 s del periodo de simulación. Desde el inicio de la simulación se crean las rutas explícitas 5-7-9-10 con el identificador de ruta 1000 y 10-8-6-4-1000 con el identificador de ruta 1005, dentro del entorno MPLS. A través de esta última ruta se re direcciona el tráfico cuando se cae el enlace 6-8. De otro lado, se usa un protocolo de enrutamiento del tipo vector distancia (rtProtoDV de NS2) para redireccionar el tráfico cuando de forma simultánea se cae el enlace 7-9 y se restablece el enlace 6-8. La simulación en NS2 del escenario de red descrito, se ejecutó de dos formas. La primera, con el pronóstico del RTO mediante la estrategia estadística. La segunda, con el pronóstico del RTO por medio de la estrategia difusa diseñada.

4. RESULTADOS

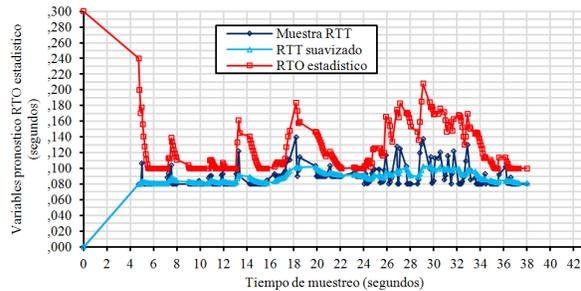
En la figura 10 se pueden apreciar las gráficas del seguimiento de las variables Sample_RTT, Smothed_RTT y RTO, para cada conexión, durante el periodo de simulación.



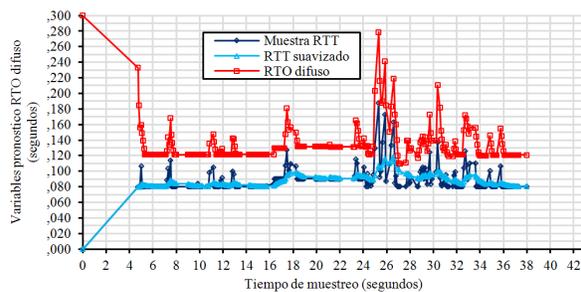
(a) Con el uso de la estrategia estadística, conexión 0.



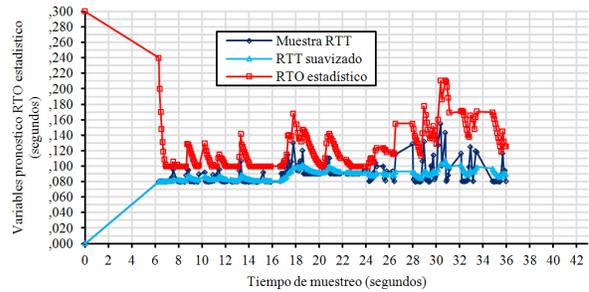
(b) Con el uso de la estrategia difusa, conexión 0.



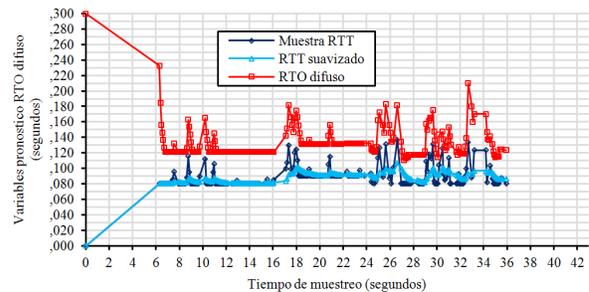
(c) Con el uso de la estrategia estadística, conexión 1.



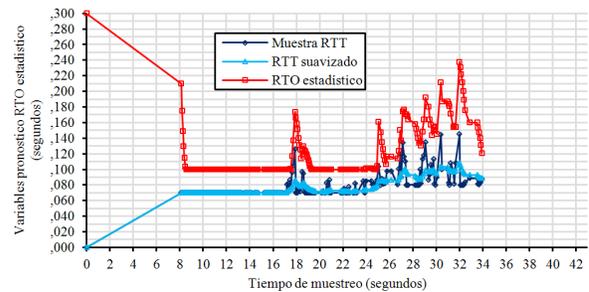
(d) Con el uso de la estrategia difusa, conexión 1.



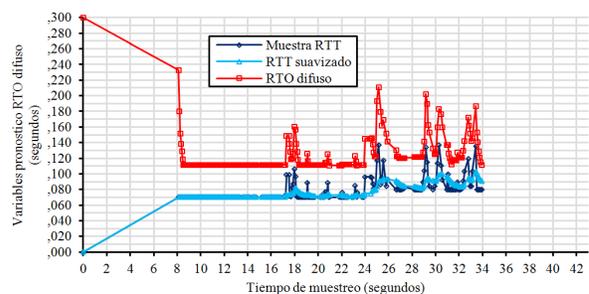
(e) Con el uso de la estrategia estadística, conexión 2.



(f) Con el uso de la estrategia difusa, conexión 2.



(g) Con el uso de la estrategia estadística, conexión 3.



(h) Con el uso de la estrategia difusa, conexión 3.

Figura 10. Comportamiento del RTO en la simulación 2.

Al comparar las figuras (a) y (b), (c) y (d), (e) y (f), (g) y (h), de la figura 10, se observa que el RTO pronosticado mediante la estrategia difusa hace un seguimiento más ajustado a las varia-

ciones del RTT. De nuevo, la expresión “seguimiento más ajustado”, se refiere a que la variación de la diferencia entre el RTO pronosticado para el siguiente paquete de datos a enviar y el RTT (Δ_RTO_RTT'), es menos drástica durante el periodo de simulación, en comparación a la variación que arroja el uso de la estrategia estadística. Para confirmar esta observación se hace un análisis descriptivo de Δ_RTO_RTT' , con base en los resultados de cada conexión. La variable estadística idónea para este propósito es la desviación estándar relativa de Δ_RTO_RTT' , que representa el porcentaje de variación promedio respecto la media de la variable en cuestión. A mayor desviación estándar relativa, mayor variación de Δ_RTO_RTT' y viceversa.

Tabla 6. Desviación estándar relativa de D_RTO_RTT' .

Id Conexión	Desviación estándar de D_RTO_RTT' (uso de la estrategia estadística)	Desviación estándar de D_RTO_RTT' (uso de la estrategia difusa)
0	73,09%, media de 0,0366 s	18,69% media de 0,0422 s
1	64,74%, media de 0,0326 s	23,53% media de 0,0444 s
2	64,84%, media de 0,0326 s	18,03% media de 0,0426 s
3	57,05%, media de 0,0393 s	18,91% media de 0,0436 s

De los resultados presentados en la tabla 6, se observa que la desviación estándar relativa de Δ_RTO_RTT' siempre es menor al utilizar la estrategia difusa en cada una de las conexiones. Por lo tanto, se confirma que el RTO pronosticado mediante esta estrategia hace un seguimiento más ajustado a las variaciones del RTT.

De otro lado, se analizó el comportamiento de la tasa de transferencia del TCP emisor en cada conexión. Esta tasa se calcula como el cociente entre la ventana de congestión y el RTT promedio de FAST TCP, atendiendo lo especificado en los script de simulación que hacen parte del código fuente del módulo FAST TCP para NS2 [14].

Las figuras 11, 12, 13 y 14 presentan las gráficas de la tasa de transferencia de cada co-

nexión, cuando se usa la estrategia estadística y difusa, respectivamente. Igualmente, en la tabla 7 se muestra la tasa de transferencia promedio de cada una. Finalmente, en esta misma tabla se presenta el promedio ponderado total de la tasa de transferencia para cada una de las estrategias, ya que la diferencia en la duración de cada una de las conexiones obliga al uso de este estimador.

Los resultados de la tabla 7 muestran que el promedio ponderado total de la tasa de transferencia es levemente superior al utilizar la estrategia de pronóstico difusa.

Con el propósito de aproximar la cantidad de paquetes retransmitidos en el contexto de cada conexión se tuvieron en cuenta dos indicadores: La cantidad de paquetes con más de un reconocimiento y la cantidad de paquetes que fueron reconocidos por una única vez.

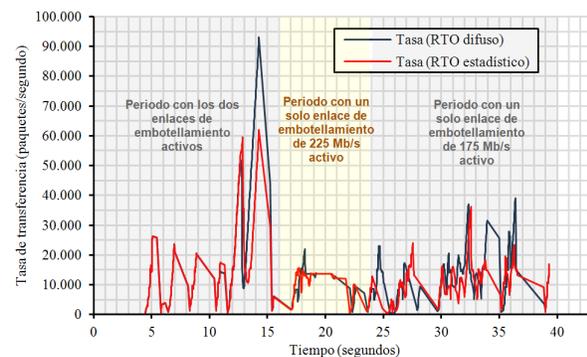


Figura 11. Tasa de transferencia del emisor TCP de la conexión 0.

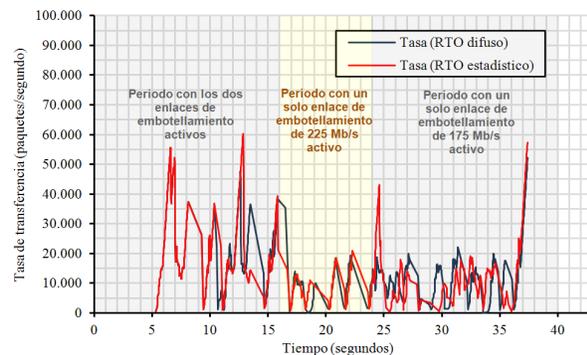


Figura 12. Tasa de transferencia del emisor TCP de la conexión 1.

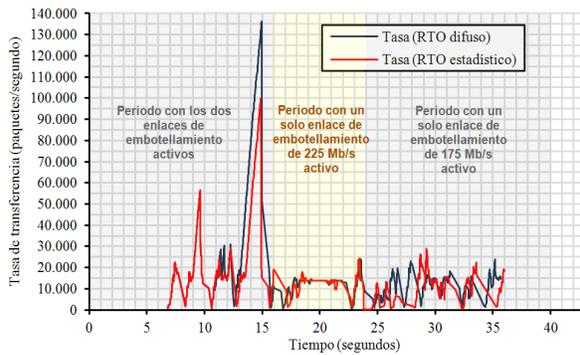


Figura 13. Tasa de transferencia del emisor TCP de la conexión 2.

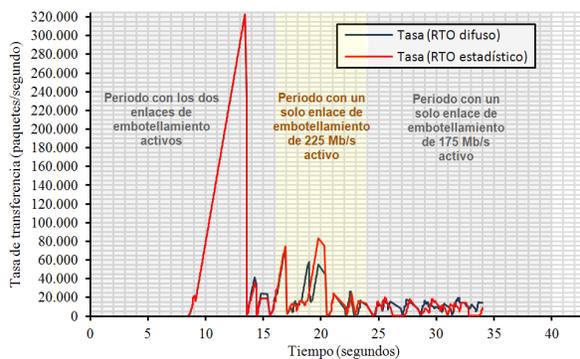


Figura 14. Tasa de transferencia del emisor TCP de la conexión 3.

Tabla 7. Tasa de transferencia promedio de la fuente TCP para cada conexión en la simulación 2.

Id. Conexión	Duración conexión (s)	Tasa de transferencia media (paquetes/s)	
		Estrategia estadística	Estrategia difusa
0	38	14.394,59	16.701,34
1	34	16.540,63	15.718,61
2	30	17.214,57	19.377,86
3	26	52.549,78	49.657,62
Promedio ponderado		23.375,84	23.761,85

El primero de ellos permite tener una idea aproximada de la cantidad de paquetes retransmitidos por el emisor TCP y por lo tanto cuanto menor sea esta cantidad, menor es el número de paquetes que efectivamente fueron retransmitidos.

El segundo indicador también permite tener una idea aproximada de la cantidad de paquetes retransmitidos, ya que cuanto mayor sea este indicador, menor es el número de paquetes retransmitidos, y viceversa.

La tabla 8 muestra que mediante el uso de la estrategia difusa, en comparación con el uso de la estrategia estadística en la simulación 2, se registraron más paquetes reconocidos una única vez (2,56% más que con el uso de la estrategia estadística) y menos paquetes con más de un reconocimiento (54,11% menos que con el uso de la estrategia estadística). Lo anterior se puede interpretar como que en la simulación con la estrategia difusa para el pronóstico del RTO se retransmitieron menos paquetes, y por lo tanto, se logró una tasa de transferencia más efectiva, aunque un 1,65% superior a la tasa de transferencia registrada con el uso de la estrategia estadística.

Tabla 8. Cantidad de paquetes reconocidos por una única vez y cantidad de paquetes con más de un reconocimiento, en el contexto de cada conexión.

Id	Cantidad de paquetes reconocidos por una única vez		Cantidad de paquetes con más de un reconocimiento	
	Estrategia estadística	Estrategia difusa	Estrategia estadística	Estrategia difusa
0	97.075	99.852	314	136
1	98.420	102.424	399	105
2	118.655	120.455	252	193
3	96.497	98.414	153	79
Totales	410.647	421.145	1.118	513

En FAST TCP, la estimación del RTT presenta una alta oscilación cuando flujos con diferentes RTT's comparten un mismo ancho de banda [15]. Estas oscilaciones son el resultado del proceso cíclico presentado en la figura 15.

La variación de la ventana de congestión en función del aumento o la disminución de los RTT's afecta directamente la tasa de transferencia [16].

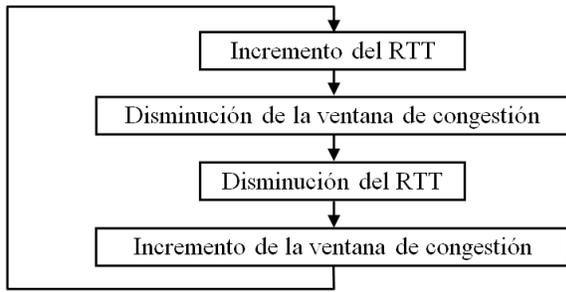


Figura 15. Ciclo de variación del RTT en función de la ventana de congestión y viceversa, en FAST TCP.

Fuente: Adaptado de [15].

La variación de la ventana de congestión en función del aumento o la disminución de los RTT's afecta directamente la tasa de transferencia [16].

La tabla 9 presenta el RTT promedio en cada conexión. Estos datos muestran que el promedio ponderado del RTT es inferior al utilizar la estrategia difusa para el pronóstico del RTO, en comparación al mismo promedio obtenido con el uso de la estrategia estadística. Así, esta disminución global del RTT implica un incremento global en la ventana de congestión, lo que explica el incremento en la tasa de transferencia.

Tabla 9. RTT promedio registrado por la fuente TCP de cada conexión, en la simulación 2.

Id. Conexión	Duración conexión (s)	RTT promedio (s)	
		Estrategia estadística	Estrategia difusa
0	38	0,0882	0,0873
1	34	0,0870	0,0881
2	30	0,0882	0,0875
3	26	0,0773	0,0766
Promedio ponderado		0,0857	0,0854

5. CONCLUSIONES

Al pronosticar el RTO mediante la estrategia difusa, en cada una de las fuentes TCP de la simulación 2 se logró hacer un seguimiento más ajustado a las variaciones del RTT.

Igualmente se logró reducir el número de paquetes con más de un reconocimiento y au-

mentar el número de paquetes que no tienen asociados reconocimientos duplicados, lo que se traduce en una disminución del número de paquetes retransmitidos.

De esta manera se logró una tasa de transferencia más efectiva en cuanto a la reducción de la retransmisión de paquetes.

Estos resultados demuestran que esta nueva forma de pronóstico del RTO permite a FAST TCP hacer un mejor seguimiento a las condiciones reales de congestión en la red, y por lo tanto, mejorar las prestaciones del protocolo TCP. Aunque esta nueva estrategia presentó un drástico aumento en el tiempo de ejecución comparada con la estrategia estadística, el impacto sobre el intervalo de muestreo de RTT's válidos para actualizar el RTO no es significativo, y por lo tanto, no afecta el comportamiento del TCP simulado en NS2.

6. TRABAJO FUTURO

La estrategia difusa aquí diseñada, al igual que el trabajo realizado en [6] es aplicable a redes cuyos los tiempos de ida y vuelta son relativamente bajos, del orden de ms.

La estrategia presentada en [5] se aplica a redes con un orden de magnitud de RTT's muy superior. Para lograr que una estrategia de pronóstico del RTO mediante inteligencia computacional, aplique a cualquier tipo de red, se plantean dos alternativas de trabajos futuros. La primera, podría considerar que los universos en discurso de las variables de entrada y de salida del sistema difuso, se adapten a los diferentes órdenes de magnitud de los RTT's, y por lo tanto a cualquier tipo de red, lo cual se podría lograr mediante la utilización de un sistema neuro-difuso. La segunda alternativa es plantear una nueva estrategia que utilice un diseño coevolutivo o evolutivo.

Otra opción de trabajo futuro, consiste en realizar este mismo trabajo utilizando lógica difusa tipo dos.

De otro lado, se podría evaluar la utilización de

técnicas de inteligencia computacional en diferentes algoritmos de TCP, como inicio lento, prevención de la congestión, control de ventana o el propio FAST TCP.

Finalmente, en un trabajo futuro se propondría

la modificación de los RFCs 1122 y 2988, en relación al replanteamiento del orden de magnitud de las variables RTO inicial, RTO máximo y RTO mínimo, las cuales son acotadas en estos documentos.

Referencias Bibliográficas

- [1] C. Jin, et al; FAST TCP: from theory to experiments, *IEEE Network*, vol. 19, No. 1, pp. 4-11, Jan/Feb. 2005.
- [2] C. Jin, D. Wei, S. Low; FAST TCP: motivation, architecture, algorithms, performance, *Proceedings of IEEE INFOCOM*, pp. 2490–2501, March 2004.
- [3] V. Jacobson, B. Braden, D. Borman; TCP extensions for high performance, *Network Working Group, RFC 1323*, May 1992.
- [4] V. Paxson, M. Allman; Computing TCP's retransmission Timer, *Network Working Group, RFC 2988*, November 2000.
- [5] Z. Zhang, Z. Li, S. Suthaharan; Fuzzy logic strategy of prognosticating TCP's timeout and retransmission, *Studies in Computational Intelligence*, chapter 22, vol. 2, pp. 309–319, January 2005.
- [6] K. Ho; Method for network transmission, *U.S. Patent US7911949B2*, March 22, 2011.
- [7] J. Postel; Transmission control protocol, *DARPA Internet Program Specification, RFC 793*, September 1981.
- [8] A. Parra, E. Piedrahita, O. Salcedo; Aplicaciones del modelo ON/OFF al tráfico agregado en las redes de comunicaciones, *Revista Tecnura*, vol. 15, no. 30, pp. 129-147, Diciembre 2011.
- [9] P. Karn, C. Partridge; Improving round-trip time estimates in reliable transport protocols, *Proceeding of SIGCOMM'87*, vol. 17, No. 5, pp.1–9, October 1987.
- [10] V. Jacobson; Interpacket arrival variance and mean. Letter to the TCP-IP mailing list. [En línea], consultado en Junio 10 del 2012, disponible en: <http://ee.lbl.gov/tcp.html#variances>.
- [11] R. Braden; Requirements for internet hosts – communication layers, *Network Working Group, RFC 1122*, October 1989.
- [12] D. Andersen, G. Ganger, G. Gibson, E. Krevat, A. Phanishayee, S. Seshan, V. Vasudevan; Measurement and analysis of TCP throughput collapse in cluster-based storage systems, *6th USENIX Conference on File and Storage Technologies (FAST '08)*, September 2007, disponible en: http://www.pdl.cmu.edu/PDL-FTP/Storage/FASTincast_abs.shtml.
- [13] L. Wang; *A course in fuzzy systems and control*, New Jersey: Prentice Hall, 1997, ISBN 0-13-540882-2.
- [14] T. Cui, L. Andrew; FAST TCP simulator module for ns-2, version 1.1, URL: <http://www.cubinlab.ee.unimelb.edu.au/ns2fasttcp/>.
- [15] S. Belhaj, M. Tagina; VFAST TCP: An improvement of FAST TCP, *Proceedings of the Tenth International Conference on Computer and Modeling Simulation (Uksim'08)*, pp. 88–93, Cambridge, England, April 2008.
- [16] J. Wang, D. Wei, S.Low; Modelling and stability of FAST TCP, *Proceedings of IEEE INFOCOM*, vol. 2, pp. 938–948, March 2005.