

Contienda entre las variantes del protocolo TCP Vegas y Reno por los recursos de la red en un modelo híbrido simple

Diego R. Rodríguez Herlein, Carlos A. Talay, Claudia N. González y Franco

A. Trinidad

Campus Universitario – Oficina 18/ Dpto. Ciencias Exactas e Informática UARG / UNPA
{dherlein, ctalay, cgonzalez}@uarg.unpa.edu.ar, tfrancoalejandro@gmail.com

Luis A. Marrone

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

Resumen

El presente trabajo aborda la problemática que se observa al convivir diferentes variantes de los algoritmos de control de congestión en la competencia por los recursos compartidos de la red. Dada la naturaleza de las redes IP, es deseable el uso justo de estos recursos por los distintos flujos TCP, sin perder de vista las posibles degradaciones del rendimiento en los enlaces inalámbricos. De esta manera, se propone analizar cómo interactúan dos flujos TCP similares que utilizan algoritmos de control de congestión conceptualmente diferentes, como son Reno y Vegas, compitiendo por los recursos de una red híbrida con enlaces cableados e inalámbricos

Palabras clave: TCP, red híbrida, contienda, ancho de banda

Contexto

Este estudio está enmarcado en el proyecto de investigación 29/A396 “Evaluación de desempeño del protocolo TCP en topologías mixtas cableadas-inalámbricas” radicado de la UNPA-

UARG, continuación del proyecto 29/A358-1 “Análisis de performance del protocolo TCP utilizado en redes móviles”. El proyecto está compuesto mayoritariamente de docentes de la UNPA-UARG, dirigido por el Sr. Carlos A. Talay y cuanta como Co-director al Sr. Luis A. Marrone perteneciente a la UNLP y se financia íntegramente con fondos destinados a proyecto de investigación de la UNPA-UARG.

Introducción

El protocolo TCP [1] (Transmission Control Protocol) es el encargado de transportar la mayor parte del tráfico de internet, es por ello que el rendimiento de internet depende, en gran medida, de cómo funciona TCP. Las características de rendimiento de una versión particular de TCP se definen por el algoritmo de control de congestión que implementa. En las redes que incluyen enlaces inalámbricos, el control de congestión puede ser aún más determinante en el rendimiento de este protocolo. En el presente proyecto de investigación se analizan algunas de las propuestas de

control de congestión para mejorar el rendimiento pero que preservan su principio fundamental de comunicación host a host, es decir, que no dependen de ningún tipo de señalización explícita de la red.

La característica fundamental de TCP es su capacidad para proporcionar un canal fiable, bidireccional y virtual entre dos hosts en Internet. Dado que el protocolo funciona a través de la red IP, es TCP el que debe proporcionar esta fiabilidad, implementando un control de flujo basado en ventana deslizante. [2]

Desafortunadamente, el TCP original carecía de algún medio para ajustar la velocidad de transmisión de acuerdo al estado de la red. Esto crea varios efectos inesperados, siendo el más importante el colapso por congestión. Por ello el uso inteligente de los recursos disponibles en las redes de conmutación de paquetes, no es un problema trivial.

Para resolver el problema del colapso de congestión, se han propuesto varias soluciones, donde todas ellas comparten la misma idea: la introducción de un mecanismo que limite para la tasa de envío. Para ello se introdujo el concepto de ventana de congestión, cuyo propósito es estimar de la cantidad de datos que la red puede aceptar para su entrega sin que se produzca congestión.

Internet es un entorno heterogéneo y el uso efectivo de la red no sólo depende de que un único flujo TCP pueda utilizar la capacidad de la red, sino también de qué tan bien coopera con otros flujos a través

de la misma red. La eficacia no es el único parámetro importante de los algoritmos de control de congestión, también deben hacer valer el uso justo de recursos compartidos.

Es por ello que el análisis de dos flujos de datos, regidos por el protocolo TCP, pero con distintas variantes, puede aportar una visión acerca de cómo es utilizado el ancho de banda bajo distintas estrategias, y ver cómo interactúan entre sí ante una competencia por la utilización de este recurso. Explorando este camino, planteamos aquí un modelo simple, con dos flujos que compiten entre sí en la misma trayectoria de red en una topología híbrida, sin otros flujos presentes.

El caso de estudio

Los algoritmos de control de congestión, tratan de determinar dinámicamente el ancho de banda disponible y la latencia de la red. Luego en función de su análisis se modificar la tasa de envío del emisor TCP para evitar el colapso de la subred.

El control de congestión [3] comienza con el algoritmo Slow Start, en donde se incrementa la ventana de congestión (cwnd) en forma exponencial [4] hasta que se alcanza el valor del umbral (ssthresh). A partir de este punto, comienza el algoritmo de Congestion Avoidance [5], que incrementa el valor de cwnd en forma lineal. Si se retrasa el ACK de un paquete o bien recibe un ACK duplicado, TCP asume que el paquete se perdió y lo retransmite. Es en este momento que se activa el algoritmo

de retransmisión rápida (Fast Retransmit) [6].

Dentro de las implementaciones de TCP existentes [7] se analizaran:

TCP Reno [8]: Cuando se detecta una pérdida de paquete, TCP Reno dispara el algoritmo de Fast Recovery [9], reenviando el paquete perdido y reduciendo el umbral ssthresh a la mitad, evitando disparar el algoritmo Slow Start. Cuando Reno sale del algoritmo Fast Recovery, da paso al algoritmo Congestión Avoidance, lo que permite que se recupere más rápido de la congestión.

TCP Vegas [10], [11]: Hasta ahora la ventana de congestión crecía hasta que ocurría una pérdida de paquete. TCP Reno asume que hay congestión cuando se detectan estas pérdidas. La idea del TCP Vegas es controlar y mantener el tamaño adecuado de la ventana de manera de no llegar a la pérdida de paquetes. El TCP Vegas administra el tamaño de la ventana observando el RTT (round-trip-time) de los paquetes que el emisor envió con anterioridad. Si el RTT crece, TCP Vegas reconoce que la red comienza a estar congestionada y reduce su ventana de congestión. Por el contrario, si los valores de RTT decrecen, el emisor determina que la red no está congestionada e incrementa el tamaño. Para evitar las oscilaciones, se define un intervalo en donde se mantiene el tamaño de la ventana.

TCP fue desarrollado y posteriormente optimizado para redes cableadas, lo que

implica que la pérdida de paquetes se debe casi con exclusividad a la congestión en la red, suponiendo una tasa de error de bits en tránsito prácticamente inexistente. Con el crecimiento en la demanda de movilidad, se comenzó a utilizar en enlaces inalámbricos. Estos enlaces tienen características muy diferentes, de manera que los retrasos y las pérdidas de paquetes, ya no se deben exclusivamente a la congestión, sino también al daño de paquetes en tránsito [12] y la desconexión temporal de los nodos, lo que puede ocasionar un comportamiento muy poco eficiente del protocolo en ese entorno. Esta complejidad adicional se contempla en el modelo del presente artículo, al utilizar una topología híbrida (cableado e/inalámbrico), lo que plantea un caso más general de análisis.

Diseño del modelo de prueba

El modelo implementado en el simulador NS-2 (Network Simulator 2, v2.35) representa una topología híbrida de 5 nodos (2 nodos cableados, 1 estación base y 2 nodos inalámbricos), como se observa en la siguiente figura (Fig.1)

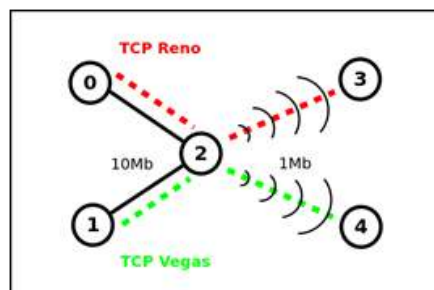


Figura 1. Modelo de estudio

Los nodos 0 y 1 están conectados a la estación base (nodo 2) mediante enlaces cableados full dúplex, 10Mb/s, retardo de propagación de 2ms y política de cola DropTail. Para el enlace inalámbrico se establece un ancho de banda de 1 Mb/s, modo de propagación TwoRayGround, capa física WirelessPhy, MAC 802.11, antena OmniAntena y los dos nodos (3 y 4) no poseen movimiento.

Se realizaron tres pruebas con dos tráficos FTP de 3.000 paquetes de 1.000 bytes cada uno, utilizando las variantes de TCP Reno y Vegas. En todos los casos, el flujo del TCP Reno es desde el Nodo 0 al Nodo 3 y para el TCP Vegas del Nodo 1 al Nodo 4,

En la primera simulación, ambos flujos TCP comienzan simultáneamente, a los 5 seg. de iniciado el ensayo (fig. 2). En la segunda corrida, solo el flujo de TCP Vegas comienza a los 5 seg. y se demora hasta los 10 seg. el comienzo del flujo TCP Reno (fig. 3). En la tercera simulación se invierten los flujos, y es TCP Reno quién comienza a transmitir a los 5 seg. y TCP Vegas a los 10 seg. (fig. 4).

De cada una de las prueba se obtuvo throughput instantáneo (cantidad de Bytes en tránsito en la red en determinado momento), utilizando un script AWK que filtró los valores correspondientes del archivo de traza generado por NS2. Luego se utilizó gnuplot para graficar los resultados.

Resultados obtenidos y futuras líneas de investigación

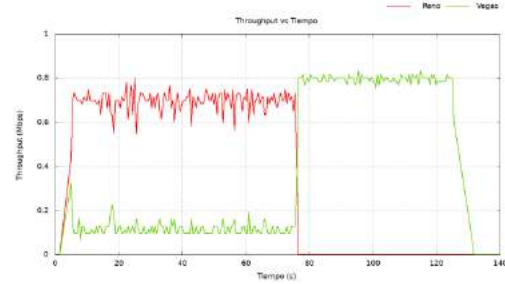


Figura 2. Reno y Vegas inicio simultáneo

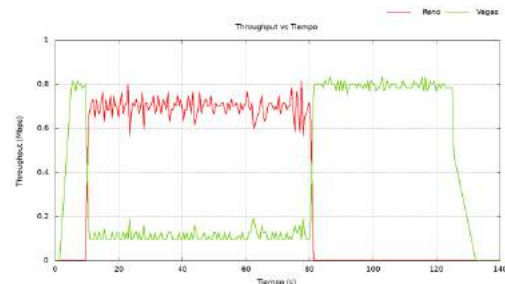


Figura 3. Inicia Vegas y luego Reno con 5 seg. de retraso

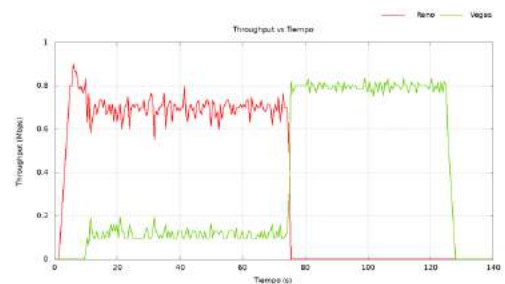


Figura 4. Inicia Reno y luego Vegas con 5 seg. de retraso

En la confrontación por recursos en un esquema de transmisión bajo estas condiciones son muy claras. La agresividad del protocolo TCP Reno acapara gran parte del ancho de banda al momento de competir contra su rival, en este caso el protocolo TCP Vegas. En la Figura 2, vemos un ensayo donde las transmisiones comienzan en forma simultánea y desde el comienzo Reno captura el mayor ancho de banda posible, relegando al protocolo Vegas al desarrollo de un tráfico mínimo, retrasando notablemente su ritmo de

transmisión de datos. Esto se confirma en cuanto, comenzando al mismo tiempo y habiendo transmitido el mismo volumen de datos, para el caso de TCP Reno emplea 77 seg. para completar el envío, cuando para el protocolo Vegas se tiene 132 seg. De la misma forma, en las Figuras 3 y 4, cuando los comienzos de los flujos no son simultáneos, se puede observar idéntico comportamiento.

En base al ejemplo planteado, vemos que si por competencia de recursos de tratase, la variante más agresiva es la que domina la competencia en la transmisión de datos. La agresividad en el mecanismo de transferencia de datos marca la diferencia y es el factor distintivo que hace que un protocolo domine por sobre otro. Esto se verifica además si analizamos el tiempo total de transmisión de los datos. A pesar de ello hay un hecho a resaltar, si consideramos el nivel instantáneo de throughput, observamos que cuando el protocolo Vegas se encuentra transmitiendo solo en el enlace, su valor promedio es más alto, por tanto podemos inferir que posee un mejor aprovechamiento de los recursos disponibles de la red.

Si bien la prueba es concluyente, ampliarse el estudio con ensayos donde se presente otras topologías con un mayor espectro de variantes del protocolo TCP y estudiar si se puede introducir propuestas que brinden más equidad al funcionamiento de la red.

Formación de Recursos Humanos

El grupo de investigación conformado se caracteriza una constitución heterogénea de profesionales vinculados a la informática. Entre ellos podemos enumerar una Licenciada en Informática, un Ingeniero en Electrónica y un Ingeniero Electricista con un máster en Sistemas y Redes de Comunicaciones. También integra el grupo dos alumnos de la carrera Lic. en Sistemas de la UNPA-UARG. En el transcurso del proyecto se tiene como objetivo consolidar la formación en investigación de los integrantes de menos antecedentes en proyectos y también está contemplado que uno de los integrantes complete su trabajo final de maestría.

Referencias

- [1] Postel J., "RFC 793: Transmission Control Protocol", September 1981.
- [2] M. Handley, J. Padhye and S. Floyd, "TCP Congestion Window Validation", RFC 2861, June 2000.
- [3] V. Jacobson, "Congestion Avoidance and Control," ACM SIGCOMM '88, Vol. 118 N°4, August 1988.
- [4] Stevens, W., "TCP slow start, congestion avoidance, fast retransmit, and fast re-recovery algorithms". RFC 2001, 1997.
- [5] Jacobson V., "Congestion Avoidance and Control", ACM SIGCOMM Computer Communication Review, 25(1), 1995, pp. 157-187.
- [6] W. Stevens, "RFC2001—TCP Slow Start, Congestion Avoidance, Fast Retransmit," RFC, 1997
- [7] Saleem-ullah Lar, Xiaofeng Liao, "An initiative for a classified bibliography on TCP/IP congestion control", Journal of Network and Computer Applications 36, 126–133, 2013
- [8] V. Jacobson, "Modified TCP congestion avoidance algorithm," email to the end2end list, April 1990.
- [9] M. Allman, V. Paxson, and W. Stevens, "RFC2581—TCP congestion control" RFC, 1999.
- [10] L. Brakmo and L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet," IEEE Journal Selected Areas Communications. vol. 13, no. 8, pp. 1465–1480, October 1995.
- [11] G. Hasegawa, K. Kurata, and M. Murata, "Analysis and improvement of fairness between TCP Reno and Vegas for deployment of TCP Vegas to the Internet" in Proc. IEEE ICNP, 2000, pp. 177–186.
- [12] Elaarag, H., "Improving TCP Performance over Mobile Networks", ACM Computing Surveys, Vol. 34, No. 3, September 2002.