

Evaluación de algoritmos para la estimación del Round-Trip Delay Time y una propuesta de mejora

Daniel Docmac M., Eduardo I. Silva, Agustín J. González*

Departamento de Electrónica, Universidad Técnica Federico Santa María, Avenida España, 1680, Valparaíso, Chile.

Resumen

En la literatura se han propuesto diversos algoritmos para la estimación del RTT (por sus siglas en inglés: Round-Trip Delay Time). Éstos buscan obtener un mejor desempeño que el del algoritmo de estimación adoptado por el estándar TCP (por sus siglas en inglés: Transfer Control Protocol), propuesto en los años ochenta. En general, los resultados de la evaluación del desempeño de las distintas alternativas en la literatura, se presentan en forma gráfica, considerando un único escenario de conexión o a través de simulación. No se cuenta con métricas de evaluación que permitan determinar, cuantitativamente, el rendimiento de los distintos algoritmos o realizar una comparación entre ellos. Lo anterior motiva el presente trabajo en que se estudian e implementan diversos algoritmos de estimación del RTT en la literatura, se define un marco experimental de prueba y se realizan comparaciones usando una métrica y escenarios comunes. Como segunda contribución, se propone un algoritmo adaptativo de estimación del RTT, capaz de reaccionar rápido frente a cambios repentinos en dicha señal y, a la vez, menos sensible a ruidos transitorios que alternativas en la literatura. Copyright © 2013 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.

Palabras Clave: ACUSUM, estimación, filtros adaptativos, filtro de Kalman, PEM, redes de comunicación, Round-Trip Delay Time (RTT), TCP.

1. Introducción

Diariamente se establecen millones de conexiones a través de Internet. Dado que los volúmenes de datos son cada vez mayores, resulta de gran importancia que estas conexiones sean eficientes. El protocolo de comunicación más utilizado es TCP (por sus siglas en inglés: Transfer Control Protocol), el cual ya era usado en el año 2000 en más de un 70 % de las conexiones y, en el año 2006, en un 95 % de las conexiones (Lei et al., 2006; McCreary and Claffy, 2000). Dada la naturaleza de una conexión, es preciso contar con algoritmos para la estimación de ciertos parámetros que determinan el desempeño del protocolo y, por lo tanto, el funcionamiento de la red.

En el caso del protocolo TCP, un parámetro de relevancia para su desempeño es el RTT (por sus siglas en inglés: Round-Trip Delay Time). La mayoría de las implementaciones TCP existentes utilizan el algoritmo de estimación del RTT introducido por Jacobson en 1988 (Jacobson, 1988; Paxson and Allman, 2000; Haeri and A.H.M.Rad, 2004).

Existen diversos trabajos que indican que el uso del algoritmo de Jacobson se traduce en un rendimiento insatisfactorio (Balakrishnan et al., 1998; Ludwig and Sklower, 2000; Seddigh, 2000). Dicho algoritmo se basa en un filtro lineal e invariante en el tiempo. Este tipo de estimadores son óptimos cuando la señal a estimar puede modelarse como un proceso Gaussiano y estacionario (Anderson and Moore, 1979). Sin embargo, la señal RTT no puede ser modelada de forma precisa como un proceso Gaussiano ni estacionario pues, usualmente, presenta un comportamiento impulsivo (Li, 2000). Dado lo anterior, diversos métodos de estimación del RTT han sido propuestos en la literatura. Estos se basan, principalmente, en filtros adaptativos y algoritmos de detección de cambios, lo que permite lidiar con las características no estacionarias del RTT. Así, (Ma et al., 2004) proponen un método de estimación del RTT basado en filtros de medianas ponderadas (vea (Arce and Paredes, 2000; Yin et al., 1996)). Los autores de (Ma et al., 2004) afirman que el algoritmo propuesto es adecuado para conexiones cortas en las cuales se envían pocos paquetes. Sin embargo, dada la descripción del algoritmo y la forma en que ha de implementarse, son necesarias al menos cinco mediciones del RTT antes de poder realizar una predicción. Esto se traduce en que el método propuesto no sea aplicable a conexiones cortas. Un problema similar

* Autor en correspondencia.

Correos electrónicos: ddocmac@gmail.com (Daniel Docmac M.), eduardo.silva@usm.cl (Eduardo I. Silva), agustin.gonzalez@usm.cl (Agustín J. González)

presenta el método propuesto por (Li et al., 2009). Dicho método se basa en un filtro de respuesta a impulso finita (*finite impulse response* (FIR)), cuyos coeficientes son actualizados en función del error de estimación del RTT obtenido en instancias anteriores. De acuerdo a la implementación del algoritmo, éste precisa contar con un número relativamente elevado de mediciones pasadas del RTT para efectuar una predicción. En (Li et al., 2009), el desempeño del estimador es evaluado usando una realización particular del RTT, y los resultados son presentados en forma gráfica, no siendo éstos concluyentes. Los autores de (Li et al., 2009) afirman que el algoritmo de estimación propuesto no logra reaccionar rápidamente frente a cambios significativos en el valor del RTT, los cuales pueden surgir ante la aparición (o desaparición) de tráfico de larga duración en algún punto de la red. Es importante, entonces, que el estimador utilizado reaccione rápidamente ante dichos cambios. Motivados por esto, (Ngwenya and Hancke, 2004) proponen dos métodos de estimación del RTT basados en algoritmos de detección de cambios (vea (Basseville and Nikiforov, 1993; Gustafsson, 2000)). El primero de ellos define un intervalo de confianza (Silvey, 1975) y supone que si el valor del RTT medido se encuentra fuera de éste, entonces la media del proceso de RTT ha cambiado. En estos casos, el valor del RTT estimado se reinicia apropiadamente. El segundo método, se basa en el *log-likelihood ratio test* (Basseville and Nikiforov, 1993), suponiendo una distribución normal para el RTT. Nuevamente, se toman medidas apropiadas si un cambio se detecta en las estadísticas del proceso. Los resultados de (Ngwenya and Hancke, 2004) son presentados en forma gráfica, evaluándose sólo la habilidad de cada algoritmo para seguir rápidamente la media o cambios en el nivel del RTT. Un enfoque similar es el adoptado por Jacobsson et al. (Jacobsson et al., 2004), proponiendo el uso combinado de un filtro de Kalman (Anderson and Moore, 1979), junto al algoritmo de sumas acumulativas para la detección de cambios (CUSUM; (Gustafsson, 2000; Page, 1954)). Sin embargo, los autores de (Jacobsson et al., 2004) suponen que la varianza de los ruidos de proceso y de medición, en un modelo de primer orden para el RTT, son iguales a la varianza del RTT y suponen ésta conocida. Por otra parte, los parámetros del algoritmo CUSUM son considerados constantes en (Jacobsson et al., 2004). Esto afecta el rendimiento del algoritmo en escenarios en que el RTT sufra cambios de magnitudes considerables (Shu and Jiang, 2006). La evaluación del algoritmo de estimación en (Jacobsson et al., 2004), al igual que en (Li et al., 2009; Ngwenya and Hancke, 2004), es hecha utilizando una realización particular del RTT.

La discusión anterior muestra que no se ha evaluado el desempeño de los distintos algoritmos de estimación del RTT en una forma uniforme, que permita llegar a conclusiones significativas. Se hace, por lo tanto, necesario contar con una métrica específica para evaluar el desempeño de los distintos algoritmos de estimación del RTT, utilizando escenarios comunes. Por otra parte, concluimos que no se ha propuesto un algoritmo de estimación del RTT completamente adaptativo, capaz de seguir rápidamente los cambios de media del RTT y, al mismo tiempo, poco sensible a condiciones de ruido transitorio.

En este trabajo, se evalúan diversos algoritmos propuestos en la literatura para la estimación del RTT. Para ello se usa una métrica específica y diversos escenarios de conexión comunes a todos los algoritmos. Asimismo, se presenta un nuevo método de estimación adaptativo, basado en el filtro de Kalman y complementado con un algoritmo adaptativo de sumas acumulativas (ACUSUM; (Sparks, 2000)) para la detección de cambios, y con un algoritmo de minimización del error de predicción para la estimación recursiva de los parámetros en el filtro de Kalman (*prediction error method* (PEM); (Goodwin and Sin, 1984)). El algoritmo propuesto logra un mejor desempeño cuando se le compara con algoritmos de parámetros estáticos.

La organización del artículo es la siguiente: En la Sección 2 se discute la importancia de una correcta estimación del RTT y cómo ésta influye en el rendimiento de una conexión TCP. La Sección 3 presenta algunos de los algoritmos de estimación del RTT propuestos en la literatura. Lo anterior motiva la propuesta del nuevo algoritmo de estimación del RTT presentado en la Sección 4. La Sección 5 define la métrica de evaluación de desempeño y se presentan los resultados obtenidos. Finalmente, en la Sección 6, se presentan conclusiones.

2. Importancia de la estimación del RTT

El Round-Trip Delay Time (RTT) se define como el tiempo transcurrido entre el envío de un paquete y la recepción del acuse de recibo (ACK) correspondiente por parte del receptor (Kurose and Ross, 2010). Por otra parte, se define como Timeout (RTO) al tiempo máximo durante el cual el transmisor espera el arribo de un ACK, antes de considerar que el paquete se ha perdido y re-enviarlo (Kurose and Ross, 2010).

En el contexto descrito más arriba, surge la pregunta de cómo elegir el RTO. Dada la definición del RTT, es claro que el RTO deberá ser, al menos, igual al RTT para evitar la retransmisión innecesaria de paquetes. De no ser así, el transmisor esperará un ACK durante un intervalo menor al que le toma, efectivamente, a este mensaje viajar hasta el receptor. Esto se interpretará erróneamente como la pérdida de un paquete, generándose la retransmisión del mismo, con el consiguiente mal uso de la capacidad del canal. Por otra parte, la elección de un RTO mucho mayor al RTT, se traducirá en un tiempo de espera innecesario luego de la pérdida de un paquete, aumentando así la latencia de la conexión. Lo anterior se ilustra en la Fig. 1, donde se presenta la densidad de probabilidades del RTT para una conexión TCP en particular. En ella se observa cómo la elección de un RTO muy pequeño, digamos T1, producirá retransmisiones innecesarias, mientras que una elección muy holgada, digamos T2, llevará al retardo en el reenvío de un paquete perdido. Por otro lado, la media y varianza de la distribución del RTT puede cambiar rápidamente dentro de unos pocos segundos debido a un aumento o disminución de la congestión en la red (Tanenbaum, 2003). Esto impide fijar un valor para RTO que garantice buen desempeño en todos los escenarios posibles.

En la práctica, el RTO se elige en función de estimaciones del RTT. Un algoritmo popular sugiere utilizar la relación

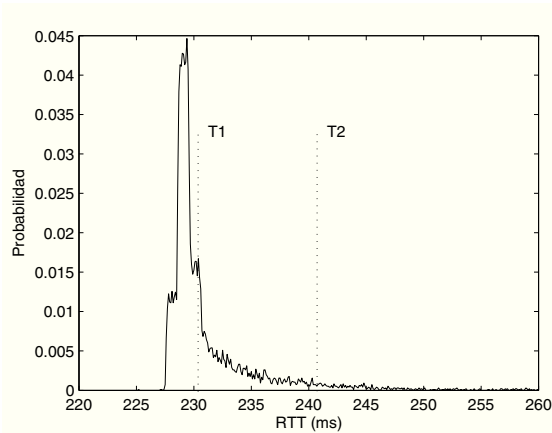


Figura 1: Densidad de probabilidades del tiempo de arribo de acuse de recibo (RTT) en una conexión TCP particular (estimada usando datos reales capturados según se indica en la Sección 5.1).

siguiente (Paxon and Allman, 2000)

$$RTO(i+1) \triangleq \hat{RTT}(i+1) + k\hat{\sigma}(i+1), \quad (1)$$

donde $RTO(i+1)$ corresponde al RTO a utilizar en el instante $i+1$, $\hat{RTT}(i+1)$ corresponden a una estimación del RTT en el instante $i+1$, obtenida en base a mediciones del RTT en instantes anteriores, k es un parámetro (usualmente se elige $k=4$ (Paxon and Allman, 2000)), y $\hat{\sigma}(i+1)$ es una estimación de la varianza del error de estimación $RTT(i+1) - \hat{RTT}(i+1)$. Esta estimación se puede construir de acuerdo a la recursión (Jacobson, 1988)

$$\hat{\sigma}(i+1) \triangleq (1 - \beta)\hat{\sigma}(i) + \beta|RTT(i) - \hat{RTT}(i)|, \quad (2)$$

donde β es un parámetro (típicamente $\beta = 0.25$ (Jacobson, 1988)) y $RTT(i)$ es el RTT medido en el instante i .

Tal como se discutió en la sección introductoria, existen diversos trabajos que muestran que el algoritmo de estimación del RTT utilizado en el protocolo TCP (Paxon and Allman, 2000), es decir, el algoritmo de Jacobson (Jacobson, 1988), alcanza un desempeño insatisfactorio. En una prueba llevada a cabo en (Balakrishnan et al., 1998), se observaron 857'142 retransmisiones de paquetes TCP. Entre ellas, sólo 436'747 paquetes fueron realmente perdidos. Esto significa que casi la mitad de las retransmisiones fue innecesaria, e ilustra la importancia de una correcta estimación del RTT en conexiones basadas en el protocolo TCP.

3. Algoritmos para la Estimación del RTT

Esta sección describe varios algoritmos para la estimación del RTT propuestos en la literatura. Usaremos $x(i)$, con $i \in \mathbb{N}_0 \triangleq \{0, 1, \dots\}$, para denotar al i -ésimo RTT, $y(i)$ para denotar a la medición de $x(i)$, y $\hat{x}(i)$ para referirnos a una estimación de $x(i)$, obtenida en función de las mediciones de RTT en instantes anteriores (i.e., obtenida en base a $y(i-1), y(i-2), \dots$). Definimos, además, el error de estimación como

$$\varepsilon(i) \triangleq y(i) - \hat{x}(i). \quad (3)$$

Usaremos subíndices para distinguir las estimaciones obtenidas con distintos algoritmos.

3.1. Método de estimación de Jacobson (Jacobson, 1988)

Cuando se usa este algoritmo para la estimación del RTT,

$$\hat{x}_J(0) \triangleq \phi, \quad \hat{x}_J(1) = y(0), \quad (4a)$$

$$\hat{x}_J(i+1) \triangleq [1 - \alpha]\hat{x}_J(i) + \alpha y(i), \quad i \geq 1, \quad (4b)$$

donde α corresponde a un factor ajustado en forma empírica (se ha propuesto $\alpha = 0.125$ (Jacobson, 1988)) y ϕ es cualquier número real.

Como se mencionó en la sección anterior, las estimaciones del RTT se usan para definir el parámetro RTO. Inicialmente (es decir, si $i = 0$), este valor es definido de forma empírica y por lo tanto $\hat{x}(0)$ no juega rol alguno (Paxon and Allman, 2000).

Las estimaciones entregadas por el algoritmo descrito en (4) corresponden a la salida de un filtro estrictamente causal, de primer orden, lineal e invariante en el tiempo, excitado por las mediciones del RTT. Por lo tanto, es razonable esperar que dicho algoritmo se desempeñe razonablemente bien si el RTT pudiese modelarse como un proceso Gaussiano y estacionario (Anderson and Moore, 1979). Lamentablemente, éste no es el caso: La señal RTT presenta un comportamiento que puede variar rápidamente y cuya distribución dista de ser Gaussiana (Li, 2000). Lo anterior se ilustra en la Fig. 2.

La discusión anterior explica, al menos en forma preliminar, el mal desempeño que el algoritmo de Jacobson alcanza en algunos regímenes de conexión (Balakrishnan et al., 1998; Ludwig and Sklower, 2000; Seddigh, 2000). A continuación se describen varias alternativas al algoritmo de Jacobson propuestas en la literatura. Cada una de ellas apunta a la consideración de algún aspecto no tomado en cuenta por Jacobson y, en ocasiones, permiten obtener mejores estimaciones del RTT.

3.2. Estimador del RTT basado en un filtro de medianas ponderadas (Ma et al., 2004)

(Ma et al., 2004) afirman que la distribución del RTT puede aproximarse como una distribución Laplaciana. Motivados por esto, proponen la utilización de un filtro de medianas ponderadas (WM, por sus siglas en inglés: Weighted Medians) para estimar el RTT. Este tipo de filtro presenta un buen desempeño cuando la señal a estimar obedece una distribución con cola pesada (*heavy tail*) como por ejemplo Laplaciana (Arce and Paredes, 2000; Yin et al., 1996). El estimador propuesto en (Ma et al., 2004) puede ser expresado como

$$\hat{x}_{WM}(i+1) \triangleq WM\left(\left[\hat{x}_{WM}(i-\ell)\right]_{\ell=0}^{N-1}, y(i-j)\right]_{j=0}^{M-1}, W^T\right), \quad (5)$$

donde N es el número de estimaciones pasadas consideradas, M es el número de mediciones pasadas consideradas, y W es el vector de pesos asociados a cada una de ellas, definido como $W \triangleq [A_0, A_1, \dots, A_{N-1}, B_0, B_1, \dots, B_{M-1}]^T$, donde A_ℓ , con $\ell \in \{0, 1, \dots, N-1\}$, corresponde a los pesos asociados a las estimaciones pasadas, y B_j , con $j \in \{0, 1, \dots, M-1\}$, son los pesos asociados a las mediciones pasadas. (Ma et al.,

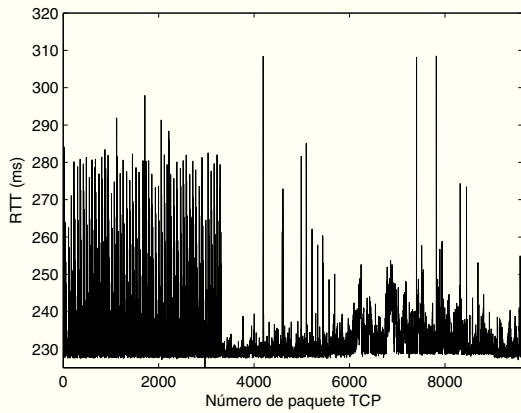


Figura 2: Señal RTT en una conexión TCP en que se envía un paquete de 20 (bytes) cada 500 (ms) (intervalo determinístico, fijado en forma arbitraria).

2004) proponen la utilización de un filtro WM con pesos reales positivos, basando en las ideas presentadas por (Arce and Paredes, 2000). El algoritmo puede describirse como sigue:

1. Cálculo del umbral $T_0 \triangleq 1/2 \left[\sum_{\ell=0}^{N-1} A_{\ell} + \sum_{j=0}^{M-1} B_j \right]$.
2. Ordenamiento en forma ascendente, según magnitudes, de las estimaciones pasadas, $\hat{x}_{WM}(i), \hat{x}_{WM}(i-1), \dots, \hat{x}_{WM}(i-N+1)$, junto a las mediciones pasadas, $y(i), y(i-1), \dots, y(i-M+1)$.
3. Suma de las magnitudes de los pesos asociados a cada una de las muestras ordenadas previamente, comenzando con el peso asociado a la muestra de valor máximo, y continuando en forma descendente.
4. Finalmente, la salida del filtro, y por tanto la estimación del RTT $\hat{x}_{WM}(i+1)$, será igual a la muestra cuyo peso asociado causó que la suma fuese mayor o igual a T_0 .

Los parámetros del estimador descrito son N, M y los coeficientes A_{ℓ}, B_j . Los autores de (Ma et al., 2004) realizan simulaciones que sugieren que elegir $N > 1$ no provee mejoras significativas en el desempeño del algoritmo y, por lo tanto, recomiendan utilizar $N = 1$. Asimismo, determinan que $M = 5$ es un valor razonable para capturar la correlación entre las mediciones del RTT. Los coeficientes A_{ℓ}, B_j son elegidos según

$$A(1) = \beta, \quad B(j) = \alpha^{j-1}, \quad (6)$$

donde $j = \{0, \dots, M-1\}$, (recuerde que $N = 1$) y $\alpha = 0.875$ y $\beta = 0.5$ corresponden a una elección adecuada en un amplio rango de escenarios de conexión (Ma et al., 2004).

Descrito el funcionamiento del algoritmo, y utilizando los valores para los parámetros N y M sugeridos en (Ma et al., 2004), es posible notar la necesidad de contar con 1 predicción previa, al igual que con la observación de al menos 5 mediciones de RTT antes de lograr una estimación del mismo. Dado esto, los autores de (Ma et al., 2004) sugieren fijar los valores de RTO de forma empírica hasta lograr registrar las mediciones de RTT necesarias para ejecutar el algoritmo. En el presente trabajo, el foco es la estimación del RTT. Por lo tanto,

la implementación del algoritmo fue realizada tomando como condiciones iniciales los primeros 5 valores estimados por el algoritmo de Jacobson (Jacobson, 1988). Así, $\hat{x}(i+1) = \hat{x}_j(i+1)$ si $i \in \{0, 1, \dots, 4\}$, y $\hat{x}_{WM}(i+1)$ estará dado por (5) en otro caso.

3.3. Estimador del RTT basado en un filtro de respuesta a impulso finita (Li et al., 2009)

A diferencia de los métodos de estimación del RTT basados en filtros invariantes en el tiempo propuestos por Jacobson o Ma et al., (Li et al., 2009) proponen el uso de filtros adaptativos. Específicamente, proponen un filtro de respuesta a impulso finita (FIR), cuyos coeficientes se ajustan en línea. Defina el vector de dimensión p

$$Y_i \triangleq [y(i) \ y(i-1) \ \dots \ y(i-p+1)]^T. \quad (7)$$

El estimador propuesto en (Li et al., 2009) está dado por

$$\hat{x}_{FIR}(i+1) \triangleq W_i^T Y_i, \quad (8)$$

donde $W_i \triangleq [w(i) \ w(i-1) \ \dots \ w(i-p+1)]^T$ corresponde al vector de coeficientes del filtro FIR.

La adaptación de los coeficientes del filtro se hace en base al algoritmo Widrow-Hoff LMS (Zaknich, 2005). Así,

$$W_i = W_{i-1} + 2\mu \varepsilon(i) Y_i, \quad (9)$$

donde $\varepsilon(i)$ corresponde al error de estimación del RTT, i.e., $\varepsilon(i) = y(i) - \hat{x}_{FIR}(i)$, y μ es un factor responsable de controlar la agresividad de la adaptación. En (Li et al., 2009), no se propone un valor para μ , sin embargo se establece que

$$\frac{1}{\sum_{j=1}^p E[Y_i(j)^2]} \geq \mu > 0, \quad (10)$$

garantizaría la convergencia de W si el RTT fuese estacionario. En el presente trabajo, se adopta

$$\mu = \frac{1}{2 \sum_{j=1}^p E[Y_i(j)^2]}, \quad (11)$$

elección que se encuentra en el centro de la región que garantiza la convergencia de W (vea (10)).

El algoritmo en (8) precisa contar con p mediciones pasadas del RTT para realizar las estimaciones. El trabajo (Li et al., 2009) no menciona las condiciones iniciales utilizadas. Por lo tanto, supondremos $\hat{x}_{FIR}(0) = 0$, mientras que $Y_0 = [y(0), y(0), \dots, y(0)]$, $Y_1 = [y(1), y(0), \dots, y(0)]$, $Y_2 = [y(2), y(1), y(0), \dots, y(0)]$, etc. De esta forma, las mediciones faltantes del RTT serán reemplazadas por la primera medición hecha, es decir, por $y(0)$, hasta la instancia $i = p$. Por otra parte, la elección de p debe hacerse considerando que un mayor valor de p permitirá una predicción más suave y menos sensible a condiciones de ruido transitorio, pero con una reacción más lenta frente a cambios repentinos, lo que es común en el caso del RTT. (Li et al., 2009) proponen $p = 15$ y $W_0 = [1/15, \dots, 1/15]$.

3.4. Estimador del RTT basado en intervalos de confianza (Ngwenya and Hancke, 2004)

Este método define, en cada instate, un intervalo de confianza para el RTT. Denote el estimador del RTT en el instante i por $\hat{x}_{IDC}(i)$ y suponga que se cuenta con estimaciones de la varianza del error de estimación correspondiente, denotadas por $\hat{\sigma}(i)$. Defina el intervalo

$$[\hat{x}_{IDC}(i) - k\hat{\sigma}(i), \hat{x}_{IDC}(i) + k\hat{\sigma}(i)], \quad (12)$$

donde k es una constante que define el nivel de confianza deseado (Silvey, 1975). Suponiendo que el RTT es Gaussiano, que $\hat{x}_{IDC}(i)$ es la media condicional de $x(i)$ dadas las mediciones pasadas (i.e., que $\hat{x}_{IDC}(i)$ es el mejor estimador de $x(i)$ dadas las mediciones pasadas), y que $\hat{\sigma}(i)$ es la varianza del error de estimación asociado, entonces se puede determinar la probabilidad de que, condicionado a las mediciones pasadas del RTT, el intervalo anterior contenga al RTT $x(i)$. En particular, si $k = 3$, dicha probabilidad es del 99.75 % (Silvey, 1975).

Los autores de (Ngwenya and Hancke, 2004) proponen estimar la varianza del estimador usando la relación recursiva (Jacobson, 1988)

$$\hat{\sigma}(i) = [1 - \beta]\hat{\sigma}(i - 1) + \beta|x(i) - \hat{x}_{IDC}(i)|, \quad (13)$$

con $\beta = 0.25$.

En cada instancia i , se construye el intervalo de confianza descrito en (12) y se verifica la pertenencia de la medición $y(i)$ a dicho intervalo. Si esta medición está contenida en el intervalo, entonces se usa el algoritmo de Jacobson (ver Sección 3.1) para estimar el RTT en la instancia $i + 1$, es decir, para estimar $x(i + 1)$. Si la medición $y(i)$ no está contenida en el intervalo, entonces se usa la medición $y(i)$ como estimación de $x(i + 1)$. Concretamente, el algoritmo propuesto está dado por

$$\hat{x}_{IDC}(i + 1) \triangleq \begin{cases} y(i) & , \text{ si } |y(i) - \hat{x}_{IDC}(i)| > k\hat{\sigma}(i), \\ [1 - \alpha]\hat{x}_{IDC}(i) + \alpha y(i), & \text{ en otro caso,} \end{cases} \quad (14)$$

donde $\alpha = 0.125$ (Jacobson, 1988) y las condiciones iniciales, al igual que en el algoritmo de Jacobson, se eligen como $\hat{x}_{IDC}(0) = 0$ y $\hat{x}_{IDC}(1) = y(0)$, comenzando la ejecución del algoritmo a partir de $i = 1$. El algoritmo necesita, además, la condición inicial $\hat{\sigma}(1) = 0$.

3.5. Estimación del RTT usando el log-likelihood ratio test (Ngwenya and Hancke, 2004)

Los autores de (Ngwenya and Hancke, 2004) proponen, también, un algoritmo similar al descrito en la sección anterior, pero reemplazando el test de pertenencia a un intervalo de confianza por el test *log-likelihood ratio* (Basseville and Nikiforov, 1993).

La idea central de este algoritmo es suponer que el RTT está distribuido de forma normal, con media $\mu(i)$ variante en el tiempo y varianza constante σ^2 . Defínase

$$s(i) = \ln \frac{p_{\hat{\mu}(i), \hat{\sigma}(i)}(\hat{x}(i))}{p_{\hat{\mu}(i-1), \hat{\sigma}(i-1)}(\hat{x}(i))}, \quad (15)$$

donde $\hat{\mu}(i)$ es una estimación de $\mu(i)$, $\hat{\sigma}(i)$ es una estimación de σ y $p_{\mu, \sigma}(y) \triangleq \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$ es la verosimilitud (*likelihood*) de y , dados los parámetros μ y σ . Si $s(i) > 0$, entonces la verosimilitud de $\hat{x}(i)$ es mayor con parámetros $(\hat{\mu}(i), \hat{\sigma}(i))$ y por lo tanto se puede suponer que ha habido un cambio en la media del RTT entre las instancias $i - 1$ e i . En caso contrario, es decir, si $s(i) \leq 0$, entonces se puede suponer que no ha habido cambios en la media del RTT. Al detectarse cambios en la media, (Ngwenya and Hancke, 2004) fuerza la estimación del RTT a ser igual a la nueva medición $y(i)$. Si no es así, la estimación del RTT se construye usando el algoritmo de Jacobson (ver Sección 3.1). Explotando la suposición de Gaussianidad del RTT, el procedimiento anteriormente descrito puede resumirse como sigue:

$$\hat{x}_{LLR}(i + 1) = [1 - \alpha]\hat{x}_{LLR}(i) + \alpha y(i) \quad (16a)$$

$$s(i + 1) = \frac{\hat{x}_{LLR}(i + 1) - \hat{x}_{LLR}(i)}{\hat{\sigma}(i + 1)^2} \times \left[y(i) - \frac{\hat{x}_{LLR}(i) + \hat{x}_{LLR}(i + 1)}{2} \right] \quad (16b)$$

Si $s(i + 1) > 0$ y $s(i + 1) > s(i)$ entonces

$$\hat{x}_{LLR}(i + 1) = y(i), \quad (16c)$$

Fin

donde $\alpha = 0.125$ (Jacobson, 1988), $\hat{\sigma}(i + 1)$ se calcula como en (13), mientras que las condiciones iniciales del algoritmo son $\hat{x}_{LLR}(0) = 0$, $\hat{x}_{LLR}(1) = y(0)$, $\hat{\sigma}(1) = 0$ y $s(1) = 0$.

Note que, para evitar el efecto de *outliers* en las estimaciones, los autores consideran cambios sostenidos en la media del RTT para gatillar la detección de cambios. Esto se captura en la condición adicional $s(i + 1) > s(i)$.

3.6. Estimación del RTT usando un filtro de Kalman

Uno de los algoritmos más populares para la estimación de señales es el filtro de Kalman (Anderson and Moore, 1979). Para usar el filtro de Kalman, es preciso contar con un modelo adecuado de la señal a estimar. En el caso de estimar RTT, se puede utilizar el modelo propuesto por (Jacobsson et al., 2004), quienes suponen que el RTT es una señal suave junto a una componente de ruido de alta frecuencia. El modelo propuesto en (Jacobsson et al., 2004) tiene la forma

$$x(i + 1) = x(i) + v(i), \quad (17a)$$

$$y(i) = x(i) + e(i), \quad (17b)$$

donde v y e son procesos de ruido blanco de media cero, no correlacionados entre sí. La característica ruidosa de la medición del RTT es capturada por el ruido $e(i)$, de varianza R_e , mientras que los cambios repentinos de la señal RTT son modelados por el ruido $v(i)$, de varianza R_v .

Usando como base el modelo en (17a) y las ecuaciones del filtro de Kalman (Anderson and Moore, 1979), se puede proponer el siguiente estimador para el RTT:

$$\hat{x}_K(0) \triangleq \phi, \quad \hat{x}_K(1) \triangleq y(0), \quad (18a)$$

$$\hat{x}_K(i+1) \triangleq \hat{x}_K(i) + K(i)[y(i) - \hat{x}_K(i)], \quad i \geq 1, \quad (18b)$$

donde

$$K(i) = \frac{P(i)}{P(i) + R_e}, \quad P(i+1) = [1 - K(i)]P(i) + R_v. \quad (19)$$

Note que la inicialización del algoritmo anterior se hace como en el algoritmo de Jacobson descrito anteriormente. Dado que la ejecución del filtro de Kalman comenzará a partir de $i = 1$, es preciso contar con $P(1) = P_0$ como condición inicial. Así, el algoritmo descrito en (18a) posee tres parámetros de diseño: las varianzas de los ruidos v y e , y la condición inicial P_0 .

El desempeño alcanzable con este método de estimación depende de cuan acertado sea el modelo en (17a), y de la precisión con que se conozcan los parámetros R_v , R_e y P_0 . Los autores de (Jacobsson et al., 2004) proponen elegir las varianzas R_v y R_e iguales a una estimación de la varianza del RTT. Sin embargo, no presentan un valor para el parámetro P_0 , cuya configuración se discute en la Sección 5.1.

Independientemente de la elección que se haga para los parámetros del algoritmo en (18a), éste sufre de un problema fundamental: Supone, implícitamente, que las varianzas de v y e son constantes. Esta suposición no es razonable pues, tal como se discutió en la Sección 2, el RTT cambia sus estadísticas en el tiempo. A continuación se propone un algoritmo para la estimación dinámica de R_v y R_e .

3.7. Estimación del RTT usando un filtro de Kalman con estimación adaptativa de varianzas

Considere el siguiente modelo para el RTT:

$$x(i+1) = x(i) + bw_1(i), \quad (20a)$$

$$y(i) = x(i) + dw_2(i), \quad (20b)$$

donde w_1 y w_2 son secuencias de ruido blanco de media cero y varianza unitaria. En (20a), bw_1 y dw_2 juegan el rol de los ruidos v y e presentados previamente en (17a). Supondremos que ambos ruidos no se encuentran correlacionados. Claramente, $R_v = b^2$ y $R_e = d^2$, es decir, los parámetros b y d definen las varianzas de los ruidos v y e .

En esta sección usaremos un filtro de Kalman para estimar el RTT, suponiendo que b y d son parámetros desconocidos. Para ello definimos

$$\theta \triangleq \begin{pmatrix} b \\ d \end{pmatrix}, \quad (21)$$

y denotamos por

$$\hat{\theta}(i) \triangleq \begin{pmatrix} \hat{b}(i) \\ \hat{d}(i) \end{pmatrix} \quad (22)$$

a las estimaciones de b y d obtenidas con las mediciones $y(i), y(i-1), \dots$. Consideraremos el siguiente estimador del RTT:

$$\hat{x}_{KP}(i+1) \triangleq \hat{x}_{KP}(i) + K(i)|_{\theta=\hat{\theta}(i)} [y(i) - \hat{x}_{KP}(i)], \quad i \geq 1, \quad (23)$$

donde la ganancia del filtro, es decir $K(i)|_{\theta=\hat{\theta}(i)}$, es una función de los valores estimados de b y d . Como estimaciones iniciales

para el RTT se consideran las mismas usadas en la sección anterior, es decir, $\hat{x}_{KP}(0) = 0$ y $\hat{x}_{KP}(1) = y(0)$.

Para estimar θ y, como consecuencia de ello, determinar la ganancia del filtro de Kalman en (23), proponemos utilizar el método de minimización del error de predicción (PEM (Goodwin and Sin, 1984, pp. 308–315)). Procediendo como en (Goodwin and Sin, 1984), obtenemos el siguiente estimador recursivo para θ :

$$\hat{\theta}(i) = \hat{\theta}(i-1) + M(i-1)\psi(i-1)[y(i) - \hat{x}(i)], \quad (24a)$$

$$M(i-1) = M(i-2)$$

$$- \frac{M(i-2)\psi(i-1)\psi(i-1)^T M(i-2)}{1 + \psi(i-1)^T M(i-2)\psi(i-1)}, \quad (24b)$$

$$\psi(i-1) = \psi(i-2)[1 - K(i-1)]$$

$$+ \frac{\partial}{\partial \theta} K(i-1)[y(i-1) - \hat{x}_{KP}(i-1)], \quad (24c)$$

donde $\hat{x}_{KP}(i)$ corresponde al estimador en (23), la derivada debe evaluarse en $\theta = \hat{\theta}(i-1)$ y,

$$\frac{\partial K(i)}{\partial \theta} = \begin{pmatrix} \frac{\partial K(i)}{\partial b} \\ \frac{\partial K(i)}{\partial d} \end{pmatrix}, \quad (25)$$

con

$$\frac{\partial K(i)}{\partial b} = \frac{\frac{\partial P(i)}{\partial b} d^2}{[P(i) + d^2]^2}, \quad \frac{\partial K(i)}{\partial d} = \frac{\frac{\partial P(i)}{\partial d} d^2 - P(i)2d}{[P(i) + d^2]^2}, \quad (26)$$

donde

$$\begin{aligned} \frac{\partial P(i)}{\partial b} &= [1 - K(i-1)] \frac{\partial P(i-1)}{\partial b} \\ &\quad - P(i-1) \frac{\partial K(i-1)}{\partial b} + 2b, \end{aligned} \quad (27a)$$

$$\frac{\partial P(i)}{\partial d} = [1 - K(i-1)] \frac{\partial P(i-1)}{\partial d} - P(i-1) \frac{\partial K(i-1)}{\partial d}. \quad (27b)$$

Dadas las definiciones anteriores, la ganancia del filtro de Kalman en (23) puede escribirse como

$$K(i)|_{\theta=\hat{\theta}(i)} = \frac{P(i)}{P(i) + \hat{d}(i)^2}, \quad (28)$$

donde

$$P(i) = [1 - K(i-1)|_{\theta=\hat{\theta}(i-1)}] P(i-1) + \hat{b}^2(i-1). \quad (29)$$

El uso del estimador en (23) implica contar con ciertas condiciones iniciales para la ejecución del algoritmo PEM. A saber, se necesita conocer $\hat{\theta}(1)$, $M(1)$, $\psi(1)$ y $\frac{\partial P(1)}{\partial \theta}$. $\hat{\theta}(1)$ puede elegirse igual a los parámetros utilizados por el estimador basado en el filtro de Kalman sin uso del algoritmo PEM (vea la Sección 3.6). Las condiciones iniciales restante pueden determinarse en forma empírica (vea la Sección 5.1).

3.8. Estimación del RTT usando el filtro CUSUM-Kalman (Jacobsson et al., 2004)

Este algoritmo se inspira en el hecho de que las estadísticas del RTT pueden experimentar cambios repentinos en el tiempo.

Esto motiva el uso de algoritmos de detección de cambios en la estimación del RTT. (Jacobsson et al., 2004) proponen el uso del algoritmo de suma acumulativa (CUSUM) en su versión *one-sided* (Gustafsson, 2000), en conjunto con un filtro de Kalman (Anderson and Moore, 1979). A continuación se presenta el algoritmo.

Defínase

$$g(i+1) = \begin{cases} 0 & , \text{ si } \max(0, g(i) + \varepsilon(i+1) - \xi) > h \\ \max(0, g(i) + \varepsilon(i+1) - \xi), & \text{ en otro caso.} \end{cases} \quad (30)$$

La señal $g(i+1)$ corresponde, esencialmente, a la suma acumulativa de la entrada auxiliar $\varepsilon(i+1)$. En caso de que esta suma supere cierto umbral de control h , una alarma será activada, indicando un posible cambio de media en el proceso. Para los propósitos de este trabajo, la señal auxiliar se define como el error de estimación del RTT $\varepsilon(i)$. Para evitar la activación de una falsa alarma, el algoritmo CUSUM substrahe un valor pequeño ξ en cada ciclo del algoritmo (ver (30)). Por otro lado, la versión de CUSUM en (30) iguala la señal $g(i+1)$ a cero si ésta toma valores negativos. Esto implica que (30) sólo detecta cambios ascendentes en la media del RTT, justificando el nombre de *one-sided* CUSUM.

El modelo del sistema utilizado por (Jacobsson et al., 2004), es similar al presentado en (17a), pero considera, además, una variable adicional $\varphi(i)$ que permite la integración del filtro de Kalman con el algoritmo CUSUM:

$$x(i+1) = x(i) + \varphi(i)v(i), \quad \varphi(i) \in \{0, 1\}, \quad (31a)$$

$$y(i) = x(i) + e(i), \quad (31b)$$

donde la característica ruidosa de las mediciones del RTT queda capturada por el ruido de medición $e(i)$, mientras que los cambios repentinos de la señal RTT son modelados por el ruido $v(i)$, de varianza R_v , junto a la variable discreta $\varphi(i)$. Esta última tomará el valor 1 si se detecta un cambio significativo en las estadísticas del RTT, y 0 en otro caso.

Como se mencionó anteriormente, la estimación de $x(i+1)$ es hecha a través de un filtro de Kalman, mientras que la detección de cambios en el RTT se realiza usando el algoritmo CUSUM en su versión *one-side* (Gustafsson, 2000). Combinando ambos algoritmos, se logra el siguiente filtro denominado filtro CUSUM-Kalman:

$$\hat{x}_{CK}(i+1) \triangleq \hat{x}_{CK}(i) + K(i)[y(i) - \hat{x}_{CK}(i)] \quad (32a)$$

$$K(i) = P(i)(P(i) + R_e)^{-1} \quad (32b)$$

$$P(i+1) = [1 - K(i)]P(i) + \varphi(i)R_v \quad (32c)$$

$$\varepsilon(i+1) = y(i+1) - \hat{x}_{CK}(i+1) \quad (32d)$$

$$g(i+1) = \max(0, g(i) + \varepsilon(i+1) - \xi) \quad (32e)$$

Si $g(i+1) > h$ entonces

$$\varphi(i+1) = 1 \quad (32f)$$

$$g(i+1) = 0 \quad (32g)$$

Si no

$$\varphi(i+1) = 0 \quad (32h)$$

Fin

Dado que los autores de (Jacobsson et al., 2004) no presentan condiciones iniciales, definimos, nuevamente, $\hat{x}_{CK}(0) = 0$ y $\hat{x}_{CK}(1) = y(0)$, iniciándose la ejecución del algoritmo a partir de $i = 1$, con $g(1) = 0$, $\varphi(1) = 1$ y $P(1) = P_0$.

El filtro en (32a) necesita, al igual que el filtro de Kalman, contar con los valores de las varianzas del ruido de medición e y del ruido del proceso v . Jacobsson et al. realizan el supuesto de que ambas varianzas son iguales a la varianza del RTT observado. Además, el algoritmo en (32a) cuenta con dos parámetros de diseño adicionales: ξ y h . ξ corresponde a la mitad del cambio de media esperado para el RTT y h , al umbral de control (Gustafsson, 2000). Jacobsson et al. ajustan ambos parámetros en forma empírica, proponiendo $\xi = 0.005$ y $h = 0.05$. Una elección incorrecta de estos parámetros afecta considerablemente el desempeño del algoritmo CUSUM (Shu and Jiang, 2006).

Resulta evidente que la propuesta anterior puede mejorarse utilizando el algoritmo PEM descrito en la Sección 3.7, para estimar, para cada valor de i , los parámetros R_v y R_e . Asimismo, se podría considerar un mecanismo adaptativo que permita modificar, en línea, los parámetros ξ y h . Las observaciones anteriores motivan el algoritmo propuesto en la Sección 4.

4. Algoritmo Propuesto: CUSUM-Kalman Adaptativo

4.1. Motivación

El algoritmo descrito en la Sección 3.8 busca reaccionar rápidamente ante cambios abruptos en la media del RTT y, a la vez, ser poco sensible a ruido transitorio. Para lograr esto, se utiliza, en forma combinada, el algoritmo CUSUM y un filtro de Kalman (ver Sección 3.8).

Desde su introducción por Page (Page, 1954), el algoritmo CUSUM ha sido utilizado para la detección de cambios en la media de procesos, existiendo diversos estudios que han demostrado su buen desempeño (Lorden, 1971; Moustakides, 1986; Pollak, 1985; Siegmund, 1985). Sin embargo, este desempeño depende grandemente de la elección de los parámetros h y ξ (Shu and Jiang, 2006). Se han propuestos diversos criterios de ajuste de estos parámetros. Por ejemplo (Sparks, 2000) afirma que, si se desea detectar cambios de magnitud δ en la media, entonces $\xi = \delta/2$ es una buena elección. Ésta y otras elecciones en la literatura, requieren información precisa sobre magnitud del cambio de media esperado. Dada la cantidad de escenarios de conexión posibles, y la naturaleza no estacionaria del RTT, resulta imposible contar con esta información cuando se estima el RTT.

Por otro lado, el trabajo (Jacobsson et al., 2004) utiliza una versión *one-sided* de CUSUM y, por lo tanto, sólo considera cambios ascendentes en la media del RTT. Sin embargo, la media del RTT se ve fuertemente influenciada por el aumento o disminución de la congestión en la red. Así, es esperable que la consideración de una versión de CUSUM que detecte tanto cambios ascendentes como descendentes, permita lograr mejores estimaciones del RTT. Asimismo, sería deseable contar con un mecanismo que pueda elegir en línea los parámetros de dicho algoritmo. Un algoritmo con estas características se presenta a continuación.

4.2. Algoritmo CUSUM Adaptativo (ACUSUM) (Sparks, 2000)

(Sparks, 2000) propone un algoritmo CUSUM adaptativo que logra un desempeño satisfactorio cuando se desconocen las magnitudes de los cambios en las medias de la señal a monitorear. La propuesta de Sparks activa una alarma para señalar cambios significativos en la media del RTT si

$$g^+(i+1) > h(\hat{\delta}^+(i+1)) \quad \text{o} \quad g^-(i+1) < -h(-\hat{\delta}^-(i+1)), \quad (33)$$

donde

$$g^+(i+1) \triangleq \max(0, g^+(i) + \varepsilon(i+1) - \hat{\delta}^+(i+1)/2), \quad (34a)$$

$$g^-(i+1) \triangleq \min(0, g^-(i) + \varepsilon(i+1) - \hat{\delta}^-(i+1)/2), \quad (34b)$$

$$\hat{\delta}^+(i+1) \triangleq \max(\lambda\varepsilon(i+1) + [1-\lambda]\hat{\delta}^+(i), \delta_{\min}), \quad (34c)$$

$$\hat{\delta}^-(i+1) \triangleq \min(\lambda\varepsilon(i+1) + [1-\lambda]\hat{\delta}^-(i), -\delta_{\min}), \quad (34d)$$

$g^+(0) = g^-(0) = 0$, $\varepsilon(i+1)$ corresponde al error de estimación del RTT, $\lambda \in [0, 1]$ es un parámetro de diseño, δ_{\min} es el menor cambio de media que se desea detectar, y $\hat{\delta}^+(0)$ y $\hat{\delta}^-(0)$ son condiciones iniciales para $\hat{\delta}^+$ y $\hat{\delta}^-$, las cuales deberán ser definidas de forma apropiada.

Note que el algoritmo anterior detecta tanto cambios ascendentes como descendentes en la media del RTT, es decir, es una versión *two-sided* de CUSUM. Es importante notar, además, que la definición de $\hat{\delta}^+$ y $\hat{\delta}^-$ implica no detectar cambios de magnitud menor a δ_{\min} .

En el algoritmo propuesto, el umbral de control h no está fijo, sino que cambia en función de $\hat{\delta}^+$ y $\hat{\delta}^-$. Resta, entonces, obtener una expresión para el umbral de control en función de dichos parámetros. Para ello, se pueden usar los resultados en (Shu and Jiang, 2006), donde se prueba que

$$h(\hat{\delta}(i)) = \frac{\ln \left[1 + (\hat{\delta}(i)^2/2) ARL_{ic} + 1.166 \hat{\delta}(i) \right]}{\hat{\delta}(i)} - 1.166, \quad (35)$$

es una elección razonable. En (35), ARL_{ic} puede interpretarse como el número promedio de observaciones del RTT, antes de la activación de una falsa alarma (Ryu et al., 2010). Se desea, por lo tanto, un valor de ARL_{ic} alto.

Los autores de (Shu and Jiang, 2006) proponen la siguiente guía para la sintonía del algoritmo ACUSUM: Sea $[\delta_1, \delta_2]$ el rango de cambios de media posibles.

1. Elija $\delta_{\min} = \delta_1$.
2. Fije $\hat{\delta}^+(0) = \delta_{\text{mean}}$ y $\hat{\delta}^-(0) = -\delta_{\text{mean}}$, con $\delta_{\text{mean}} \triangleq (\delta_1 + \delta_2)/2$.
3. Elija λ en $[0.05, 0.25]$, siendo usual $\lambda = 0.1$ ó $\lambda = 0.2$.
4. Elija un valor para ARL_{ic} .

Al igual que en el diseño del algoritmo de CUSUM tradicional, la propuesta de (Shu and Jiang, 2006) y (Sparks, 2000) necesita que el usuario elija ciertos parámetros. Sin embargo, no se necesita contar con información exacta respecto a la señal que se desea monitorear.

4.3. Algoritmo de estimación propuesto

El algoritmo propuesto corresponde a una versión adaptativa del algoritmo presentado por Jacobsson et al. (Jacobsson et al., 2004). Concretamente se propone usar el algoritmo PEM para estimar en línea las varianzas de los ruidos de proceso y medición, y utilizar el algoritmo de CUSUM adaptativo descrito anteriormente para detectar cambios. Es posible resumir la ejecución del algoritmo como sigue:

Ejecute el algoritmo PEM y defina

$$R_v \triangleq \hat{b}(i)^2, R_e \triangleq \hat{d}(i)^2 \quad (36a)$$

$$\hat{x}_{ACK}(i+1) \triangleq \hat{x}_{ACK}(i) + K(i)[y(i) - \hat{x}_{ACK}(i)] \quad (36b)$$

$$K(i) = \frac{P(i)}{P(i) + R_e} \quad (36c)$$

$$P(i+1) = [1 - K(i)]P(i) + \varphi(i)R_v \quad (36d)$$

$$\varepsilon(i+1) = y(i+1) - \hat{x}_{ACK}(i+1) \quad (36e)$$

Ejecute el algoritmo ACUSUM para obtener

$$\hat{\delta}^{+,-}(i+1), g^{+,-}(i+1), h(\hat{\delta}^+(i+1)), h(\hat{\delta}^-(i+1)). \quad (36f)$$

$$\text{Si } g^+(i+1) > h(\hat{\delta}^+(i+1))$$

$$\quad \text{ó } g^-(i+1) < -h(\hat{\delta}^-(i+1)) \quad (36g)$$

entonces

$$\varphi(i+1) = 1 \quad (36h)$$

$$\hat{\delta}^+(i+1) = \delta_{\text{mean}} \quad (36i)$$

$$\hat{\delta}^-(i+1) = -\delta_{\text{mean}} \quad (36j)$$

$$g^+(i+1) = g^-(i+1) = 0 \quad (36k)$$

Si no

$$\varphi(i+1) = 0 \quad (36l)$$

Fin

Al igual que los algoritmos presentados hasta ahora, la ejecución del estimador comenzará a partir de $i = 1$ y se elige $\hat{x}_{ACK}(0) = 0$, $\hat{x}_{ACK}(1) = y(0)$. Las condiciones iniciales del algoritmo son $P(1) = P_0$, $g^+(1) = g^-(1) = 0$, $\hat{\delta}^+(1) = \delta_{\text{mean}}$ y $\hat{\delta}^-(1) = -\delta_{\text{mean}}$, con $\delta_{\text{mean}} \triangleq (\delta_1 + \delta_2)/2$. Las condiciones iniciales del algoritmo PEM son las usadas en la Sección 3.7.

5. Evaluación de los algoritmos de estimación

5.1. Escenarios de prueba y definición de la métrica de evaluación

La evaluación de los algoritmos de estimación del RTT puede ser hecha de dos formas: Mediante simulación, con el uso de programas como *ns-2* (ns-2, 2009), o utilizando mediciones reales del RTT. Dada la complejidad de Internet, y los diversos escenarios de conexión posibles, resulta imposible crear un escenario de simulación que refleje las propiedades estadísticas del RTT (Paxson and Floyd, 1997). Así, se optó por considerar un enfoque experimental para la evaluación de los algoritmos discutidos anteriormente. Para ello, se utilizó la herramienta de análisis *hping3* (hping, 2004), con la cual se establecieron conexiones TCP entre la Universidad Técnica

Federico Santa María (www.utfsm.cl), y el sitio de CAIDA (por sus siglas en inglés: The Cooperative Association for Internet Data Analysis, www.caida.org). Concretamente, se ejecutó el comando `hping3 -I eth0 -S -i u500000 -c 10000 www.caida.org -p 80`, desde un terminal Linux (vea detalles en (hping, 2004)). La conexión fue establecida el día Miércoles 27 de Abril de 2011 a las 18:20 h. (GMT-4). Los datos fueron capturados y analizados usando los programas de código abierto *Wireshark* y *Tshark* (Wireshark, 2011).¹

Las condiciones iniciales y elección de parámetros de los distintos algoritmos se resumen en la Tabla 1. En la primera columna de la tabla se presentan los valores utilizados por el filtro de Kalman, presente tanto en el estimador del RTT basado en este filtro, como en el estimador CUSUM-Kalman propuesto por (Jacobsson et al., 2004). El valor de $P(1)$ fue elegido de forma empírica, mientras que los valores de las varianzas R_v y R_e fueron fijados de forma empírica, en base al supuesto realizado en (Jacobsson et al., 2004), es decir, ambas varianzas se consideran iguales a una estimación de la varianza del RTT. A continuación, en la columna dos, se presentan las condiciones iniciales usadas en la ejecución del algoritmo PEM utilizado para la estimación adaptativa de las varianzas, donde $\theta = (b \ d)^T$, $R_v = b^2$ y $R_e = d^2$ (ver Sección 3.7). Así, se definió $\hat{\theta}(1) = (\sqrt{R_v} \ \sqrt{R_e})^T$, donde R_v y R_e son los valores utilizados por el filtro de Kalman, presentados en la primera columna. Los valores elegidos para $\psi(1)$, $\frac{\partial P(1)}{\partial \theta}$ y $M(1)$ fueron elegidos en forma empírica. Finalmente, en la tercera columna, se presentan los valores usados para la implementación del algoritmo de estimación del RTT propuesto, donde nuevamente los parámetros fueron elegidos en forma empírica.

Es importante señalar que, para la elección de aquellos parámetros ajustados de forma empírica, se utilizó un conjunto de datos distinto al usado para la evaluación de los algoritmos de estimación.

Para evaluar el desempeño de los algoritmos estudiados se utilizó la métrica de evaluación MAE (por sus siglas en inglés: Mean Absolute Error), la cual corresponde al promedio del valor absoluto del error de estimación, es decir, $MAE \triangleq 1/m \sum_{i=1}^m |\varepsilon(i)|$, donde $\varepsilon(i) = y(i) - \hat{x}(i)$ (Hyndman and Koehler, 2006).

5.2. Resultados

En el presente trabajo se optó por la evaluación de los algoritmos utilizando mediciones reales del RTT. Para ello, y reconociendo las limitaciones de trabajar con realizaciones específicas, se tomaron 96 series de datos, i.e., se consideraron 96 realizaciones del RTT. Para cada realización, se calculó el índice MAE y luego se promediaron los índices sobre todas las realizaciones. Reconocemos que los datos experimentales utilizados para la evaluación de los algoritmos no representan a todos los escenarios de conexión posibles, y sólo representan el estado de la conexión entre los servidores mencionados anteriormente en un momento específico. Sin perjuicio de lo

Tabla 1: Parámetros de diseño y condiciones iniciales.

Kalman	PEM	ACUSUM-Kalman
$P(1) = 1$	$\hat{\theta}(1) = (0.0187 \ 0.0187)^T$	$P(1) = 1$
$R_v = R_e = 4.67 \times 10^{-4}$	$M(1) = 10^{-3} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ $\psi(1) = (0 \ 0)^T$ $\partial P(1)/\partial \theta = (0 \ 0)^T$	$\delta_1 = 10^{-9}$ $\delta_2 = 0.01$ $\lambda = 0.1$ $ARL_{ic} = 400$

anterior, nuestro procedimiento contrasta con los resultados presentados en la literatura donde es usual que los autores obtengan conclusiones basadas en sólo una realización del RTT y donde, en general, no se hace una evaluación cuantitativa de las mejoras obtenidas.

Las Tablas 2 a 5 muestran el desempeño alcanzado por cada uno de los algoritmos. En cada caso, la columna \overline{MAE} presenta el promedio del MAE sobre las 96 realizaciones. Se indica, además, el porcentaje de mejoría o deterioro del desempeño de cada algoritmo, para las realizaciones consideradas, cuando se compara con el algoritmo de Jacobson (1988). (Porcentajes positivos (negativos) implican que el algoritmo en cuestión alcanza un desempeño peor (mejor) que el que es alcanzado por el algoritmo de Jacobson.) Los valores numéricos reportados en este trabajo corresponden a los obtenidos en el escenarios descrito anteriormente, y mostraron consistencia con series de datos obtenidas en condiciones alternativas.

En las Tablas 2 a 5 es posible observar que la utilización de algoritmos de estimación adaptativos basados en el filtro de Kalman, logran mejoras que van desde un 2.7 % a un 12.7 % con respecto al algoritmo de Jacobson.

Comenzando por el algoritmo de estimación del RTT propuesto por Jacobson, el cual puede ser visto como un filtro de Kalman estacionario de ganancia $K(i) = 0.125$, es posible apreciar como los resultados mejoran a medida que se utilizan algoritmos adaptativos de estimación. Así, de las Tablas 2 y 3, se observa una mejoría de hasta un 8.9 % al utilizar un filtro de Kalman junto con la estimación adaptativa de las varianzas, a través del algoritmo PEM. No obstante, las Tablas 4 y 5, presentan resultados desfavorables para Kalman frente a Jacobson. Esto puede ser explicado por la sensibilidad del filtro de Kalman frente a *outliers* (Farahmand et al., 2010), los cuales tienen mayor probabilidad de aparecer cuando se considera una cantidad elevada de datos, como es el caso de envíos de 50 y 100 paquetes. Lo anterior, se ve contrarrestado al incorporar el algoritmo de detección de cambios CUSUM, el cual evita la reacción del estimador frente a cambios aislados en la media del RTT, y con el que se logra en promedio, una mejoría de un 5.7 % respecto a Jacobson. Nuevamente, es posible observar una mejoría en los resultados al incorporar la estimación adaptativa de las varianzas, sin embargo, ésta no resulta significativa. Por otra parte, el algoritmo de estimación del RTT propuesto en el presente trabajo, compuesto por la estimación adaptativa de los parámetros de diseño del algoritmo de CUSUM, en conjunto con el uso de su versión *two-sided*, logra una mejoría de hasta un 8.9 %, frente al estimador CUSUM-Kalman original

¹Las mediciones realizadas y los algoritmos considerados se encuentran disponibles en: <http://profesores.elo.utfsm.cl/~agv/DDocmac>

presentado en (Jacobsson et al., 2004), y entre un 8.9 % y un 12.7 % frente a Jacobson. Finalmente, aquellos algoritmos de estimación del RTT que no se encuentran basados en el filtro de Kalman, presentan, en general, una mal desempeño, a excepción del filtro de medianas ponderadas. Este algoritmo logra, en promedio, una mejoría de un 17 % con respecto al algoritmo de Jacobson. Este valor es cercano al presentado por los autores de este algoritmo en (Ma et al., 2004).

Los resultados presentados permiten concluir que, si bien el uso de estrategias adaptativas implica una mejora en el desempeño de estrategias de estimación del RTT, éstas no son lo suficientemente exitosas como para lograr un desempeño superior al alcanzado por el algoritmo de medianas ponderadas propuesto en (Ma et al., 2004). Lo anterior no es sorprendente pues los filtros de medianas ponderadas son óptimos con respecto a la métrica MAE considerada en este trabajo (Yli-Harja et al., 1991).

6. Conclusión

En este trabajo se evaluaron diversos algoritmos para la estimación del RTT, considerando para ello una métrica particular y escenarios de conexión comunes. Además de considerar propuestas en la literatura, se presentó un método de estimación adaptativo, basado en un filtro de Kalman complementado con algoritmos de detección de cambios y de estimación de parámetros en línea (ACUSUM y PEM). El algoritmo propuesto logró un desempeño mejor que aquel alcanzado por procedimientos de estimación con parámetros estáticos. En particular, el algoritmo alcanza un desempeño en promedio 10.7 % mejor que el alcanzado por el algoritmo clásico de Jacobson. Sin embargo, debemos mencionar que el desempeño del algoritmo propuesto es, en general, inferior al alcanzado por el algoritmo de medianas ponderadas propuesto en (Ma et al., 2004).

Es relevante recordar que las estimaciones del RTT no tienen valor *per se*. Su utilidad radica en que permiten dar un mejor uso a la red, evitando congestión derivada de la retransmisión innecesaria de paquetes, o excesiva latencia derivada de la espera innecesaria de ACKs asociados a paquetes perdidos. Cuantificar el modo en que la calidad de las estimaciones del RTT influyen en el uso del ancho de banda de una red, o en la latencia de la misma, así como un estudio acabado de la complejidad computacional de los algoritmos propuestos, son temas para trabajo futuro.

English Summary

Assessment of algorithms for round-trip delay time estimation and a proposal.

Abstract

Several algorithms have been proposed for the estimation of the Round-Trip Delay Time (RTT). They aim at achieving better performance than the estimation algorithm adopted by

Tabla 2: Desempeño de algoritmos para conexiones de largo 10 paquetes.

\hat{x}	$\overline{MAE} (ms)$	% vs. \hat{x}_J
\hat{x}_J (Jacobson, 1988)	4.1406	-
\hat{x}_{WM} (Ma et al., 2004)	3.6632	-11.5294
\hat{x}_{FIR} (Li et al., 2009)	4.2306	2.1731
\hat{x}_{IDC} (Ngwenya y Hancke, 2004)	4.1385	-0.0509
\hat{x}_{LLR} (Ngwenya y Hancke, 2004)	4.9024	18.3988
\hat{x}_K (Sección 3.6)	3.7716	-8.9122
$\hat{x}_K + PEM$ (Sección 3.7)	3.7710	-8.9262
\hat{x}_{CK} (Jacobsson et al., 2004)	3.6816	-11.0857
$\hat{x}_{CK} + PEM$ (Sección 3.8*)	3.6810	-11.1011
\hat{x}_{ACK} (Sección 4)	3.6147	-12.7008

*Este algoritmo fue insinuado al final de la Sección 3.8.

Tabla 3: Desempeño de algoritmos para conexiones de largo 20 paquetes.

\hat{x}	$\overline{MAE} (ms)$	% vs. \hat{x}_J
\hat{x}_J (Jacobson, 1988)	3.9325	-
\hat{x}_{WM} (Ma et al., 2004)	3.2514	-17.3209
\hat{x}_{FIR} (Li et al., 2009)	4.3354	10.2445
\hat{x}_{IDC} (Ngwenya y Hancke, 2004)	4.6448	18.1123
\hat{x}_{LLR} (Ngwenya y Hancke, 2004)	5.5624	41.4459
\hat{x}_K (Sección 3.6)	3.8254	-2.7254
$\hat{x}_K + PEM$ (Sección 3.7)	3.8248	-2.7401
\hat{x}_{CK} (Jacobsson et al., 2004)	3.5937	-8.6154
$\hat{x}_{CK} + PEM$ (Sección 3.8*)	3.5933	-8.6268
\hat{x}_{ACK} (Sección 4)	3.5089	-10.7731

*Este algoritmo fue insinuado al final de la Sección 3.8.

Tabla 4: Desempeño de algoritmos para conexiones de largo 50 paquetes.

\hat{x}	$\overline{MAE} (ms)$	% vs. \hat{x}_J
\hat{x}_J (Jacobson, 1988)	3.3570	-
\hat{x}_{WM} (Ma et al., 2004)	2.6859	-19.9915
\hat{x}_{FIR} (Li et al., 2009)	3.8969	16.0818
\hat{x}_{IDC} (Ngwenya y Hancke, 2004)	4.8015	43.0299
\hat{x}_{LLR} (Ngwenya y Hancke, 2004)	5.7686	71.8367
\hat{x}_K (Sección 3.6)	3.4242	2.0023
$\hat{x}_K + PEM$ (Sección 3.7)	3.4240	1.9947
\hat{x}_{CK} (Jacobsson et al., 2004)	3.2449	-3.3413
$\hat{x}_{CK} + PEM$ (Sección 3.8*)	3.2445	-3.3510
\hat{x}_{ACK} (Sección 4)	3.0042	-10.5098

*Este algoritmo fue insinuado al final de la Sección 3.8.

Tabla 5: Desempeño de algoritmos para conexiones de largo 100 paquetes.

\hat{x}	$\overline{MAE} (ms)$	% vs. \hat{x}_J
\hat{x}_J (Jacobson, 1988)	3.2060	-
\hat{x}_{WM} (Ma et al., 2004)	2.5905	-19.1968
\hat{x}_{FIR} (Li et al., 2009)	3.8427	19.8596
\hat{x}_{IDC} (Ngwenya y Hancke, 2004)	4.6933	46.3931
\hat{x}_{LLR} (Ngwenya y Hancke, 2004)	5.8852	83.5689
\hat{x}_K (Sección 3.6)	3.3595	4.7894
$\hat{x}_K + PEM$ (Sección 3.7)	3.3598	4.7969
\hat{x}_{CK} (Jacobsson et al., 2004)	3.2090	0.0958
$\hat{x}_{CK} + PEM$ (Sección 3.8*)	3.2087	0.0864
\hat{x}_{ACK} (Sección 4)	2.9205	-8.9036

*Este algoritmo fue insinuado al final de la Sección 3.8.

the Transfer Control Protocol (TCP) standard proposed in the late eighties. In general, the algorithms in the literature are assessed qualitatively, observing the estimation errors for a single connection scenario or via simulations. There are no works considering a quantitative evaluation metric, comparing the performance achieved by different algorithms. This motivates the present work in which several RTT estimation algorithms are compared using a common performance metric, and common experimental scenarios. As a second contribution, we propose an adaptive algorithm to estimate RTT, which is able to react quickly against sudden changes in the signal and, simultaneously, is less sensitive to noise interference than alternatives in the literature.

Keywords:

ACUSUM, adaptive filters, communication networks, estimation, Kalman filter, PEM, Round-Trip Delay Time (RTT), TCP.

Agradecimientos

Los autores agradecen el apoyo económico recibido de la UTFSM a través de su programa PIIC y de CONICYT a través del proyecto Anillo ACT53.

Referencias

- Anderson, B. D. O., Moore, J. B., 1979. Optimal filtering. Prentice-Hall.
- Arce, G., Paredes, J., Mar. 2000. Recursive weighted median filters admitting negative weights and their optimization. *Signal Processing, IEEE Transactions on* 48 (3), 768–779.
DOI: 10.1109/78.824671
- Balakrishnan, H., Padmanabhan, V., Seshan, S., Stemm, M., Katz, R., Apr. 1998. TCP behavior of a busy Internet server: analysis and improvements. In: *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Vol. 1. pp. 252–262 vol.1.*
DOI: 10.1109/INFCOM.1998.659661
- Basseville, M., Nikiforov, I., 1993. Detection of Abrupt Changes: Theory and Applications. Prentice-Hall.
- Farahmand, S., Angelosante, D., Giannakis, G. B., nov. 2010. Doubly robust Kalman smoothing by controlling outlier sparsity. In: *Signals, Systems and Computers (ASIOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on. pp. 691–695.*
DOI: 10.1109/ACSSC.2010.5757650
- Goodwin, G. C., Sin, K. S., 1984. Adaptive filtering prediction and control. Prentice Hall, Englewood Cliffs.
- Gustafsson, F., 2000. Adaptive filtering and change detection. John Wiley & Sons, Ltd.
- Haeri, M., A.H.M.Rad, 2004. TCP retransmission timer adjustment mechanism using model-based RTT predictor. In: *Control Conference, 2004. 5th Asian. Vol. 1. pp. 686–693 Vol.1.*
DOI: 10.1109/ASCC.2004.184838
- hping, 2004. hping3.
URL: <http://www.hping.org/>
- Hyndman, R. J., Koehler, A. B., 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting* 22, 679–688.
- Jacobson, V., 1988. Congestion avoidance and control. In: *Symposium proceedings on Communications architectures and protocols. SIGCOMM '88. ACM, New York, NY, USA, pp. 314–329.*
DOI: 10.1145/52324.52356
- Jacobsson, K., Hjalmarsson, H., Möller, N., Johansson, K. H., 2004. Round-Trip Time estimation in communication networks using adaptive Kalman filtering. In: *Reglermöte.*
- Kurose, J. F., Ross, K. W., 2010. Computer networking - a top-down approach featuring the Internet, 3rd Edition. Addison-Wesley.
- Lei, Y., Zhu, R., Wang, W., 2006. A survey on TCP protocol and RTT estimation. In: *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on. Vol. 1. pp. 4410–4414.*
DOI: 10.1109/WCICA.2006.1713211
- Li, G., Zhao, N., Liu, C., 2009. Round Trip Time estimation based on adaptive filtering. In: *Information Science and Engineering (ICISE), 2009 1st International Conference on. pp. 1842–1846.*
DOI: 10.1109/ICISE.2009.1040
- Li, Q., May 2000. Delay characterization and performance control of wide-area networks. Ph.D. thesis, Univ. Of Delaware, Newark.
URL: <http://www.ece.udel.edu/~qli>
- Lorden, G., 1971. Procedures for reacting to a change in distribution. *Annals of Mathematical Statistics* 42, 1897–1908.
- Ludwig, R., Sklower, K., 2000. The Eifel retransmission timer. *Computer Communication Review* 30 (3), 17–27.
- Ma, L., Arce, G., Barner, K., 2004. TCP retransmission timeout algorithm using weighted medians. *Signal Processing Letters, IEEE* 11 (6), 569–572.
DOI: 10.1109/LSP.2004.827957
- McCreary, S., Claffy, K., 2000. Trends in wide area IP traffic patterns - a view from Ames Internet exchange. In: *13th ITC specialist seminar: IP Traffic measurement, modeling and management.*
- Moustakides, G. V., 1986. Optimal stopping times for detecting changes in distributions. *Annals of Statistics* 14, 1379–1387.
- Ngwenya, D., Hancke, G., 2004. Estimation of SRTT using techniques from the practice of SPC and change detection algorithms. In: *AFRICON, 2004. 7th AFRICON Conference in Africa. Vol. 1. pp. 397–402 Vol.1.*
DOI: 10.1109/AFRICON.2004.1406702
- ns-2, 2009. Network Simulator.
URL: <http://www.isi.edu/nsnam/ns/>
- Page, E. S., 1954. Continuous inspection schemes. *Biometrika* 42, 243–254.
- Paxon, V., Allman, M., 2000. RFC 2988 - Computing TCP's retransmission timer.
URL: <http://www.faqs.org/rfcs/rfc2988.html>
- Paxson, V., Floyd, S., Dec. 1997. Why we don't know how to simulate the Internet. In: *Simulation Conference, 1997., Proceedings of the 1997 Winter. pp. 1037–1044.*
DOI: 10.1109/WSC.1997.640988
- Pollak, M., 1985. Optimal detection of a change in distribution. *Annals of Statistics* 13, 206–227.
- Ryu, J.-H., Wan, H., Kim, S., 2010. Optimal design of a CUSUM chart for a mean shift of unknown size. *Journal of Quality Technology* 42, 311–326.
- Seddigh, N., 2000. Performance analysis of TCP's retransmission timeout mechanism. Master's thesis, University of Waterloo.
- Shu, L., Jiang, W., 2006. A Markov chain model for the adaptive CUSUM control chart. *Journal of Quality Technology* 38, 135–147.
- Siegmund, D., 1985. Sequential analysis: tests and confidence intervals. Springer-Verlag.
- Silvey, S. D., 1975. Statistical inference. Chapman and Hall/CRC.
- Sparks, R. S., 2000. CUSUM charts for signalling varying locations shifts. *Journal of Quality Technology* 32, 157–171.
- Tanenbaum, A. S., 2003. Computer network, 4th Edition. Prentice Hall.
- Wireshark, 2011. Network Protocol Analyzer.
URL: <http://www.wireshark.org/>
- Yin, L., Yang, R., Gabbouj, M., Neuvo, Y., Mar. 1996. Weighted median filters: a tutorial. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on* 43 (3), 157–192.
DOI: 10.1109/82.486465
- Yli-Harja, O., Astola, J., Neuvo, Y., feb 1991. Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation. *Signal Processing, IEEE Transactions on* 39 (2), 395–410.
DOI: 10.1109/78.80823
- Zaknich, A., 2005. Principles of adaptive filters and self-learning systems. Springer.