

.Análisis del comportamiento de variantes de TCP cuando se producen desconexiones de un nodo móvil de una red heterogénea

Diego R. Rodriguez Herlein¹, Carlos A. Talay¹ and Luis A. Marrone²

¹ Instituto de Tecnología Aplicada I.T.A.
Universidad Nacional de la Patagonia Austral – U.A.R.G. – U.N.P.A.
Río Gallegos, Argentina

² L.I.N.T.I
Universidad Nacional de La Plata
La Plata, Argentina
{dherlein@uarg.unpa.edu.ar, ctalay@uarg.unpa.edu.ar and lmarrone@info.unlp.edu.ar}

Abstract. El presente trabajo está dedicado al análisis de desempeño de distintas variantes del protocolo TCP en redes heterogéneas, cuando se producen desconexiones del host móvil. El objetivo es analizar el comportamiento de variantes de TCP, influenciadas en su desarrollo por el escenario de implementación, en una red híbrida con enlaces cableados e inalámbricos y desconexiones de distinta duración. Para ello, se implementó un modelo simple de una red de acceso WLAN en el simulador NS-2 y se realizaron distintas pruebas que permitieron clasificar y cuantificar los efectos negativos para el rendimiento de TCP que producen las desconexiones frecuentes.

Keywords: TCP, Desconexiones, NS2, Rendimiento.

1 Introduction

El protocolo TCP se diseñó específicamente para proporcionar un flujo de bytes confiable de extremo a extremo, a través de una red no confiable que puede tener diferentes topologías, anchos de banda, retardos, tamaños de paquete y demás características, a través del camino [1]. Transporta la mayor parte del tráfico de internet, por lo que su rendimiento depende, en gran medida del propio rendimiento del este protocolo.

Para lograr esa confiabilidad extremo a extremo, asigna un número de secuencia a cada byte transmitido, y espera una confirmación positiva (ACK) de la capa de transporte del receptor. Para administrar la cantidad de datos que se envían a la vez, utiliza el denominado algoritmo de ventana deslizante [2]. El tamaño de la ventana establece el número máximo de bytes que pueden ser transmitidos sin haber sido aún reconocidos.

El protocolo TCP fue diseñado originalmente para redes cableadas donde las pérdidas y los retrasos de paquetes se deben casi con exclusividad a la congestión de la red.

La congestión es uno de los principales problemas que se afronta en la transmisión de datos, e implica que se saturan los recursos de la red, degradándose su utilización [3]. Cuando la congestión empieza a producirse, el tiempo de transmisión a través de la red aumenta. Conforme la congestión se hace más severa, los nodos de la red descartan paquetes, llegando al extremo en que esta puede colapsar. A partir de lo que se denominó el colapso por congestión, se incorporó un mecanismo de control, conformado por distintos algoritmos. Estos algoritmos de control de congestión, basados en ventanas, permiten adaptar la tasa de envío en forma dinámica, evitando saturar la capacidad de la red y, al mismo tiempo, utilizar eficientemente el ancho de banda disponible, proporcionando una parte justa del ancho de banda de la red a todas las conexiones [4].

De esta forma, el control de congestión primero debe detectar la congestión y después, tomar las acciones necesarias. En las redes cableadas, TCP supone que hay congestión cuando se pierde un paquete.

Las redes inalámbricas han experimentado un importante auge en los últimos años debido, principalmente al desarrollo tecnológico. Estas redes brindan flexibilidad y portabilidad al usuario, sin tener que sacrificar la conexión a Internet o a la red del lugar de trabajo.

El control de congestión estaba diseñado a medida de las redes cableadas, donde los datos normalmente llegan en orden y prácticamente sin errores. Este diseño está claramente influenciado por el escenario. De esta manera, se encuentra con nuevos desafíos en los enlaces inalámbricos, debido a su naturaleza más impredecible [5].

Una de las principales características del enlace inalámbrico es su alta tasa de error (BER). La tasa de error en tránsito es importante debido a que presentan características de transmisión diferentes. Esto tiene su origen en la propia naturaleza del medio físico, que depende de las condiciones del entorno, dado que las señales que se propagan sufren de atenuación, interferencia y ruido, por lo que los paquetes que se reciben pueden estar dañados y se descartan, produciendo la pérdida de paquetes en tránsito. Además, se le suman los efectos de la movilidad de los hosts y del entorno, que pueden modificar las condiciones de transmisión, produciendo otras pérdidas de paquetes en tránsito. Por lo tanto, el reordenamiento de paquetes es más frecuente. De esta manera, es habitual observar largos retardos y pérdida de paquetes que no se deben exclusivamente a la congestión y el control de congestión enfrenta nuevos desafíos en el entorno inalámbrico.

Históricamente la detección de la congestión es por pérdida de paquetes. Esta suposición desdibuja en un escenario inalámbrico porque ya no es una medida confiable para detectar la congestión [6]. En este nuevo escenario hay un porcentaje elevado de pérdidas que no tiene que ver con la congestión. Esto se debe a que la mayoría de las técnicas de control de congestión están diseñadas para un escenario a nivel de red, pero estas pérdidas ocurren a nivel de enlace.

Por esta razón, TCP reacciona inadecuadamente ante las pérdidas de paquetes no relacionadas con la congestión, por ejemplo, si se pierde un paquete de datos debido a interferencias de radiofrecuencia de corta duración en el canal de transmisión. A pesar de que no hay desbordamientos de buffer, TCP tomara la decisión de reducir la

ventana de congestión en forma innecesaria y, por ende, su tasa de transmisión y el rendimiento. En cambio, lo que debería haber sucedido es recuperarse de la pérdida y continuar la transmisión a la misma tasa de envío pues no había congestión en la subred.

Los problemas de rendimiento de TCP no se circunscriben exclusivamente a los enlaces inalámbricos. Otros escenarios requirieron modificaciones del control de congestión para mejorar su rendimiento en esos enlaces. Se puede observar que la búsqueda de mejora de rendimiento es en función de escenarios en particular [7]. Como en su diseño original, las variantes de TCP están fuertemente influenciadas por las características del escenario en particular para el que fueron desarrolladas.

Sin embargo, TCP debe poder desplegarse en cualquier tipo de red y debe ser flexible para lograr un buen rendimiento en redes donde del origen a destino puede transitar distintos tipos de medio de transmisión. Hoy es muy probable que la conexión comience en un enlace inalámbrico (cliente) y termine en un servidor conectado a una red cableada. Para lograr esa flexibilidad, ha tenido que renunciar a conseguir un rendimiento óptimo en distintos escenarios como son las redes de muy alta velocidad, las redes con valores altos de BDP (producto de ancho de banda – latencia), y en medios con valores de pérdidas altas como los entornos inalámbricos.

Durante los últimos años, se han realizado distintas propuestas para mejorar el protocolo y adecuarlo a estos nuevos ambientes. A pesar de todas las propuestas, que dieron lugar a tantas otras variantes del protocolo TCP, no han tenido una gran aceptación, debido a que debe transitar todo tipo de redes.

Analizando algunas de las distintas implementaciones de TCP [8], se pudo observar que están ligadas a escenarios particulares. Detrás de cada variante hay una búsqueda de mejora de la performance en función de un conjunto acotado de características de estos escenarios y se observa su fuerte influencia en el desarrollo de las variantes de TCP. Es decir, las mejoras fueron desarrolladas para mejorar la performance en escenarios de particulares características comunes.

Una mejora en un escenario es de esperar que mejore el rendimiento para esos requerimientos. A pesar de todas las propuestas que se pueden encontrar en la literatura, estas no han tenido una gran aceptación, dado que su aplicación será, en general, en una red donde un segmento TCP es inevitable que atraviese escenarios diferentes.

Como se está evaluando el rendimiento de una sesión TCP, esta se evalúa de extremo a extremo. Por esta razón, resulto relevante analizar qué es lo que sucede cuando estas variantes deben atravesar un escenario con distintos tipos de enlace en su camino. De esta forma, es interesante ensayar el rendimiento de distintas variantes de TCP desarrolladas para distintos tipos de enlaces y escenarios, como lo que se plantean en una red heterogénea de acceso WLAN.

Por esta razón se utilizaron en los ensayos, versiones o implementaciones de TCP pensados para mejorar el rendimiento en escenarios tanto fijos como inalámbricos.

2 Casos de Prueba

En una conexión que incluye enlaces inalámbricos, se producen frecuentemente desconexiones debido a la movilidad, crisis de energía o variaciones en el entorno. Este tipo de desconexiones puede tener distinto tiempo de duración, aunque suele durar más que un RTO y es más corto que la vida útil de una conexión TCP.

El emisor ignora los motivos de las pérdidas e invoca los algoritmos de control de congestión erróneamente, reduciendo así el rendimiento efectivo tras la reconexión.

Es así que resulta interesante analizar el comportamiento de una red WLAN de acceso ante la presencia de desconexiones de diferentes tiempos de duración. Se simula un modelo sencillo (Figura 1) de 3 nodos y un solo flujo para poder visualizar el efecto en el rendimiento que producen estas desconexiones. Las redes de acceso inalámbricas típicas comprenden un host móvil conectado de forma inalámbrica a una estación base, que a su vez está conectada a la red troncal cableada, posiblemente a Internet. En la Figura 1 se observa el modelo implementado en el simulador. La razón de recurrir a este modelo sencillo, responde a que permite introducir, en forma controlada, los efectos individuales pudiéndose observar sus respuestas en forma específica y, de esta manera, aislar el efecto de las desconexiones de las otras causas de pérdida de rendimiento de TCP.

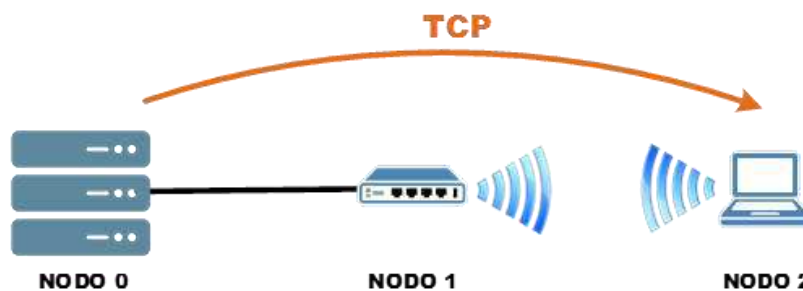


Fig. 1. Modelo Simple de tres nodos y un flujo TCP. (WLAN).

En cuanto a las simulaciones planteadas en este trabajo, no se tiene en cuenta el efecto de la movilidad, ya que en definitiva se traducirá a errores en la comunicación.

Para explorar el efecto de las desconexiones se implementó el modelo propuesto en el simulador de redes de eventos discretos (NS-2, Network Simulator 2), en su versión 2.35 (released nov. 4 2011) [9]. La selección de esta topología es una aproximación a un escenario inalámbrico con un nodo fijo, una estación base y un nodo móvil, con la simplificación práctica que el enlace inalámbrico no presenta errores y solo tiene desconexiones de distinta duración producidas sobre el enlace inalámbrico.

El nodo fijo y la estación base están vinculados por un enlace cableado que se configuró como dúplex, con un ancho de banda 10 Mb/s, retardo de propagación 2 ms. y política de servicio de las colas DropTail. El enlace entre la estación base y el nodo móvil es inalámbrico y se configuró como modo de propagación TwoRayGround, la capa física WirelessPhy, MAC 802.11, la antena OmniAntenna, 2Mb./s con MAC 802.11. El nodo inalámbrico no posee movimiento.

El flujo TCP se define para cada una de las simulaciones entre el nodo fijo (emisor) y el nodo móvil (receptor). El flujo comienza a transmitir y las desconexiones comienzan a partir del paquete número 1000. En cada simulación se transmitieron 3.000 segmentos de TCP de 1.000 Bytes cada uno. Las simulaciones fueron realizadas en forma independiente para cada variante del protocolo y para cada una de los tiempos de desconexión. Las desconexiones ensayadas tuvieron una duración de 0,05, 0,1, 0,5, 1, 2, 5, 10 y 20 segundos.

El flujo TCP se define para cada una de las simulaciones entre el nodo fijo (emisor) y el nodo móvil (receptor). El flujo comienza a transmitir y las desconexiones comienzan a partir del paquete número 1000. En cada simulación se transmitieron 3.000 segmentos de TCP de 1.000 Bytes cada uno. Las simulaciones fueron realizadas en forma independiente para cada variante del protocolo y para cada una de los tiempos de desconexión. Las desconexiones ensayadas tuvieron una duración de 0,05, 0,1, 0,5, 1, 2, 5, 10 y 20 segundos.

En cuanto a las variantes ensayadas, se consideraron las variantes más utilizadas tales como Compound (Windows) [10], CUBIC (Linux) [11]. También se incorporó TCP Westwood [12] desarrollado para las redes con enlaces inalámbricos y TCP Reno, como referencia de control de congestión basado en la pérdida de paquetes. Además, se incluyó TCP vegas [13] como un caso particular, tanto por sus respuestas en los ensayos, como su particular mecanismo de control de congestión que sentó las bases para desarrollar distintas técnicas que permitieran diferenciar las causas de pérdidas por congestión de las que no lo son, utilizando solo la retroalimentación del receptor sin ayuda de la red.

3 Resultados Obtenidos

Para poder analizar los efectos de las desconexiones de distinta duración sobre variantes de TCP, a continuación, se presentan distintas métricas tales como Throughput vs. Tiempo, Throughput Promedio y el Tiempo Total de Transmisión en función de la duración de la desconexión, la evolución del tamaño de la ventana de congestión y del número de secuencia del segmento TCP en función del tiempo.

La Figura 2 representa el throughput instantáneo de TCP Reno para el ensayo con una desconexión de 5 segundos de duración.

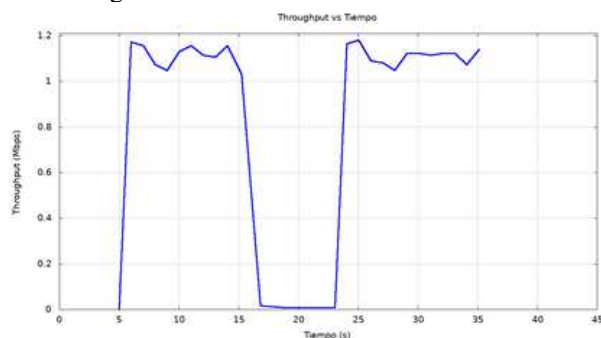


Fig. 2. Throughput vs Tiempo - TCP Reno (Desconexión 5s.)

Como puede observarse, la desconexión con una duración de 5 s., produce una caída abrupta del throughput y recupera su ritmo de transferencia de datos casi inmediatamente recuperada la conexión.

Una de las métricas de interés para análisis es la evolución del tamaño de la ventana de congestión en función del tiempo de la simulación, pues tiene directa relación con el rendimiento de TCP.

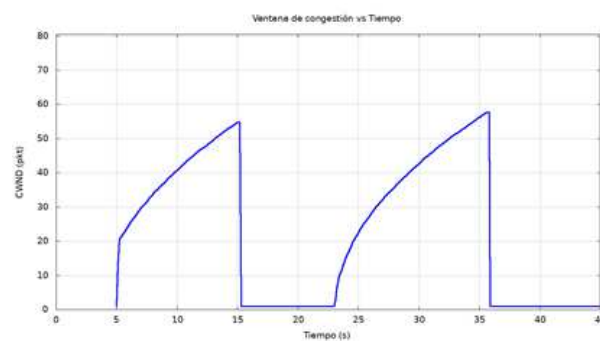


Fig. 3. CWND vs Tiempo - TCP Reno (Desconexión 5s.)

En la Figura 3 se observa la evolución de la ventana de congestión en función del tiempo para la variante TCP Reno para desconexión de 5 segundos. Se puede observar cómo actúan los algoritmos de control de congestión de TCP Reno, reduciendo el valor de la ventana a 1 durante la desconexión y disparando el algoritmo de Slow Start.

Así mismo, en la Figura 4 se puede observar los valores del throughput promedio en Mbit/s, para cada uno de los casos considerando.

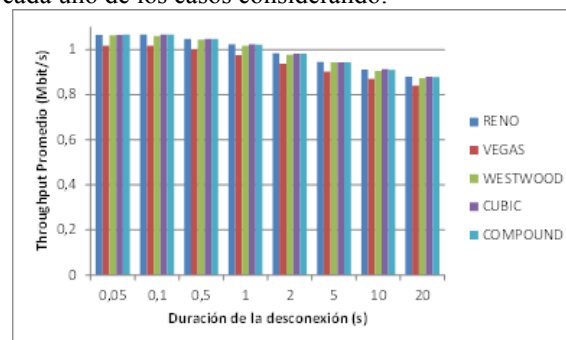


Fig. 4. Throughput promedio vs duración de las desconexiones

En esta figura se observa un patrón común para todas las variantes ensayadas: el throughput promedio disminuye a medida que el tiempo de desconexión aumenta. Más allá de alguna pequeña variación en la disminución relativa entre ellas, de los resultados se desprende que TCP Vegas es de las variantes ensayadas, la que presenta el menor valor de throughput promedio en todos los casos.

Dado que las simulaciones se realizan transmitiendo una cantidad de información constante, resulta interesante analizar el tiempo que demora cada una de las variantes estudiadas en transmitirlos. Esta medida puede tener directa relación con el rendimiento, lo que permitirá observar cómo afecta esas desconexiones a cada una de las variantes analizadas desde otra perspectiva.

En la siguiente figura se observa el Tiempo Total de Transmisión en función del tiempo para cada una de las variantes ensayadas y para cada uno de los tiempos de desconexión.

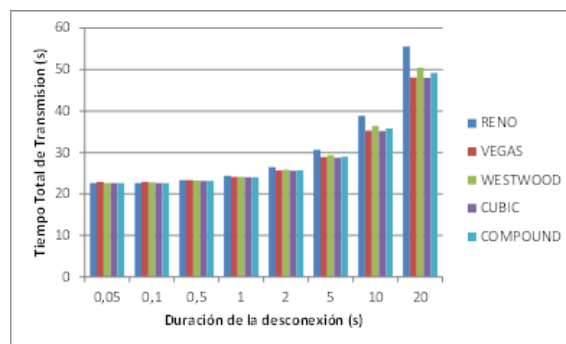


Fig. 5. Tiempo total de transmisión vs duración de las desconexiones

Ahora observamos que a medida que el tiempo de desconexión crece, el tiempo necesario para transmitir todos los datos también lo hace, de una forma exponencial. Además, para desconexiones de baja duración el tiempo necesario para transmitir todos los paquetes es prácticamente del mismo orden. Sin embargo, a medida que aumenta el tiempo de desconexión, empiezan a diferenciarse distintos valores de tiempo total de transmisión, siendo la variante de TCP Reno la que más tiempo necesita para concluir la transmisión de los datos.

Como análisis complementario al tiempo total de transmisión, resulta interesante observar la evolución del número de secuencia del segmento TCP. Al ser un valor instantáneo permite observar la dinámica de la transmisión.

A continuación, se observan una serie de 3 figuras (Figura 6 a Figura 8) que representan la evolución del número de secuencia del segmento TCP en función del tiempo de simulación. La serie muestra a las variantes de Reno, Vegas y Westwood para la desconexión de 20 segundos de duración.

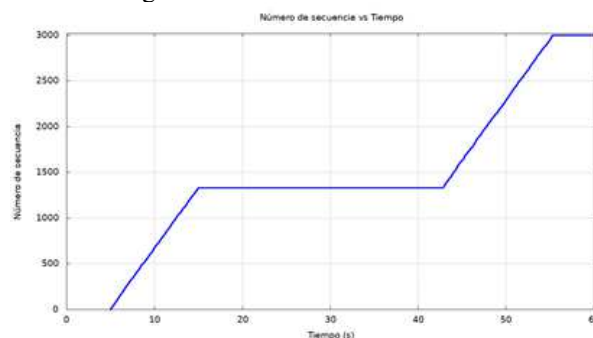


Fig. 6. N° de Secuencia vs. Tiempo - TCP Westwood (Desconexión 20s.)

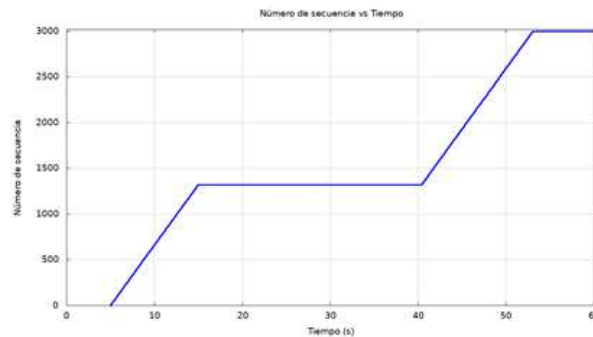


Fig. 7. N° de Secuencia vs. Tiempo - TCP Vegas (Desconexión 20s.)

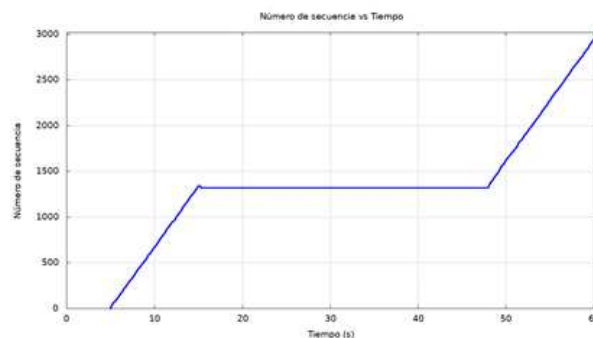


Fig. 8. N° de Secuencia vs. Tiempo - TCP Reno (Desconexión 20s.)

Para el mismo tiempo de desconexión, comparando TCP Westwood (Figura 6) y TCP Vegas (Figura 7), con TCP Reno (Figura 8) es este último el que demora más tiempo en retomar el envío de los datos, lo que repercute directamente en el tiempo total de transmisión.

En la Figura 8 se puede determinar el tiempo necesario desde el momento de la desconexión para que TCP Reno continúe la transmisión de los datos. Si bien la desconexión es de 20 segundos, TCP Reno demora casi el doble de ese tiempo en retomar el envío.

Teniendo en cuenta las consideraciones anteriormente expuestas, se observa que la interrupción del envío de datos afecta en menor medida al protocolo Vegas y en mayor medida al protocolo Reno, y la variante Westwood, es la que presenta una respuesta intermedia.

4 Trabajos Futuros

Se prevé ampliar el número de variantes del protocolo TCP analizadas, como así incluir otras causas de pérdidas de paquetes en tránsito de las redes inalámbricas, y estudiar cómo afectan su rendimiento.

5 Conclusiones

En los resultados de las simulaciones se pudo observar que, si bien el throughput promedio disminuye a medida que el tiempo de desconexión aumenta, en forma similar para las variantes ensayadas. Sin embargo, al analizar el tiempo total de transmisión se observa allí las diferencias en la recuperación de cada variante, siendo TCP Reno la que requiere mayor tiempo para completar el envío de datos.

Durante la desconexión, se descartan tanto los paquetes de datos como los ACK, y cada intento de retransmisión conduce a una retransmisión fallida. Esta condición solo se detecta por la caducidad de los RTO. Estos intentos consecutivos fallidos harán que el emisor TCP aumente exponencialmente los tiempos de acuerdo al algoritmo de Karn, Esto dará como consecuencia que se espacien los reintentos y por tanto el canal permanezca inactivo desperdiciando recursos de red.

Además del tiempo sin actividad, el rendimiento de TCP también se degrada por el disparo de los algoritmos de control de congestión cuando no hay descarte de paquetes en los nodos intermedios. Esto produce que el TCP emisor reduzca su tasa de envío cuando no es necesario. Esto se observa en la gráfica de la evolución del tamaño de la ventana de congestión en función del tiempo (Figura 3)

En los ensayos se observó que TCP Reno requiere mayor tiempo para recuperar la tasa de envío, debido a los crecientes períodos de inactividad producidos por el backoff exponencial que hace crecer los valores de RTO. Esto se produce porque cada timeout en serie reduce su ventana de congestión, lo que implica iniciar la fase de Slow Start en cada ocasión. Dado que las desconexiones también impiden el arribo de los ACK, no se disparan los algoritmos de Fast Retransmit ni Fast Recovery, pues no puede recibir ningún ACK duplicado. La recuperación se inicia reduciendo el valor de la ventana de congestión (en 1) e invoca al algoritmo de Slow Start, causando un impacto muy negativo en el rendimiento de TCP.

En los casos más severos, al producirse una desconexión y posterior reconexión del host móvil, el emisor podría seguir a la espera hasta que ocurra un timeout para reiniciar la retransmisión.

Además, una vez que deja de estar en espera, requiere de un tiempo para llegar a las tasas de transmisión anteriores a la desconexión del nodo. Esto se debe no solo al disparo del control de congestión, por más que no la hubiera, sino también debido a la continua reducción a la mitad del umbral (ssthresh) con los timeout en serie.

Agradecimientos

Agradecemos al alumno de la Lic. en Sistemas de la UNPA-UARG Franco A. Trinidad su colaboración en la realización de las pruebas con el simulador.

Referencias

1. J. Postel, RFC 793: Transmission Control Protocol, September 1981.
2. 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
3. Rodríguez Herlein, D. R.; Talay, C. A.; González, C. N.; Trinidad, F. A.; Almada, L.; Marrone, L. A., Un análisis de comportamiento entre distintos mecanismos de control de congestión ensayados sobre una topología mixta. CACIC 2018. Tandil, Argentina. <http://sedici.unlp.edu.ar/handle/10915/73349>
4. Rodríguez Herlein D. R., Talay C. A., González C. N., Trinidad F. A., Almada M .L., Marrone L. A. (2019) Contention Analysis of Congestion Control Mechanisms in a Wireless Access Scenario. In: Pesado P., Aciti C. (eds) *Computer Science – CACIC 2018*. CACIC 2018. Communications in Computer and Information Science, vol 995. Springer, Cham. https://doi.org/10.1007/978-3-030-20787-8_18
5. Teja F. R., Vidal, L, Alves, L., TCP sobre enlaces wireless – Problemas y algunas posibles soluciones existentes, Curso de posgrado y actualización, Instituto de Ingeniería Eléctrica, Facultad de la República, marzo 2004. <https://studylib.es/doc/8541837/articulo-sobre-tcp-en-wireless>
6. D. R. Bhadra, C. A. Joshi, P. R. Soni, N. P. Vyas and R. H. Jhaveri, Packet loss probability in wireless networks: A survey, 2015 International Conference on Communications and Signal Processing (ICCSP), 2015, pp. 1348-1354, doi: 10.1109/ICCSP.2015.7322729
7. D. R. Rodríguez Herlein, Análisis del rendimiento del protocolo TCP en redes de acceso wireless, Master Thesis, Fac. Informática., UNLP, La Plata, Bs.As., Argentina, 2020
8. Saleem-ullah Lar, Xiaofeng Liao, An initiative for a classified bibliography on TCP/IP congestion control, *Journal of Network and Computer Applications*, Volume 36, Issue 1, 2013, Pages 126-133, ISSN 1084-8045, <https://doi.org/10.1016/j.jnca.2012.04.003>
9. T. Issariakul, E. Hossain, *Introduction to Network Simulator 2*, Segunda Edición. New York: Editorial Springer. 2012
10. S. Mascolo, C. Casetti, M. Gerla, S. Lee, and M. Sanadidi, TCP Westwood: congestion control with faster recovery, Univ. California, Los Angeles, Tech. Rep. CSD TR, vol. 200017, pp. 1–14, 2000
11. Sangtae Ha; Injong Rhee; Lisong Xu (July 2008). CUBIC: A New TCP-Friendly High-Speed TCP Variant, *ACM SIGOPS Operating Systems Review*. 42 (5): 64–74
12. S. Mascolo, C. Casetti, M. Gerla, S. S. Lee and M Sanadid, TCP Westwood: Congestion control with faster recovery. Technical Report 200017, UCLA CSD 2000
13. Low, Steven; Peterson, Larry & Wang, Limin. *Understanding TCP Vegas: Theory and Practice*, <https://www.cs.princeton.edu/research/techreps/TR-616-00>