

RED, ALGORITMO DE CONTROL DE CONGESTION EN REDES IP

**Ramiro Algozino, Nahuel Alincaastro, Fernando Corteggiano, Hector Magnago y
Marcelo Gioda**

Universidad Nacional de Rio Cuarto

Palabras Clave: RED, Control de Congestión, TCP, IP.

Resumen.

El control de congestión en redes IP ha sido un problema recurrente en los últimos años. A fines de los años 80, problemas de colapsos por congestión encontrados en los primeros protocolos TCP/IP, llevaron a realizar estudios y nuevas propuesta de los algoritmos de control de congestión extremo a extremo, los cuales forman las bases del actual sistema de control de congestión en redes TCP/IP.

La esencia de este esquema de control está basada en el hecho de que el emisor TCP ajusta su tasa de envío en función de la tasa (probabilidad) de paquetes descartados en la red, la cual es una medida de la congestión de la misma.

Una técnica alternativa para anticiparse a la congestión es la implementación de un algoritmo llamado RED (Random Early Detection) que permite medir los paquetes almacenados en la cola del dispositivo más solicitado y producir el descarte aleatorio en función de esa cantidad sin esperar la información remota.

El objetivo de este trabajo es analizar la convergencia del modelo analítico de RED a través de un sistema de control realimentado realizado en Simulink y compararlo con los resultados obtenidos en el simulador de redes TCP por eventos discretos, NS-2.

Se utiliza este modelo y los requisitos de estabilidad y eficiencia para derivar los parámetros adecuados para la configuración de RED en un rango dado de características de tráfico y de la velocidad de transmisión de la línea.

1. RED, ALGORITMO DE CONTROL DE CONGESTIÓN EN REDES IP

1.1. Introducción

Tradicionalmente la gestión de colas en los routers, se realiza descartando paquetes cuando el buffer se llena. (este mecanismo se denomina Drop-Tail). Recientemente, se han propuesto nuevos mecanismos de gestión de colas, uno de los más populares es RED (Random Early Detection) propuesto por [3]. RED tiene el potencial de resolver algunos de los problemas encontrados en Drop-Tail tales como sincronización de flujos TCP y correlación de los eventos de descarte dentro de un flujo TCP (se descartan múltiples paquetes en secuencia). En RED los paquetes son aleatoriamente descartados antes de que el buffer esté lleno, y la probabilidad de descarte se incrementa con el tamaño promedio de la cola.

RED es un mecanismo poderoso para controlar el tráfico. Puede proveer una mejor utilización de la red que Drop-Tail si se usa apropiadamente, pero puede inducir inestabilidad en la red e importantes perturbaciones del tráfico si no está configurado adecuadamente. La configuración de RED ha sido un problema, en varios estudios se trata este inconveniente [2],[1],[5].

El problema común es que cada configuración propuesta solo es buena para unas determinadas condiciones de tráfico pero puede producir efectos no deseados si es usado bajo otras condiciones.

El presente trabajo simula la convergencia del modelo analítico de RED a través de un sistema de control realimentado como el propuesto en [7] realizado en Simulink y lo compara con los resultados obtenidos en el simulador de redes TCP por eventos discretos ns-2. Se utiliza este modelo y los requisitos de estabilidad y eficiencia para derivar los parámetros adecuados para la configuración de RED en un rango dado de características de tráfico y velocidad de la línea.

1.2. CONTROL DE CONGESTION BASADO EN EL TAMAÑO DE LA COLA VISTO COMO UN SISTEMA DE CONTROL REALIMENTADO

A continuación se analiza la dinámica de control de congestión de TCP en presencia de un módulo de control de congestión basado en el tamaño de la cola [6]. Primero desarrollaremos un modelo de tamaño de cola promedio cuando el flujo TCP pasa a través de un sistema de colas con una probabilidad de descarte fija. Luego verificaremos este modelo a través de simulaciones, a continuación combinaremos este modelo con los elementos de control de RED y derivaremos el comportamiento en estado estacionario del sistema de control realimentado resultante y por último analizaremos la estabilidad del sistema de control RED.

1.2.1. Modelo de Tamaño de Cola Promedio como una función de la media probabilística de descarte de paquetes

En la figura 1 se muestra el sistema de n flujos TCP, que pasan a través de un enlace común l de capacidad c . El flujo f_i , $1 \leq i \leq n$, es establecido entre A_i y D_i , mientras que el tráfico en sentido opuesto consiste solamente de paquetes de reconocimiento (ACKs). El enlace cuello de botella es $B - C$ para cualquier flujo f_i . Se supone también que el número de flujos n se mantiene constante por un largo período de tiempo y que los mismos envían datos por un período prolongado. Se asume que el sistema implementa TCP Reno como el propuesto en [9].

Cada flujo f_i envía a una tasa $r_{s,i}$ los cuales se combinan en el nodo B y generan una cola de tamaño q . El módulo de descarte en el enlace l descarta paquetes con probabilidad p la cual es función del tamaño medio de la cola \bar{q} para cada flujo f_i , los paquetes que no son descartados

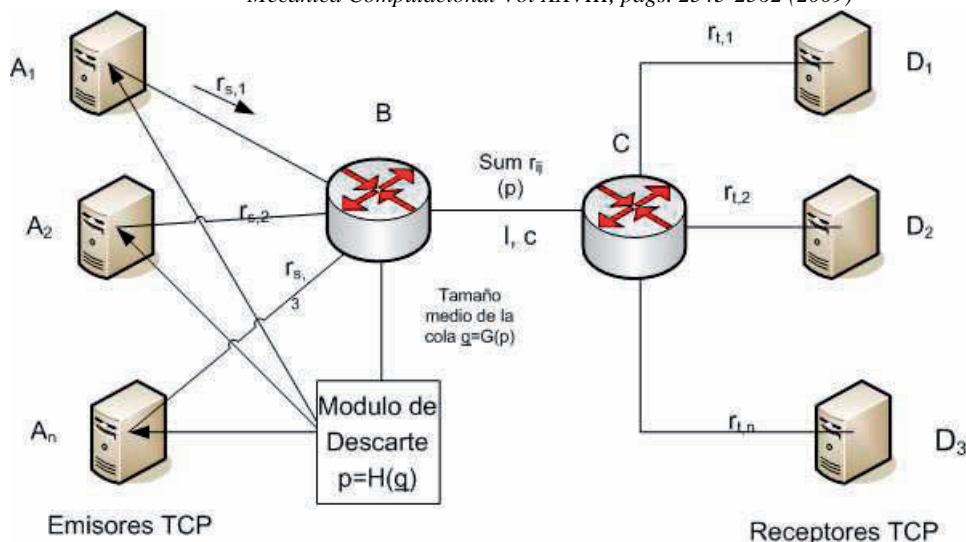


Figura 1: Sistema realimentado con n flujos

son enviados sobre el enlace l a una tasa $r_{t,i}$ que es mas pequeña que la tasa de envío $r_{s,i}$. Cada emisor TCP ajusta su tasa de envío (variando el tamaño de la ventana) acorde a la probabilidad de descarte p .

Este modelo se puede ver como un sistema de control realimentado, los emisores TCP son el sistema controlado, el módulo de descarte es el controlador, la señal de realimentación es la probabilidad de descarte y la variable controlada es la tasa de envío TCP [7].

En este sistema de control el número de flujos (variable controlada) puede variar en el tiempo. El propósito del elemento controlador es mantener el throughput (λ) de todos los flujos por debajo de la capacidad del enlace c .

$$\sum_{j=1}^n r_{tj} \leq c \tag{1}$$

Dado que hemos asumido que el flujo TCP se mantiene por largo tiempo y que el numero de ellos no cambia, el throughput de cada flujo puede ser derivado de [6]:

$$r_{t,i}(p, R_i) = T(p, R_i)$$

donde

$$T(p, R) = \begin{cases} M \frac{\frac{1-p}{p} + \frac{W(p)}{2} + Q(p, W(p))}{R(\frac{b}{2}W(p)+1) + \frac{Q(p, W(p))F(p)T_0}{1-p}} & \text{si } W(p) < W_{\text{máx}} \\ M \frac{\frac{1-p}{p} + \frac{W_{\text{máx}}}{2} + Q(p, W_{\text{máx}})}{R(\frac{b}{8}W_{\text{máx}} + \frac{1-p}{pW_{\text{máx}}} + 2) + \frac{Q(p, W_{\text{máx}})F(p)T_0}{1-p}} & \text{otro} \end{cases} \tag{2}$$

donde T es el throughput de un flujo TCP (en bits/seg) y depende de la probabilidad de descarte de paquetes p , del Round Trip Time medio R , del tamaño medio del paquete M (en bits), del número medio de paquete reconocidos por un ACK b (usualmente 2), del tamaño máximo de la

ventana de congestión soportado por el receptor $W_{\text{máx}}$ (en paquetes) y la duración del Time out T_0 (el cual es típicamente $5R$). W , Q , y F tienen la siguiente expresión:

$$W(p) = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \quad (3)$$

$$Q(w, p) = \min\left(1, \frac{(1 - (1-p)^3)(1 + (1-p)^3(1 - (1-p)^{w-3}))}{3}\right) \quad (4)$$

$$F(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6 \quad (5)$$

Se considera que todos los flujos tienen el mismo RTT, $R_i = R$, el mismo tamaño medio de paquete, $M_i = M$ y que $W_{\text{máx}}$ es lo suficientemente grande como para no influenciar en $T(p, R)$, así tenemos:

$$r_{t,i}(p, R) = r_{t,j}(p, R), \quad 1 \leq i, j \leq n$$

entonces (1) se convierte en:

$$r_{t,i}(p, R) \leq c/n, \quad 1 \leq i \leq n$$

En la figura 2 se muestra el throughput (T) en función de p para el caso de un flujo TCP con $W_m = 12$, $RTT = 470\text{mseg}$, y $T_0 = 3,2\text{ seg}$.

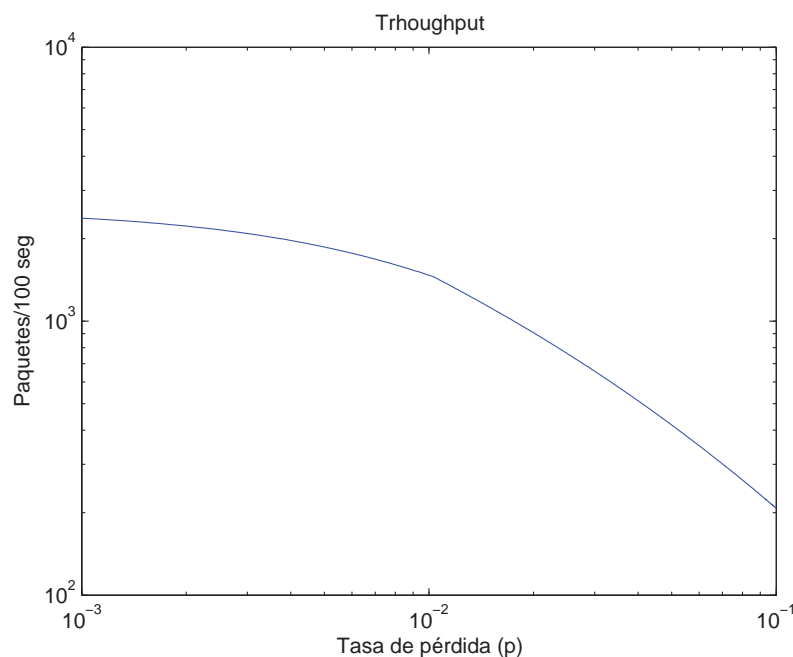


Figura 2: Gráfica de $T(p)$

De esta forma se puede reducir el sistema realimentado de n flujos en uno de un simple flujo como se muestra en la figura 3

Para determinar el estado estacionario del sistema realimentado, es decir obtener los valores medios de r_t , \bar{q} , p , cuando el sistema está en equilibrio necesitamos determinar la función de

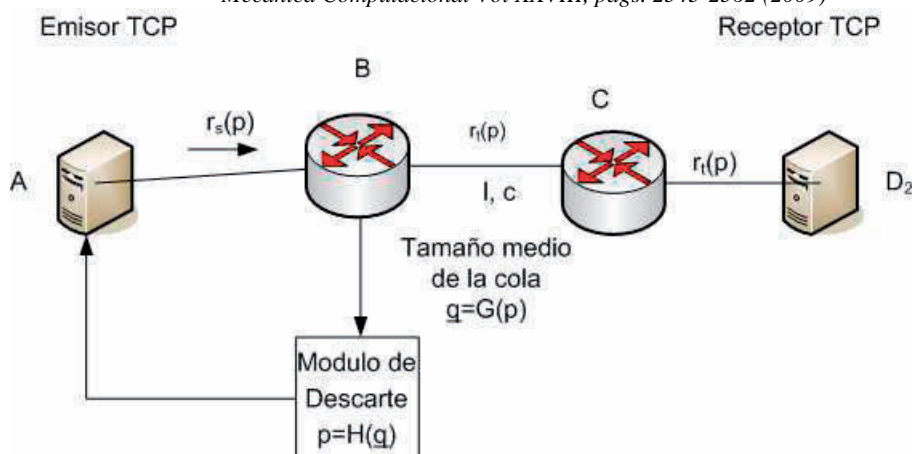


Figura 3: Sistema de control realimentado de un flujo

la cola $\bar{q} = G(p)$ y la función de control $p = H(\bar{q})$. La función de control H está dada por la arquitectura del módulo de descarte, por ejemplo puede ser Drop-Tail o RED.

Para determinar $\bar{q} = G(p)$ se analiza la figura 3 a lazo abierto, es decir sin el módulo de descarte, donde p es el parámetro independiente. Si se asume que l es el único enlace cuello de botella para todos los flujos TCP, el Round Trip Time (RTT) promedio de un paquete es la suma del tiempo de espera medio en la cola (\bar{q}/c) del enlace l más R_0 (tiempo de propagación). Así R queda dado por:

$$R = R_0 + \bar{q}/c$$

Dependiendo del valor de p , el sistema puede estar en uno de dos estados [7]:

1) Para $p > p_0$, el ancho de banda del enlace cuello de botella está subutilizado y

$$r_t(p, R) < c/n,$$

en tal caso, el tamaño medio de la cola es despreciable y $R = R_0$ y la utilización del enlace es:

$$u(p) = \frac{r_t}{c/n} = \frac{T(p, R_0)}{c/n}, \quad p > p_0$$

2) Para $p \leq p_0$, el ancho de banda del enlace está completamente utilizado, es decir $u(p) = 1$, y el tamaño medio de la cola puede ser derivado de la condición $r_t(p, R_0 + \bar{q}/c) = c/n$:

$$\bar{q}(p) = c(T_R^{-1}(p, c/n) - R_0)$$

donde $T_R^{-1}(p, y)$ es la inversa de $T(p, R)$ en R . Si la probabilidad de descarte p es suficientemente pequeña para que $\bar{q}(p) > B$ (tamaño del buffer), entonces se descartan paquetes adicionales debido a overflow del buffer. Obviamente el tamaño promedio de la cola no puede ser mayor que el tamaño del buffer B , así:

$$\bar{q}(p) = \max(B, c(T_R^{-1}(p, c/n) - R_0)), \quad p \leq p_0$$

A continuación se determina p_0 , el valor de probabilidad de descarte al cual el régimen del enlace cambia de subutilizado a completamente utilizado.

Si $p = p_0$, $R = R_0$:

$$r_t(p_0, R_0) = c/n$$

Denotando $T_p^{-1}(x, R)$ la inversa de $T(p, R)$ en p , se tiene que:

$$p_0 = T_p^{-1}(c/n, R_0) \quad (6)$$

Las expresiones del tamaño medio de la cola y la utilización del enlace en función de la probabilidad de descarte p son:

$$\bar{q}(p) = \begin{cases} \text{máx}(B, c(T_R^{-1}(p, c/n) - R_0)), & p \leq p_0 \\ 0, & \text{otro} \end{cases} \quad (7)$$

$$u(p) = \begin{cases} 1, & p \leq p_0 \\ \frac{T(p, R_0)}{c/n}, & \text{otro} \end{cases} \quad (8)$$

Dada una red como la de la figura 1, donde la capacidad del enlace l es $c = 1,5Mbps$, el número de flujos $n = 20$ generados por una aplicación FTP de duracion “infinita”, un RTT $R_0 = 100 \text{ mseg}$ (el cual no incluye el tiempo de espera en la cola), un tamaño medio de paquete de 500 Bytes y un buffer de tamaño $B = 2cR_0$ (se supone que el buffer es lo suficientemente grande para evitar overflow). Para poder obtener el valor de $\bar{q}(p)$ usando la ecuación (7), primero se debe calcular p_0 (6) iterando (2) y evaluando el valor de p_0 para $T = c/n$ y $R = R_0$. El resultado obtenido se muestra en la figura 4.

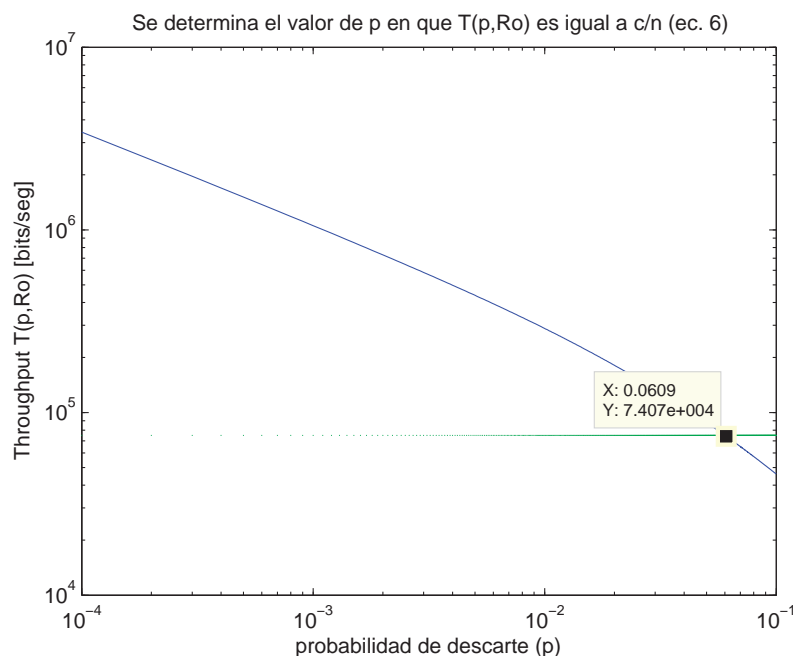


Figura 4: Cálculo de p_0

A partir de p_0 , es posible obtener la función de utilización $u(p)$ (8) mostrada en la figura 5 y el tamaño medio de la cola $\bar{q}(p)$ (7) mostrado en la figura 6.

A continuación se analiza la dinámica de RED como un sistema de control realimentado.

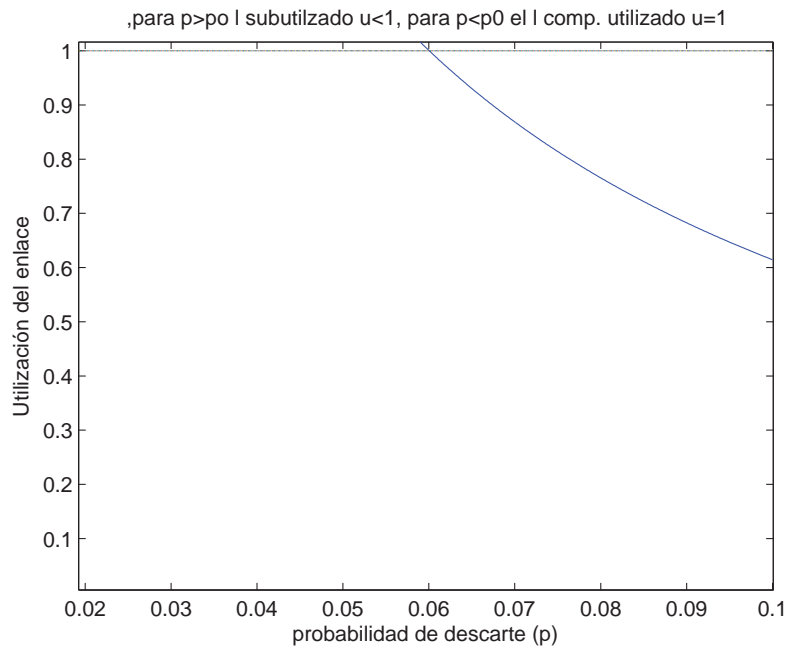


Figura 5: Utilización del enlace

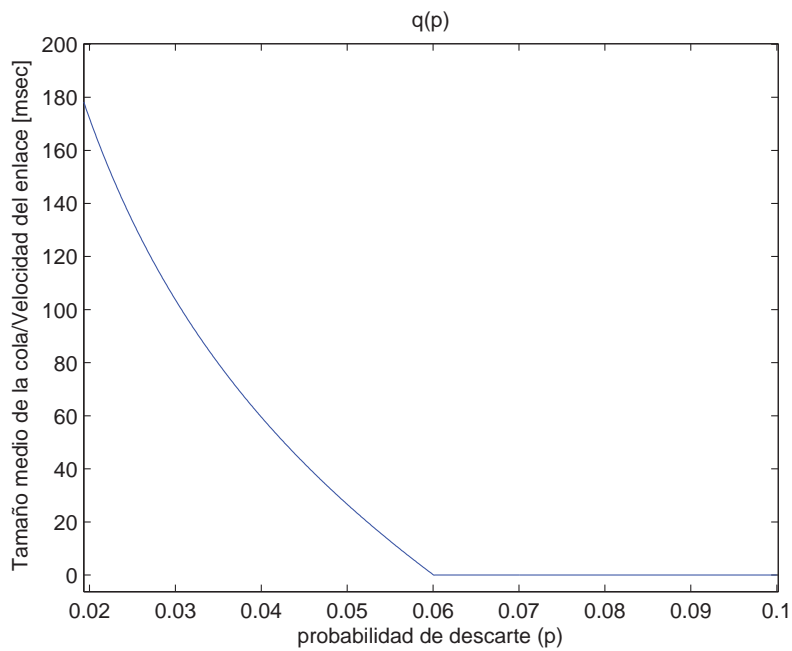


Figura 6: Tamaño medio de la cola para $c=1.5\text{Mbps}$

1.2.2. Operación en estado estacionario de RED

Retornando al sistema de control realimentado de la figura 3, en la sección anterior se obtuvo una expresión del tamaño medio de la cola en estado estacionario en función de la probabilidad de descarte de paquetes, representada por $\bar{q}(p) = G(p)$ dada por (7). Asumiendo que el módulo de descarte tiene una función de realimentación dada por $p = H(\bar{q}_e)$, donde \bar{q}_e es un estimador del tamaño medio de la cola y si el siguiente sistema de ecuaciones tiene una solución única (p_s, \bar{q}_s) , entonces el sistema realimentado de la figura 3 tiene un estado de equilibrio (p_s, \bar{q}_s) .

$$\begin{aligned}\bar{q} &= G(p) \\ p &= H(\bar{q})\end{aligned}\quad (9)$$

El sistema opera en promedio en (p_s, \bar{q}_s) , es decir a largo plazo el número medio de paquetes descartados es p_s y el número medio de paquetes en la cola es \bar{q}_s . En la figura 7 se ilustra el concepto.

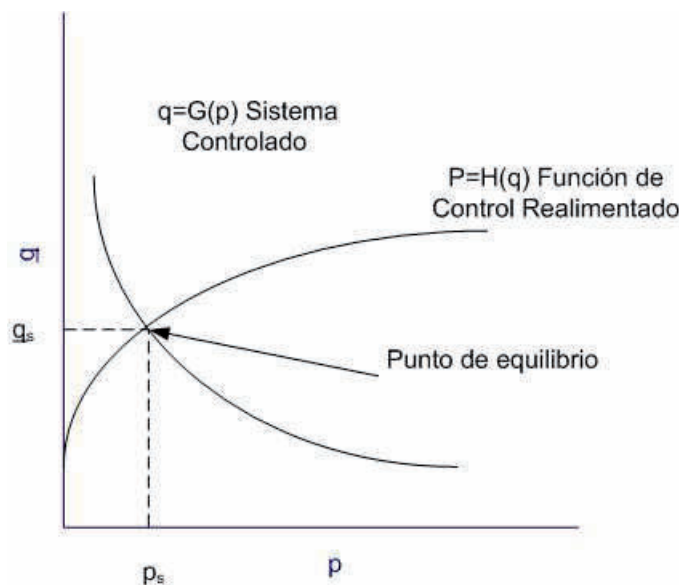


Figura 7: Punto de equilibrio de un sistema de control realimentado

La razón de que el sistema se estabilice en el punto de equilibrio, es que está condicionado por un lado por la ley del tamaño de la cola $\bar{q} = G(p)$ y por otro lado por el módulo de control a través de la función $p = H(\bar{q})$. Es importante considerar que el sistema reside en promedio en el punto de equilibrio y que no necesariamente está todo el tiempo en este estado.

A continuación se aplican estos conceptos para determinar el estado de operación del sistema cuando el módulo de control a utilizar es el algoritmo RED.

En este caso la función de control es:

$$p = H(\bar{q}_e) = \begin{cases} 0, & 0 \leq \bar{q}_e < q_{\min} \\ \frac{\bar{q}_e - q_{\min}}{q_{\max} - q_{\min}} p_{\max}, & q_{\min} \leq \bar{q}_e < q_{\max} \\ 1, & q_{\max} \leq \bar{q}_e < B \end{cases} \quad (10)$$

donde \bar{q}_e es el promedio móvil de la cola ponderada exponencialmente, q_{\min} , q_{\max} , p_{\max} , son parámetros configurables de RED y B es el tamaño del buffer.

Se ha implementado en Matlab un programa para obtener el punto de operación (p_s, \bar{q}_s) como solución al sistema de ecuaciones (10) y (7).

En la figura 8, se muestra la curva del tamaño de la cola dada por (7) y la función de control RED dada por (10), con los parámetros sugeridos en [2] cuyos valores son: $p_{\text{máx}} = 0,1$, $q_{\text{mín}} = 12,5$ paquetes, $q_{\text{máx}} = 37,5$ paquetes, el tamaño del buffer $B = 75$ paquetes y el tamaño del paquete $M = 500$ Bytes.

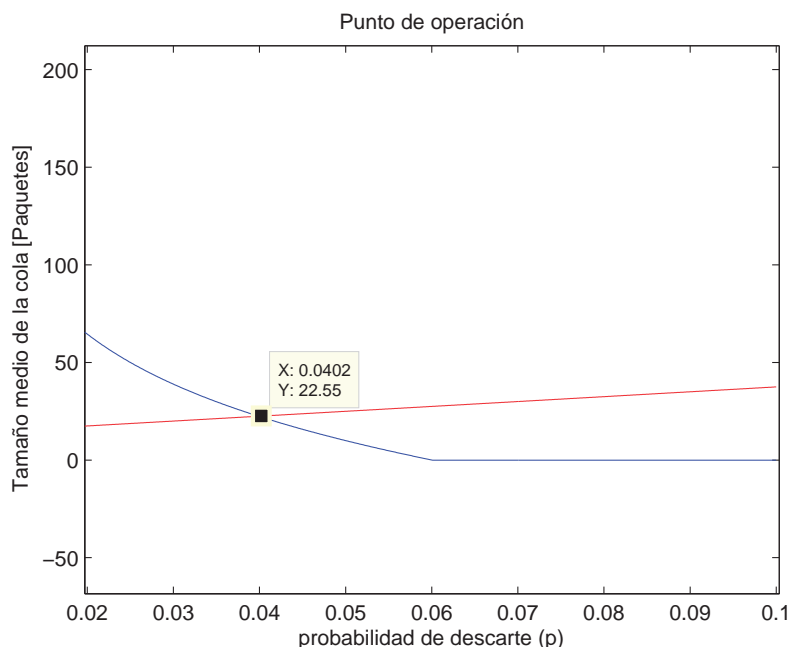


Figura 8: Punto de operación media usando RED, $c=1.5\text{Mbps}$, $n=20$

1.2.3. Operación transitoria del control de congestión RED

En (9) se ha definido un sistema dinámico donde los parámetros de estado son el tamaño medio de la cola y la tasa media de descarte de paquetes. Este sistema puede o no ser estable alrededor del punto de equilibrio dependiendo de las funciones H y G, por lo que se analizará la evolución instantánea del tamaño de la cola en el tiempo, es decir se describirá la dinámica del sistema con RED.

El emisor TCP ajusta su ventana de congestión (por ende su tasa de envío) dependiendo si ha sentido o no descarte de paquetes. Si se descarta un paquete en el enlace l , este evento es detectado en el emisor TCP aproximadamente un RTT después que el paquete ha sido descartado. Por lo tanto el sistema realimentado que se pretende modelar tiene un tiempo de retardo de aproximadamente un RTT entre el momento en que la señal es enviada por el modulo de control y el momento en que el sistema controlado (emisor TCP) reacciona a esta señal. El incremento o decremento en la tasa de envío del emisor TCP produce un incremento o decremento de la longitud de la cola en el enlace cuello de botella l , quien indica al módulo RED que cambie nuevamente su tasa de descarte, y así el proceso se repite.

Se modela el sistema de control realimentado como un sistema de tiempo discreto cuyo paso es un $RTT = R$. Se asume que en el tiempo t_k la probabilidad de descarte es p_k . En el tiempo $t_{k+1} = t_k + R$, el emisor TCP reacciona a p_k , y ajusta la tasa de envío a r_{k+1} . El resultado es que

el tamaño de la cola en t_{k+1} es $q_{k+1} = G(p_k)$ de acuerdo a (7), luego el módulo RED computa una nueva estimación del tamaño de la cola.

$$\bar{q}_{e,k+1} = A(\bar{q}_{e,k}, q_{k+1})$$

donde A es el promedio móvil ponderado exponencialmente (del inglés, EWMA (exponential weighted moving average)):

$$A(\bar{q}_{e,k}, q_{k+1}) = (1 - w)\bar{q}_{e,k} + wq_{k+1}$$

Luego RED actualiza su tasa de descarte $p_{k+1} = H(\bar{q}_{e,k+1})$ de acuerdo a su ley de control expresada en (10), esto es un sistema dinámico en tiempo discreto definido con las siguientes ecuaciones de recurrencia:

$$\begin{aligned} q_{k+1} &= G(p_k) \\ \bar{q}_{e,k+1} &= A(\bar{q}_{e,k}, q_{k+1}) \\ p_{k+1} &= H(\bar{q}_{e,k+1}) \end{aligned}$$

Para realizar un estudio cuantitativo de la evolución transitoria de este sistema dinámico se implementó el modelo en Matlab mostrado en la figura 9.

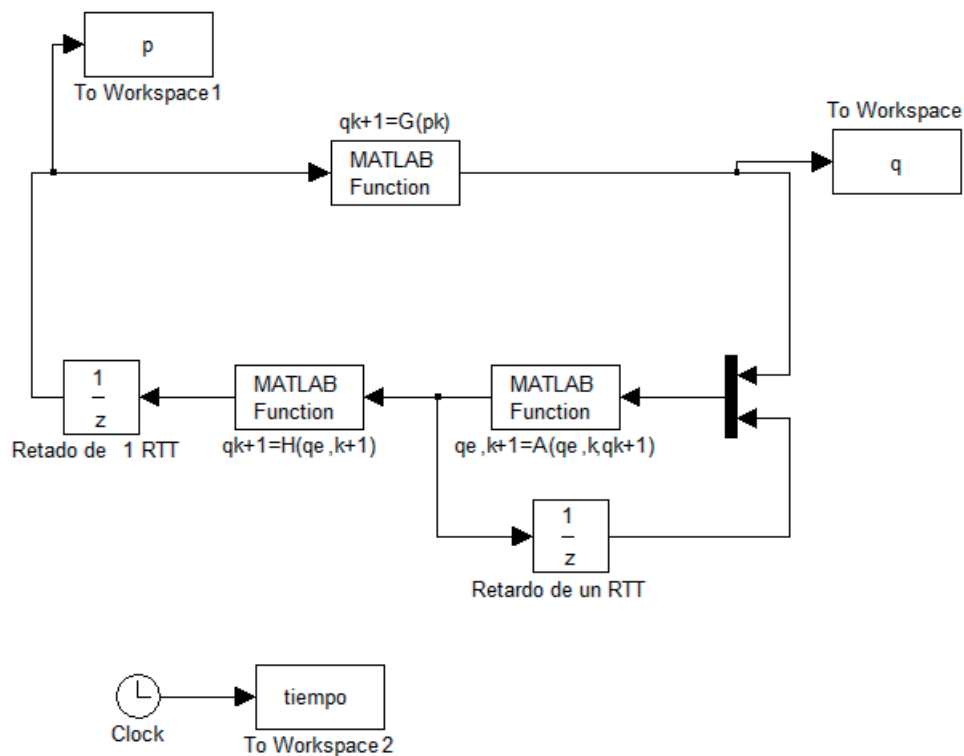


Figura 9: Sistema Dinamico con función de control RED

En la figura 10 se observa como converge el sistema al punto de equilibrio para el ejemplo de la sección anterior y un valor de $w = 0,002$ sugerido en [3], las condiciones iniciales de dicho sistema son $q_0 = 0, \bar{q}_{e,0} = 0$ y $p_0 = 0$.

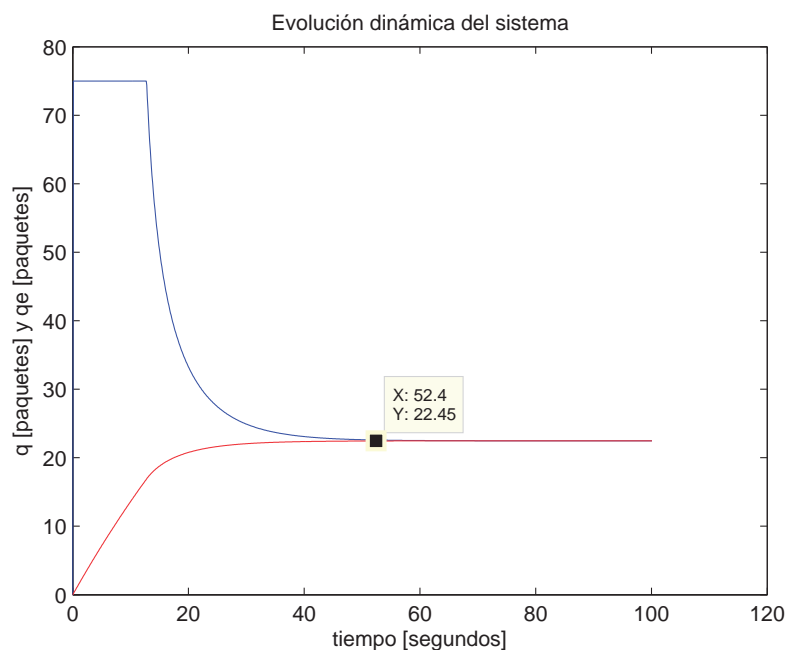


Figura 10: Convergencia del Sistema

Otro análisis es estudiar el comportamiento del sistema cuando el punto de estabilidad está por encima de $p_{\text{máx}} = 0,1$. En la figura 11 se muestra el punto de equilibrio para el caso en que $R_0 = 0,1$, $c = 1,5\text{Mbps}$, $n = 200$ usuarios (para el caso anterior $n = 20$) se observa que el punto de equilibrio $(p_s, q_s) \simeq (0,3431, 37,5)$ está situado por encima de $p_{\text{máx}} = 0,1$. Claramente es un punto inestable dado que para la función de control este valor de p no es un punto de operación definido para RED.

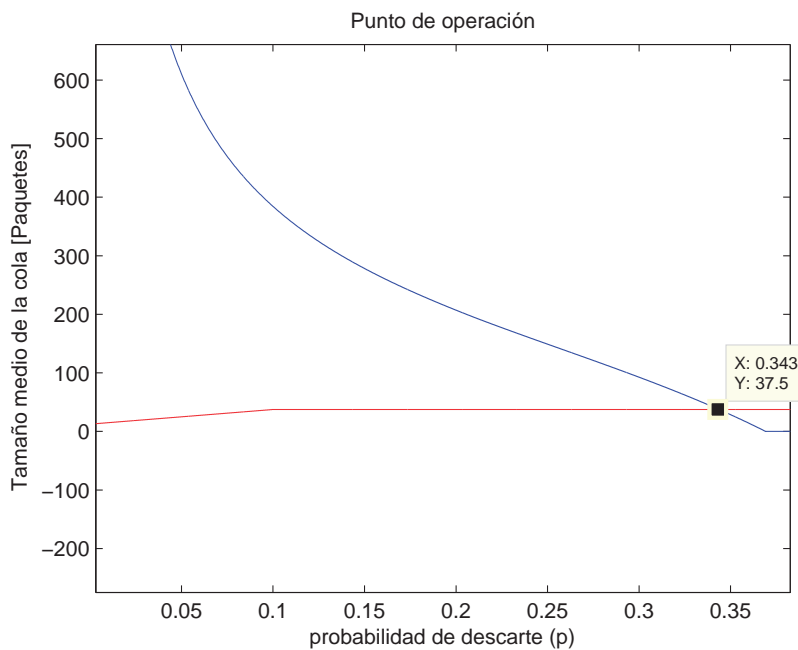


Figura 11: Punto de operación para $c=1.5\text{Mps}$, $R_0=0.1$, $n=200$

Esto produce oscilaciones en el tamaño de la cola entre 0 y el tamaño del buffer B como se observa en la figura 12.

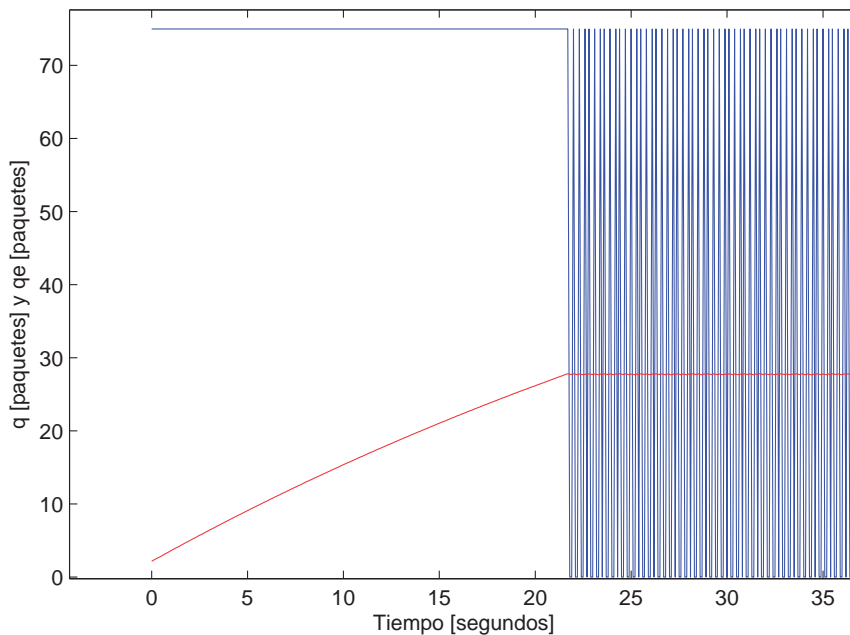


Figura 12: Evolución temporal para $n=200$

Por lo tanto, este tipo de operación es peligrosa y debería ser evitada por medio de una correcta configuración del controlador RED, es decir que tal configuración debe evitar que el sistema trabaje alrededor de $p_{\text{máx}}$ (punto de discontinuidad).

Otro variable que influye en la estabilidad del sistema, o el tiempo de convergencia, es la pendiente entre $q_{\text{mín}}$ y $q_{\text{máx}}$ ($\alpha = \partial q / \partial p$). Si la pendiente de la función de control RED es pequeña, el sistema resulta en una rápida convergencia. Caso contrario, un sistema con gran pendiente, el sistema es más inestable. En la figura 13 se observa el comportamiento con la pendiente modificada $q_{\text{mín}} = 12,5$ a $q_{\text{mín}} = 15,5$. Se observa como el sistema evoluciona más rápido que el comportamiento mostrado en la figura 10.

2. SIMULACIÓN EN NS2

Como continuación del presente trabajo se realizaron dos simulaciones utilizando ns2, para comprobar la exactitud de los modelos analíticos obtenidos en las secciones anteriores. En primera instancia se representó el modelo de la sección 1.2.1 y luego el propuesto en la sección 1.2.2.

Se consideró un sistema de 4 flujos TCP que pasan a través de un enlace común (cuello de botella) de capacidad c y se dirigen hacia un receptor también común (Sink) que produce tráfico en sentido opuesto, el cual consiste solamente de paquetes de reconocimiento (ACKs). Esta arquitectura se muestra en la figura 14.

Los parámetros tanto para las simulaciones como para los cálculos analíticos usados son:

- $c = 1,5$ Mbps
- $R_0 = 6$ ms

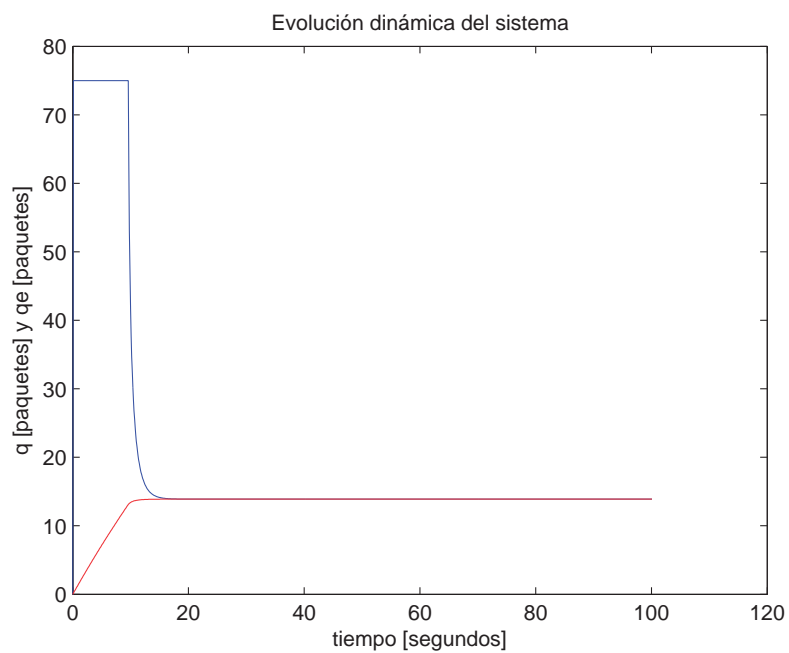


Figura 13: Evolución temporal

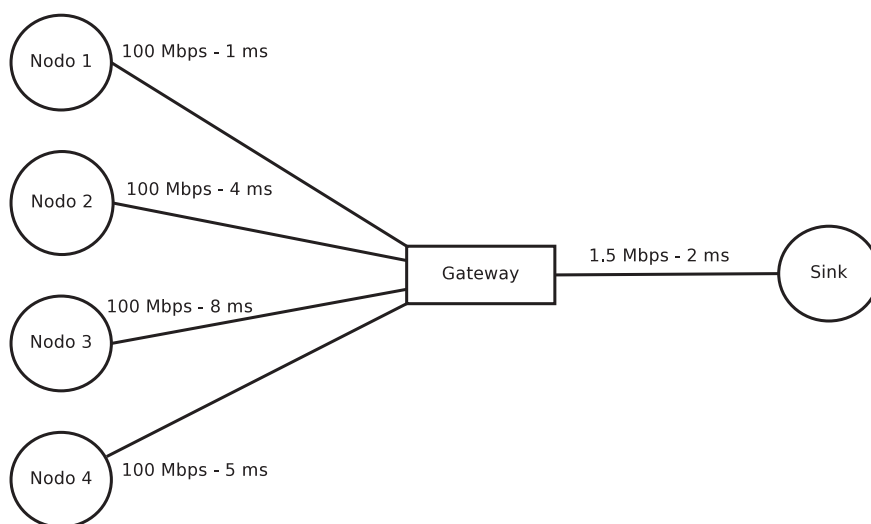


Figura 14: Arquitectura de la simulación

- $n = 4$ (cantidad de nodos)
- $p_{max} = 0,1$
- $min_{th} = 12,5$
- $max_{th} = 37,5$
- $M = 500$ bytes (tamaño del paquete)

2.1. Simulación TCP/Droptail

En esta simulación el buffer de la cola del gateway es del tipo Droptail, con capacidad suficiente para que no haya overflow. Los descartes son producidos por un “módulo de descarte” que desecha paquetes de la cola con una probabilidad definida. Esta probabilidad se va variando desde 0 hasta 0.1 (p_{max}) para generar la curva del tamaño medio de la cola en función de la probabilidad de descarte, la cual representa el modelo del sistema a controlar. Otra posibilidad para generar esta curva es la de ir aumentando progresivamente la cantidad de nodos emisores de paquetes en vez de usar el módulo de descarte y se obtendría una curva de características similares. Nótese que haciéndolo de este modo además de tener que calcular el valor medio de la cola, también habría que calcular el valor de la probabilidad de descarte para cada caso.

2.1.1. Generación de la Curva de Operación

Para graficar la curva, como se mencionó, se varió el valor de la probabilidad del módulo de descarte y se fue generando una tabla con el respectivo valor medio de la cola para cada valor de dicha probabilidad (Tabla 2.1.1), cuyos resultados se muestran en la figura 15. En la figura 16 se compara esta grafica con la curva generada por el modelo analítico sobre el mismo par de ejes.

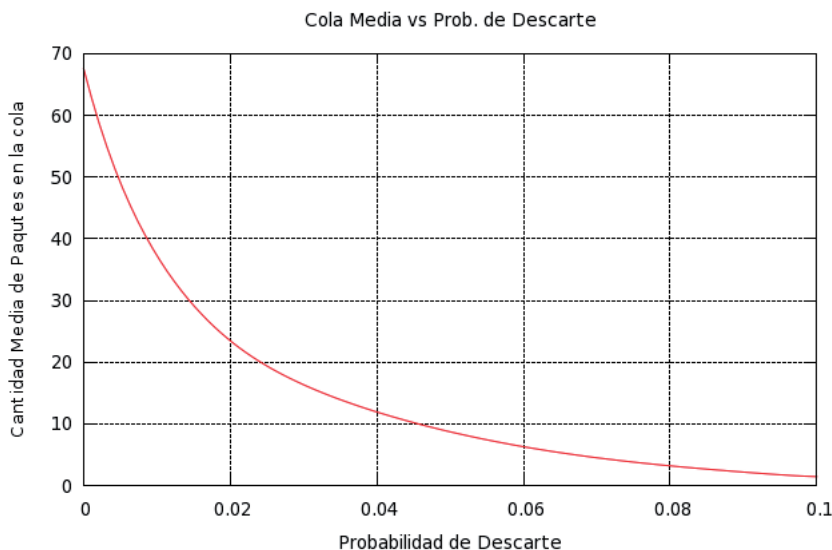


Figura 15: q_{medio} vs p_{medio}

Probabilidad	Cola Media en Bytes
0.0	36504.4
0.005	24027.1
0.01	17676.3
0.015	14530.7
0.02	12136.3
0.02259	10552.6
0.025	9854.23
0.03	8264.27
0.035	6940.94
0.04	6128.44
0.045	5309.02
0.05	4469.28
0.055	3629.46
0.06	3243.12
0.065	2675.54
0.07	2415.48
0.075	1885.76
0.08	1677.77
0.085	1434.4
0.09	1221.71
0.095	877.02
0.1	793.318

Tabla 1: p_{media} vs. q_{medio}

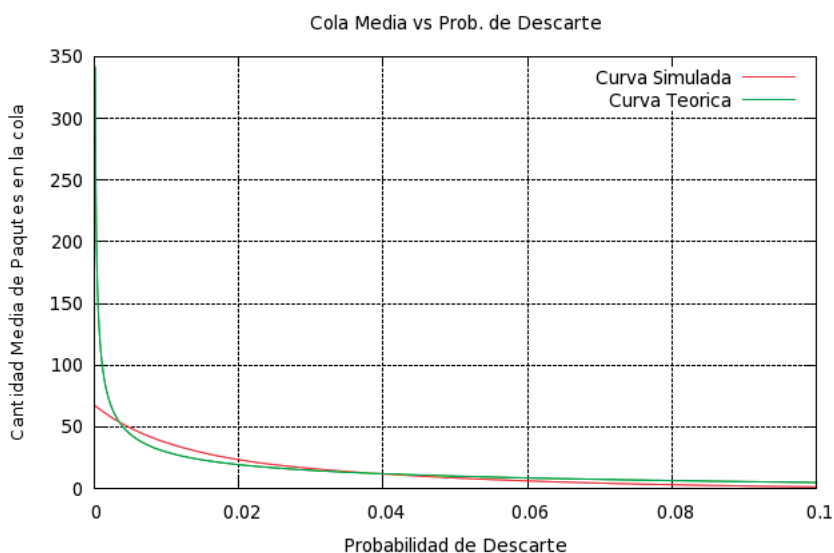


Figura 16: Comparación

2.2. Simulación TCP/RED

Se definió la misma red que en el caso anterior, salvo que ahora el buffer de la cola del gateway, en vez de hacer droptail y de usar el módulo de descarte, usa el algoritmo de control RED, es decir, que al control de congestión de tcp/droptail, se le suman las bondades del algoritmo

red.

Se eligieron los parámetros de operación de RED para la posterior comparación con los valores obtenidos en el modelo analítico. De acuerdo con las recomendaciones de [2] y [3], se eligió el umbral máximo (37.5) como tres veces el umbral mínimo (12.5), la probabilidad máxima de marcado 0.1, y *mean_pkt_size* a 500 bytes para los cálculos de la cola media.

2.2.1. Generación de la Curva de Operación

La curva de operación de RED se muestra en la figura 17. Nótese que la variable independiente de este gráfico es la dependiente del gráfico generado para tcp/droptail. Es evidente en esta figura que cuando la cantidad media de paquetes en el buffer supera el umbral mínimo (12.5 paquetes) aumenta linealmente la probabilidad de descarte. También se ve que no se supera el umbral máximo, y, por lo tanto, la probabilidad nunca llega a *pmax*. En la figura 18 se compara la curva simulada con la curva generada analíticamente.

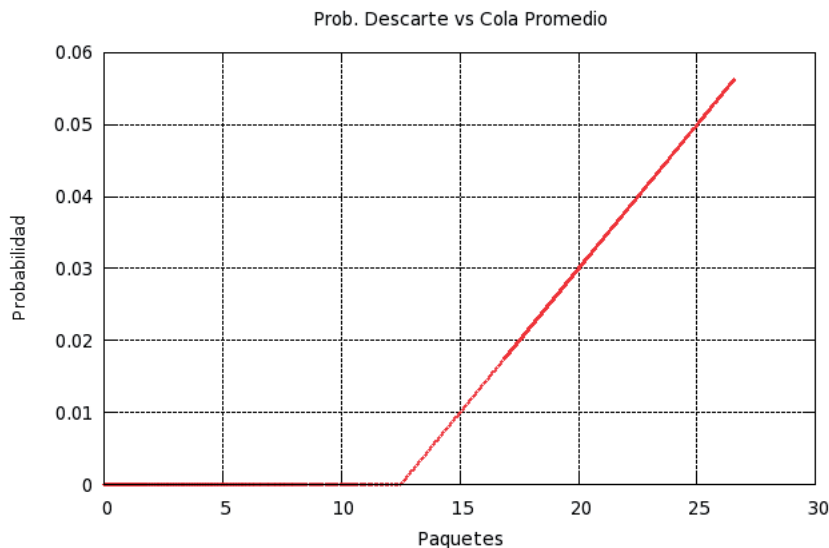


Figura 17: Curva de operación de RED

Observamos aquí que la curva de operación de RED obtenida mediante las simulaciones y la obtenida de acuerdo al modelo planteado en la sección 1.2.2 (calculada con los mismos valores de los parámetros que se usaron en la red anteriormente descrita) se corresponden.

2.2.2. Desempeño de RED

En la figura 19 se puede valorar como se desempeña RED en función de los valores de la cola. Se ve que el valor promedio de la cola (línea de color rojo) se mantiene por debajo del umbral máximo (línea de color rosa) y por encima del umbral mínimo (línea de color azul) cuando se alcanza el estado estacionario. Además se puede apreciar como el peso del promedio móvil permite que las ráfagas de datos, que se pueden observar sobre todo en el período transitorio sobre la curva de la cola instantánea (línea de color verde), que sobrepasan el umbral máximo no son influenciadas por RED y, por lo tanto, no sufren descarte de sus paquetes.

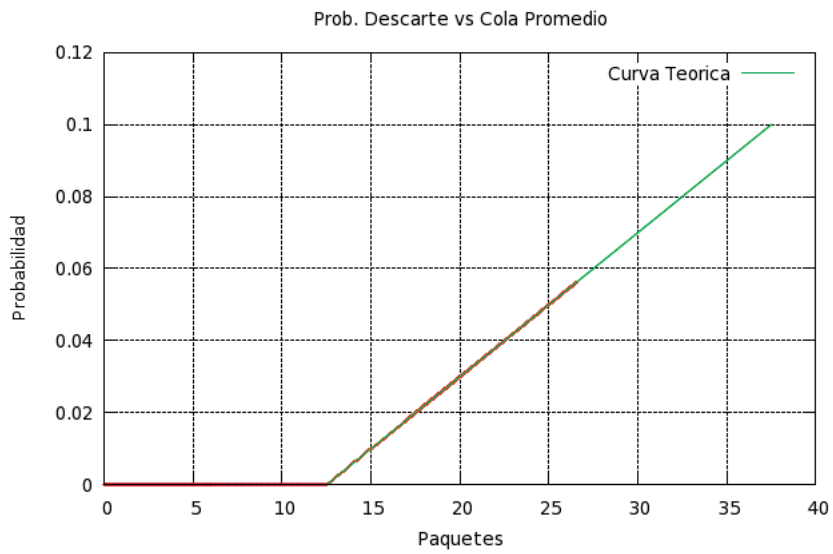


Figura 18: Comparación de las curvas de operación de RED

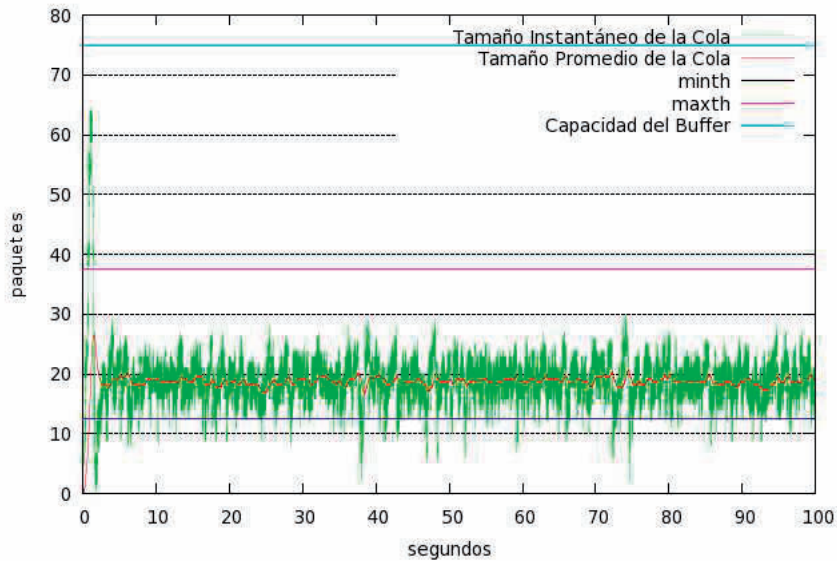


Figura 19: Desempeño de RED

2.3. Punto de Equilibrio

Como se demostró en la introducción teórica, el punto donde se cruzan las dos curvas (la de tcp/droptail y la de tcp/red) es el punto de equilibrio al cual tiende el sistema si está bien configurado. Del resultado de las simulaciones, obtuvimos los siguientes valores:

- q_{medio} : 18.65 paquetes
- p_{media} : 0.0223

La probabilidad de descarte fué calculada como la cantidad de paquetes descartados sobre la cantidad total de paquetes transmitidos, donde paquetes descartados son los descartados en la cola entre el gateway y el sink, y los paquetes totales los calculamos como todos aquellos que llegan al nodo 4.

En la figura 20 se muestran las curvas de operación generadas mediante la simulación. De la misma se desprende que el punto donde se cruzan corresponden, o están muy cercanos a los valores obtenidos.

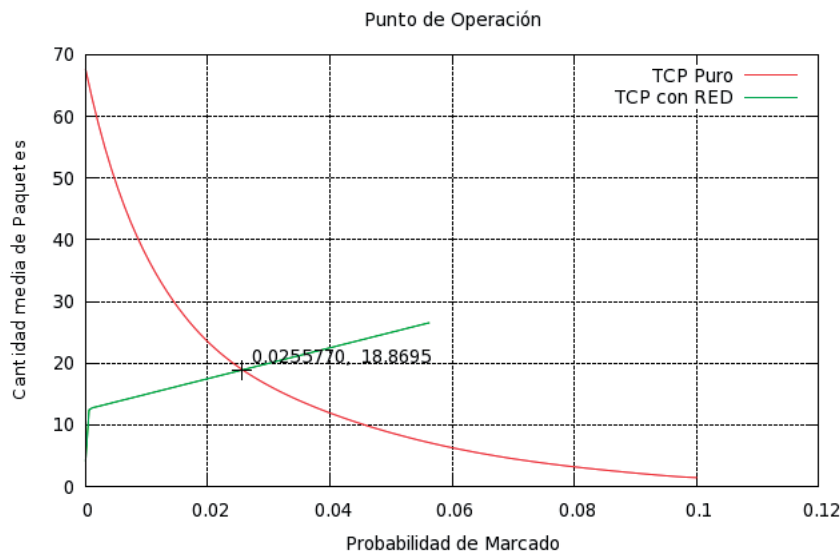


Figura 20: Punto de operación de RED

Los valores de q_{medio} y p_{media} para el caso teórico se obtienen de la figura 21, se observa que el punto donde se cruzan las curvas es:

- q_{medio} : 18.56 paquetes
- p_{media} : 0.0228

Finalmente, se hace una comparación de los puntos de operación teóricos y analíticos, estos valores se ven graficamente en la figura 22.

	Simulado	Analítico
q_{medio}	18.86	18.56
p_{media}	0.0223	0.0228

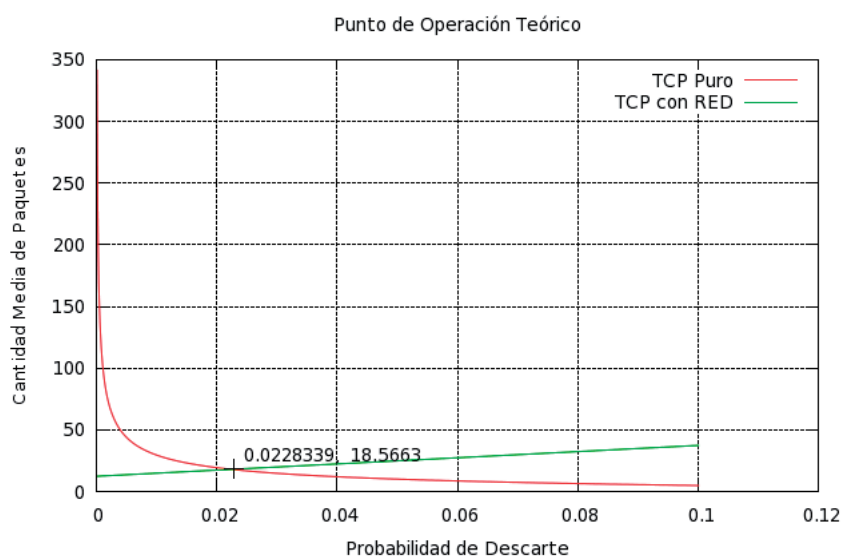


Figura 21: Operación de RED analítica

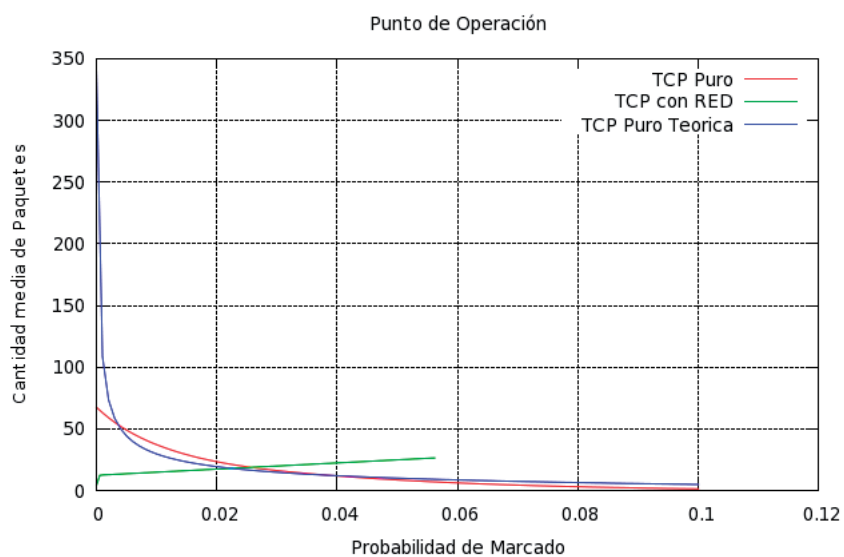


Figura 22: Comparación puntos de equilibrio analítico y simulado.

CONCLUSIONES

Se puede concluir que el sistema de control RED es aplicable siempre y cuando el sistema se mantenga dentro de ciertos parámetros. En caso de que esto no ocurriese se sugiere utilizar un control RED dinámico, es decir que modifique los parámetros de RED (por ejemplo p_{max} , o α) en función del estado de la red (por ejemplo, punto de operación).

Se concluye además que al validar mediante la herramienta de simulación ns2 el modelado de TCP como un sistema de control realimentado, se podría reemplazar el módulo de control RED por algún otro algoritmo que controle el sistema realimentado.

REFERENCIAS

- [1] W. C. Feng, D. Kandlur, D. Saha, and K. Shin. A Self-configuring RED Gateway. In Infmom'99, 1999.
- [2] S. Floyd. Notes on RED in the end-to-end-interest mail list. 1998.
- [3] S. Floyd and V. Jacobson. Random Early Detection gateways for Congestion.
- [4] V. Jacobson and M. 1. Karek. Congestion Avoidance and Control. In SFGCOMM'88, 1988.
- [5] D. Lin and R. Moms. Dynamics of Random Early Detection. In SIGCOMM' 97,1997.
- [6] J. Padhye, V. Fmiu, D. Towsley, and Kuruse. A Stochastic Model of TCP Reno Congestion Avoidance and Control. Technical Report CMPSCI TR 99-02. Univ. of Massachusetts, Amherst, 1999.
- [7] Victor Firoiu, Marty Borden, A Study of Active Queue Managment for Congestion Control. IEEE Infocom 2000.