



Micro Transport Protocol (μ TP)

Redes de Computadores I - ELO322
1er Semestre 2017

Integrantes:

José Rojel
Rodolfo Jaramillo

28 de Agosto, 2017

1. Resumen

BitTorrent Inc. innovó el 2009 con un nuevo protocolo de transporte llamado *Micro Transport Protocol* (μ TP) para asegurar la escalabilidad de sus aplicaciones p2p. μ TP es un protocolo de código abierto basado en UDP, con la particularidad de tener un control de congestión. Su implementación no estuvo exento de polémica, sin embargo mostró su eficacia para convertirse en el protocolo de transporte más utilizado en la actualidad por los clientes BitTorrent.

Se expone al protocolo en acción, detallando cómo se lleva a cabo el control de congestión y las diferencias que tiene con los protocolos de transporte de uso común (TCP y UDP).

2. Introducción

Las tecnologías *peer to peer* (p2p) hacen referencia a un tipo de arquitectura para la comunicación entre aplicaciones que permite a individuos comunicarse y compartir información con otros individuos sin necesidad de un servidor central que facilite la comunicación.

Una de las aplicaciones de uso más extendido son los clientes BitTorrent, utilizados para compartir grandes archivos. Se estimó el 2009 que utilizaban un 40-50% del tráfico global de internet con una tendencia al aumento. Por tal motivo se decidió dejar atrás el protocolo TCP en la capa de transporte e innovar con un nuevo protocolo de código abierto basado en UDP llamado *Micro Transport Protocol* (μ TP) para asegurar la escalabilidad de sus aplicaciones.

3. Motivación

¿Por qué crear un protocolo de transporte para aplicaciones p2p que compartan grandes archivos?

La motivación principal para la creación de un protocolo de transporte distinto a TCP es usar el ancho de banda completo cuando no haya interferencias, teniendo cuidado de no competir con procesos o aplicaciones paralelas que sean sensitivas al retardo o reducción de su ancho de banda (“foreground app’s”).

Cuando un usuario decide usar algún cliente Torrent, por lo general lo utiliza para la subida o descarga de archivos de gran tamaño, es por eso que este proceso pasa a segundo plano en el uso cotidiano del tráfico de internet de un usuario, teniendo menos prioridad que revisar correos, ver algún streaming o revisar páginas web. Cuando se utiliza TCP en estas aplicaciones de “background” que mueven grandes archivos, los paquetes llenan rápidamente los buffers, agregando considerable tiempo de retardo a todo el tráfico paralelo.

μ TP resuelve estos problemas del uso compartido usando un retardo fijo como objetivo o **TARGET**, de modo que cuando es sobrepasado reduce su tasa de envío y aumenta más lento cuando se acerca al objetivo. De ese modo permite utilizar la capacidad completa de envío cuando no hay competición y pone un freno en su throughput cuando hay mucho tráfico.

4. Características de μ TP

μ TP es un protocolo de código abierto basado en UDP, el cual es un protocolo de transporte de “best effort” no orientado a la conexión y sin control de congestión. Todo

intento de mejora en esos aspectos UDP le traspasa la responsabilidad a la aplicación y eso fué precisamente lo que hizo BitTorrent, implementó un control de congestión sobre UDP ganando todas las ventajas que ello implica (reducción de retardo y no competir con TCP). Del mismo modo que TCP, μ TP verifica la integridad de los datos con números de secuencia pero inicia la conexión con solo un handshake (en vez de tres).

La principal diferencia de μ TP con TCP es el tipo de control de congestión que implementan, μ TP utiliza un control basado en el retardo en una dirección, intenta estimar con *time-stamps* en sus paquetes cuánto demora el paquete en llegar al receptor y ver cómo varía este valor para tomar una decisión. Al igual que TCP, μ TP implementa una ventana de envío similar a **CongWin** la cual será modificada según la diferencia entre el retardo y el **TARGET** [1].

5. Low Extra Delay Background Transport (LEDBAT)

LEDBAT es el novedoso algoritmo de control de congestión que se implementa en μ TP, está basado en la estimación del retardo en una dirección a modo de reaccionar de forma temprana a las congestiones de la red, de forma que permite a las aplicaciones de “background” usar plena la capacidad de la red sin entrar en interferencia con flujos paralelos, minimizando el retardo que provocan sus aplicaciones [2].

A continuación se presenta de forma simplificada el modo de operación del algoritmo:

- **Lado del emisor:** Inicialización

```
base_delay = +INFINITY
```

El emisor setea la variable `base_delay` la cual se irá actualizando a medida que

responde el receptor

- **Lado del receptor:** Recibido el paquete, el receptor calcula la diferencia entre el timestamp del paquete con su hora local `local_time()`

```
remote_timestamp = data_packet.timestamp
ack.delay = local_time() - remote_timestamp
acknowledgement.send()
```

Envía el ACK de vuelta con el la estampa `ack.delay`.

- **Lado del emisor:**

```
current_delay = ack.delay
base_delay = min(base_delay, current_delay)
queuing_delay = current_delay - base_delay
off_target = (TARGET - queuing_delay) / TARGET
cwnd += GAIN * off_target * bytes_newly_acked * MSS / cwnd
```

Recibido el ACK, estima cuanto fué la variación entre el ACK actual y uno anterior. Si hay un aumento significa que el retardo está aumentando conforme se envían paquetes, esto se traduce en que el factor `off_target` será menor a medida que `queuing_delay` se aproxima al `TARGET`. Nótese que si el retardo en algún momento supera al `TARGET`, el cual se setea comúnmente a 100[ms], `off_target` será negativo traduciéndose en una reducción de la ventana `cwnd` proporcional a la variación.

6. Polémica y desventajas

El 2008 cuando BitTorrent anunció su retirada del uso de TCP a una variación de UDP se encendieron las alarmas pues UDP no posee un control de congestión y se pensó que colapsaría la red por esta drástica decisión. Recordemos que en ese entonces se estimaba que el tráfico de Torrent era cerca de la mitad del tráfico global, sin

embargo, como muchas de las polémicas que surgen hoy en día, sólo se trataba de una especulación desinformada pues desde el primer minuto el cambio al nuevo protocolo estaba respaldado con un control de congestión que detallamos anteriormente [3].

Desventajas de μ TP

Una de las premisas que asume el protocolo estudiado, la cual asegura un correcto funcionamiento es que el receptor entrega un confiable feedback al emisor, de otro modo se deteriora la tasa de envío o se propicia la corrupción de los datos. A través de ataques simulados se comprobó que un receptor intervenido puede incrementar la tasa de envío del emisor hasta 5 veces, esto puede provocar un colapso y desperdiciar gran parte del ancho de banda del usuario [4].

7. Conclusiones

- μ TP es uno de los protocolos más usados hoy en día en los clientes Torrent, el cual desde su implementación, en pleno auge del file-sharing, a visto mermado su uso por las políticas anti-piratería que se llevan a cabo alrededor del mundo. Sin embargo se demuestra con este caso cómo vasta que se junte un grupo de innovadores con una visión conjunta para sacar adelante soluciones que cambian la manera tradicional de hacer las cosas.

- En la investigación realizada para llevar a cabo este trabajo pudimos percatarnos que aun hay mucho espacio para el desarrollo de nuevas tecnologías y métodos en lo que la red respecta, notamos que hasta el día de hoy tanto μ TP como el algoritmo LEDBAT siguen estando en la categoría de experimentales con mucho espacio de crecimiento, pero también con la esperanzadora vulnerabilidad de que el día de mañana sean reemplazados con alguna novedosa herramienta.

Referencias

- [1] Arvid Norberg, «*uTorrent transport protocol*». 2009
http://www.bittorrent.org/beps/bep_0029.html
- [2] S. Shalunov, G. Hazel, BitTorrent, Inc. «*RFC 6817 - Low Extra Delay Background Transport (LEDBAT)*» 2012. <https://tools.ietf.org/pdf/rfc6817.pdf>
- [3] R. Bennett. «*BitTorrent declares War on VoIP, Gamers - The Next Internet Meltdown*» 2008.
http://www.theregister.co.uk/2008/12/01/richard_bennett_utorrent_udp
- [4] Florian Adamsky, Syed Ali Khayam (2012) «*Security Analysis of the Micro Transport Protocol with a Misbehaving Receiver.*»
<https://florian.adamsky.it/research/publications/2012/PID2441647.pdf>