

Estudio del control de congestión en redes telemáticas mediante el algoritmo RED y su evolución

Claudio Eymard Parra Rojas¹

UNESR. Núcleo Caracas | claudio.parra01@gmail.com

Fecha de recepción; 17 de agosto 2023

Fecha de aceptación; 23 octubre 2023

RESUMEN

El objetivo de este artículo es el estudio del protocolo de comunicación TCP y al gestor de colas llamado RED, para lograr dicho objetivo describiré brevemente la evolución del manejo de la congestión en redes telemáticas, ya que en conjunto regulan la congestión en las redes. Investigando en la Internet encontré que el algoritmo RED, desde sus comienzos en 1993, fue diseñado por Sally Floyd. Sin embargo, surgieron muchas variantes en el tiempo, tanto de RED como de TCP, cada una con algunas mejoras al algoritmo original. Puedo decir que el algoritmo RED viene a aliviar el problema de los enrutadores y el uso de memoria para contener la congestión, que deteriora los servicios ofrecidos por tales redes. En mi trabajo describo diferentes algoritmos llamados AQM tal como es RED y sus variantes, que por razones de espacio no los incluyo a todos. Finalmente llamo la atención de la importancia de estos protocolos en las redes de datos alámbricas e inalámbricas, y sobre algunos de los usos de este algoritmo en las redes telemáticas del presente.

Palabras Clave: RED (Detección Temprana Aleatoria); Control de Congestión; TCP (Protocolo de Control de Transmisión); AQM (Active Queue Management).

¹ Nací en Caracas 1966, y soy venezolano. Graduado de Ingeniero Electricista 1990 UCV y Magister en Ing. Eléctrica 1998 UCV. Soy Consultor en CANTV desde 2007 y Profesor UCAB. He estudiado las telecomunicaciones por más de 30 años.

INTRODUCCIÓN

La congestión en Internet sucede cuando hay demanda de recursos agregada por la fuente que transmite, excediendo la capacidad del recurso produciéndose retardos grandes en la entrega de los paquetes o peor que eso, la pérdida de paquetes. Puedo decir que existen dos conceptos claves:

1. El control de la congestión y
2. La prevención de la congestión.

El control de la congestión involucra mecanismos de diseño y algoritmos que limitan la demanda de capacidad o control dinámico de las fuentes, que son de mucha importancia para los técnicos e ingenieros que tienen que poner en sintonía los equipos de enrutamiento que usan las redes y los protocolos que consideren dichos equipos.

El uso de la red Internet mantiene tráfico TCP en aplicaciones como telnet, FTP, tráfico WEB, correo electrónico, etc. Para los efectos prácticos, podemos decir que, en redes TCP/IP un tiempo fuera o un triple duplicado de paquetes de confirmación se consideran como presencia de congestión.

Mi objeto último de este trabajo, y no menos importante, es contribuir a la sociabilidad en idioma castellano [11] del conocimiento de alto nivel técnico entre profesionales del área de telecomunicaciones y especialmente los administradores de redes de datos, que para la fecha de hoy está incluido en cualquier red el algoritmo RED. Hago énfasis que las variantes de RED también utilizan parámetros similares para la sintonía en una red que cursa tráfico.

TCP Y LA RED DE REDES: LA INTERNET

Los programas que forman redes de datos en una red de computadoras emplean el protocolo TCP para crear conexiones entre sí, de forma que se pueda garantizar el flujo de datos entre los usuarios. Pero, ¿Quién garantiza que los datos lleguen a un destino? A través de este protocolo TCP se asegura que los datos lleguen a su destino en el mismo orden que se transfirieron y sin errores. En la literatura consultada encontré que se manejan cinco capas del modelo de internet y que en el protocolo usado en la capa de transporte (TCP) se utiliza una ventana de flujo adaptativa.

Básicamente puedo decir que el Control de congestión de Jacobson usa 2 algoritmos [4]: 1.- Arranque lento y 2.- Prevención de la congestión. Cuando hay pérdidas de paquetes y se detecta, el tamaño de la ventana disminuye. Esto no es de manera definitiva, ya que más adelante incluyo otras versiones de TCP, las cuales modifican los algoritmos de Jacobson.

Según otras fuentes encontré que las versiones de TCP, usan una reducción particular de las ventanas W [10]. De algunas versiones de TCP puedo decir:

TCP Tahoe: incluye nuevos algoritmos predecesor con arranque lento, prevención de la congestión y retransmisión rápida.

TCP Reno: implementa modificaciones para recuperación rápida.

TCP Vegas: tiene entre un 40 y un 70% más eficiencia que su predecesor.

TCP New Reno: Solamente transmite paquetes luego de una interrupción en estos o después de 3 duplicados que confirman la retransmisión rápida.

TCP Westwood: esta versión modifica el protocolo para determinar el ancho de banda disponible usando esto para recuperación rápida.

TCP Friendly es la más reciente versión y permite ancho de banda igual para los usuarios que usan este algoritmo [2].

Un aspecto importante que encontré al estudiar las versiones de TCP es que la ventana w de TCP da un aspecto clave para el control de la congestión ya que ejecuta la acción de fraccionar el tráfico entrante, como se ve a continuación en la Tabla 1. Para un administrador de red es de suma importancia que conozca la actuación de la versión TCP y cómo reacciona a la congestión. Ellos pueden encontrar en los manuales de los enrutadores estas consideraciones.

Tabla 1. Manejo de la ventana de TCP y sus variantes.
Elaboración propia. 2023

Protocolo	Ajuste de la ventana	Cuando se produce	Reacción a la pérdida
TCP Reno	$w \leftarrow w + 1/w$	Por confirmación	$w \leftarrow w * 0,5$
HSTCP	$w \leftarrow w + w(a)/w$	Por confirmación	$w \leftarrow w + b(w)w$
STCP	$w \leftarrow w + a$	Por confirmación	$w \leftarrow w - bw$
BIC TCP	$w \leftarrow w + \frac{a}{w}, a \in \{S_{min}, S_{max}\}$	Por confirmación	$w \leftarrow w * \beta$
Fast TCP	$w \leftarrow \min \left\{ 2w, (1 - y)w + y \left(\left(\frac{baseRRT}{RRT} \right) * w + a \right) \right\}$	periódico	$w \leftarrow w + 0,5 * w$

FUNDAMENTOS DEL ALGORITMO RED

He encontrado en toda la literatura y documentos consultados que uno de los mecanismos más importantes y utilizado para lograr el control de congestión en redes de internet es el algoritmo RED (Random Early Detection) [4][5][10][12]. La idea que proponen sus creadores es la de usar un servidor con el algoritmo RED programado, y así mantener un promedio de longitud de la cola de paquetes bajo control (esto lo pueden observar los administradores en la memoria buffer a la entrada del servidor). Puedo decir, que lo que hace el algoritmo RED, es que descarta aleatoriamente los paquetes de entrada con una probabilidad proporcional al promedio de la longitud de cola. También podemos decir, que según los creadores del algoritmo RED [4] éste tiene ventajas sobre otros métodos como Drop Tail, las cuales es posible resumir de la siguiente manera:

1. El promedio de la longitud de cola lo mantiene bajo, por lo que el retardo de extremo a extremo de una conexión TCP es bajo también. [4]
2. RED no tiene desventajas para el tráfico en ráfagas. [4]
3. La sincronización global de conexiones TCP ha sido prevista. [4]

Después de haberme familiarizado en RED y lo que hace. Diseñé un modelo analítico, que usado en [4] para las redes de datos y se muestra en la Figura 1 donde los acrónimos tienen el siguiente significado:

Figura 1. Modelo Analítico. Elaboración propia, 2023.

$N(1,...n)$ es el número de conexiones TCP. τ es la latencia. B es la velocidad de procesamiento. $W_n(k)$ es el tamaño

de la ventana del servidor **n** (fuente) y la ranura **k** (ranura es una porción de espacio que ocupa un número determinado de Bits en la señal que se estudia). **F₁-F_n** son las fuentes de tráfico. **S₁-S_n** son los sumideros.

Después de leer varias fuentes, encontré, coinciden en que una dificultad importante para la aplicación del algoritmo RED consiste en que sus parámetros son difíciles de ajustar para la red donde se implementa [4]. Considero que una mala sintonía de RED es que no controla el tráfico creciente de manera que la pérdida de paquetes sea baja. Cada vez que un paquete nuevo llega a la cola del enrutador (Servidor con RED), se estima una longitud promedio de la cola usando un proceso EWNA (Exponencially Weighted Moving Average) utilizando la siguiente ecuación:

$$avg = (1 - w_q) * avg + w_q * q \quad avg = (1 - w_q) * avg + w_q * q \quad (1)$$

Donde:

avg: longitud promedio de cola estimada. **q**: longitud instantánea de cola. **w_q**: factor de peso (constante de tiempo del filtro pasa bajos).

Este valor de **avg** es comparado por dos umbrales **min_{th}** y **max_{th}** que representan el valor mínimo y máximo de la ventana por donde entran los paquetes y son marcados (Figura 2). Si el valor promedio es menor que **min_{th}**, no pasa nada, pero si es más grande que **max_{th}** el paquete es desechado. Para los valores intermedios es marcado con un valor entre 0 y **max_p**. Donde **max_p** es un valor probabilístico que se asocia a los paquetes marcados y es directamente proporcional al promedio de la longitud de la cola.

Figura 2. Algoritmo RED [De elaboración propia, 2023].

Para el estudio de RED, necesitamos definir un modelo analítico donde en [4][12] indica:

1. $q(k)q(k)$ que es el valor actual de la cola medido en paquetes.
2. $\bar{q}(k)\bar{q}(k)$ que es el tamaño de cola promedio en la ranura (espacios de tiempo de la señal) k medido en paquetes.

Asumimos que $q(k)q(k)$ y $\bar{q}(k)\bar{q}(k)$ no cambian durante cada ranura de tiempo (una ranura es un bit). Ésto ha sido confirmado por los autores para valores pequeños de $w_q w_q$ ya que se usa un filtro pasa bajos para calcular $\bar{q}(k)\bar{q}(k)$ (Véase Ecuación 1).

Encontramos también que el servidor que ejecuta el algoritmo RED (Servidor RED), calcula una probabilidad de marcado de paquetes p_b usando el promedio de longitud de cola para los paquetes que llegan. A continuación, se muestra el mecanismo de marcado:

$$p_b(k) \quad p_b(k) \text{ es } 0 \text{ si } \bar{q}(k) \bar{q}(k) < \min_{th} \quad (2)$$

$$\text{es } 1 \text{ si } \bar{q}(k)\bar{q}(k) = \text{ó } > \max_{th} \quad (3)$$

$$p_b(k) = \frac{\max_p * (\bar{q}(k) - \min_{th})}{(\max_{th} - \min_{th})}$$

$$p_b(k) = \frac{\max_p * (\bar{q}(k) - \min_{th})}{(\max_{th} - \min_{th})} \quad (4)$$

Luego, el servidor RED aleatoriamente descarta cada paquete (k) de entrada con probabilidad p_a según la siguiente ecuación:

$$p_a = \frac{p_b}{1 - \text{contador} * p_b}$$

$$p_a = \frac{p_b}{1 - \text{contador} * p_b} \quad (5)$$

El contador indica el número de paquetes no marcados que han llegado desde el último paquete marcado. El mecanismo de marcado de paquetes no es por flujo sino por el número de paquetes. Para encontrar más información sobre RED es posible consultar [4]. Floyd y Jacobson recomiendan según sus experimentos, los siguientes valores para el uso del algoritmo RED. Se puede decir que estos valores son de manera práctica importantes, ya que cuando el administrador de la red pase a configurar a RED los usará o los cambiará dependiendo del resultado del tráfico en la red. (Tabla 2):

Valores recomendados	$\min_{th}=5$	$\max_{th}=15$	$\max_p=0,1$	$w_q=0,002$
----------------------	---------------	----------------	--------------	-------------

Tabla 2. Valores típicos de RED por Floyd, Elaboración propia, 2023

MODELO PARA EL USO DEL ALGORITMO RED

A continuación, se describe cómo se usa RED en una red cualquiera. Estas consideraciones las obtuve después de consultar un experto de la red Internet 2 y me indicó lo que se plantea aquí. Yo me identifico con los administradores de esta red porque me explicaron abiertamente que el algoritmo RED tiene amplia vida aún en esas redes de la Internet 2, inclusive me enviaron un material que me ayudó a comprender la congestión en ese momento. Todo esto completa los estudios en sistemas digitales de telecomunicaciones que inicié en 1990, cuando me diplomaba de Ingeniero.

Los siguientes ejemplos, permiten comprender el uso de RED. Si suponemos una red de 2 nodos X y Y (uno inicial(A) y uno final(B)), el algoritmo RED se usa en el borde de una red, es decir, en la llegada de los datos y programado en el servidor B. Para ajustar su uso debo conocer τ (retardo de X a Y) y B (velocidad de procesamiento) en el enrutador que tiene programado el algoritmo RED. Esto es debido a que el tráfico es aleatorio y el cursado puede tomar cualquier ruta que los protocolos de enrutamiento determinen. Para los usos de RED en una red amplia, se observa: X, la fuente de datos; Y, el sumidero; la ruta que siguen los paquetes de datos en las redes propuestas pasa por otras redes. Figura 3.

Figura 3. Usos de RED en una red. Elaboración propia, 2023

1. Red única pasando por extremo (A) a Extremo(B) sin intermediarios.
2. Red única pasando de (A) a (B) pasando por otros elementos de conexión.

3. Redes separadas por la nube (A)-nube-(B).

(RED siempre va en (B))

VARIANTES DE RED

En la literatura especializada he encontrado varias extensiones del método de RED, que presenta dificultad en la sintonía de los parámetros, y han sido propuestas posteriormente desde que fue presentado este. Puedo mencionar algunas como GRED [5], que usa varios umbrales, siendo más receptivo a la congestión repentina que RED no puede manejar eficientemente, aunque en congestión severa no tiene gran respuesta. ARED [6] tiene un incremento adaptativo de los umbrales resolviendo los problemas de retardo. DRED [3] usa un solo umbral de comparación para el umbral que usan otros. SRED [9] aumenta el valor de la probabilidad de descarte dependiendo de la cantidad de flujos que maneja. Para referencia de uno de los más actuales: EnRED[1] (Enhanced RED) que modifica parámetros de RED considerados, que no estrictos por otros, que sí, para la respuesta a tráfico pesado, también reduce el número de parámetros y sus sensibilidades; y AURED [8] (Autonomus RED for TCP Congestion Control) que presenta un autoajuste con dos períodos de muestreo permitiendo un ajuste más flexible para los escenarios que se presenten. Existen otras versiones de AQM como BLUE [Feng, W.], que han tenido éxito desde su publicación y no serán discutidos aquí. Otras variantes populares son:

ARED. Adaptive RED [6] reduce tanto la pérdida de paquetes como la variación de retraso de la cola al minimizar la posibilidad de mantener el promedio de la cola dentro del rango objetivo y funcionará para no dejar por debajo del 1% y no exceder el 50%.

Esto se hace en un tiempo transitorio aun cuando la longitud promedio de la cola se mueve en la zona de destino.

Stabilized RED [7] está diseñado para estabilizar el tamaño de la cola a nivel independiente del número de las conexiones activas en la red y la probabilidad de caída de los paquetes se hace calculando los flujos activos para ajustar la cola a la instancia del tamaño. SRED tiene un contenedor con una lista de “zombies” con la fuente y la dirección de destino, así como lo hace BLUE.

Flow RED realiza un seguimiento de los flujos que tienen paquetes en la memoria por lo que penaliza el flujo no adaptativo imponiendo un número máximo de paquetes en la memoria. El cálculo del promedio de la cola se hace con los paquetes de llegada y salida para evitar inconvenientes. FRED protege los flujos débiles de manera determinística al aceptarlos bajo una condición de ancho de banda. FRED usa una variable global para estimar el flujo promedio en la memoria.

Dynamic RED estabiliza el tamaño de cola mientras mantiene una alta utilización del enlace y controla la tasa de pérdida de paquetes. La probabilidad de descarte de paquetes es lo suficiente sensible al tráfico por lo que no habrá desbordamiento de la memoria. La operación de DRED es simple y puede ser presentada en el listado de actividades de la red en un tiempo “t”.

Beta RED [7] y Dinámico es la versión más reciente de RED donde se utiliza la distribución probabilística

Beta para la probabilidad de descarte, adecuando a las características de la red donde se va utilizar.

Después de obtener cierta experiencia con los algoritmos de diferentes versiones a la RED original, puedo decir que en general tienen tres formas:

1. Modifican la curva de marcado.
2. Agregan información de la red.
1. Cambian la forma de marcar la probabilidad de descarte.

En el mundo real, he encontrado que algunos fabricantes usan una versión de RED llamada WRED (Weighted RED) que tiene múltiples curvas de transferencia. El efecto de RED original es mejorado por la adaptabilidad que presenta este manejo de varias curvas simultáneas.

A continuación, una clasificación de los gestores de cola. Como he trabajado el tema en varios años puedo decir que actualmente hay muchas variantes, ya que cada año publican diferentes versiones, pero todos se centran en el control de congestión.

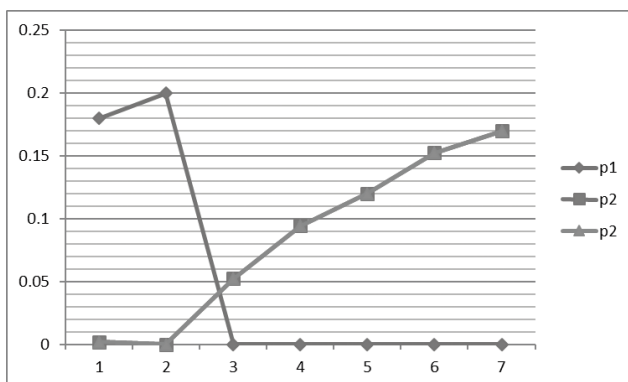
Tabla 3. Clasificación de los gestores de cola. Elaboración propia, 2023

PARTE EXPERIMENTAL

Dentro de mis trabajos he podido realizar aporte de estudios del sistema TCP-RED desde el punto de vista de la congestión y la sintonía de los parámetros del sistema. Aquí voy a introducir un breve estudio de estabilidad con parámetros de una red que utilizar dicho sistema. Para el diseño de los experimentos consideré la variación de N , de 1 a 20; la variación de \max_{con} con valores de 0.01 y 0.05; \min_{th} - \max_{th} con 5-10; B (velocidad de procesamiento) con 2; (τ) (Round Trip Time) con 1; y w_q con 0.02. Todo esto apoyándome en estudios propios anteriores referenciados en [10]

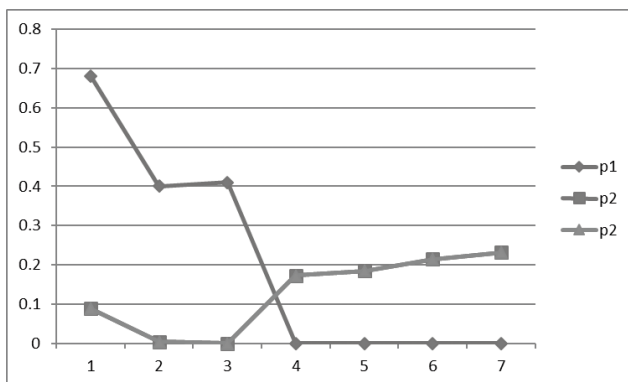
En los casos para el análisis de estabilidad del Sistema TCP-RED, es usada la posición (módulo) de los Autovalores de la Matriz A en el plano z . El sistema es estable si los módulos de los Autovalores son menores que 1 (uno). El círculo unitario no fue representado como tal y se usó un gráfico de línea para mostrar los resultados observándose claramente los valores mayores o menores a 1 para poder concluir sobre la estabilidad del sistema. En las Figuras 1 a la 4 se observa estabilidad total. Estable significa que cursa tráfico normalmente.

Figura 1. Estabilidad del sistema TCP-RED. Elaboración propia, 2023.



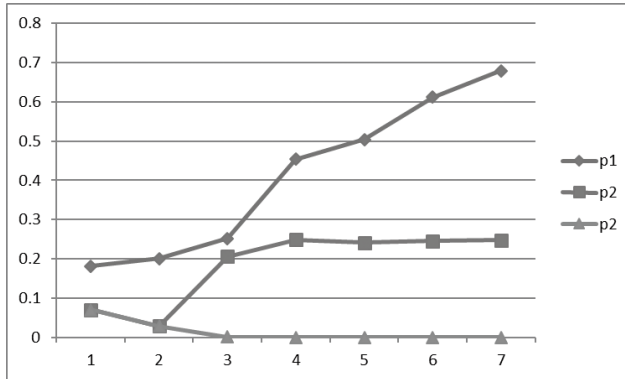
Maxp	min	max	B	tao	wq	x=N,p1,p2,p3.polos.
0,01	5	10	2	1	0,02	beta 0,4

Figura 2. Estabilidad del sistema TCP-RED. Elaboración propia, 2023.



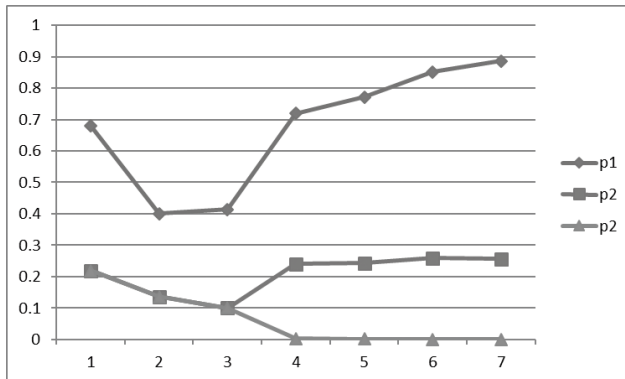
Maxp	min	max	B	tao	wq	x=N,p1,p2,p3.polos.
0,05	5	10	2	1	0,02	beta 0,4

Figura 3. Estabilidad del Sistema TCP-RED. Elaboración propia, 2023.



Maxp	min	max	B	tao	wq	x=N,p1,p2,p3.polos.
0,01	5	10	2	1	0,02	beta 0,5

Figura 4. Estabilidad del Sistema TCP-RED. Elaboración propia, 2023.



Maxp	min	max	B	tao	wq	x=N,p1,p2,p3.polos.
0,05	5	10	2	1	0,02	beta 0,5

CONCLUSIONES

A pesar de que el tema es muy extenso, encontré a lo largo del estudio objetivos comunes como: 1.- Reducción de retardo 2.- Reducción de pérdida de paquetes 3.- Reducción del RTT y otros enfocados en la parametrización y la sintonía del sistema TCP-RED. Todos éstos objetivos tiene un destino final: 1.- optimizar el servicio de internet nivel de la red y el enrutador, 2.- Mejorar la calidad de servicio del usuario final y 3.- Aumentar los anchos de banda del usuario. Todo mediante la estabilidad probada del sistema TCP-RED.

NUEVOS TRABAJOS

Estudiar alternativas para la sintonía de los parámetros de los algoritmos incluidos.

BIBLIOGRAFÍA

- Abu-Sharaha, Admad. (2019) ***“Enhanced Random Early Detection using Responsive Congestion Indicator”***. IJACSA, Vol. 10 No. 3.
- Adly, Mohammad. (2013) ***“Enhanced Congestion Control for Internet Media Traffic”*** Scholar Press.
- Aweya, J. y otros.(2001) ***“A control theoretic approach to active queue management”***, Comp. Network. p. 203-235.
- S. Floyd, y V. Jacobson. (1993) ***“Random Early Detection Gateways for Congestion Avoidance,”*** IEEE/ACM Transactions on Networking, Vol. 1, No. 4, August, pp.397-413.
- Floyd, S. (2000) “Recommendations of using the Gentle Variant of RED.” <http://www.aciri.org/floyd/red/gentle.html> 2000.

- Floyd, S. y otros. (2001) “**Adaptative RED: Algorithm for increasing the robustness of RED’s active queue management**”. AT&T center for Internet research at ICSI.
- Giménez, A. y otros (2022) **New RED-Type TCP-AQM algorithms based on Betha Distribution Drop functions**. Pre-print 2022. arXiv:2201.01105v1 Applied Science. doi. 10.3390/app122111176.[8] Ho, Hsiu-Jy y otros.(2008)” **AURED-Autonomous RED for TCP**”. 3th International Conference on System and Networks Communications. IEEE.
- Ott, T.J. y otros. (1999) **Stabilized RED**. Infocom 99. IEEE.
- Parra R. Claudio E. (2015) “A novel method for stability estimation of TCP-RED systems based on state variable”, Rev. Téc. Ing. Univ. Zulia. Vol. 38, Nº 2,. pp 169 – 177.
- Rodríguez, Simón. (1840). LUCES Y VIRTUDES SOCIALES Obras Completas. UNESR.
- Srikant. R. (2004) **The Mathematics of Internet Congestion Control**. Birkhauser.