

Explorando posibles mejoras de protocolo TCP en redes móviles

Diego R. Rodríguez Herlein Carlos A. Talay

Campus Universitario – oficina 18 - Dpto. Ciencias Exactas e Informática UARG / UNPA
diegorh@gmail.com carlostalay@yahoo.com.ar

Luis Marrone

L.I.N.T.I. – Universidad Nacional de La Plata
Calle 50 y 115 – 1er. Piso – Edificio Bosque Oeste
lmarrone@info.unlp.edu.ar

Resumen

En este trabajo se intenta realizar una mínima recopilación de los esfuerzos que se han realizado en adaptar al protocolo TCP (Transmission Control Protocol) a redes de datos en donde las tasas de error son mucho mayores que las tasas para el cual fue pensado originalmente este protocolo, debido a la congestión de datos. Así mismo intentaremos analizar que posibles mejoras pueden realizarse en la aplicación del protocolo TCP en redes móviles, en donde un enlace wireless interviene en uno o más de los trayectos de los datos e introduce dos fenómenos típicos de este enlace como son, la pérdida de paquetes por diversas caudas o directamente el desacople de uno de los nodos.

Palabras clave: TCP, performance, wireless, control congestión

Contexto

Este estudio está enmarcado en el PI 29/A358-1 “Análisis de performance del protocolo TCP utilizado en redes móviles” radicado de la UNPA-UARG, que se encuentra actualmente en etapa inicial de desarrollo. El proyecto está compuesto mayoritariamente de docentes de la

UNPA-UARG y coordinado por su Director el Sr. Luis Marrone perteneciente a la UNLP. Este proyecto se financia íntegramente con fondos destinados a proyecto de investigación de la UNPA-UARG.

Introducción

TCP [1] es un protocolo orientado a la conexión, seguro y el más utilizado en la capa de transporte. Soporta la mayoría de las aplicaciones más populares de internet. Es fiable y se adapta adecuadamente a las redes cableadas, de nodos estáticos, que regula el número de paquetes enviados mediante el método de ventanas deslizantes [2], el cual lleva el control de paquetes enviados y controla su recepción mediante el envío de un mensaje de ACK. De esta manera, TCP se adapta a los problemas de las redes cableadas, debidos al retraso y pérdida de paquetes por congestión [3,4]. Los algoritmos de control de congestión, tratan de determinar dinámicamente el ancho de banda y la latencia de la red, modificando el envío de paquetes y su rendimiento, a los cambios de tráfico para evitar el colapso de la subred.

Este protocolo 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. TCP realiza el control de congestión regulando el tráfico que inyecta a la red. Para esto, debe conocer cuando es posible enviar más tráfico y cuando se alcanzó su capacidad máxima.

El control de congestión comienza con el algoritmo de arranque lento (slow start), en donde el TCP emisor comienza con una Ventana de Congestión (cwnd) con valor 1 [5]. Por cada ACK recibido incrementa el valor de dicha ventana en forma exponencial hasta un valor umbral (ssthresh), en donde comienza el algoritmo evitar congestión (congestion avoidance) [6]. En este punto se comienza a incrementar cwnd en forma lineal hasta el tamaño máximo de la ventana del receptor. Asimismo, TCP mide el tiempo que tardan en llegar los ACK, manteniendo un promedio de este retraso y la desviación esperada de ese promedio. Entonces, si el retraso del ACK de un paquete es mayor que el promedio incrementado en 4 veces el desvío calculado o bien recibe un ACK duplicado, el TCP asume que el paquete se perdió y lo retransmite. Es en este momento que se activa el algoritmo de Retransmisión Rápida (fast retransmit).

TCP reacciona a cualquier pérdida de paquetes reduciendo el umbral (ssthresh) a la mitad o a 2 (lo que sea

mayor), restableciendo el valor de la ventana de congestión a 1, activando el algoritmo de arranque lento y, estableciendo el temporizador de retransmisión al intervalo de backoff de acuerdo al algoritmo de Karn [7].

Dentro de las implementaciones de TCP existentes, se destacan [8]:

TCP Tahoe: Posee los algoritmos de slow start y congestion avoidance y posteriormente se le agrego el algoritmo de fast retransmit. La principal desventaja de esta implementación es que cada vez que se pierde un segmento, comienza nuevamente con el algoritmo de slow start, lo que demora demasiado el retorno al valor de las tasas anteriores.

TCP Reno: Cuando se detecta congestión, el TCP Tahoe reduce la ventana de congestión a un segmento lo que dispara el algoritmo de slow start, mientras que TCP Reno dispara el algoritmo de Fast Recovery, reenviando el paquete perdido y reduce el umbral a la mitad, evitando disparar el algoritmo de slow start. Cuando recibe un nuevo ACK, sale de fast recovery y dispara el algoritmo de congestión avoidance. Esto permite que se recupere más rápido de la congestión.

TCP New Reno: La optimización en esta versión de TCP mejora la pérdida individual y aislada de segmentos pero no mejora las pérdidas en segmentos sucesivos. La diferencia con el TCP Reno es que cuando se está en la etapa de fast recovery, el arribo de uno nuevo ACK no hace que se salga de esta, hasta que todos

los segmentos pendientes de confirmar antes del disparo del algoritmo tengan su acuse de recibo. De esta forma un ACK parcial implica que el segmento siguiente se perdió y debe ser retransmitido.

TCP Selective ACK (SACK):

Implementa un mecanismo por el cual el emisor recibe información de los segmentos recibidos e, implícitamente, los perdidos en el receptor. Para esto se implementa un campo opcional SACK en el encabezado de TCP que se envía cuando se reciben segmentos fuera de orden. Contiene la información de los bloques recibidos y sus números de secuencia. Esta versión de TCP utiliza un control de congestión básico y solo utiliza el temporizador de retransmisión como última opción de recuperación. En forma análoga a TCP Reno, TCP SACK dispara el algoritmo de fast recovery al recibir ACK duplicados. Cuando el emisor recibe un ACK duplicado con la opción SACK, almacena los ACK especificados en SACK, permitiendo retransmitir solamente los paquetes implícitamente perdidos en el receptor. De la misma forma que TCP New Reno, sale de la etapa de fast recovery cuando todos los paquetes retransmitidos tienen su acuse de recibo. La ventaja es que, de esta forma, mejora el manejo de la pérdida de segmentos en la misma ventana de forma más eficiente.

TCP Vegas: TCP Vegas posee grandes modificaciones con respecto a las versiones anteriores de TCP. Hasta ahora la ventana de congestión crecía hasta que ocurría una pérdida de paquete y cuando

esto pasaba, el throughput de la conexión se degrada. El TCP New Reno y anteriores asumen que hay congestión solamente cuando detectan estas pérdidas de paquetes. La idea del TCP Vegas es controlar y mantener el tamaño adecuado de la ventana de manera que no ocurra la pérdida de paquetes y se evite que se degrade el throughput. El TCP Vegas administra el tamaño de la ventana observando RTT (round-trip-times) de los paquetes que el TCP emisor envió con anterioridad. El RTT es el tiempo desde que el emisor envía el paquete hasta que arriba el ACK correspondiente. Si el RTT es mayor, TCP Vegas reconoce que la red comienza a estar congestionada y achica la ventana. Por el contrario, si los valores de RTT son más pequeños, el TCP emisor determina que la red no está en congestión e incrementa el tamaño. Otra de las modificaciones importantes del TCP Vegas es en el algoritmo de control de congestión, en la fase de slow start. La tasa de crecimiento del tamaño de la ventana, en esta fase, es menor que en TCP Tahoe y TCP Reno que aumenta su tamaño cada vez que reciben un ACK. Contrariamente, si se recibe un ACK y el RTT es idéntico al promedio, la ventana se mantiene sin cambios.

Con el crecimiento en la demanda de movilidad, el protocolo TCP se comenzó a utilizar sobre enlaces inalámbricos. Los enlaces inalámbricos tienen características de transmisión muy diferentes a los enlaces tradicionales cableados. Estas características propias tienen su origen tanto en la naturaleza del medio físico como en los efectos debidos

a movilidad de los host. Un host es móvil cuando puede moverse libremente alrededor de un área y permite a los usuarios acceder a los datos en cualquier momento y en cualquier lugar, sin diferenciar la operación ni la performance entre la movilidad y un host fijo. De esta manera, los retrasos y las pérdidas de paquetes en las redes inalámbricas, ya no se deben exclusivamente a la congestión, sino también a la pérdida de paquetes por daño en tránsito y la desconexión temporal y frecuente de los nodos, lo que degrada el rendimiento del TCP. El comportamiento de los enlaces inalámbricos está limitado por las frecuentes ráfagas de error causadas por los desvanecimientos, atenuaciones e interferencias. Durante los periodos de ráfaga, todos los intentos de transmisión de datos resultan fallidos con muy alta probabilidad. Durante las desconexiones frecuentes, producto de la misma movilidad, se produce un intervalo de tiempo en el cual no se puede enviar ni recibir paquetes lo que produce aumento en la latencia. Las pérdidas en enlaces inalámbricos se deben principalmente a la pérdida de paquetes en tránsito y no a la congestión. Dado que el protocolo TCP supone que las pérdidas se deben a la congestión, en cada uno de estos casos activara el algoritmo de control de congestión pudiendo ocasionar un comportamiento muy poco eficiente del protocolo en ese entorno [9].

Uno de los esquemas interesantes para diferenciar la causa de la pérdida de los paquetes, se basa en la coherencia de la congestión entre paquetes consecutivos,

llamado Congestion Coherence [10,11]. Se basa en la idea que la congestión no ocurre repentinamente, es decir, ya hay congestión antes de que se torne severa y se descarte un paquete y tampoco desaparece repentinamente luego de que un segmento es desechado.

Otro enfoque proactivo, pensado para evitar la congestión, es el algoritmo denominado RED (Random Early Detection). Mientras que los algoritmos tradicionales descartan paquetes a partir de un buffer lleno, RED utiliza un enfoque probabilístico para el descarte de paquetes de ese buffer. Este procedimiento de descarte de paquetes se realiza antes de producirse un período de alta congestión. El cálculo lo realiza en base al tamaño promedio de la cola, el tamaño del paquete, la cantidad de paquetes desde el último descarte y valores umbrales de referencia máximo y mínimo.

Líneas de Investigación, Desarrollo e Innovación

El protocolo TCP es claramente el más utilizado en sistemas de transmisión sobre redes de datos. Sin embargo la evolución de los sistemas de comunicaciones ha generado una serie de variantes en las cuales se mezclan sistemas cableados con wireless, o sencillamente redes wireless con nodos de mucha movilidad [12]. Esta combinación a dado origen a nuevos problemas basados fundamentalmente en la enorme diferencia de tasa de errores que normalmente estos dos sistemas tienen, más los problemas típicos de desconexión de las redes wireless. Estos problemas han puesto en un verdadero aprieto a la versión original del protocolo TCP, el cual ha sido modificado, generado

distintas versiones que tienden a dotar de estabilidad a este protocolo bajo las distintas variantes de conexionado. En este contexto, diferentes grupos de investigación siguen proponiendo distintas modificaciones o técnicas que tienen a mejorar las condiciones de comunicación. Sin embargo es muy dificultoso llegar a una solución general debido a la heterogeneidad de las topologías, las variantes de los sistemas de comunicación y particularmente los diferentes requerimientos que existen sobre los distintos tipos de redes de datos.

Resultados y Objetivos

Si bien el presente proyecto aún no ha registrado un grado de avance que permita mostrar resultados, el análisis de las soluciones existentes posibilita inferir que las constantes modificaciones del protocolo TCP para mejorar su rendimiento en medios wireless, plantea la seria posibilidad de desarrollar un nuevo protocolo que ataque de cero la problemática. En tal sentido las soluciones existentes del protocolo TCP que contempla una reacción proactiva, parece ser la más adecuada. Por ello las soluciones propuestas por protocolos como el TCP Vegas y RED (Random Early Detection), que reaccionan variando la ventana de congestión en función de prever el estado de congestión de la red analizando el RTT de cada paquete o variando el descarte de tramas en función del estado del buffer de retransmisión, parecen ser un buen punto de partida en desarrollos que se adapten a las exigentes demandas de las redes wireless.

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 Ingeniera en Sistemas de Información, Licenciados en Informática, Ingeniero en Telecomunicaciones, Ingeniero en Electrónica y un Ingeniero Electricista con un máster en Sistemas y Redes de Comunicaciones. También integra el grupo un alumno 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 Computer Communication Review, Vol. 25, No. 1, 1995, pp. 157-187.
- [4] Allman M., Paxson V. and Blanton E., "TCP Congestion Control", IETF, Standards Track RFC 5681, Sept. 2009.
- [5] Stevens, W., "TCP slow start, congestion avoidance, fast retransmit, and fast re-covery algorithms". RFC 2001, 1997.
- [6] Jacobson V., "Congestion Avoidance and Control", ACM SIGCOMM Computer Communication Review, 25(1), 1995, pp. 157-187.
- [7] Karn, H. & Craig P., "Improving round-trip time estimates in reliable transport protocols", ACM Transactions on Computer Systems (TOCS) 9 (4), 364-373, 1991.
- [8] 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.
- [9] Elaarag, H., "Improving TCP Performance over Mobile Networks", ACM Computing Surveys, Vol. 34, No. 3, September 2002.
- [10] C. Liu, Raj Jain, "Approaches of Wireless TCP Enhancement and A New Proposal Based on Congestion Coherence", 36th Hawaii International Conference on System Sciences. 2002.
- [11] C. Liu, Raj Jain, "Congestion Coherence: A Local Enhancement for TCP over Wireless", OSU Technical Report, November 2004.
- [12] Ho, L., "An investigation of improving the traditional TCP over MANETs". Thesis Master of Computer and Information Sciences, Auckland University of Technology, 2012.