**Estimación del temporizador de retransmisión**

**(RTO, retransmission timeout) en TCP**

**1. ¿El proceso de estimación (EstimatedRTT) observa un comportamiento suave con**

**respecto a las muestras RTT (sampleRTT)? Explique.**

Las muestras “sampleRTT” en algunos casos hay unos picos altos y en otros no hay una discontinuidad en los valores no hay tantos picos. Entonces en el proceso de estimación si tiene un comportamiento suave con las primeras 30 muestras con respecto a “sampleRTT”, se puede observar que no hay mucha discontinuidad.

**2. Aplique las expresiones de Van Jacobson para calcular el tiempo de espera (TimeoutInterval) de una fuente TCP con los valores de α y de β por defecto. ¿Son estos valores adecuados para el timeout en los cuatro procesos RTT que está estudiando? En caso de que no, ¿necesitarían α y β tener valores distintos? Explique.**

Si, ya que Los valores de α y de β para calcular el “timeoutInterval” son recomendados por el estándar RFC 6298.

Si se puede tener valores distintos de α y de β, pero no se asegura que se obtenga unos buenos resultados ya que sugieren que se realicen las pruebas con α = 0.125 y β = 0.25.

**3. Cuando se pierde un segmento o un ACK, ¿cómo se calcula el temporizador de timeout en TCP?**

Cuando TCP transmite un segmento con datos, coloca una copia en la cola de retransmisión e inicializa un temporizador. Al recibir el reconocimiento (ACK) para él, TCP lo borra de la cola. Sino se llega a recibir el ACK antes de que el temporizador expire, el segmento es retransmitido. Si no se recibe el ACK la primera retransmisión ocurrirá entre 1 y 1.5 segundos. Las siguientes retransmisiones se duplicará el tiempo hasta 64 segundos y las retransmisiones finalizaran después de N intentos.

**Gráficas**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**Conclusión**

Con los valores de α y β que nos sugiere el RFC 6298 que son α = 0.125 y β = 0.25 son a veces son buenos para calcular el RTO (retransmission timeout) ya que si tienen un error más alto que las demás trazas en este caso es la traza 23 y 27 que tiene alto su error, por otra parte si tomo los valor de α y β mayor que los de que son por defecto obtengo un error mucho menor como se muestra en la traza 24 en cambio si tomo los valores de α y β menor que los que vienen por defecto con en la traza 25 su error es mucho mayor a todas las trazas, por lo tanto α y β pueden variar sus valores para el cálculo de RTO.

**Código**

**function abs(v) {return v < 0 ? -v : v}**

**BEGIN {**

**alpha = 0.125;**

**beta = 0.25;**

**k = 4**

**}**

**FNR == 1 {**

**sampleRTT = $1;**

**estimatedRTT = sampleRTT;**

**devRTT = sampleRTT/2;**

**timeoutInterval = estimatedRTT + (k\*devRTT)**

**}**

**FNR > 1 {**

**sampleRTT = $1;**

**devRTT = (1-beta)\*(devRTT)+(beta)\*(abs(sampleRTT-estimatedRTT));**

**estimatedRTT = ((1-alpha)\*estimatedRTT)\*(alpha\*sampleRTT);**

**timeoutInterval = estimatedRTT + (k\*devRTT)**

**} ; {print \t\t,estimatedRTT, timeoutInterval}**