Redes de computadoras, 5 de diciembre de 2024

Estimación del temporizador de retransmisión (*RTO, retransmission timeout*) en TCP

Martínez Buenrostro Jorge Rafael

correo, molap96@gmail.com
Universidad Autónoma Metropolitana
Unidad Iztapalapa, México

Introduccion

El protoclo de transmisión (TCP) estima el proceso del RTT para predecir el tiempo de espera (timeout) de la fuente, a fin de ajustar el temporizador de retransmisión. El emisor TCP mide el RTT desde el momento que se envía un segmento hasta recibir el acuse de recibo (ACK) correspondiente

Procedimiento

Para poder comenzar con esta práctica selecione cuatro trazas al azar de los sets proporcionados, dichas trazas son:

SetA - hop02

SetD - hop04

SetG - hop21

SetF - hop29

El siguiente paso es crear un script en AWK para generar las trazas del proceso RTT (*sampleRTT*), su estimación (*estimatedRTT*) y el valor del temporizador (*TimeoutInterval*). La base para este script son las secciones 2.1 a 2.3 del **RFC 6298**. En la figura siguiente se puede ver el código del script

```
BEGIN {
    # Inicializacion de parametros
    alpha = 1/8   # Suavizado para SRTT
    beta = 1/4    # Suavizado para RTTVAR
    K = 4         # Factor para RTTVAR
    RTO = 1       # Valor inicial de RTO
    firstRTT = 1  # Bandera para la primera medicion RTT
    sample_count = 0 # Contador de muestras procesadas
    start_sample = 200 # Comenzar a partir de la muestra 200
    max_samples = 30   # Tomar solo 30 muestras
}

{
    # Incrementar el contador de muestras
    sample_count++

    # Solo procesar muestras a partir de la muestra 200
    if (sample_count >= start_sample && sample_count < start_sample +
        max_samples) {
        # RTT_value es el unico valor por linea
        RTT = $1

        # Primer RTT
        if (firstRTT == 1) {
            SRTT = RTT
            RTTVAR = RTT / 2
            RTO = SRTT + K * RTTVAR
            firstRTT = 0
        } else {
            # RTT subsecuentes
            RTTVAR = (1 - beta) * RTTVAR + beta * (SRTT > RTT ? SRTT - RTT :
                RTT - SRTT)
            SRTT = (1 - alpha) * SRTT + alpha * RTT
            RTO = SRTT + K * RTTVAR
        }

        # Escribir en los archivos de salida
        print RTT >> "sampleRTT"
        print SRTT >> "estimatedRTT"
        print RTO >> "timeoutInterval"
    }

    # Si ya se procesaron 30 muestras, terminamos el script
    if (sample_count >= start_sample + max_samples) {
        exit
    }
}

END {
    print "Proceso completado. Los archivos de salida son: sampleRTT,
        EstimatedRTT, TimeoutInterval."
}
```

Figura 1: Script para extraer los datos requeridos

Una vez creado lo ejecutamos para cada una de las trazas seleccionadas, a continuación se muestra la forma de ejecución

```
awk -f rfc6298.awk hop02.txt
```

Figura 2: Ejecución del script para la traza *hop02.txt*

Como se puede ver en la *Figura 1* al ejecutar el script se generan tres trazas: **sampleRTT**, **estimatedRTT** y **timeoutInterval**. El siguiente paso es crear las instrucciones en *Octave* para poder visualizar en una sola gráfica: la traza original y las trazas generadas por el script. A continuación se muestran las instrucciones propuestas, a reserva del nombre de los ejes y el titulo que cambiará al graficar cada una de las trazas. Además veremos las gráficas creadas por defecto.

```
load estimatedRTT;
load timeoutInterval;
plot(sampleRTT, '--*', 'Color', 'b', 'LineWidth', 0.5, 'MarkerSize', 8);
hold on;
plot(estimatedRTT, '--*', 'Color', 'k', 'LineWidth', 0.5, 'MarkerSize', 8);
plot(timeoutInterval, '--*', 'Color', 'r', 'LineWidth', 0.5, 'MarkerSize', 8)
    ;
legend('sampleRTT','estimatedRTT','timeoutInterval');
grid on;
xlabel('Eje x');
ylabel('Eje y');
title('Erro incurrido');
print -dpng "traza.png";
```

Figura 3: Ejecución del script para la traza *hop02.txt*

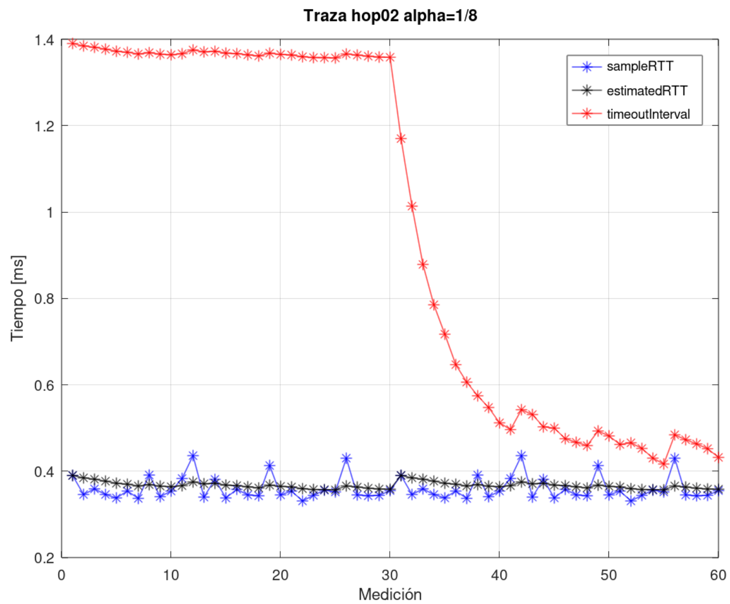

Figura 4: Gráfica de las trazas de las muetras hop02

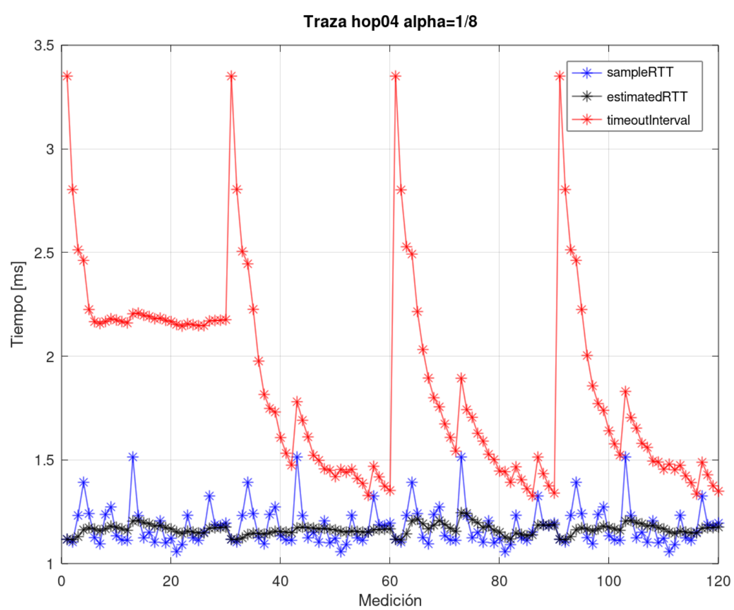

Figura 5: Gráfica de las trazas de las muetras hop04

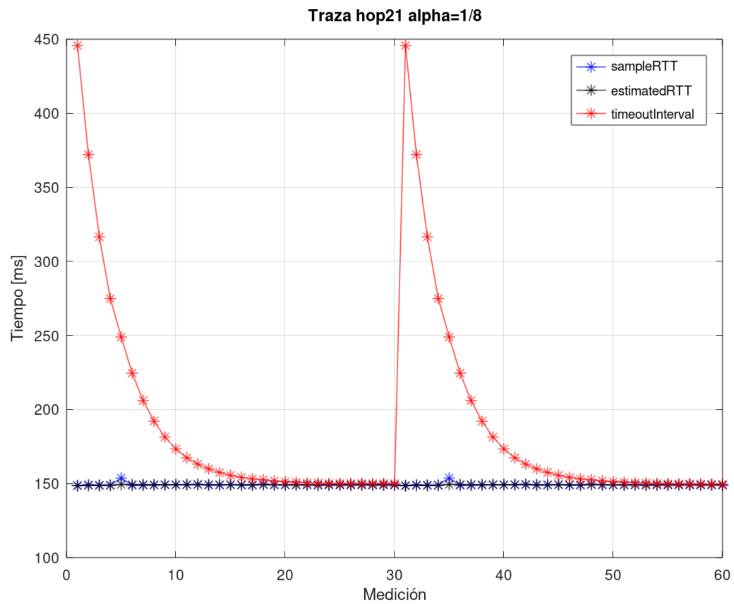

Figura 6: Gráfica de las trazas de las muetras hop21

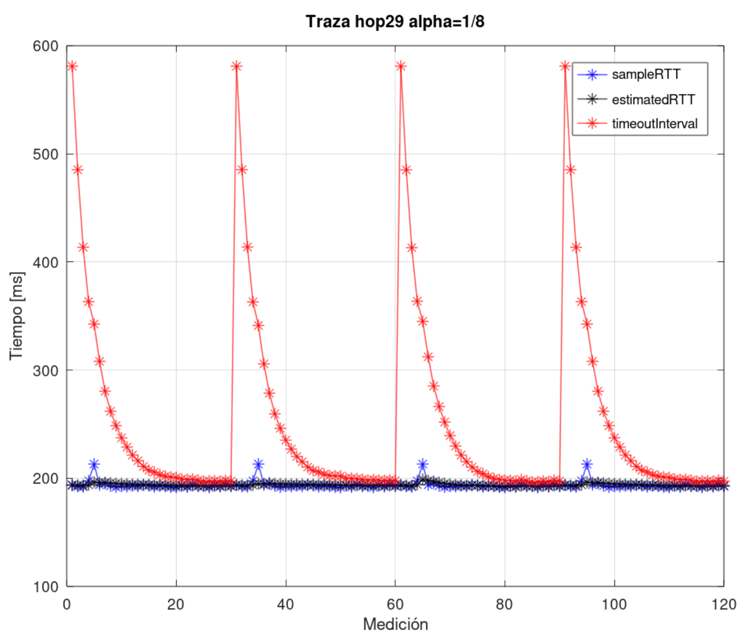

Figura 7: Gráfica de las trazas de las muetras hop29

Cuestionario

1. ¿Se observa un proceso suave en el proceso de estimación ($EstimatedRTT$) con respecto a las muestras del RTT ($SampleRTT$) tal como propone Jacobson?

   Si, el proceso de estimación de $EstimatedRTT$ debería ser suave con respectoa a las muestras de $SampleRTT$, como lo propone el algoritmo de Jacobson

2. Las fórmulas de Van Jacobson para el cálculo del RTO son las siguientes:

$$EstimatedRTT = (1 - \alpha) * EstimatedRTT + \alpha * R'$$

$$VariacionRTT = (1 - \beta) * VariacionRTT + \beta * |EstimatedRTT - R'|$$

$$TimeoutInterval = EstimatedRTT + max(G, K * VariacionRTT)$$

   Los valores por defecto son $\alpha = 1/8$ y $\beta = 1/4$ son ampliamente utilizados en la práctica debido a que equilibran adecuadamente la estabilidad de la estimación de RTT y la sensibilidad a las fluctuaciones de la red.

   α controla que tan rápido se adapra el $EstimatedRTT$ a nuevos valores de $SampleRTT$. Si el RTT flutúa constantemente debido a cambios rápidos en la red, es posible que se quiera una estimación más precisa con menor error. α podría ser ajustado ligeramente a un valor más bajo, por ejemplo, $\alpha = 1/16$ para hacer la estimación más conservadora y menos sensible a fluctuaciones grandes, pero esto hará que el $EstimatedRTT$ sea menos reactivo.

Entregables

Cálculo del Error Cuadrático Medio (ECM)

El error cuadrático medio es una métrica comúnmente utilizada para evaluar la diferencia entre valores predichos y valores reales. La fórmula del error cuadrático es:

$$ECM = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$$

donde

- N es el número total de muestras
- x_i es el valor real de $sampleRTT$ en la i-ésima medición
- $\hat{x}_i$ es el valor estimado de $estimatedRTT$ en la i-ésima medición

Este valor se utiliza para comparar los diferentes valores de α en el proceso de estimación.

Implementación de la Estimación de RTT

Para este análisis, se emplea el algoritmo de Jacobson para estimar el EstimatedRTT y el RTO a partir de las muestras de SampleRTT. Se emplean tres valores diferentes de α para estudiar su influencia en el cálculo del EstimatedRTT, a saber:

- $\alpha = \frac{1}{8}$ (valor por defecto)
- $\alpha_1 = \frac{1}{4}$
- $\alpha_2 = \frac{1}{16}$

Análisis de Resultados

Se realizaron tres simulaciones para cada valor de α, y se calcularon los valores de SampleRTT, EstimatedRTT y RTO para cada medición. A continuación, se presenta el error cuadrático medio (ECM) para cada uno de los valores de α

$$\text{MSE}(\alpha) = \frac{1}{n}\sum_{i=1}^{n}(\text{SampleRTT}_i - \text{EstimatedRTT}_i)^2 \tag{1}$$

Gráficas de la Estimación de RTT

A continuación se presentan las gráficas de los tres procesos: SampleRTT, EstimatedRTT y TimeoutInterval, para cada valor de α seleccionado. El eje x representa el número de la medición (número de vuelta), mientras que el eje y muestra los valores de los procesos.

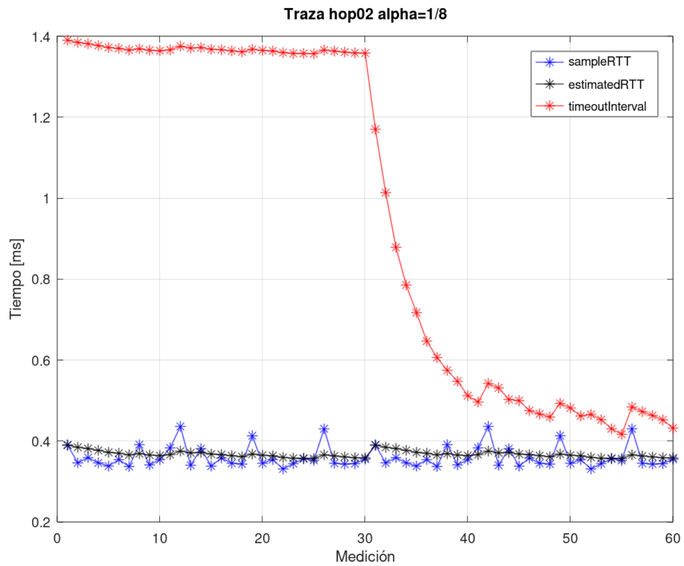

Figura 8: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{8}$ de la muestra hop02

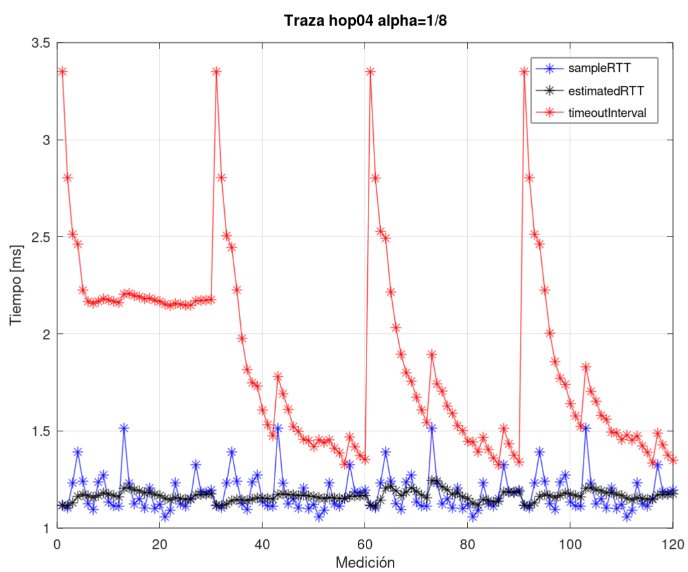

Figura 9: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{8}$ de la muestra hop04

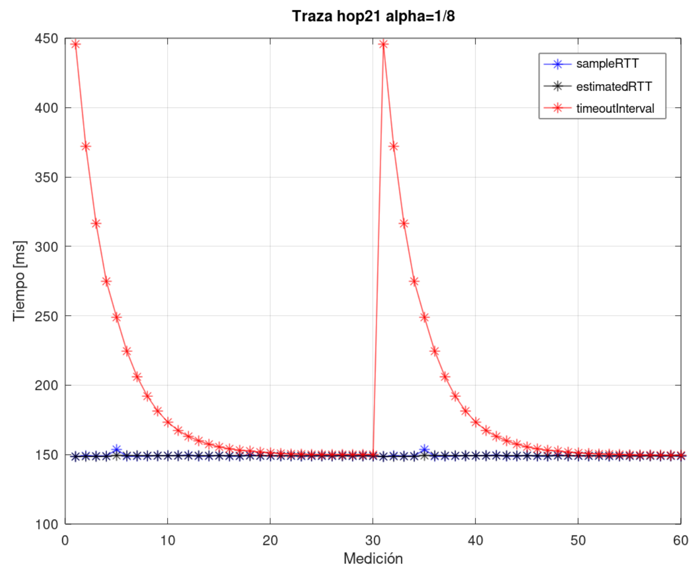

Figura 10: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{8}$ de la muestra hop21

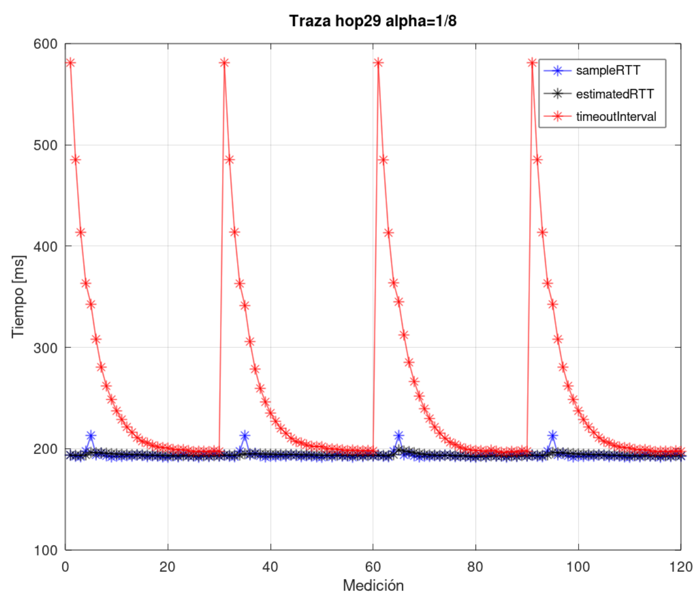

Figura 11: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{8}$ de la muestra hop29

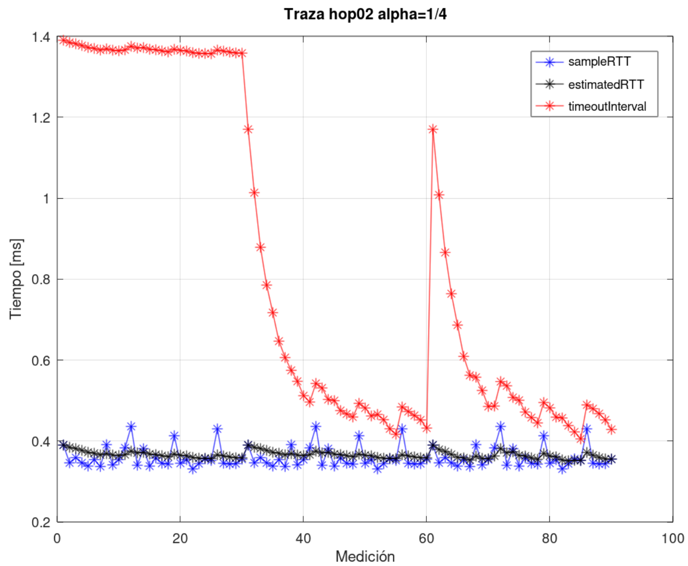

Figura 12: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{4}$ de la muestra hop02

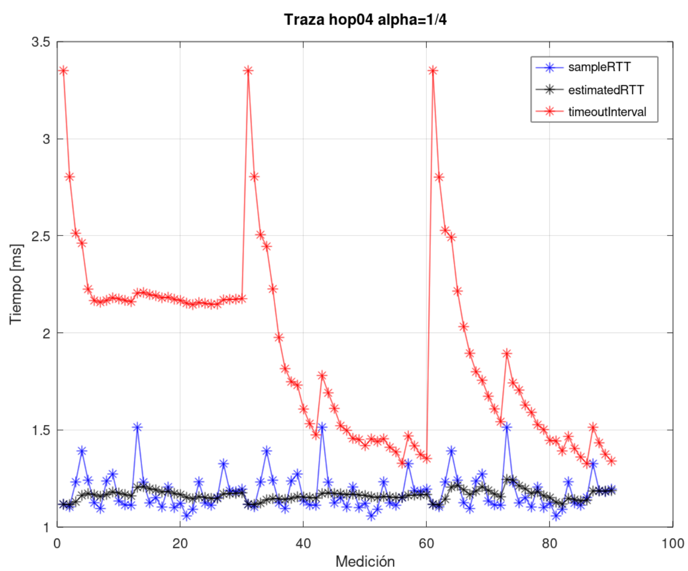

Figura 13: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{4}$ de la muestra hop04

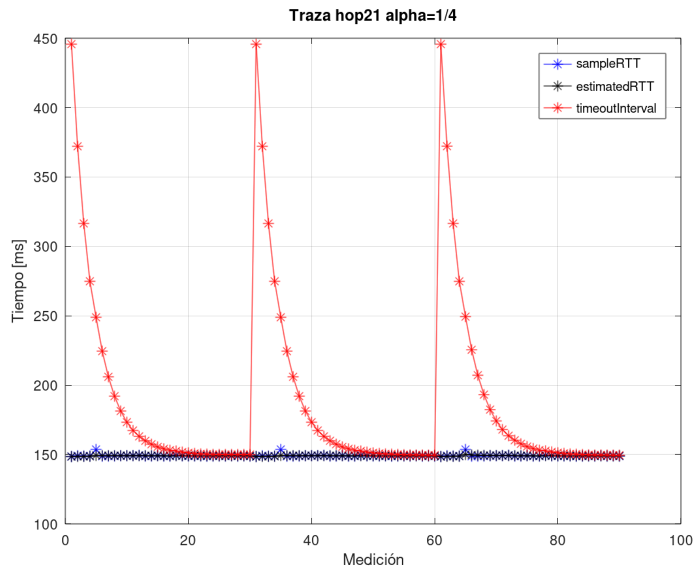

Figura 14: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{4}$ de la muestra hop21

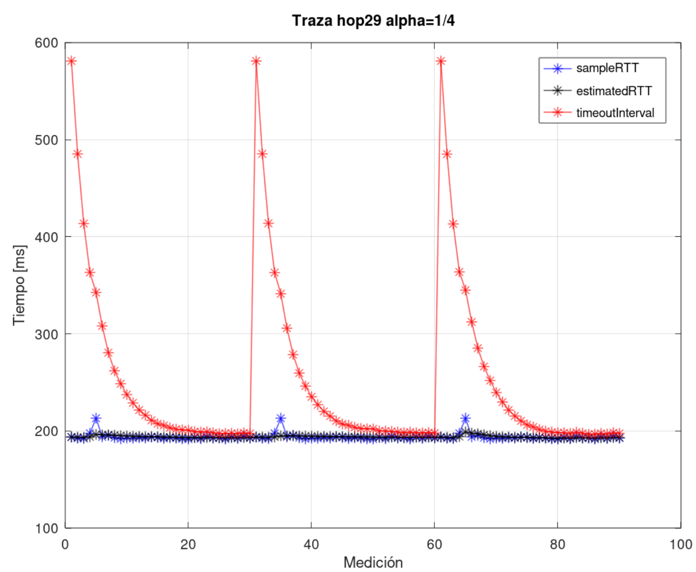

Figura 15: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{4}$ de la muestra hop29

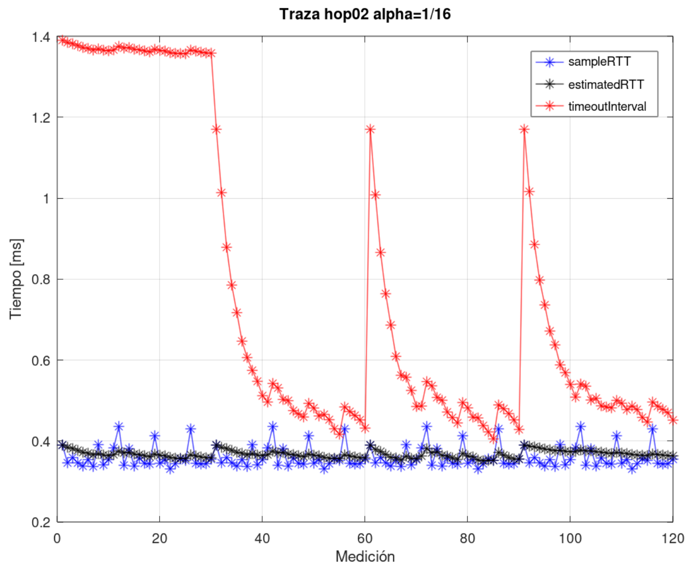

Figura 16: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{16}$ de la muestra hop02

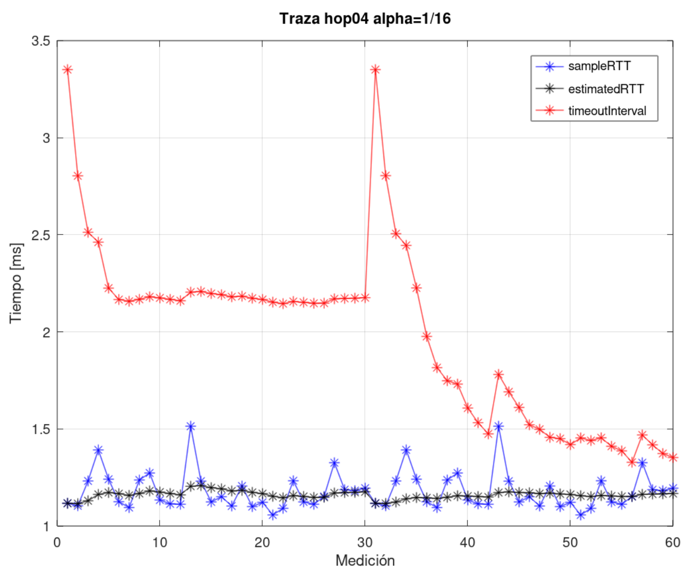

Figura 17: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{16}$ de la muestra hop04

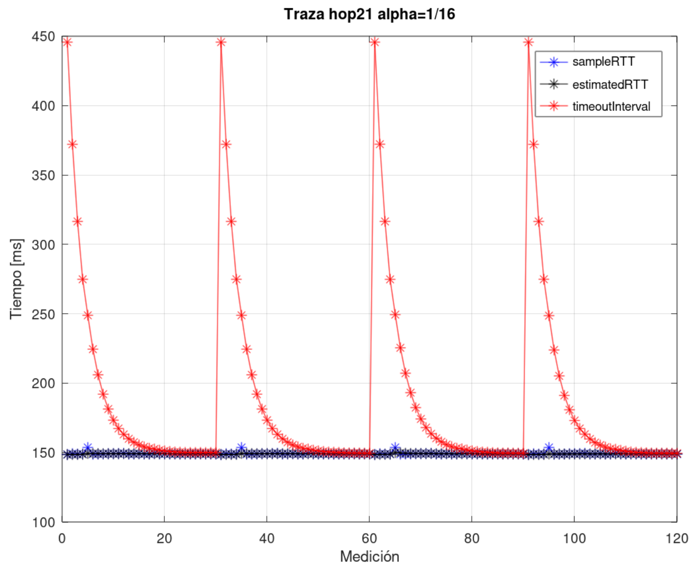

Figura 18: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{16}$ de la muestra hop21

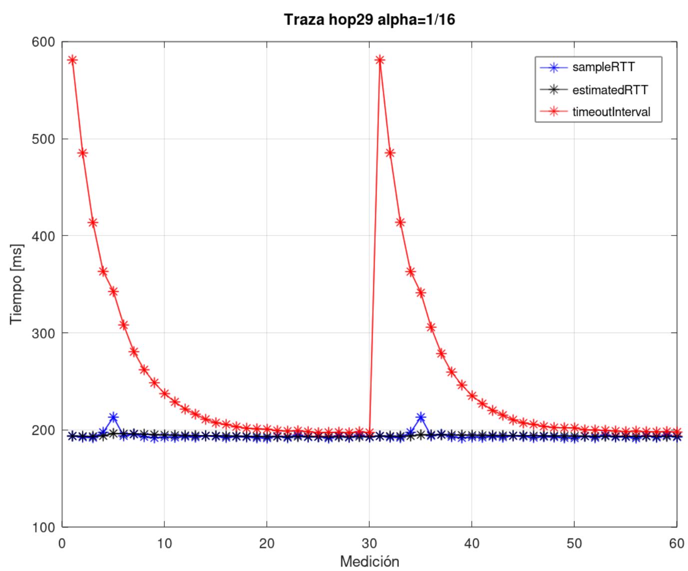

Figura 19: Gráfica de SampleRTT, EstimatedRTT y TimeoutInterval con $\alpha = \frac{1}{16}$ de la muestra hop29

Discusión y Conclusiones

A partir de las gráficas obtenidas y el análisis de los errores cuadráticos medios (ECM) para cada valor de α, podemos concluir que:

- El valor de $\alpha = \frac{1}{8}$ (por defecto) produce una estimación bastante precisa, con un ECM moderado.

- Un valor mayor de α, como $\alpha_1 = \frac{1}{4}$, aumenta la sensibilidad del algoritmo, lo que puede generar un mayor error en entornos con fluctuaciones rápidas en SampleRTT.
- Un valor menor de α, como $\alpha_2 = \frac{1}{16}$, reduce la sensibilidad del algoritmo, lo que puede ser útil en entornos estables, pero puede no reaccionar rápidamente ante cambios importantes en el RTT.