

Teoría de las Comunicaciones

Segundo Cuatrimestre de 2017

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Argentina



Lawrence Berkeley
National Laboratory

Electrical Engineering,
Berkeley University

"In October of '86, the Internet had the first of what became a series of 'congestion collapses'. During this period, the data throughput from LBL to UC Berkeley (sites separated by 400 yards and two IMP hops) dropped from 32 Kbps to 40 bps. We were fascinated by this sudden factor-of-thousand drop in bandwidth and embarked on an investigation of why things had gotten so bad."

Van Jacobson, "Congestion avoidance and control", 1988

Congestion Avoidance and Control

Van Jacobson

Lawrence Berkeley Laboratory

Michael J. Karels

University of California at Berkeley

November, 1988



Congestión

Introducción

Bibliografía Básica

- ▶ Computer Networks, Fifth Edition: A Systems Approach (The Morgan Kaufmann Series in Networking) [Larry L. Peterson](#) , [Bruce S. Davie](#) 2011
 - ▶ Capitulo 6 Control de Congestión (CC) y Alocaión de Recursos – págs. 479-499 y TCP CC págs. 499-530

Conveniente complementar con la lectura de:

- ▶ Computer Networks (5th Edition) [Andrew S. Tanenbaum](#) , [David J. Wetherall](#) 2010
 - ▶ Algoritmos de Control de Congestión pags 392-404
- ▶ Jacobson, Congestion Avoidance and Control. Proceedings of ACM SIGCOMM '88, Aug. 1988.
- ▶ M. Mathis, J. Semke, J. Mahdavi, T. Ott, “The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm”, ACM Computer Communication Review, Volume: 27, Issue: 3, July 1997

Agenda

- ▶ Introducción: multiplexación estadística, buffers y congestión
- ▶ Definiciones de Congestión
- ▶ Soluciones

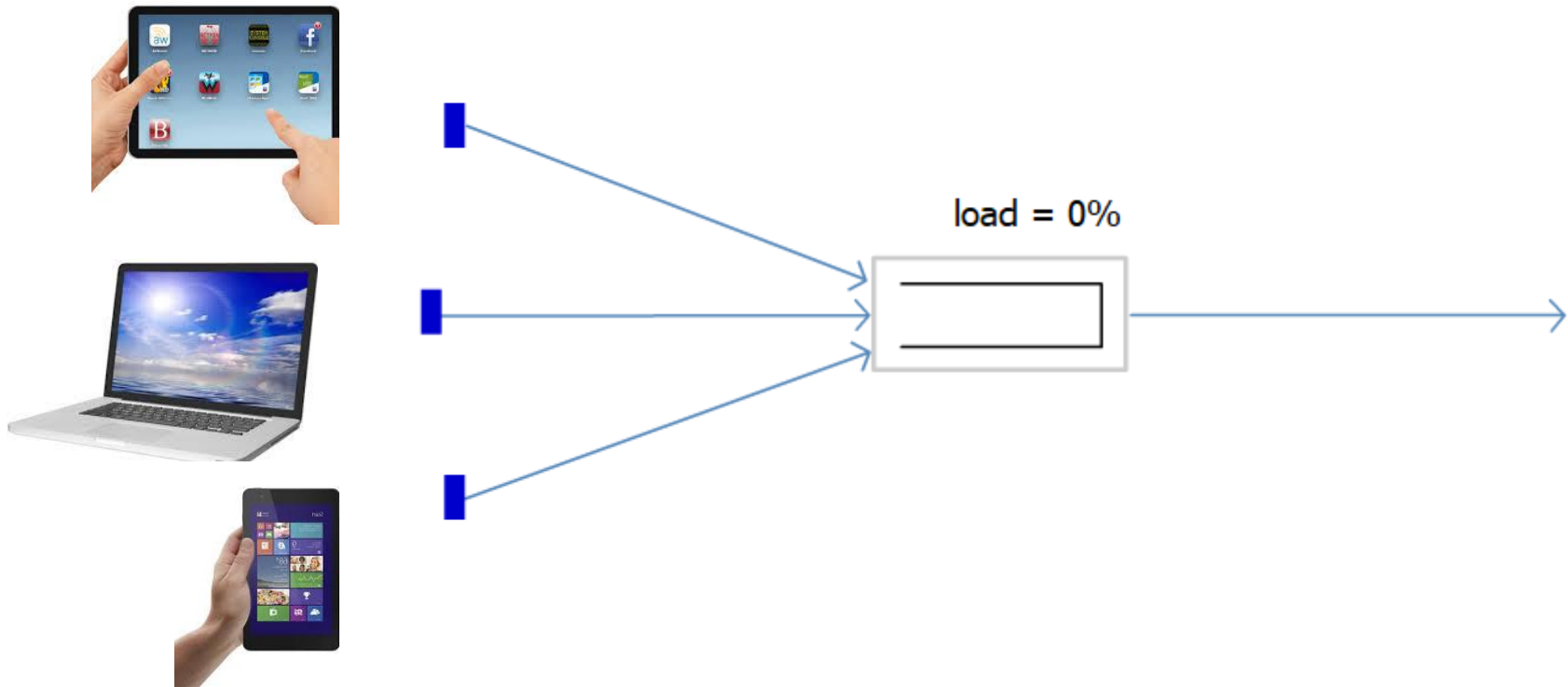


Introducción



Parte 1

Multiplexación Estadística



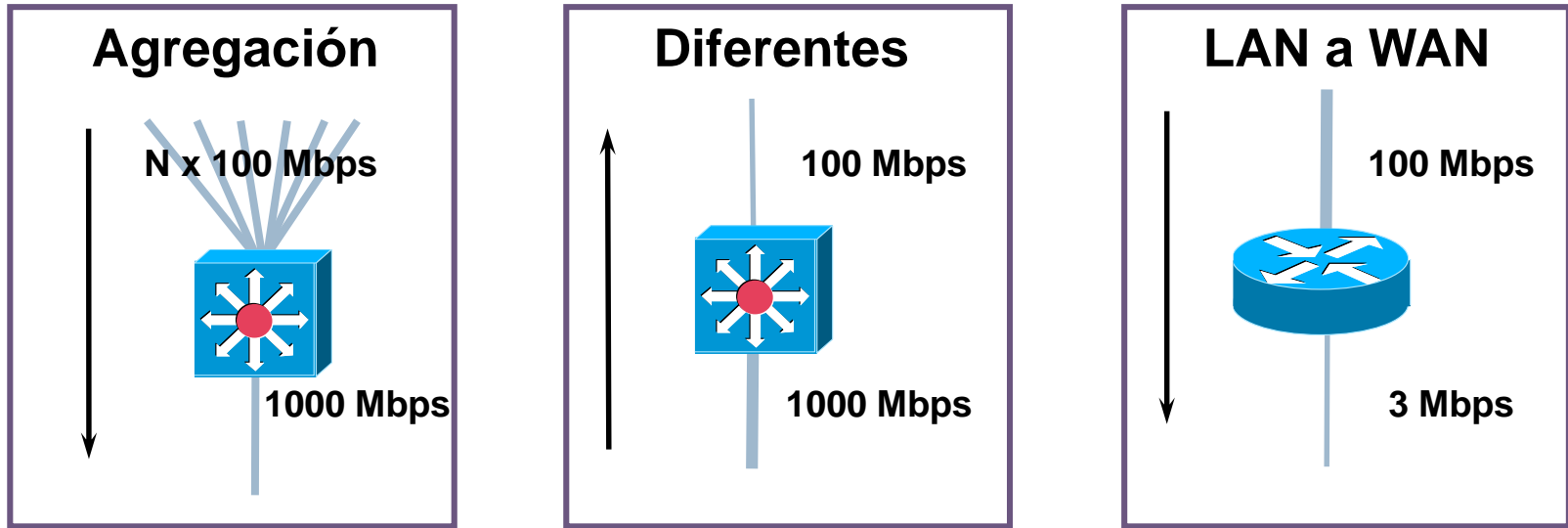
- ▶ Tenemos que pensar en términos de distribuciones estadísticas.
 - ▶ Ejemplo: El llenado de un buffer es de un 90% “en promedio”

La Naturaleza de Internet

Es una red con recursos compartidos ! ➡
Existe:



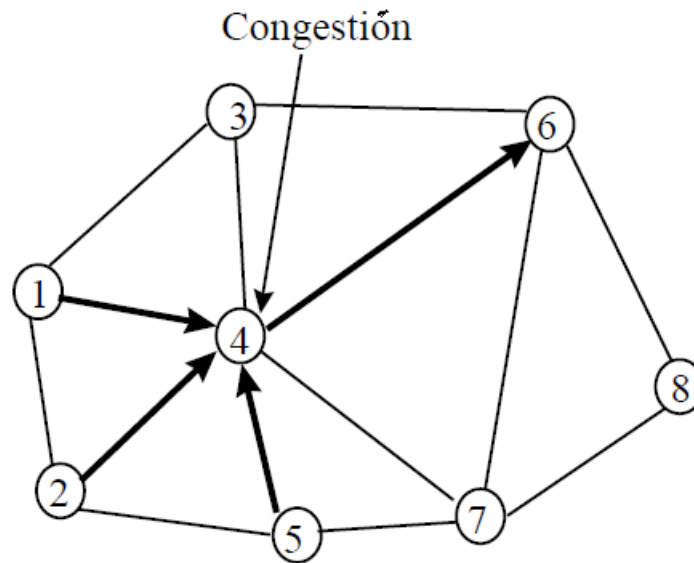
Administración de Buffers



- ▶ Buffers: Tendencia a llenarse $\text{Delay} = T_{\text{tx}} + T_{\text{prop}} + T_{\text{queue}}$
- ▶ **Buffering**: reduce pérdidas, pero incrementa el Delay
- ▶ Buffer Overflow: → **Sobrecarga** → se descartan paquetes → **Retransmisiones** → QoS degradado.
- ▶ Para garantizar QoS: se debe **reservar espacio** en los buffers.

Congestión

- ▶ Estado de **sobrecarga sostenida** de una red, donde:
 - ▶ La demanda de recursos (enlaces y buffers) se encuentra al límite o excede la capacidad de los mismos
 - ▶ La consecuencia es perceptible en términos de QoS degradada



Soluciones ?

- ▶ Sobredimensionamiento (overprovisioning)
- ▶ Diseño cuidadoso
- ▶ Control proactivo: Evitar (control preventivo)
 - ▶ Decrementar la carga ;-)
 - ▶ Cómo “convencemos” a un usuario desconocido que utilice menos la red ?
 - ▶ Ancho de banda asignado por un proveedor X a una empresa Y
 - ▶ Premiar el “uso parejo” y/o castigar el “uso excesivo”
 - Consecuencia: Algunas páginas web nos “distraen” con ofertas inverosímiles

Congestión desde la perspectiva del:

► Operador de Red

“For a CMTS (Cable Modem Termination System) **port** to enter the **Near Congestion State**, traffic flowing to or from that CMTS port must exceed a **specified level** (the “**Port Utilization Threshold**”) for a **specific period of time** (the “**Port Utilization Duration**”)

The **Port Utilization Threshold** on a CMTS port is measured as a **percentage of the total aggregate upstream or downstream bandwidth** for the particular **port** during the **relevant timeframe**. The **Port Utilization Duration** on the CMTS is measured in minutes“

► Usuario

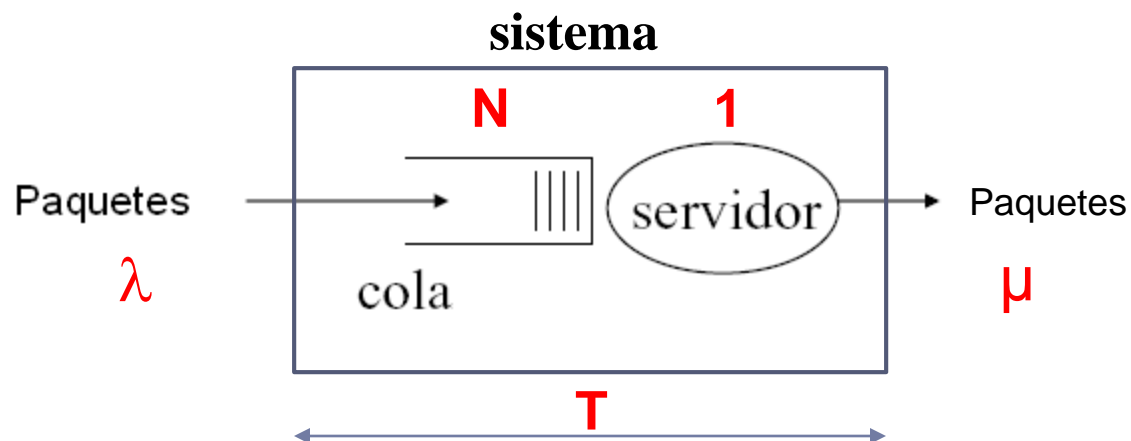
“A network is said to be congested from the perspective of a user if the **service quality noticed by the user** decreases because of an **increase in network load**”

Análisis de Congestión

Caso de un Buffer

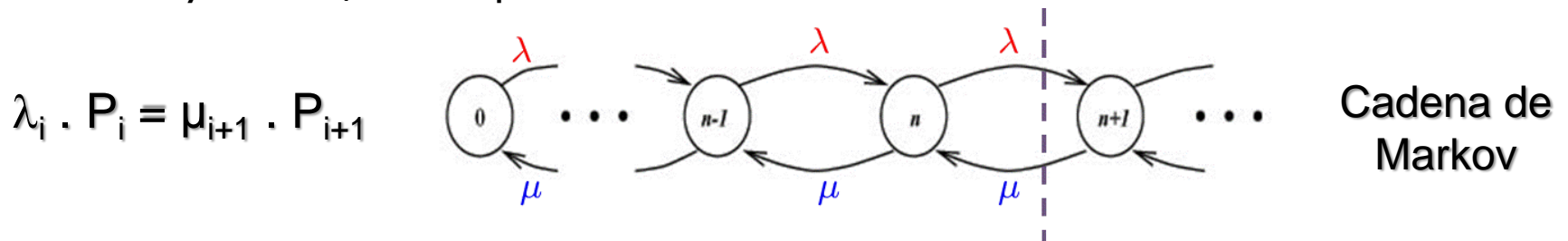
Teoría de Colas: Sistema M/M/1

- ▶ Es un sistema cola-servidor en el cual el **número de llegadas** de paquetes es un **proceso de Poisson** y el **tiempo de servicio** de paquetes es un **proceso Exponencial**.
- ▶ Hay un **único servidor**
- ▶ La cola tiene **capacidad infinita**



Teoría de Colas : Notación

- ▶ N : cantidad de paquetes en el sistema (paquetes)
- ▶ T : tiempo de estadía de cada paquete en sistema (seg)
- ▶ λ : **tasa de entrada** de paquetes a procesar en el sistema (paquetes/seg)
- ▶ μ : **tasa de servicio** de paquetes procesados por el sistema (paquetes/seg)
- ▶ $1/\lambda$: tiempo medio entre llegadas (seg/paquetes)
- ▶ $1/\mu$: tiempo medio de servicio (seg/paquetes)
- ▶ $\rho = \lambda/\mu$: **Intensidad del sistema**
 - ▶ Si y solo si $\rho < 1$ se puede alcanzar un **estado estacionario**



- ▶ P_N : probabilidad de que haya N paquetes en el sistema (**“estado N”**)
- ▶ Si tenemos que la tasa de entrada es mayor a la tasa de de servicio de la cola:
 $\rho > 1$: se desborda el sistema (Congestión)

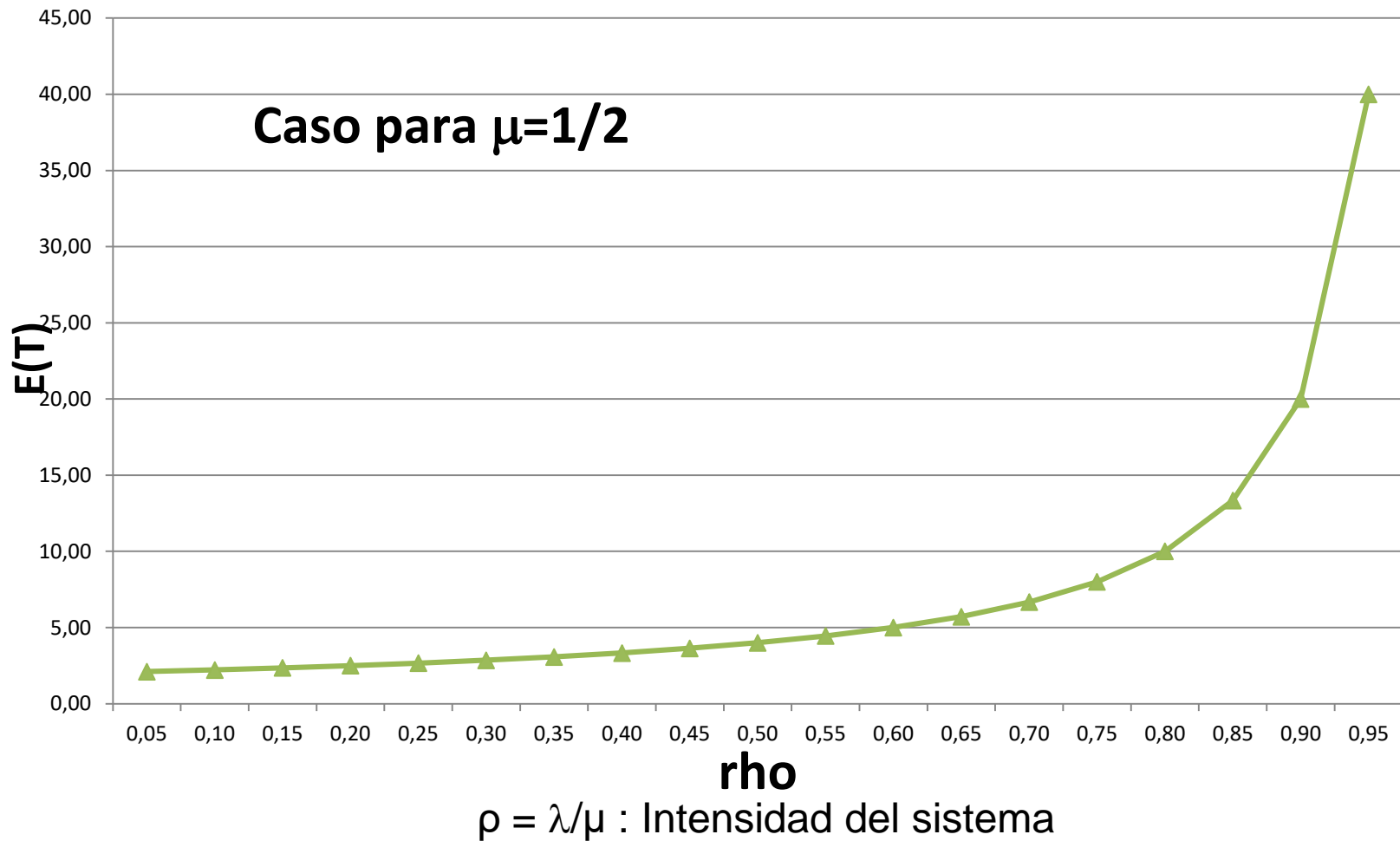
Esperanza de N y T

$$E(N) = \frac{\rho}{1-\rho}$$

$$E(N) = \lambda \cdot E(T)$$

$$\Rightarrow E(T) = \frac{E(N)}{\lambda} = \frac{\rho}{(1-\rho) \cdot \lambda} = \frac{\frac{\lambda}{\mu}}{(1-\frac{\lambda}{\mu}) \cdot \lambda} = \frac{\frac{\lambda}{\mu}}{\frac{\lambda \cdot (\mu - \lambda)}{\mu}} = \frac{1}{\mu - \lambda}$$

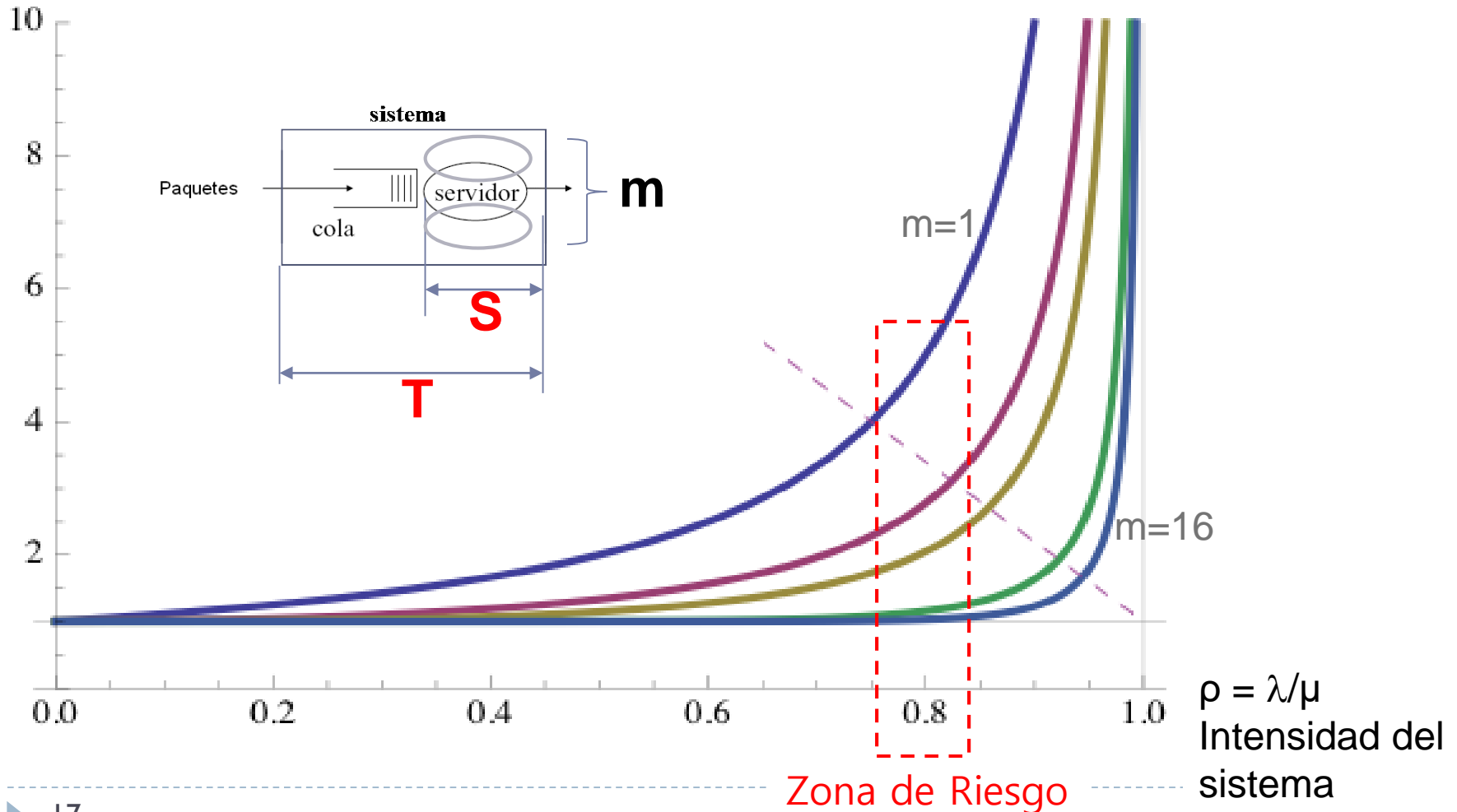
Delay promedio M/M/1



Delay medio. M/M/m

$m=1,2,3,9,16$

T/S ← Tiempo de Respuesta **Normalizado**





Fundamentos

Fundamentos del control de la congestión

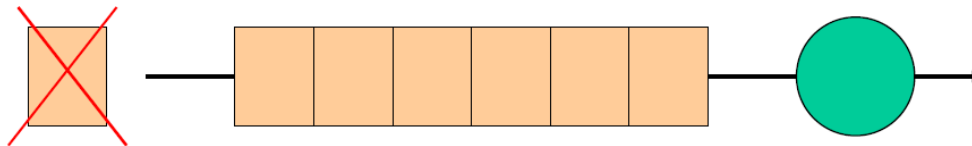
Congestión:

- ▶ Informalmente: “demasiadas fuentes utilizando una **red compartida** enviando demasiados datos demasiado de prisa como para poder ofrecer una buena calidad de servicio”
- ▶ Síntomas:
 - ▶ Pérdida de paquetes (Los buffers se saturan en routers o switches)
 - ▶ Retardos crecientes (por las colas en los buffers)
- ▶ El **control de congestión** ataca un problema muy diferente al **control de flujo**.

Consideraciones sobre los nodos

- De no expresarse lo contrario asumimos:

Drop Tail Router [FIFO]



- **FIFO**: el primer paquete que llega se transmite
 - es un mecanismo de **scheduling**
- **Drop Tail**: Se descartan paquetes cuando se llena la cola
 - es una política de **manejo de colas**.
- Combinados, producen una **sincronización global** cuando los paquetes descartados provienen desde **distintas conexiones, no sincronizadas** entre ellas (lo cual es casi siempre)

Congestion Control vs. Flow Control

- ▶ Congestion Control: debería poder evaluar la capacidad de una **subred completa** para transportar determinado **tráfico agregado**.
 - ▶ Congestión es un **efecto global**, involucra a todos los hosts y routers compartiendo una subred
- ▶ Flow Control: controla **tráfico punto-a-punto** entre un receptor y un transmisor particulares.

Métricas

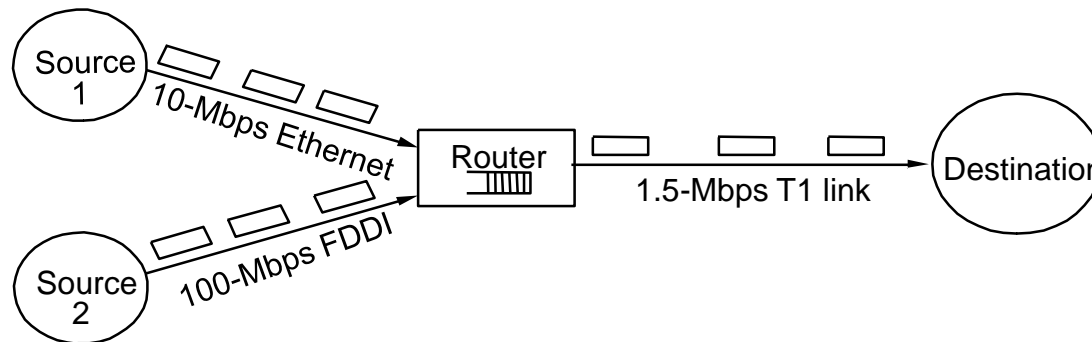
- ▶ **Varias métricas posibles para detectar congestión:**
 - ▶ % de paquetes descartados por falta de espacio en buffer
 - ▶ longitud media de una cola (*buffer*)
 - ▶ # de paquetes que generan *time out* y son retransmitidos
 - ▶ *average packet delay*
 - ▶ *standard deviation of packet delay*
- ▶ **El crecimiento de alguna o varias de estas métricas indican congestión**

Políticas que influyen en la congestión

Capa	Políticas
Transporte	<ul style="list-style-type: none">• Política de retransmisión• Política de almacenamiento en caché de paquetes fuera de orden• Política de confirmaciones de recepción• Política de control de flujo• Determinación de terminaciones de temporizador
Red	<ul style="list-style-type: none">• Circuitos virtuales vs. datagramas en la subred• Política de encolamiento y servicio de paquetes• Política de descarte de paquetes• Algoritmo de enrutamiento• Administración de tiempo de vida del paquete
Enlace de datos	<ul style="list-style-type: none">• Política de retransmisiones• Política de almacenamiento en caché de paquetes fuera de orden• Política de confirmación de recepción• Política de control de flujo

Consideraciones

- ▶ **Control de Congestión:** Es el esfuerzo hecho por los nodos de la red para prevenir o responder a sobrecargas de la red que conducen a pérdidas no controladas de paquetes.
- ▶ Los dos lados de la moneda:
 - ▶ Pre-asignar recursos (ancho de banda y espacio de buffers en routers y switches) para evitar la congestión → **Subutilización**
 - ▶ Liberar recursos y controlar la congestión sólo si ocurre (y cuando ocurra) → **A quien perjudicar ?**



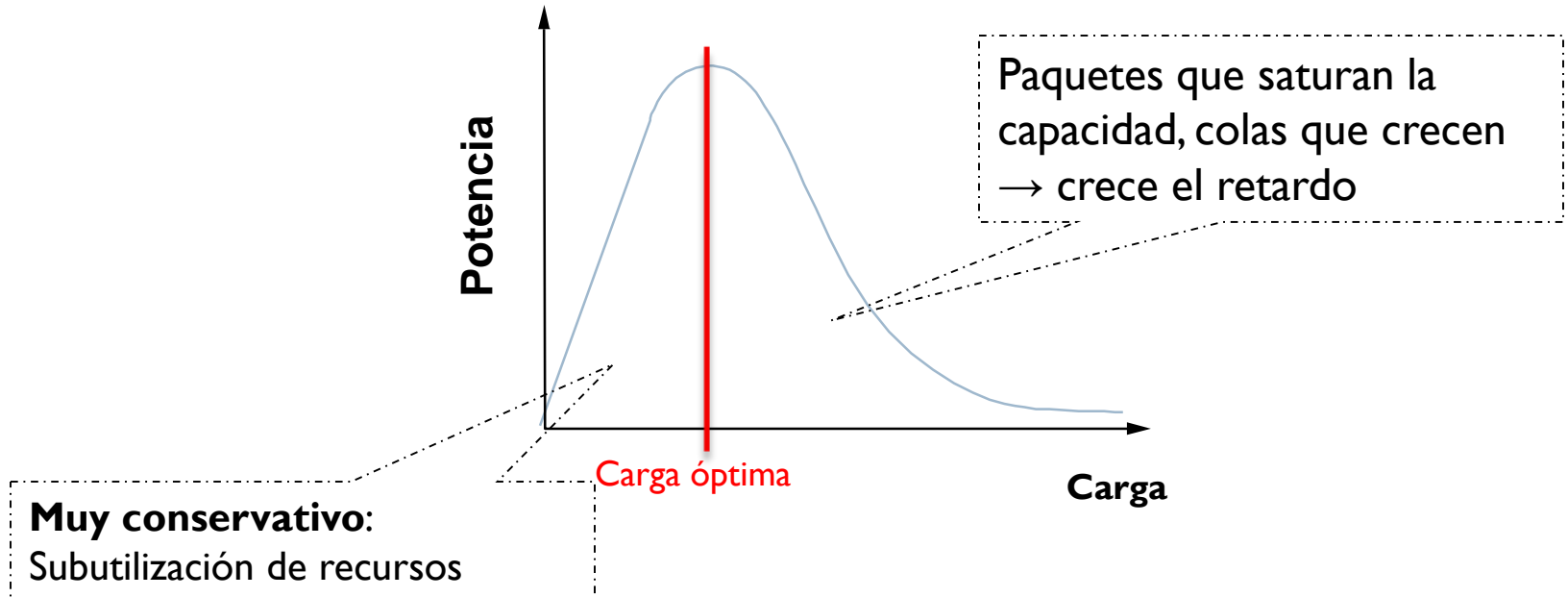
- ▶ **Solución:** Permitir el uso de recursos de la red en forma “**equitativa**”
 - ▶ cuando haya **problemas: compartir sus efectos** entre todos los usuarios, en lugar de causar un gran problema a tan solo unos pocos.

Consideraciones (cont)

- ▶ Control de flujo vs. control de congestión
 - ▶ El primero previene que los transmisores sobrecarguen a **receptores lentos**.
 - ▶ El segundo evita que los transmisores sobrecarguen **el interior de la red**.
- ▶ Dos puntos para su implementación
 - ▶ Hosts en los extremos de la red (protocolo de transporte)
 - ▶ Routers dentro de la red (disciplina de encolado, e.g. Random Early Detection, etc.)
- ▶ Modelo de servicio de los niveles inferiores
 - ▶ Best-effort o mejor esfuerzo (lo asumimos por ahora). Es el modelo de Internet.
 - ▶ Múltiples *calidades de servicio* QoS.
 - ▶ Por ejemplo ancho de banda (para video streaming) y retardo (para Voz sobre IP VoIP).

Criterios de Evaluación (1)

- ▶ La idea es que la red sea **utilizada eficientemente** y al mismo tiempo en forma **equitativa**
- ▶ Un buen indicador para “eficiencia”:
Potencia = throughput / delay



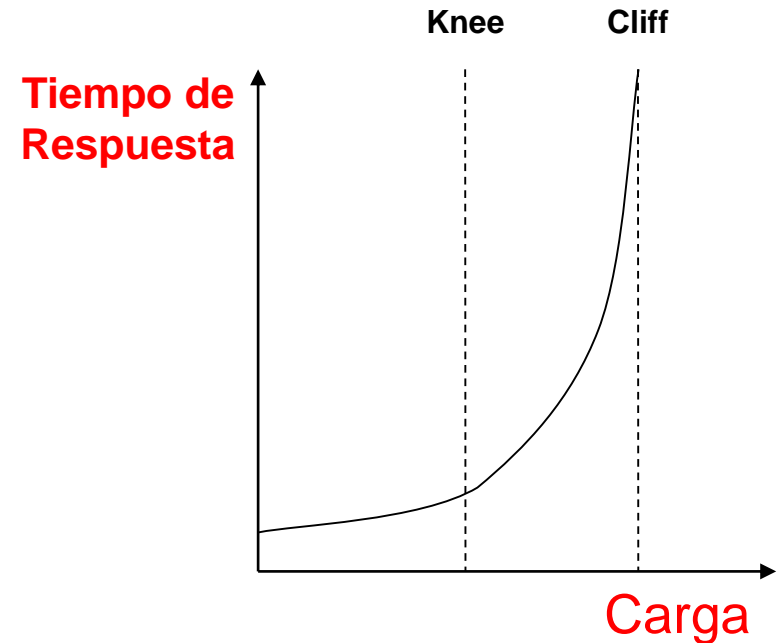
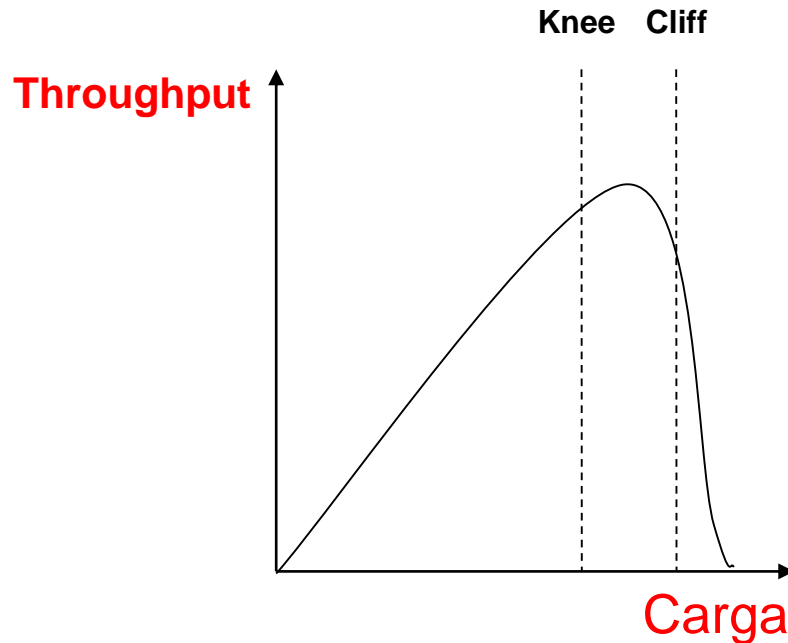
Criterios de Evaluación (2)

- ▶ **Equidad:** Se busca que los recursos sean compartidos equitativamente “en las buenas y en las malas”
- ▶ Indicador de equidad de **Jain:**
Dados n flujos por un enlace
con throughputs (x_1, x_2, \dots, x_n)

$$l/n \leq f \leq l$$

$$f(x_1, x_2, \dots, x_n) = \frac{\left(\sum_{i=1}^n x_i\right)^2}{n \sum_{i=1}^n x_i^2}$$

Performance en función de la carga (1)



“Potencia” intenta combinar ambos aspectos.

Performance en función de la carga (2)

- ▶ A medida que la **carga** ofrecida a la red aumenta, el **throughput** (tasa de datos que alcanzan el destino) se incrementa linealmente.
 - ▶ Sin embargo, a medida que la **carga alcanza la capacidad** de la red, los buffers en los routers comienzan a llenarse.
 - ▶ Esto causa el **incremento del tiempo de respuesta** (el tiempo que tardan los datos en atravesar la red entre el origen y destino) y disminuye el *throughput*.
- ▶ Una vez que los buffers de los routers comienzan a sobrecargarse ocurre la **pérdida de paquetes**.
 - ▶ Incrementos en la carga más allá de este punto incrementan la probabilidad de pérdida de paquetes.
 - ▶ **Bajo cargas extremas, el tiempo de respuesta tiende a infinito y el *throughput* tiende a cero**; este es el punto del **colapso de congestión**.
 - ▶ Es el punto conocido como *cliff* debido a la **extrema caída en el *throughput***.

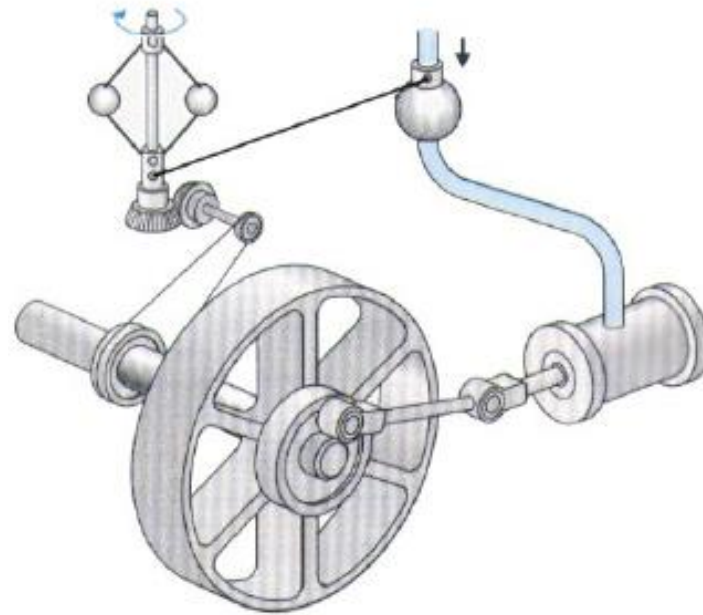
Congestión y Calidad de Servicio

- ▶ Sería muy fácil dar Calidad de Servicio (QoS) si las redes nunca se congestionaran. Para ello habría que sobredimensionar todos los enlaces, cosa no siempre posible o deseable.
- ▶ Para dar QoS con congestión es preciso tener mecanismos que permitan dar un **trato diferenciado** cierto tráfico preferencial y cumplir los **SLA (Service Level Agreement)**.
- ▶ El **SLA** suele ser estático y definido en el momento de negociación del contrato con el proveedor de servicio o ISP (Internet Service Provider).

Agenda (Parte 2)

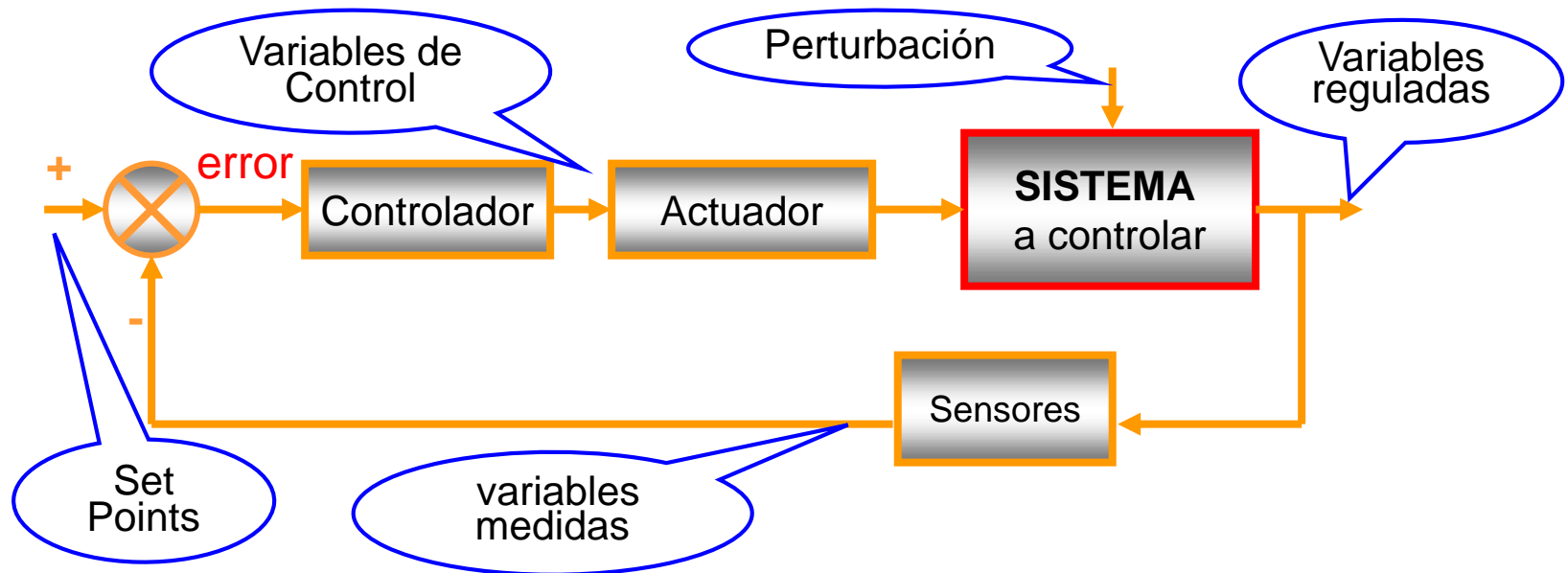
- ▶ Control de Congestión
 - ▶ Taxonomía Lazo Cerrado-Abierto
- ▶ RED
- ▶ FRED
- ▶ Políticas de tráfico

Antecedentes [1]

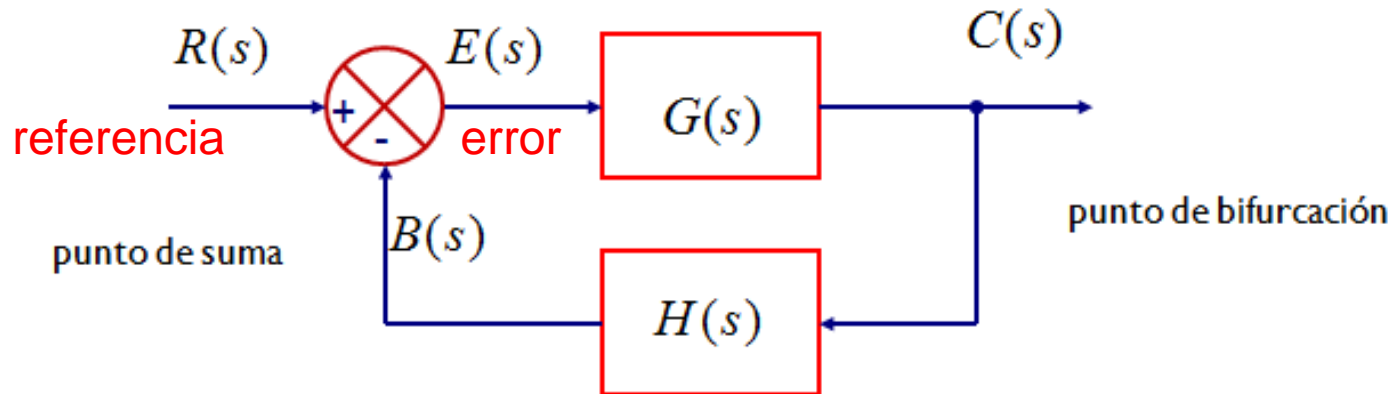


Dispositivo de control presión en una máquina de vapor
James Watt, circa 1760, http://en.wikipedia.org/wiki/Watt_steam_engine

Teoría de Control



Teoría de Control



Función de transferencia en lazo abierto

Función de transferencia trayectoria directa

Función de transferencia lazo cerrado

$$\frac{B(s)}{E(s)} = G(s)H(s)$$

error

$$\frac{C(s)}{E(s)} = G(s)$$

error

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

referencia

Taxonomía

- ▶ De acuerdo a la taxonomía de Yang y Reddy (1995), los algoritmos de control de congestión se pueden clasificar en **lazo abierto** y **lazo cerrado**.
- ▶ A su vez los de **lazo cerrado** se pueden clasificar de acuerdo a como realizan la **realimentación**.

Taxonomía [YR95]

Control Congestión

```
graph TD; CC[Control Congestión] --> LA[Lazo Abierto]; CC --> LC[Lazo Cerrado]; LC --> RI[Realimentación Implícita]; LC --> RE[Realimentación Explícita];
```

Lazo Abierto

- Principalmente en redes conmutación de circuitos (GMPLS)

Lazo Cerrado

- Usado principalmente en redes de paquetes
- Usa información de realimentación: global & local

Realimentación Implícita

- “End-to-end congestion control”
- Ej: TCP Tahoe, TCP Reno, TCP Vegas, SCTP, TCP CUBIC, etc.

Realimentación Explícita

- “Network-assisted congestion control”
- Ej: IBM SNA, DECbit, ATM ABR, ICMP source quench, **ECN**

Congestion Control and Avoidance

- ▶ “congestion control” : reactivo
- ▶ “congestion avoidance” : preventivo (o proactivo)

Feedback Implícito vs. Explícito

“Implicit feedback Congestion Control”

- “La **red descarta paquetes**” cuando se aproxima la congestión
- **La fuente infiere la congestión** en forma implícita
 - ▶ Time-Outs, ACKs duplicados, etc.
- Ej.: CC end-to-end TCP
- Implementación relativamente simple
 - ▶ Normalmente Implementada a Nivel de Transporte

Feedback Implícito vs. Explícito (cont.)

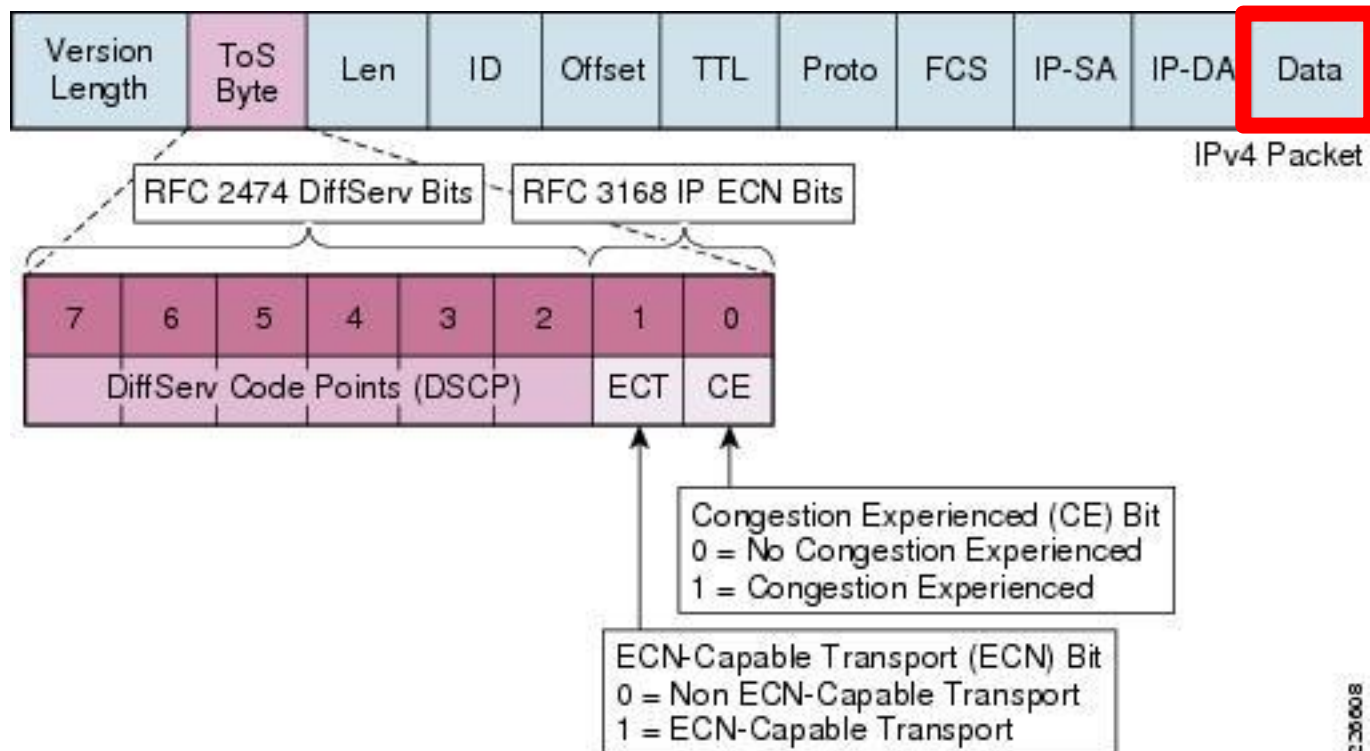
“Explicit feedback Congestion Control”

- Componentes de red (Ej., router, SW) proveen indicación explícita de la congestión a las fuentes
 - ▶ Usa “*packet marking*”
- Ej. DECbit, ECN, ATM ABR CC, etc.
- Provee información mas precisa a las fuentes
- Mas complicado de implementar
 - ▶ Se necesita cambiar la fuente y el algoritmo de red
 - ▶ Se necesita cooperación entre fuentes y componentes de red

Feedback Implícito vs. Explicito (cont.)

“**Explicit feedback** Congestion Control”

- IP ECN: Explicit Congestion Notification
- IP header: Type of Service (ToS) Byte





RED

RED (Random Early Detection)

- ▶ El algoritmo RED se tiene por objetivo **evitar la congestión** y de mantener el tamaño medio de las colas en niveles bajos.
 - ▶ Entonces: Método **Implícito** de **Congestion Avoidance**
- ▶ RED no necesita que los routers mantengan ninguna información del estado de las conexiones.
- ▶ RED fue diseñado para trabajar en colaboración con mecanismos de control de congestión en la capa de transporte (TCP, SCTP).
- ▶ Es una técnica de las conocidas como AQM (Active Queue Management). A veces denominado AQM-RED.

RED (Random Early Detection)

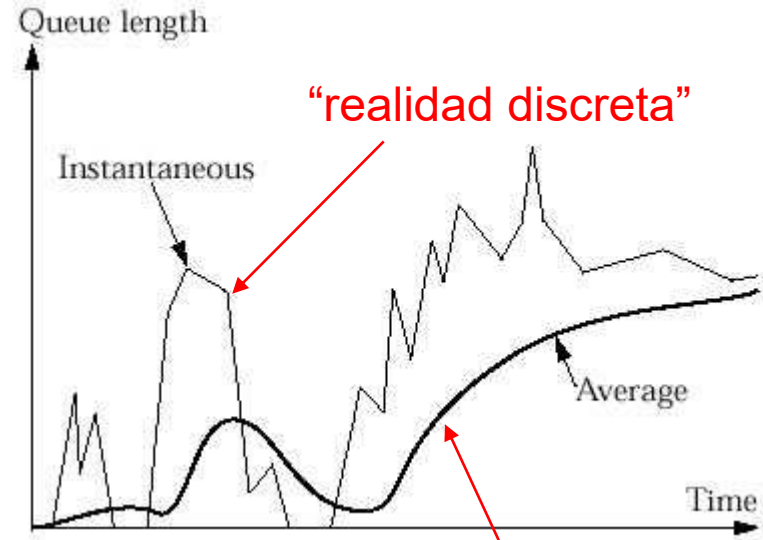
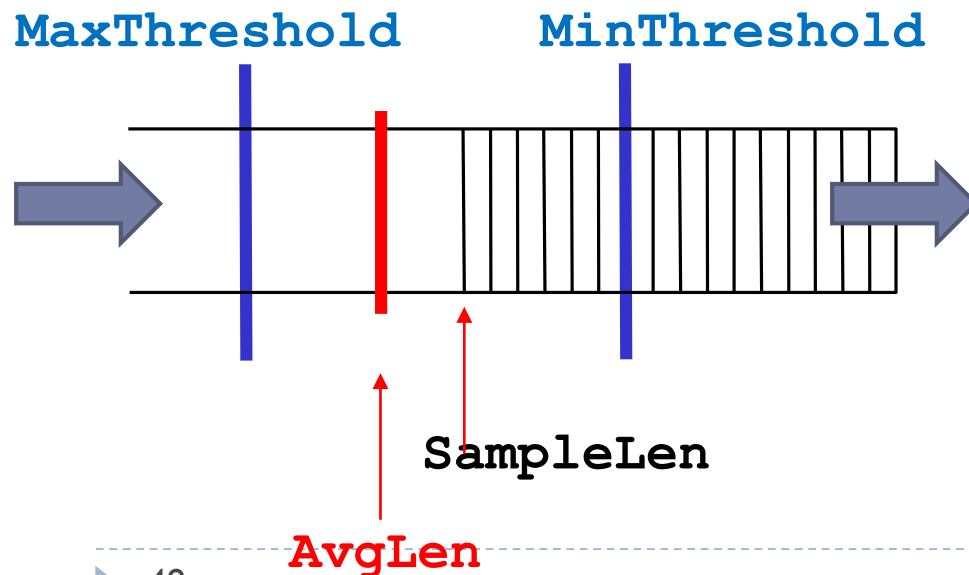
- ▶ **Notificación implícita de inminencia de congestión**
 - ▶ Simplemente descarta el paquete (luego en TCP habrá timeouts)
 - ▶ Podría hacerse explícita marcando el paquete (como ECN)
- ▶ **Descarte aleatorio temprano**
 - ▶ En lugar de esperar por que se llene la cola, descarta cada paquete de entrada con *“alguna” probabilidad de descarte* cada vez que la cola excede *“algún” nivel de descarte*

Algoritmo RED

- ▶ Calcula largo promedio de cola

$$\text{AvgLen} = (1 - \text{Weight}) * \text{AvgLen} + \text{Weight} * \text{SampleLen}$$

- ▶ $0 < \text{Weight} < 1$
- ▶ **SampleLen** es el tamaño instantáneo de la cola, actualizado **cada vez** que llega/sale nuevo paquete
- ▶ **AvgLen** es una versión suavizada de **SampleLen**



Algoritmo RED

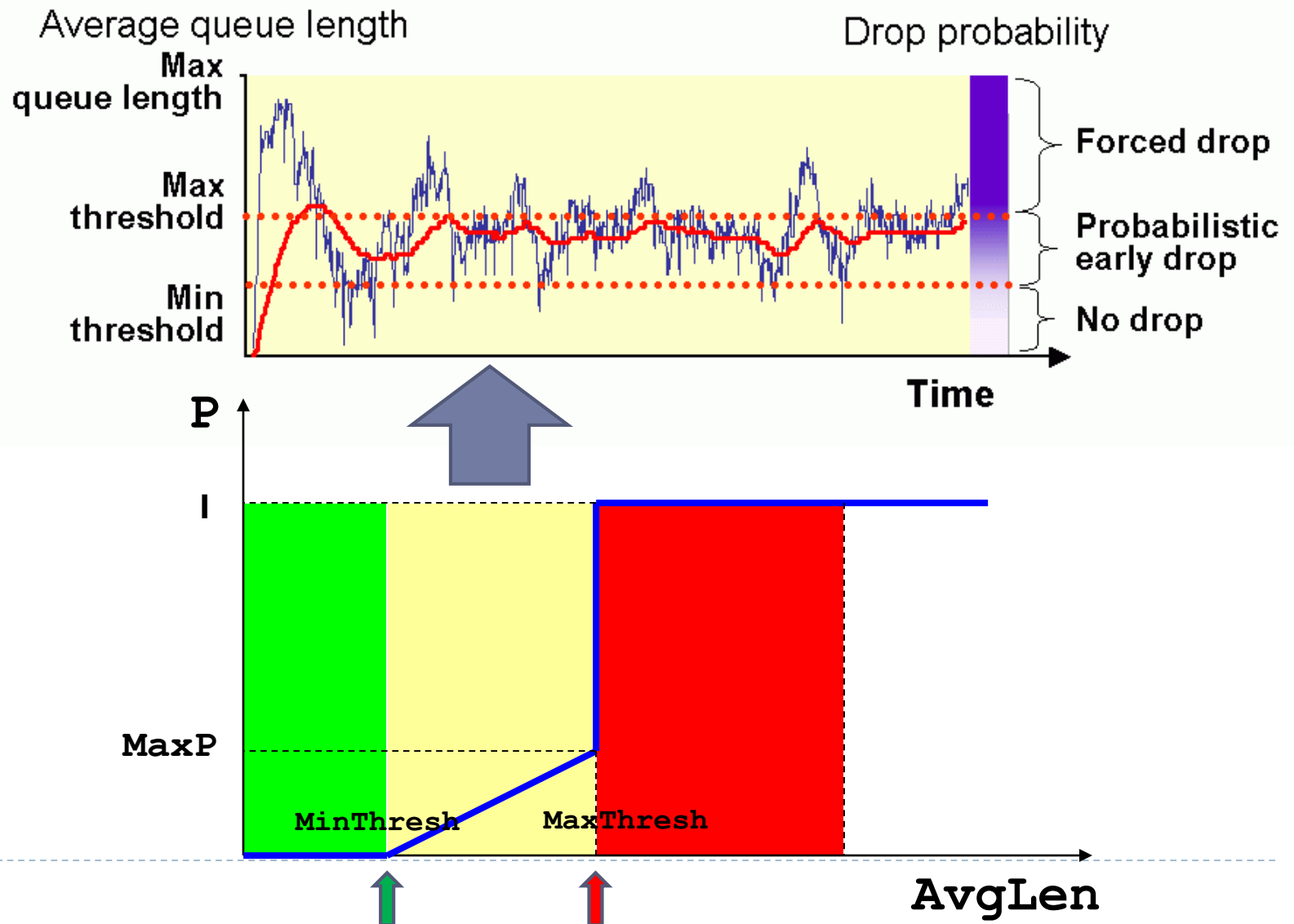
- Dos umbrales del tamaño promedio de cola

```
if MinThreshold => AvgLen then  
    encolar el paquete
```

```
if MinThreshold < AvgLen < MaxThreshold then  
    calcular probabilidad P  
    descartar paquete entrante con probabilidad P
```

```
if MaxThreshold <= AvgLen then  
    descartar paquete entrante
```

Algoritmo RED



Ajustes de RED

- ▶ Probabilidad de descartar un paquete de un flujo particular de paquetes es **aproximadamente proporcional** a la **porción del ancho de banda** que el flujo está obteniendo.
- ▶ **MaxP** es típicamente fijada en 0.02
 - ▶ Es decir en ese caso el router descarta aproximadamente **uno de cada 50 paquetes**.
- ▶ Si el tráfico es **rafagoso**
 - ▶ **MinThresh** debería **ser suficientemente grande** para permitir que la utilización del enlace sea mantenida a un nivel aceptablemente alto
- ▶ La **diferencia entre los dos umbrales** debería ser **más grande que el incremento típico** en el largo de cola promedio calculado **en un RTT**
 - ▶ Fijar **MaxThreshold = 2 * MinThreshold** es un criterio tomado como razonable para Internet.



FRED

Flow Random Early Detection (FRED)

- ▶ Una de las características más importantes RED es el hecho de que proporciona imparcialidad descartando los paquetes de una conexión según la parte que ocupa del ancho de banda.
- ▶ Sin embargo, RED tiene otros problemas de imparcialidad (fairness).
- ▶ RED no es justo (fair) con las conexiones de baja velocidad.
 - ▶ Cuando se alcanza el umbral máximo, RED descarta los paquetes aleatoriamente. Puede darse que el paquete descartado pertenezca a un flujo (conexión) que esté utilizando **menos recursos de lo que le corresponde**.
 - ▶ Cuando la longitud media de la cola está en un punto fijo dentro de los dos umbrales, todos los paquetes entrantes se descartan con la misma probabilidad.

Flow Random Early Detection (FRED)

- ▶ Control de usuarios acaparadores (bandwidth hogs)
 - ▶ Aunque se hace cumplir el umbral máximo, si el usuario no disminuye su tasa, RED no tiene ningún mecanismo para proteger a otros usuarios.
 - ▶ Los usuarios egoístas podrían terminar ocupando la totalidad del ancho de banda

FRED

- ▶ FRED soluciona estos problemas de imparcialidad manteniendo umbrales y ocupaciones del buffer **para cada flujo activo (información por conexión)**.
- ▶ FRED necesita guardar información por cada flujo para descartar paquetes produciendo un **alto costo de operación en los routers**
- ▶ FRED debe identificar cada flujo que tenga paquetes en el buffer y actualizar la información por cada paquete
 - ▶ Debe mirar la fuente del paquete y las direcciones de destino, los puertos, y la identificación del protocolo de cada paquete

Políticas de Tráfico

Repasando algunos conceptos ya vistos

Traffic Shaping

- ▶ El trafico es “*bursty*” (*rafagoso*) e impacta en la congestión.
- ▶ Traffic Shaping es un método de **lazo abierto** que trata de **guiar la congestión**, forzando a los paquetes a transmitirse a una velocidad mas **predecible**.
- ▶ Se trata de mantener el tráfico constante, es decir regular la tasa de transmisión media y su desviación estándar (el “*burstiness*” de los datos)
 - ▶ Por ejemplo en las redes de CV, en el setup de circuito el usuario y el carrier se poden de acuerdo en el “conformado” (shape) del circuito.
 - ▶ Se necesita un acuerdo del proveedor y el cliente.

Parámetros típicos

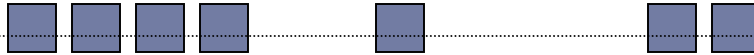
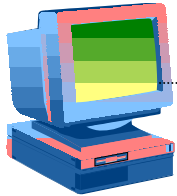
- ▶ **Average Rate:** e.g. 100 paquetes por segundo (o 6000 paquetes por minuto)
 - ▶ un aspecto crucial es la longitud del intervalo
- ▶ **Peak Rate:** e.g., 6000 ppm **Avg** y 1500 pps **Peak**
- ▶ **Burst Size:** número máximo de paquetes enviados consecutivamente (en un período corto de tiempo)

Algoritmos

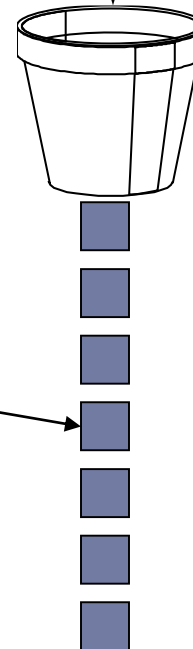
Ejemplos clásicos de control de QoS

- ▶ Leaky Bucket
- ▶ Token Bucket

Algoritmo Leaky Bucket



Flujo de paquetes sin regular



El flujo de salida tiene una velocidad constante ρ , cuando hay agua en el balde, y cero cuando esta vacío

=> Se moderan las ráfagas. Límite de throughput estricto.

Algoritmo Token Bucket

▶ **Leaky Bucket**

- ▶ Fuerza un patrón de tráfico de salida rígido (tasa máxima)

▶ **Token Bucket** permite picos de tráfico

- ▶ Se permite aumentar la velocidad durante un intervalo pequeño cuando le llega una ráfaga de paquetes
- ▶ Los baldes (buckets) mantienen fichas (tokens), que son generadas por un clock a un **rate de 1 token cada ΔT segundos**
- ▶ Para transmitir, se necesita consumir un token.
- ▶ Si no hay tokens, se espera, o se descarta.
- ▶ Hosts Idle pueden “ahorrar” tokens (“permisos”) para enviar bursts luego
- ▶ Hay proveedores de servicio de cablemodem con tecnología que permite (por unos instantes), tener un pico de tráfico superior al contratado.

Impacto del CC en TCP

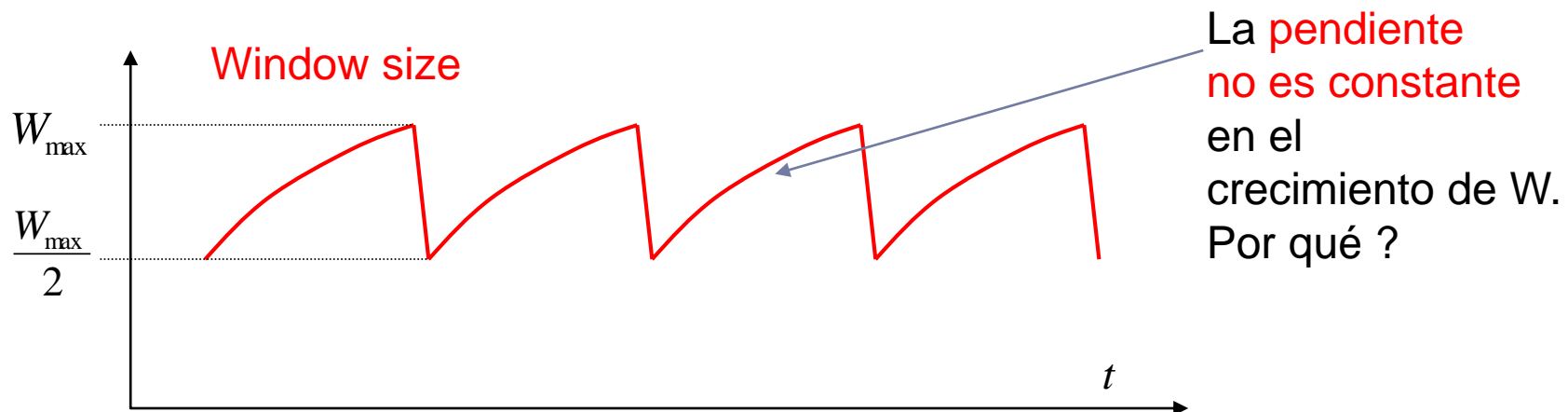
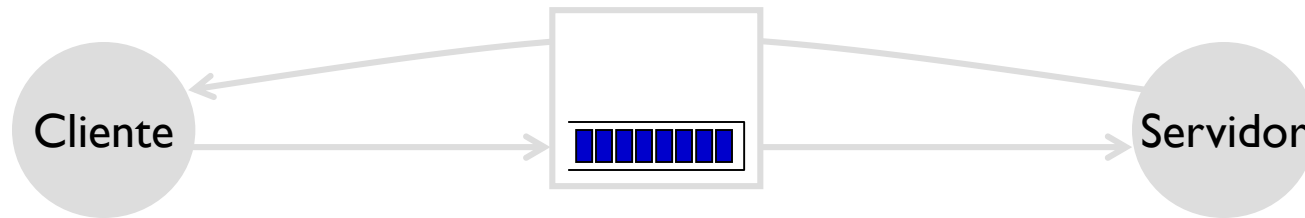
Idea básica del control de congestión en TCP

Performance de TCP

Control Congestión en TCP

Idea básica: Por cada RTT hacer:

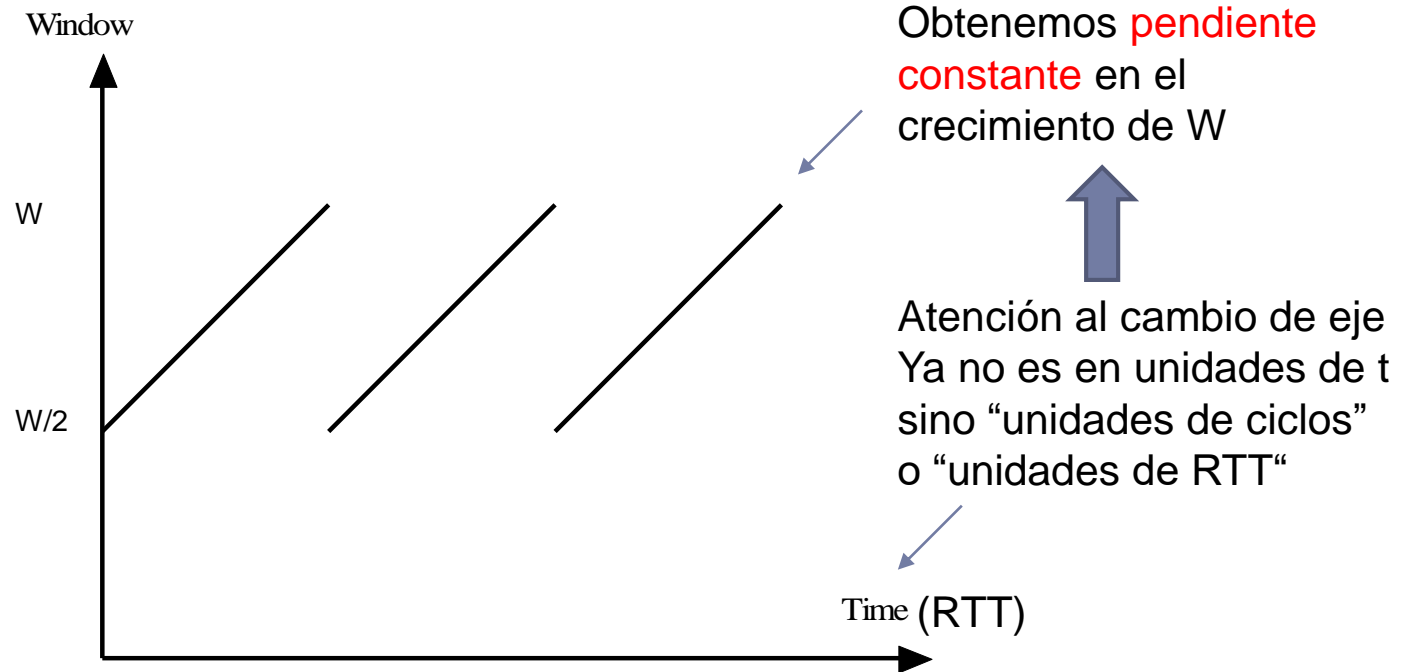
- Si se recibe un ACK : $W \leftarrow W + 1/W$ Additive Increase
- Si se pierde un ACK: $W \leftarrow W/2$ Multiplicative Decrease



Performance de TCP_[1]

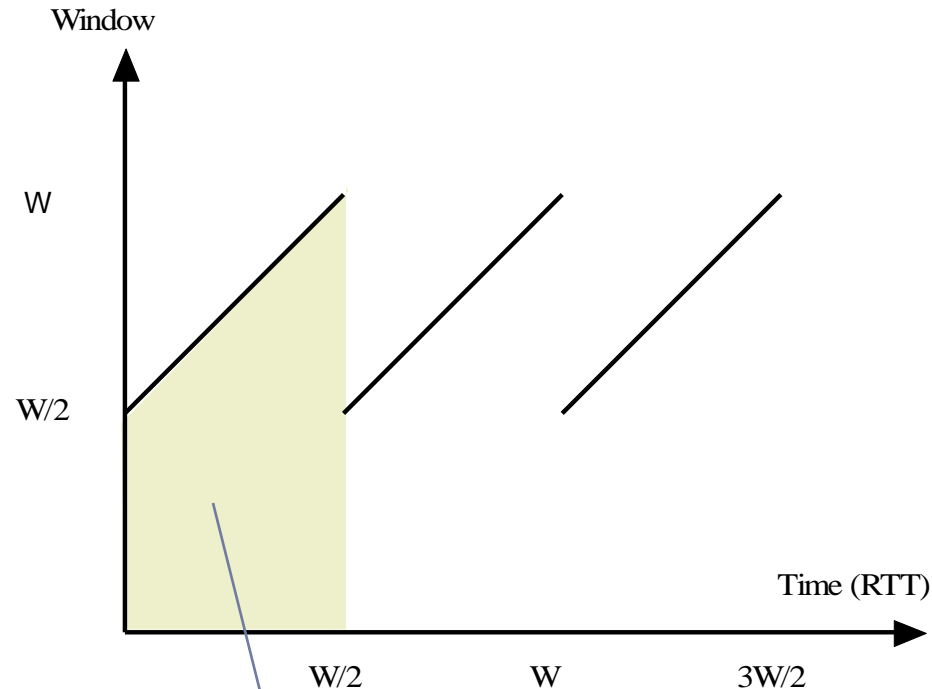
- ▶ Hallar una expresión para “steady state throughput” como una función del
 - ▶ **RTT**
 - ▶ Probabilidad **p** de pérdida de paquetes
- ▶ **Asumimos**
 - ▶ Cada paquete descartado con probabilidad **p**
 - ▶ RTT estable
 - ▶ Suficiente ancho de banda BW en el enlace
 - ▶ Señal de congestión periódica (**I/p paquetes OK seguidos por I Descartado**)
- ▶ **Metodología: régimen permanente, conexión de larga vida**
 - ▶ Cuantos paquetes se TX por ciclo ?
 - ▶ Cual es la duración de un ciclo ?

Descripción Macroscópica del comportamiento de TCP



- Sea W el máximo valor de la ventana
- Asumimos la versión de “Congestion Avoidance” que define la ventana mínima como $W/2$

Análisis de un Ciclo (I)



TCP window evolution under periodic loss

$$\text{pkts TX/ciclo} = \text{area} = \left(\frac{W}{2}\right)^2 + \frac{1}{2}\left(\frac{W}{2}\right)^2 = \frac{3}{8}W^2$$

Análisis de un Ciclo (III)

$$T = \text{Throughput} = \frac{\text{pkts TX/ciclo}}{\text{tiempo/ciclo}} = \frac{\frac{3}{8}W^2}{RTT \cdot \left(\frac{W}{2}\right)} \quad \text{pkts/seg}$$

$$BW = \frac{\text{data TX/ciclo}}{\text{tiempo/ciclo}} = \frac{MSS \cdot \frac{3}{8}W^2}{RTT \cdot \left(\frac{W}{2}\right)} \quad \text{bits/seg}$$

Análisis de un Ciclo (II)

Dado α packet losses al final del ciclo:

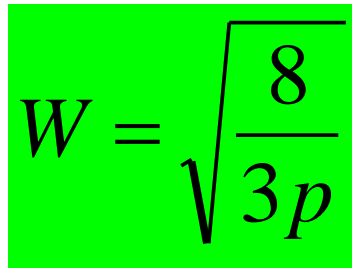
$$P(\alpha = k) = P(k - 1 \text{ pkts : NO Lost}, k_{\text{th}} \text{ pkt : Lost})$$

$$P(\alpha = k) = (1 - p)^{k-1} \cdot p$$

$$\Rightarrow E(\alpha) = \sum_{k=1}^{\infty} k \cdot (1 - p)^{k-1} \cdot p = \frac{1}{p}$$

“Se transmiten $1/p$ paquetes antes de la primer pérdida”

$$\Rightarrow \frac{1}{p} = \frac{3}{8} W^2 \Rightarrow W = \sqrt{\frac{8}{3p}}$$



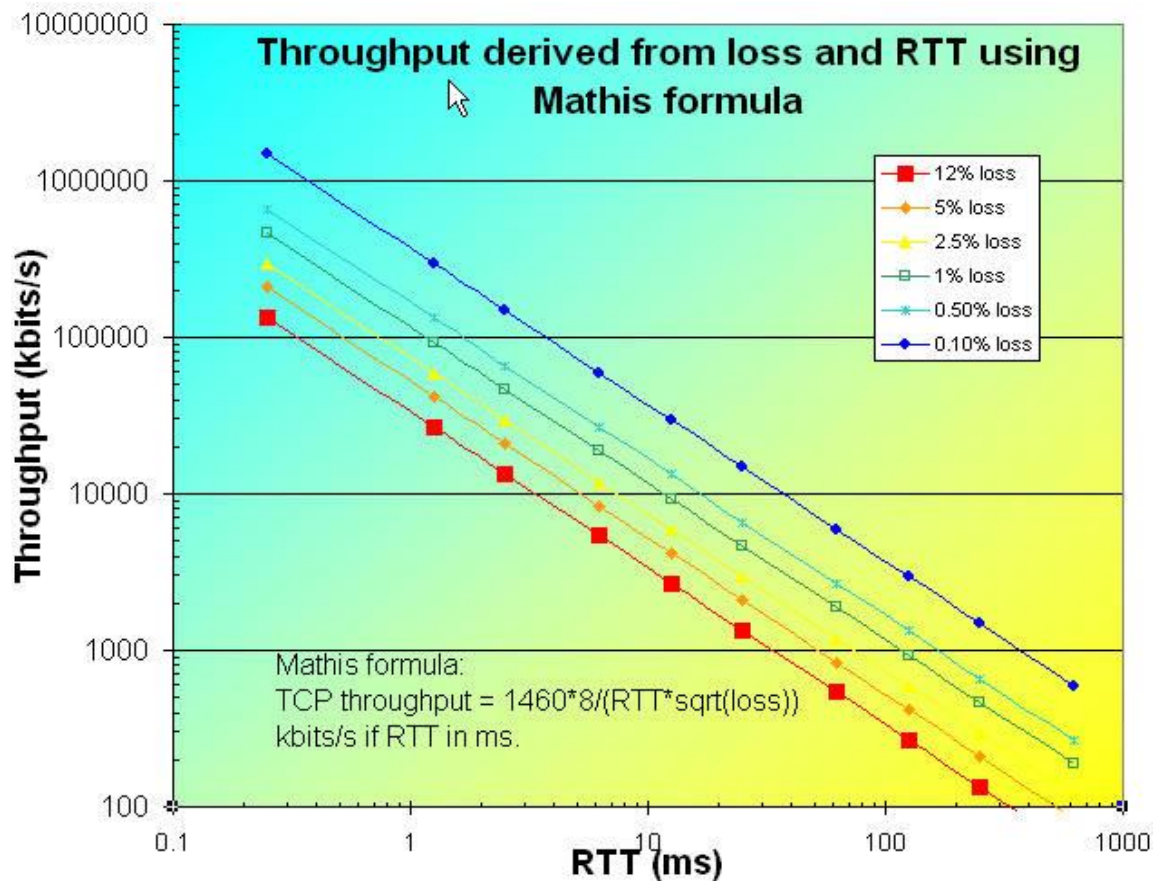
Performance TCP

$$\text{Throughput}(p) = T(p) = \frac{\frac{1}{p}}{RTT \cdot \frac{1}{2} \sqrt{\frac{8}{3p}}} = \frac{1}{RTT \sqrt{\frac{2}{3} p}}$$

$$\text{BW} = \frac{MSS * \frac{1}{p}}{RTT \cdot \frac{1}{2} \sqrt{\frac{8}{3p}}} = \frac{MSS * C}{RTT \sqrt{p}}$$

$$C = \sqrt{\frac{3}{2}}$$

Performance TCP



$$BW = \frac{MSS * C}{RTT \sqrt{p}}$$

$$C = \sqrt{\frac{3}{2}}$$

Estrictamente en el RFC 2581 se define el Sender Maximum Segment Size (SMSS): es el máximo tamaño que el emisor de un segmento puede enviar en un momento dado. El valor puede estar acotado en el MTU (máxima unidad de transmisión) del enlace de salida desde el host, el resultado del Path MTU Discovery Algorithm [RFC1191], RMSS u otros factores. Este tamaño no incluye los encabezados TCP e IP.

Temas Abiertos

Algunos temas recientes

- ▶ Es bien conocida la pobre performance de TCP en ambientes wireless, viene a solucionar TCP/NC este problema ?
- ▶ Es durante el 2011 que Jim Gettys de Alcatel-Lucent, que acuñó el termino *Bufferbloat*, comenzó a publicar diversos artículos que plantean que la “lentitud de Internet” es a causa del *Bufferbloat* [3][4] además de presentaciones en el IETF [11], en Google Tech Talk [21] o en el congreso de operadores de redes de USA [24]. Lo define formalmente como “la existencia de buffers excesivamente grandes en los sistemas, en particular en los relacionados con las redes y comunicaciones” [5].



Ejercicios

Ejercicio (1)

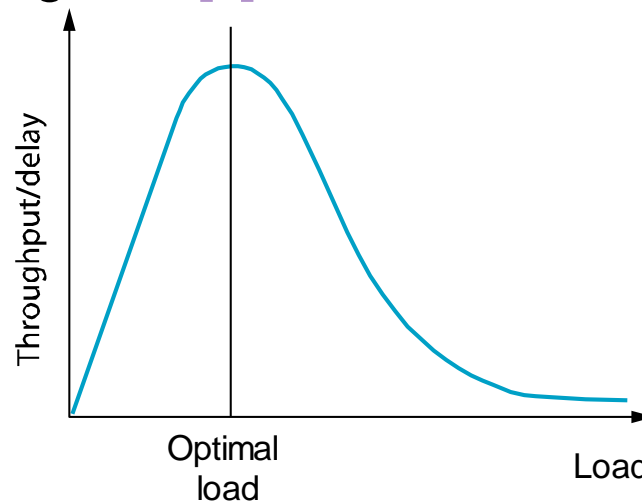
De acuerdo a la taxonomía de Yang y Reddy (1995), los algoritmos de control de congestión se pueden clasificar en lazo abierto y lazo cerrado. A su vez los de lazo cerrado se pueden clasificar de acuerdo a como realizan la realimentación.

A que categoría pertenece RED (Randomly Early Detection)?

Que impacto tiene en flujos UDP ?

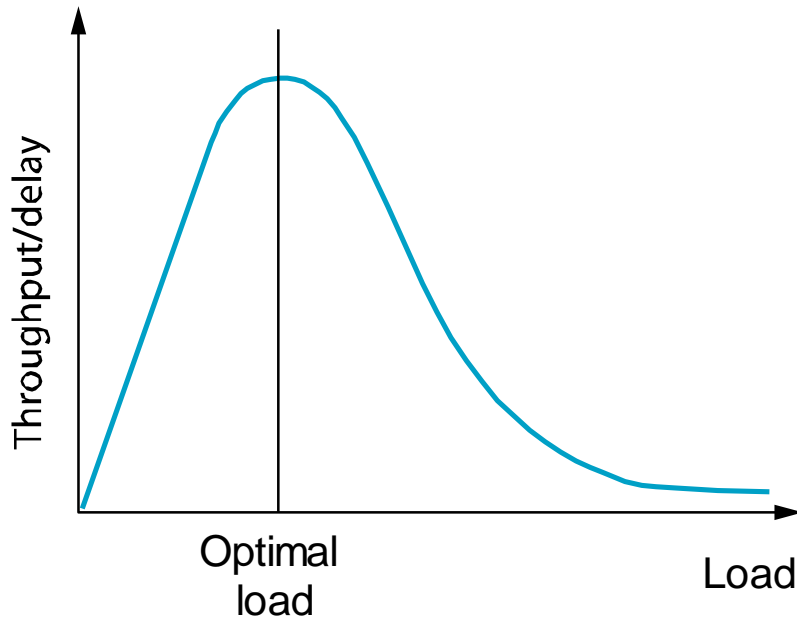
Ejercicio (2)

- ▶ Algunos autores utilizan la relación que denominan **Potencia** $P = \text{Throughput/delay}$ como una métrica para medir la eficiencia de un esquema de asignación de recursos.
- ▶ Para un flujo de paquetes que ingresa a un router de una red de conmutación de paquetes, la variación de la potencia en función de la carga (paquetes/seg.) es la siguiente[1] :



- [1] En realidad la teoría subyacente en la definición de potencia está basada en la teoría de colas. En este caso particular un router con una entrada y una salida, se modela como una cola M/M/1 (notación de Kendall) 1 : un servidor, M es que tanto la distribución de la llegada de paquetes como el tiempo de servicio son Markovianos, esto es, exponenciales. Luego, el modelo es una cola FIFO con buffer ilimitado y un servidor que despacha n paquetes/seg cuyos tiempos entre paquetes obedece a una distribución exponencial (sin memoria).

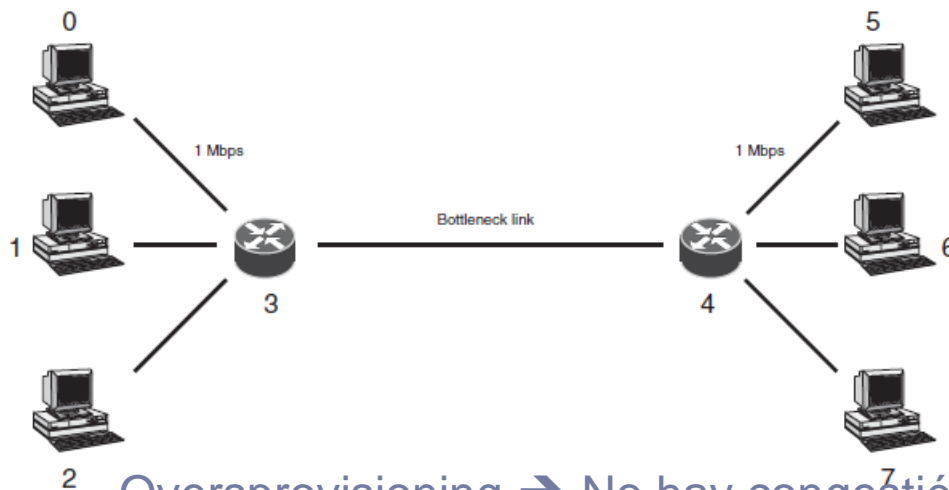
Ejercicio(2 cont.)



- ▶ a) ¿En que zona de la función Potencia se presenta el fenómeno de congestión , que significa la congestión en redes de conmutación de paquetes ?
- ▶ b) Cuales son dos soluciones posibles para evitar entrar en congestión ?
- ▶ c) una de las causas de la congestión es cuando el trafico se presenta en ráfagas , a tal efecto se han desarrollado mecanismos de “traffic shaping” que “fuerza” a un trafico mas predecible . Mencione un mecanismo de traffic shaping ? Dicho mecanismo es de lazo abierto o cerrado ?

Ejercicio (3): Efecto “Traffic Phase” (Jacobson –Floyd 1991)

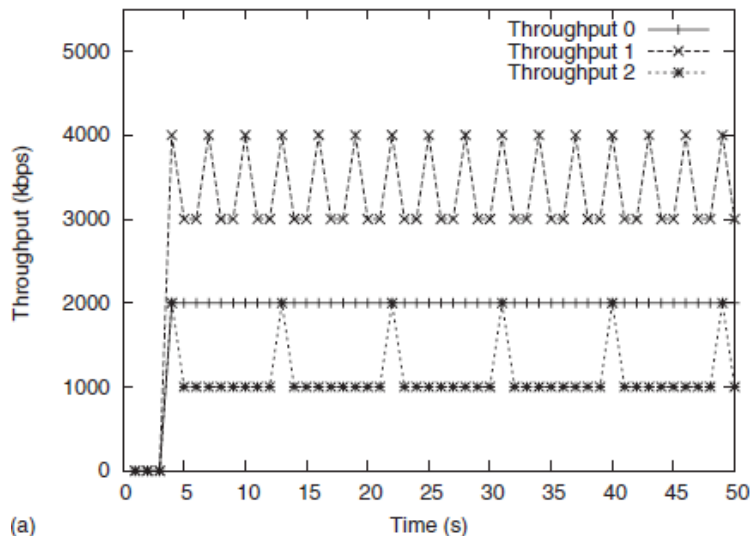
- ▶ Supongamos la siguiente topología para simular burstiness con flujos CBR
 - ▶ Tenemos tres fuentes (0,1,2) que tx paquetes de 1000 bytes a los nodos 5,6 y7
 - ▶ Todos los enlaces al router 1 Mbps y delay del enlace de 1 seg
 - ▶ El nodo 0 envia un paquete cada 500ms, 1 cada 300ms y 2 cada 900 ms → 16, 26 y 18 Kbps
 - ▶ Medimos el throughput en 5,6 y 7 cada segundo



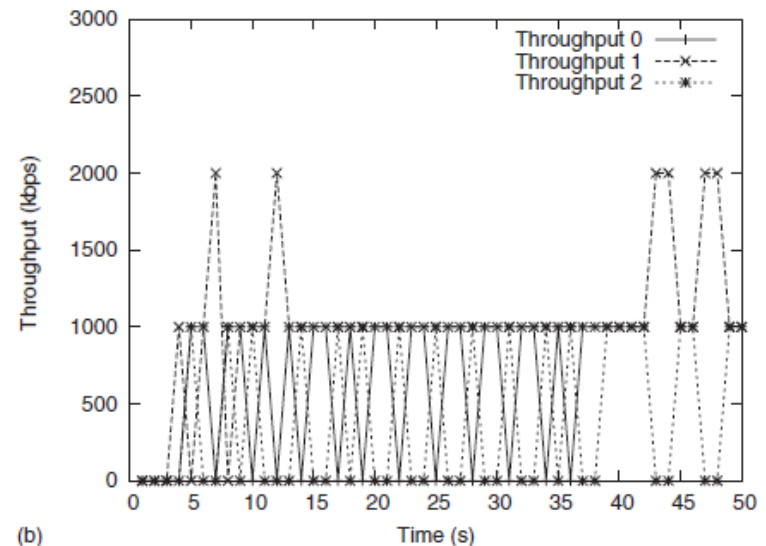
Oversprovisioning → No hay congestión

Efecto “Traffic Phase”

- ▶ Por lo tanto uno espera que todas las “lineas” de la figura (a) sean CTES , es cierto para I
- ▶ Los flujos CBR no exhiben un comportamiento cte cuando el intervalo de tiempo de muestreo no es múltiplo de su velocidad
- ▶ En la figura (b) se muestra la agregación de los tres flujos
- ▶ Que sucede si seteo el enlace de los nodos 3-4 en 16kbps ??



(a)



(b)

Procesos de Poisson de intensidad λ

- ▶ La probabilidad de que lleguen n paquetes en un intervalo de longitud t es:

$$A(t) \sim \text{Poisson}(\lambda t), t \geq 0$$

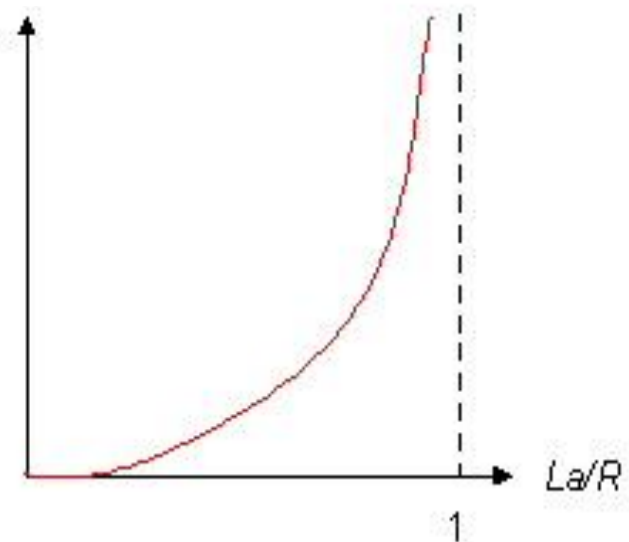
- ▶ λ : velocidad media de llegadas (paquetes/sg)
- ▶ $1/\lambda$: tiempo medio entre llegadas

$$A_n = P(A(t) = n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!}, n \geq 0$$

Retardo de una COLA – M/M/1

- ▶ R = ancho de banda del enlace (bps).
- ▶ L = longitud del paquete (bits).
- ▶ a = media de tasa de llegada del paquete.

Media de retardo
de cola



Intensidad de tráfico = La/R

- $La/R \sim 0$: media de retardo de cola pequeño.
- $La/R \rightarrow 1$: aumentan los retardos.
- $La/R > 1$: ¡Llega más “trabajo” del que puede servirse, media de retardo **infinita**

Referencias

- ▶ [1] Nagle, J. On packet switches with infinite storage. Network Working Group RFC 970 (December 1985) <http://www.ietf.org/rfc/rfc970.txt>
- ▶ [2] Nagle, J.: *On packet Switches with Infinite Storage*. IEEE vol Com-35. April 1987
- ▶ [3] J. Gettys, Bufferbloat: Dark Buffers in the Internet, IEEE Internet Computer, May/June 2011.
- ▶ [4] Jim Gettys, Kathleen Nichols Bufferbloat: Dark Buffers in the Internet Queue ACM November 29, 2011 <http://queue.acm.org/detail.cfm?id=2071893>
- ▶ [5] <http://gettys.wordpress.com/what-is-bufferbloat-anyway/>
- ▶ [6] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. Internet Measurement Conference (IMC), San Diego, CA (October 24-27).
- ▶ [7] Kreibich, C., et al. 2010. Netalyzr: illuminating the edge network. Internet Measurement Conference (IMC), Melbourne, Australia (November 1-3). conferences.sigcomm.org/imc/2010/papers/p246.pdf
- ▶ [8] J. Martin, J. Westall, J. Finkelstein, G. Hart, T. Shaw, G. White, R. Woundy, "[Cable Modem Buffer Management in DOCSIS Networks](#)", Proceedings of the IEEE Sarnoff 2010 Symposium, (Princeton University, Princeton NJ, April 2010).
- ▶ [9] Cablelabs, SP-MULPlv3.0: MAC and Upper Layer Protocols Interface Specification, Feb 10, 2011.
- ▶ [10] Cablelabs, SP-MULPlv3.0: MAC and Upper Layer Protocols Interface Specification, June, 2011. <http://www.cablelabs.com/specifications/CM-SP-MULPlv3.0-116-110623.pdf>
- ▶ [11] J. Gettys, Bufferbloat: Dark Buffers in the Internet, <http://mirrors.bufferbloat.net/Talks/PragueIETF/IETFBloat7.pdf>, April 2011.
- ▶ [12] M. Claypool, R. Kinicki, M. Li, J. Nichols, "Inferring Queue Sizes in Access Networks by Active Measurement", Proceedings of the 5th Passive and Active Network Measurement Workshop, April, 2004.
- ▶ [13] Cablelabs, CM-GL-Buffer-V01-110915: Cable Modem Buffer Control <http://www.cablelabs.com/specifications/CM-GL-Buffer-V01-110915.pdf>
- ▶ [14] L. Kleinrock, "Models for Computer Networks," in *Conference Record, IEEE International Conference on Communications*, Boulder, Colorado, June 1969, p. 21/9–21/16. [\[PDF\]](#)

Referencias

- ▶ [15] Has AT&T Wireless data congestion been self-inflicted? <http://blogs.broughtturner.com/2009/10/is-att-wireless-data-congestion-selfinflicted.html>
- ▶ [16] Reed, D.P. 2009. What's wrong with this picture; thread at <http://mailman.postel.org/pipermail/end2end-interest/2009-September/007742.html>.
- ▶ [17] Lista correo end-to end <http://www.postel.org/e2e.htm>
- ▶ [18] V. Jacobson and M. Karels, Congestion Avoidance and Control, Proceedings of SIGCOMM '88, August 1988
- ▶ [19] Allman, M., Paxson, V., Stevens, W.: *TCP Congestion Control*. RFC 2581 April 1999.
- ▶ [20] Iman, M., Paxson, V., and Blanton, E., "TCP Congestion Control", RFC 5681, September 2009, <http://tools.ietf.org/html/rfc5681.txt>
- ▶ [21] Bufferbloat: Dark Buffers in the Internet. Google Tech Talk April 26, 2011 <http://www.youtube.com/watch?v=qblozKVz73g>
- ▶ [22] Sally Floyd, Van Jacobson *Random early detection gateways for congestion avoidance* IEEE/ACM Transactions on Networking (TON), August 1993
- ▶ [23] V. Jacobson, "Notes on Using RED for Queue Management and Congestion Avoidance", talk at NANOG <http://www.nanog.org/mtg-9806/agen0698.html>
- ▶ [24] Bufferbloat: Dark Buffers in the Internet. postado agosto del 2011 en el IETF
- ▶ [25] James Gettys, Bufferbloat: "Dark" Buffers in the Internet. Presentation Date: June 14, 2011, 11:30 AM - 12:00 PM NANOG52
- ▶ [26] Jim Gettys, Kathleen Nichols Bufferbloat: "Dark" Buffers in the Internet Communications of the ACM January 2012 Vol. 55 No. 1, Pages 57-65
- ▶ [27] Comcast Powerboost press release, Jun 2006. <http://www.comcast.com/About/PressRelease/PressReleaseDetail.ashx?PRID=65>
- ▶ [28] Comcast, "Method and packet-level device for traffic regulation in a data network," Patent 7289447, Oct 30, 2007.
- ▶ **[29]** Mitigations and Solutions of Bufferbloat in Home Routers and Operating Systems <https://gettys.wordpress.com/2010/12/13/mitigations-and-solutions-of-bufferbloat-in-home-routers-and-operating-systems/>