Consideraciones sobre el comportamiento del protocolo TCP en sus variantes Vegas, Reno, Cubic y Westwood ante errores en ráfaga en una topología híbrida

Diego R. Rodríguez Herlein, Carlos A. Talay, Claudia N. González y Franco A. Trinidad
Licenciatura en informática, UNPA-UARG
Río Gallegos, Argentina
Campus universitario - Oficina B 18
diegorh@gmail.com, carlostalay@yahoo.com.ar, claudiagonzález@yahoo.com, talejandro.franco@gmail.com

Luis A. Marrone
L.I.N.T.I. – Universidad Nacional de La Plata
La Plata, Argentina
Calle 50 y 120 – 2do. Piso – Edificio Bosque Oeste
lmarrone@linti.unlp.edu.ar

Abstract. Las comunicaciones que utilizan el protocolo TCP han sido y son ampliamente estudiadas debido a la heterogeneidad de los medios y condiciones en las que tienen que desempeñarse. Esto ha dado lugar a diferentes variantes de este protocolo, que tienen que afrontar distintos tipos de escenarios con sus propias características, tales como errores en ráfaga. Estos problemas, entre otros, son característicos de los enlaces inalámbricos y tienen como consecuencia una merma en el rendimiento de las comunicaciones. Ante este inconveniente, es interesante ver cómo el protocolo recupera el funcionamiento normal después de sufrir este evento. Por lo tanto, este documento tiene como objetivo realizar un análisis de cómo las transmisiones de datos son afectadas al sufrir errores en ráfaga, cuando se utiliza alguna de las 4 variantes estudiadas, mediante simulaciones con el ns-2

Keywords: TCP, error en ráfaga, Performance, ns-2

1 Introducción

El protocolo de comunicaciones TCP (Transmission Control Protocol) [1] ha sido ampliamente utilizado en las redes de datos desde su desarrollo hasta la actualidad. Introduciéndole modificaciones se trató de adaptarlo a distintas condiciones de trabajo, permitiéndole así acompañar a las innovaciones tecnológicas que se han desarrollado en el área de las telecomunicaciones y que utilizan tanto medios

cableados como inalámbricos. Este protocolo se caracteriza por ser confiable, realizar un control de flujo y poseer un mecanismo de control de congestión de datos. También regula la secuencia de entrega de los paquetes, mediante la verificación de la recepción ordenada de dichos paquetes numerados en origen, corroborado el orden de llegada en el receptor. Este protocolo ofrece un servicio orientado a conexión, que basa su entrega confiable en un procedimiento conocido como ARQ (Automatic Repeat reQuest), en sus distintas variantes [2], que garantiza la integridad de los datos. Mediante el procedimiento ARQ y con la utilización de ACKs (acknowledgments) positivos, se logra que, con menos de un ACK por paquete de datos, se pueda confirmar la recepción de información de todo un conjunto de paquetes. Esta técnica se conoce como delayed-ACK [3] y permite lograr un importante aumento de eficiencia en el funcionamiento de la red.

A nivel de control de congestión, el protocolo TCP realiza una regulación del tráfico sobre el flujo de datos. Para lograr esto, el protocolo verifica si existe una pérdida de segmentos o si se produce una recepción de ACKs duplicados. Al analizar el resultado de esta verificación, el protocolo determina la ocurrencia de pérdida de paquetes y por consiguiente si existe o no congestión en la red [4] [5]. Mediante el perfeccionamiento de este método, se han desarrollado dos variantes para atender problemas de control de congestión. Una de ellas se basa en un control reactivo del problema, suponiendo que existe congestión en los enlaces ante la pérdida de segmentos [6] [7]. Por su parte, la otra variante trata de realizar un control proactivo de la congestión, en donde lo que se busca es desarrollar una estrategia que permita evitar que el tráfico llegue a una situación de congestión [8] [9].

En la actualidad, las tecnologías de transmisión apuntan a la calidad del flujo de datos, lo que posibilita tener una baja tasa de errores. En este escenario, las técnicas de control de congestión de la red se han basado fundamentalmente en la detección de paquetes perdidos. Por ello, bajo estas condiciones, los protocolos reactivos entienden que hay congestión en la red y accionan sus algoritmos de control de congestión. Sin embargo, existen situaciones en las cuales la pérdida de paquetes puede tener otro origen que no es la congestión y, por tanto, no deberían disparar sus mecanismos de control de congestión. Pensando en esta situación, se propone mediante la configuración de un modelo simple y utilizando la herramienta de simulación ns-2 [10] [11], analizar la respuesta de 4 protocolos ante la introducción de errores en ráfaga en una transmisión de datos.

2 Diseño del modelo de prueba

Para modelar y generar los datos del presente trabajo, se utilizó el ns-2 (Network Simulator 2), simulador de redes de eventos discretos en su versión 2.35 (released Nov. 4 2011). Con este simulador se implementó la siguiente topología de 3 nodos:

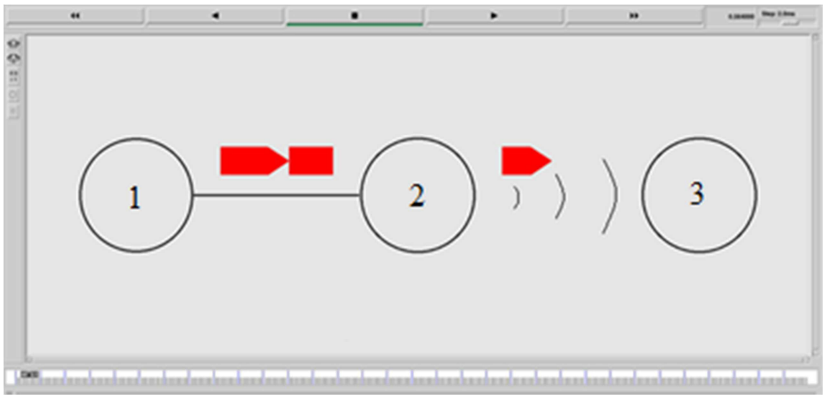

Grafico 1

Como se observa en el grafico 1, los nodos 1 y 2 están vinculados por un enlace cableado que se configuro como dúplex, con un ancho de banda 2 Mb/s, retardo de propagación 2 ms. y política de servicio de las colas DropTail. El enlace entre los nodos 2 y 3 es inalámbrico y se configuró como modo de propagación TwoRayGround, la capa física WirelessPhy, MAC 802.11, la antena OmniAntenna y el nodo inalámbrico sin movilidad.

La selección de este modelo es una aproximación a un escenario Wireless con un nodo fijo (nodo 1), una estación base (nodo 2) y un nodo móvil (nodo 3), con la simplificación práctica que el enlace inalámbrico no presenta desconexiones y solo tiene errores en forma de ráfagas.

En principio el modelo constaba de más nodos inalámbricos. Sin embargo, debido a que el presente trabajo se basó en el estudio de los errores en forma de ráfagas en medios híbridos no se consideró necesario, por el momento, aumentar la complejidad incorporando una mayor cantidad de nodos móviles.

El nodo 1 se configuro como emisor y en él un agente TCP, por otro lado el nodo 3 se configuró como receptor. A este enlace se asoció un tráfico FTP (file transfer protocol) como único tráfico.

Se realizaron simulaciones independientes sobre las implementaciones de las distintas variantes de TCP. Para cada una de ellas, se generaron simulaciones introduciendo distintas longitudes de los errores en ráfagas, con tamaños que fueron desde una prueba sin errores (0), pasando a pruebas con ráfagas de error de 5, 10, 15 y 20 paquetes. Los Agentes TCP que se utilizaron fueron Vegas, Cubic, Reno y Westwood, tal como están designados e implementados en esta versión de ns-2 (ver. 2.35), es decir sin modificación alguna. En el caso de TCP Vegas los valores de ALPHA=1 y BETA=3 son los que se utilizan por defecto.

La trasmisión de datos comienza a los 5 segundos de iniciada las simulación y están condicionadas a la transmisión de 3.000 paquetes de TCP, de 1.000 bytes c/u, independientemente de la longitud de la ráfaga de error. Estas ráfagas siempre comenzaron después de trasmitidos los primeros 999 paquetes, el ensayo concluyó al terminarse de transmitir los 3.000 paquetes TCP asociado al tráfico de FTP.

Se escribió el script TCL de manera que, al finalizar cada simulación, se obtuviera, además del archivo de traza, un archivo con el tamaño de la ventana de congestión del agente TCP en función del tiempo y el valor del número de secuencia en función del tiempo. A su vez, se utilizaron script AWK sobre el archivo de traza para obtener valores de throughput, retardo de extremo a extremo y ventana de congestión.
Para procesar estos datos, se utilizó una planilla de cálculo a través de la cual se generaron los gráficos que aquí se presentan.

3 Resultados obtenidos

Para verificar el comportamiento de los protocolos ante la ocurrencia de ráfagas, se diseñaron pruebas sobre la topología definida en el punto anterior. Simulando una transferencia de datos, y sobre el envío de los 3.000 paquetes que conforman los datos, se genera un error en ráfaga, correspondiente a los casos de 5, 10, 15, 20 y 25 paquetes perdidos en ráfaga, que comienzan a perderse a partir del paquete 999. Bajo estas condiciones, tenemos los siguientes ensayos:

3.1 Pruebas para comportamiento del modelo sin existencias de errores

Se expone a continuación el comportamiento de los parámetros retardo extremo a extremo, throughput, número de secuencia y ventana de congestión para el caso de transmisión de datos sin existencia de errores.

3.1.1 Retardo extremo a extremo

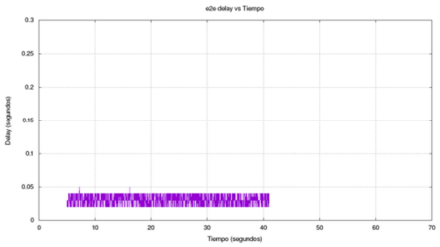

Fig. 1. VEGAS

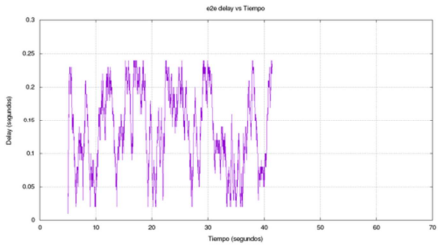

Fig. 2. CUBIC

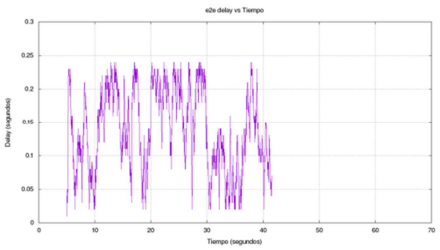

Fig. 3. RENO

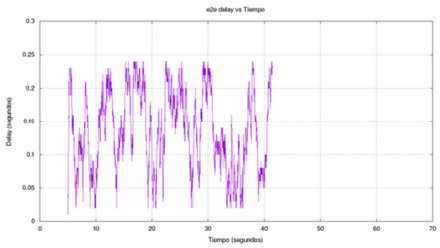

Fig. 4. WESTWOOD

3.1.2 Throughput

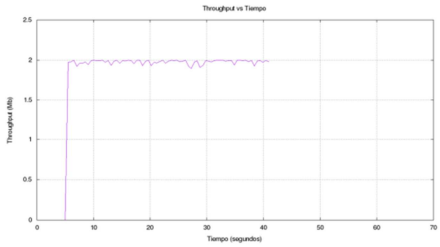

Fig. 5. VEGAS

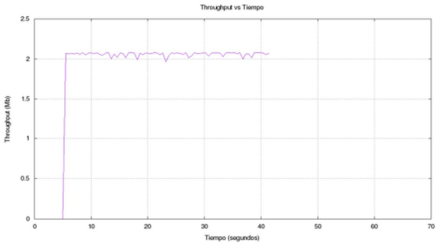

Fig. 6. CUBIC

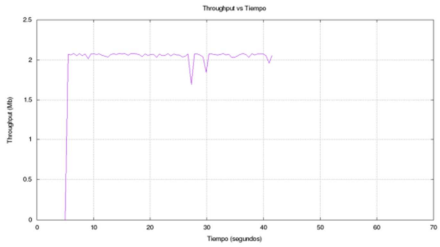

Fig. 7. RENO

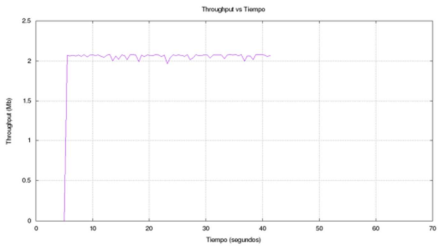

Fig. 8. WESTWOOD

3.1.3 Número de secuencia

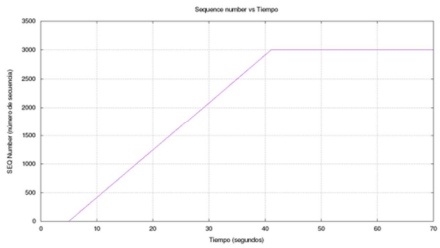

Fig. 9. VEGAS

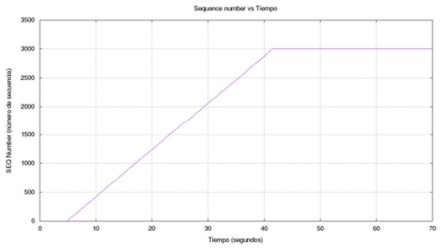

Fig. 10. CUBIC

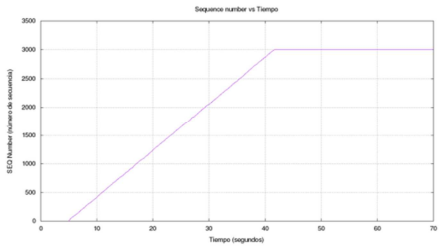

Fig. 11. RENO

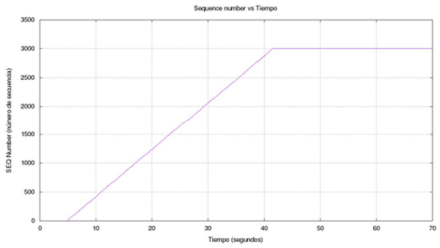

Fig. 12. WESTWOOD

3.1.4 Ventana de congestión (CW)

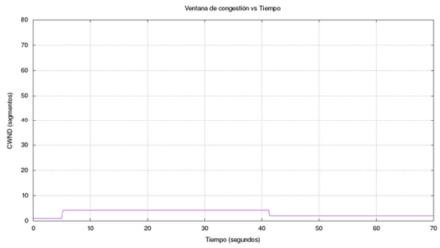

Fig. 13. VEGAS

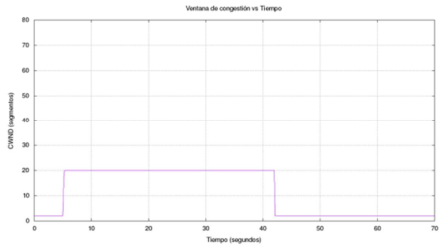

Fig. 14. CUBIC

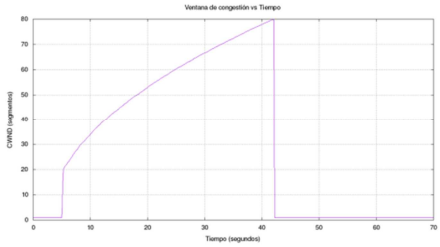

Fig. 15. RENO

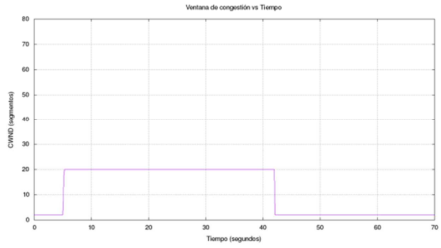

Fig. 16. WESTWOOD

3.2 Pruebas para comportamiento con una ráfaga de 15 paquetes

Se expone a continuación el comportamiento de los parámetros retardo extremo a extremo, throughput, número de secuencia y ventana de congestión introduciendo un error en ráfaga de 15 paquetes perdidos, a los efectos de comparar en comportamiento de los cuatro protocolos analizados.

3.2.1 Retardo extremo a extremo

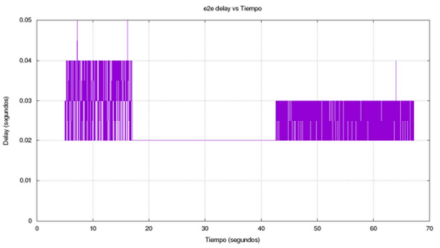

Fig. 17. VEGAS

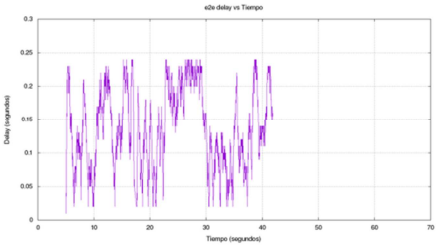

Fig. 18. CUBIC

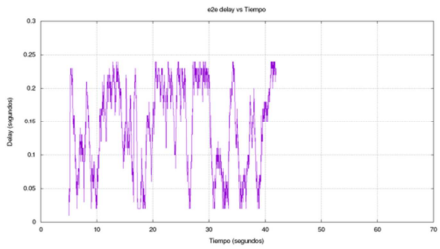

Fig. 19. RENO

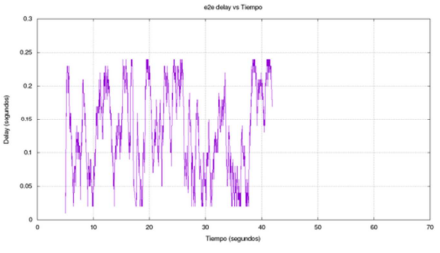

Fig. 20. WESTWOOD

3.2.2 Througput

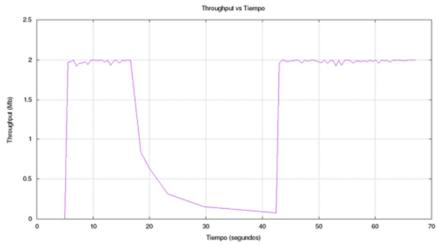

Fig. 21. VEGAS

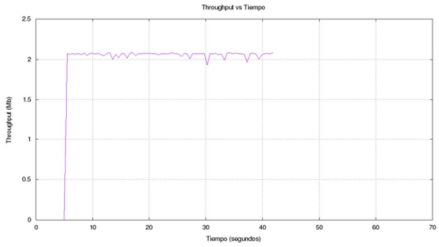

Fig. 22. CUBIC

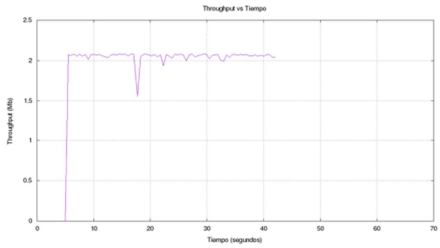

Fig. 23. RENO

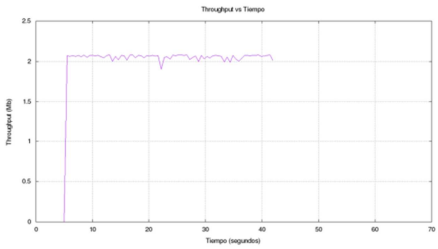

Fig. 24. WESTWOOD

3.2.3 Números de secuencia

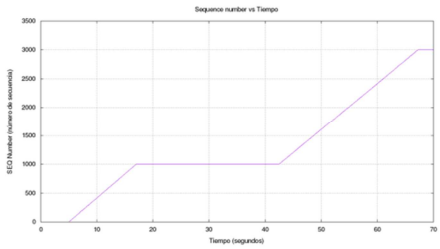

Fig. 25. VEGAS

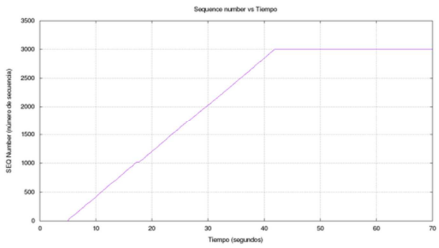

Fig. 26. CUBIC

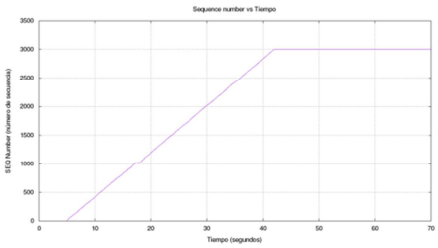

Fig. 27. RENO

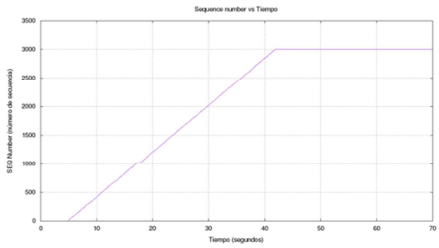

Fig. 28. WESTWOOD

3.2.4 Ventana de congestión (CW)

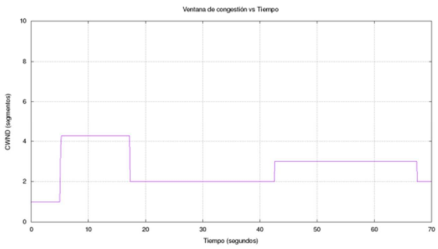

Fig. 29. VEGAS

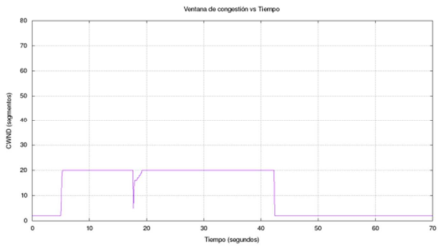

Fig. 30. CUBIC

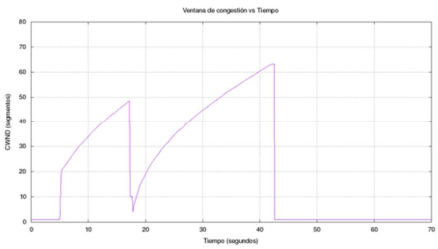

Fig. 31. RENO

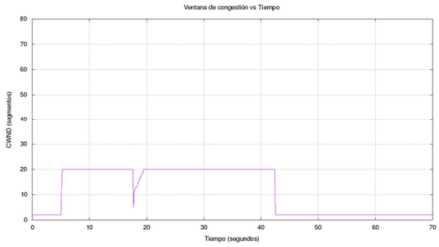

Fig. 32. WESTWOOD

3.3 Throughput de Vegas con distintos tamaños de ráfaga

Como se observa en la prueba de error en ráfaga de 15 paquetes, el agente TCP Vegas revela una sensibilidad mayor que los otros 3 protocolos, mostrando una demora en el reinicio de la trasmisión. Es por ello que, a los efectos de apreciar cómo las ráfagas afectan de manera progresiva su comportamiento, recreamos pruebas de su parámetro throughput, para los valores de error en ráfaga correspondiente a 5, 10, 15, 20 y 25 paquetes y los mostramos a continuación:

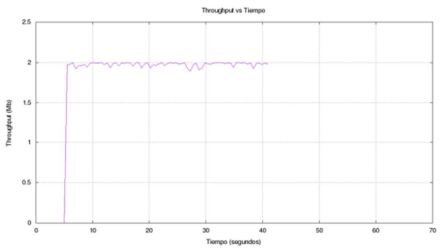

Fig. 33. Ráfaga de 0 paquetes

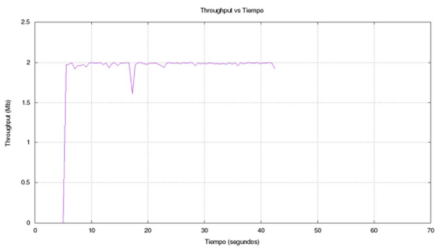

Fig. 34. Ráfaga de 5 paquetes

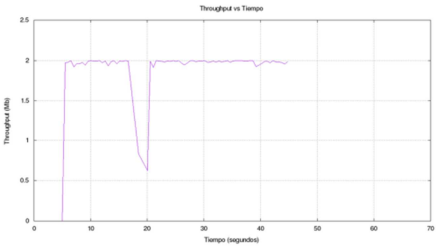

Fig. 35. Ráfaga de 10 paquetes

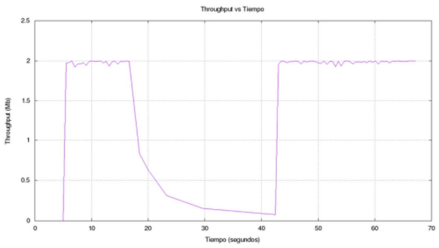

Fig. 36. Ráfaga de 15 paquetes

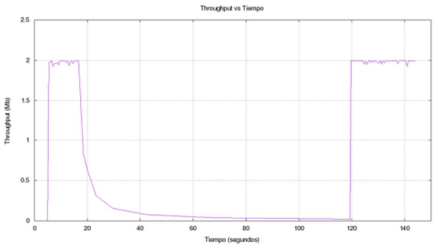

Fig. 37. Ráfaga de 20 paquetes

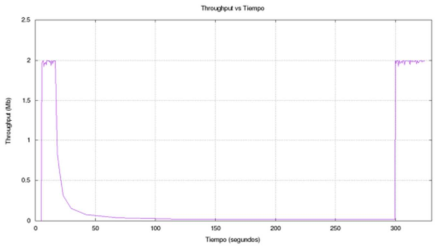

Fig. 38. Ráfaga de 25 paquetes

4 Conclusiones

Como comentábamos en la introducción, los distintos problemas que se plantean en las comunicaciones han sido solucionados en buena medida por los protocolos TCP, en sus diversas variantes, mediante el uso de distintas mejoras. Sin embargo, las variantes que solucionan determinados problemas terminan mostrando falencias bajo otras circunstancias. En este caso se ha analizado cómo los errores en ráfaga afectan a cuatro variantes características del protocolo TCP como son Vegas, Reno, Cubic y Westwood.

Observando los resultados obtenidos para todas las simulaciones, y comparando el comportamiento de los cuatro protocolos, es notoria la dificultad que presenta tempranamente el protocolo vegas para recuperarse de un evento como es el error en ráfaga. Estas dificultades no se detectan en los demás protocolos, por lo menos en pruebas de bajos valores de errores en ráfaga (iguales o menores a 20 ráfagas). Puede observarse en las gráficas que, luego de producirse el evento, el protocolo reacciona disparando sus mecanismos de congestión, interpretando que una situación de este tipo se ha producido, cuando en realidad lo que se presenta es un problema de pérdida de paquetes por errores en ráfaga.

Debido a que se detectó este comportamiento por parte del protocolo Vegas en una ráfaga de 15 paquetes, se realizó un seriado de errores simulado distintos tamaños de ráfaga, comenzando con una simulación sin errores (0) y a continuación con 5, 10, 15, 20 y 25 paquetes perdidos en ráfagas, analizando su evolución a través del throughput (figuras 33 a 38). De esta manera se puede observar cómo el protocolo extiende progresivamente la demora en iniciar nuevamente la transmisión de los datos, resultando de esta manera sensible a este tipo de errores desde un primer un momento.

5 Futuros Trabajos

En virtud de la reacción del agente Vegas ante este escenario, y para poder determinar más detalles en su comportamiento bajo condiciones similares, se propone realizar variaciones de los parámetros característicos alfa y beta de este protocolo a fin de verificar si es posible mejorar la respuesta ante los errores en ráfaga. Asimismo, se debe considerar un análisis más amplio que abarque una mayor cantidad de nodos inalámbricos y realizar ensayos con redes ad-hoc, utilizando para tal fin a la herramienta ns-3.

Referencias

[1] Postel J., "RFC 793: Transmission Control Protocol", September 1981.
[2] Stallings, William, "Comunicaciones y redes de computadoras". 6ta edición, Prentice Hall, 2000
[3] David Clark, RFC 813: Window and Acknowledgment Strategy in TCP, Julio 1982
[4] M. Allman, V. Paxson and W. Stevens, "RFC 2581: TCP Congestion Control", April 1999
[5] M. Handley, J. Padhye and S. Floyd, "TCP Congestion Window Validation", RFC 2861, June 2000.
[6] Allman M., Paxson V. and Blanton E., "TCP Congestion Control", IETF, Standars Track RFC 5681, Sept. 2009.
[7] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-Host Congestion Control for TCP," IEEE Communications Surveys Tutorials, vol. 12, no. 3, 3rd quarter 2010, pp. 304-340.
[8] Stevens, W., "TCP slow start, congestion avoidance, fast retransmit, and fast re-covery algorithms".RFC 2001, 1997.
[9] V. Jacobson, "Congestion Avoidance and Control," ACM SIGCOMM Computer Communication Review, Vol. 25, No. 1, pp. 157-187,1995.
[10] Teerawat, Issariyakul & Ekram Hossain, "Introduccion to Network Simulartor NS2", Springer, 2009
[11] The ns Manual, (formerly ns Notes and Documentation). A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. Kevin Fall & Kannan Varadhan, Editores, Nov 5, 2011 (ns-2.35)