

Entrega 1 Proyecto

Guillermo Ribero, Juan Luis Ávila, Juan José Caballero

Escuela de Ingeniería, Ciencia y Tecnología, Universidad del Rosario

Octubre 2021

1. Reporte ejecutivo

En el funcionamiento la arquitectura que conforma Internet es importante conocer el estado de congestión de la red, especialmente en la capa de transporte y de red. El protocolo TCP, utiliza una medida llamada 'Sample RTT', que mide el tiempo que transcurre desde que se envía un segmento hasta que se recibe una notificación de que ya fue entregado. Ya que simultáneamente se suelen enviar una significativa cantidad de segmentos, TCP mide el 'Sample RTT' de sólo un segmento en cada unidad de tiempo. Es importante destacar, que el valor del 'Sample RTT' fluctúa constantemente debido al tráfico en la red y los tiempos de carga, por lo que es probable que algunas veces esta estimación sea atípica por lo que se intenta tener una medida que pueda estimar con menor variación la congestión en la red (conocida como 'Estimated RTT'. Una de las maneras que existen para calcular el 'Estimated RTT', se conoce como 'exponential weighted moving average' (EWMA) y su fórmula es :

$$EstimatedRTT = (1 - \alpha)EstimatedRTT + \alpha SampleRTT$$

donde α es un valor entre 0 y 1, aunque se recomienda utilizar esta variable como 0.125. Anexado en esta carpeta, se encuentra un código de 'Python' amigable con el usuario que permite graficar los valores de 'Sample RTT' y 'Estimated RTT', dado un archivo csv con medidad del 'Sample RTT' en cada instante de tiempo o existe la opción de que estos sean calculados aleatoriamente. Además se puede cambiar el valor de α , mediante una barra deslizable.

2. Preguntas

- a **¿Qué pasa con EstimatedRTT cuando varía Alpha?** El valor estimado de RTT el cual se actualiza por cada dato nuevo de la muestra va a tener un peso respecto a los datos viejos, este peso es la influencia que tiene el dato nuevo sobre el resultado final y esta dado por el valor de alfa que es entre 0 y 1.

Si alfa de cerca a 0 hace que cada dato nuevo tenga poco impacto en el resultado final. La desventaja de un alfa pequeño es que no tenemos una visión clara de cual debe ser el RTT según la congestión actual en la red (esto para intervalos grandes de tiempo). Pero si el intervalo es muy corto, entonces será mejor calcular el valor estimado de RTT con un alfa mas grande.

Si el alfa cerca a 1, el peso de cada dato nuevo será mayor a los anteriores, lo que en intervalos largos de tiempo reflejará correctamente el estado de la congestión en red.

- b **¿Qué pasa cuando se utiliza otra metodología? (diferente a EWMA)**

EWMA, permite darle un valor mayor a los valores más recientes de una manera proporcionada ya que estos describen mejor el estado actual de la red. Las otras formas de encontrar el RTT estimado, asignan los pesos correspondientes a cada dato nuevo de distintas formas como: Define un peso por defecto, multiplicar por una constante de peso, los primeros con un mayor peso o ignorar la asignación de peso. Por lo tanto la desventaja de usar estos métodos en vez de EWMA es que no nos dan datos sobre como se comporta actualmente la red, siendo así más favorable usar EWMA.

c **¿Podría el usuario calcular el intervalo sugerido de timeout con su aplicación?**

No, pues lo único que podría saber con el valor estimado de RTT es que el timeout debe ser mayor, y cuan mayor se resuelve con esta formula:

$$TimeoutInterval = EstimatedRTT + 4.DevRTT$$

la cual requiere de la variacion entre RTT's. DevRTT se define como el valor que define que tanto varia el SampleRTT con respecto al EstimatedRTT, lo cual se calcula con la siguiente ecuacion

$$DevRTT = (1 - \beta).DevRTT + \beta.|SampleRTT - EstimatedRTT|$$

Sin embargo, si solo tenemos el EstimatedRTT se podria usar como un estimador del timeout, el cual aunque no nos de un valor exacto nos daría una idea de el valor del timeout. Esto siguiendo la logica de que un tiempo de 3 o 4 veces el EstimatedRTT es algo muy elevado para los parametros del sistema segun el caso, por lo cual podrian ser probables rangos de timeout.

3. Repositorio

<https://github.com/BufGRbrain/ProyectoRedes>

4. Bibliografia

- 1 James F., K. A. K. R. W. (2017). Computer Networking: A Top-Down Approach, 6Th Edn (6th ed.). Pearson India.
- 2 MetaTrader 5. (s. f.). Moving Average. Recuperado 6 de octubre de 2021, de <https://www.metatrader5.com/en/terminology>