

Congestión

Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Grado en Ingeniería en Tecnologías de
Telecomunicación, 4º

“Internet me va mal”

- Internet da un servicio *best-effort*
- No hay garantías para los paquetes: se pueden perder, desordenar, corromper, retrasar en “exceso”... incluso puede que lleguen al destino! 😊
- En general consideramos que la red “funciona”, pero es gracias a que muchos servicios son “elásticos”
- Nuestra tolerancia (como usuarios) no lo es tanto
- Descargar un fichero puede tardar un día el doble que otro o incluso llegar a estancarse la transferencia
- Si esa descarga es una película que estamos viendo (...)
- ¿Y esto por qué sucede?



“Internet me va mal”

- Hay varios motivos posibles:
 - El hardware de comunicaciones falla (enlaces, routers)
 - Eso lleva a periodos con diversos problemas en la red mientras se calculan otras rutas
 - <http://blog.level3.com/level-3-network/the-10-most-bizarre-and-annoying-causes-of-fiber-cuts/>
 - Debemos tener mecanismos automáticos de recuperación
 - También detectar los fallos para hacer el mantenimiento
 - Podemos hacer estimaciones de probabilidad de fallos (los equipos suelen tener tiempos de vida medios antes de fallos)



“Internet me va mal”

- Hay varios motivos posibles:
 - El hardware de comunicaciones falla (enlaces, routers)
 - **Los servidores (su hardware o su software) fallan**
 - Si el servicio es crítico debería estar redundado el hardware
 - Podemos de nuevo valorar su tiempo esperado de vida
 - Debería haber técnicas de recuperación automática de ese software
 - Y luego a depurar...



“Internet me va mal”

- Hay varios motivos posibles:
 - El hardware de comunicaciones falla (enlaces, routers)
 - Los servidores (su hardware o su software) fallan
 - **Los servidores se congestionan**, no pueden atender mejor a tantas peticiones o es debido a unas pocas con grandes demandas
 - Hay que intentar predecir la demanda
 - Dimensionar los recursos de servidor para ella
 - Probar esos recursos ante *stress*

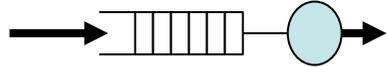


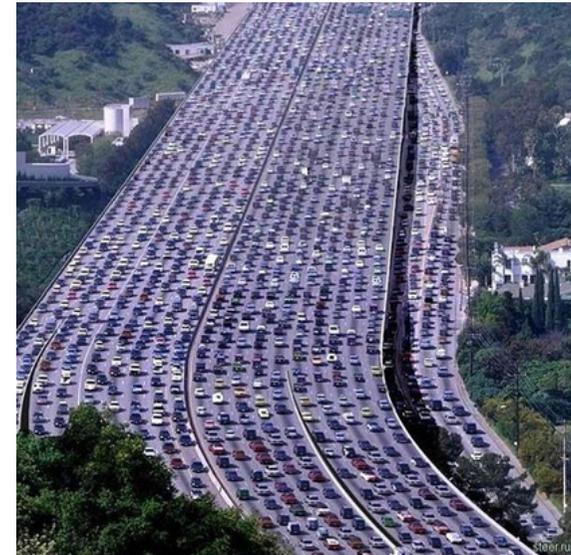
“Internet me va mal”

- Hay varios motivos posibles:
 - El hardware de comunicaciones falla (enlaces, routers)
 - Los servidores (su hardware o su software) fallan
 - Los servidores se congestionan, no pueden atender mejor a tantas peticiones o es debido a unas pocas con grandes demandas
 - **La red (el camino) se congestiona**
 - Keshav 1991: *“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.”*
 - En este tema nos centraremos en esta congestión en la red



¿Sufrimos congestión?

- Se da cuando la demanda excede la capacidad
- Los equipos de red introducen el exceso en una cola 
- Se espera que ese exceso de demanda se alivie
- Cuando la demanda es menor que la capacidad se puede cursar lo acumulado en cola pero
 - Este proceso aumenta el retardo y eso puede afectar no solo al servicio final sino a los protocolos (TCP)
 - El tráfico de Internet no es de Poisson y por lo tanto las fluctuaciones no son breves
- Las colas cumplen bien su función solo si en general se mantienen en baja ocupación
- ¿Soluciones? (...)



¿Sobredimensionar?

- También llamado *overprovisioning*
- Aumentar la capacidad para cursar el caso peor de tráfico
- Entonces no hay congestión
- La Internet ha evolucionado de un *core* congestionado a uno sobredimensionado
 - En los inicios enlaces del *core* 56Kbps, extremos LANs Ethernet y similares
 - En los 70s-80s accesos residenciales telefónicos y *core* a Mbps (PDH)
 - En los 90s-00s LANs 100Mbps y *core* a Gbps
 - Hoy LANs 1-40Gbps Ethernet y *core* DWDM, accesos evolucionando



¿Sobredimensionar?

- También llamado *overprovisioning*
- Aumentar la capacidad para cursar el caso peor de tráfico
- Entonces no hay congestión
- La Internet ha evolucionado de un *core* congestionado a uno sobredimensionado
- Overprovisioning porque:
 - BW más barato que el coste de perder clientes
 - Más sencillo controlar una red sobredimensionada
 - Así la red está preparada para absorber el crecimiento



¿Re-enrutar?

- Elegir otro camino para el tráfico
 - Un camino de mayor capacidad
 - Routing más complejo
 - ¿Y si también se congestiona ese camino? ¿*Route flapping*?
- Elegir otro camino para el tráfico en exceso
 - Si es por paquete crea diferentes RTTs en paquetes de un flujo
 - Si es por flujo requiere estado
- En la Internet actual los ISPs no enrutan para evitar la congestión
- ISPs dimensionan circuitos de gran capacidad en función de la evolución de la demanda en escalas temporales grandes



¿Control de congestión?

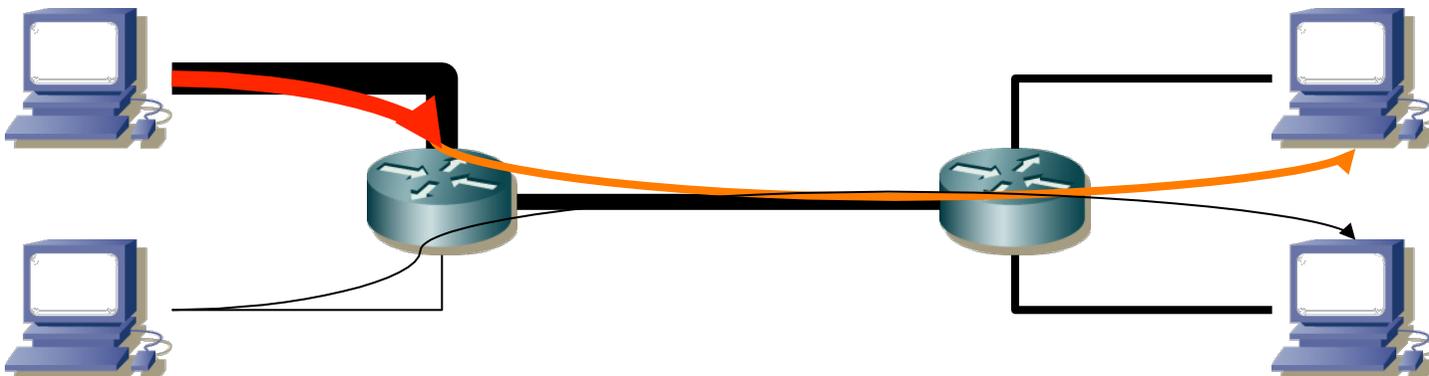
- Queremos evitar la congestión pues lleva a pérdidas y mayores retardos
- Van Jacobson, primeros trabajos en Internet (80's)
- Busca usar la red de la manera más eficiente posible
- Y estabilizarla
- Ante la falta de congestión hoy en el *core* buscamos ahora la forma de sacar provecho a la capacidad disponible
- Puede haber congestión en el acceso
- Sigue habiendo pérdidas en la red IP y hay que saber cómo reaccionar ante ellas



Van Jacobson

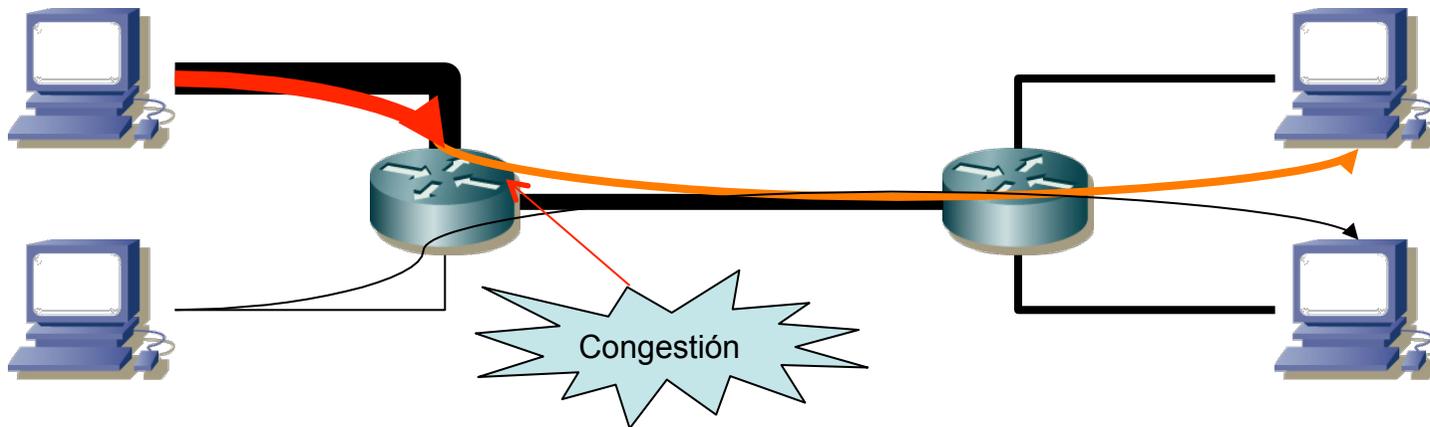
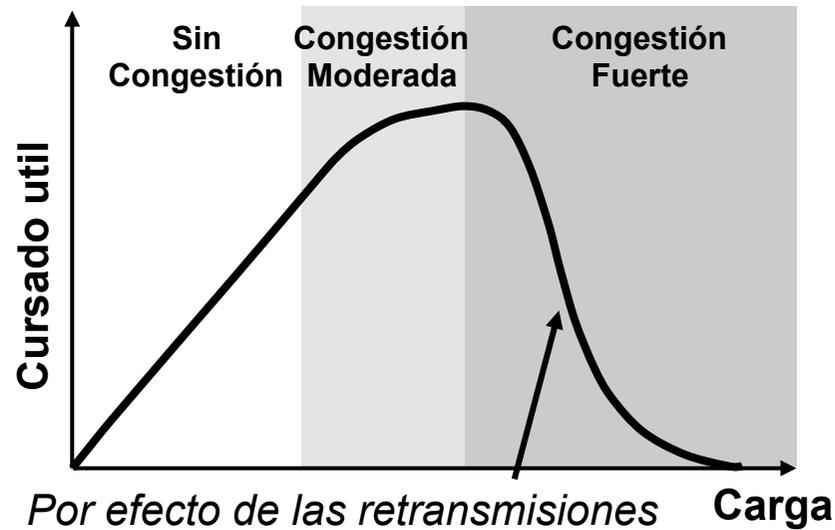
Congestion collapse

- Los primeros casos se dieron en los 80s
- ARPANET, TCP/IP protocolos experimentales
- Sin control de congestión
- LANs 10Mbps unidas por enlaces 56Kbps
- <http://www.youtube.com/watch?v=QP4A6L7CEqA>
- Elevadas tasas de tráfico introducido desde las redes de acceso
- (...)



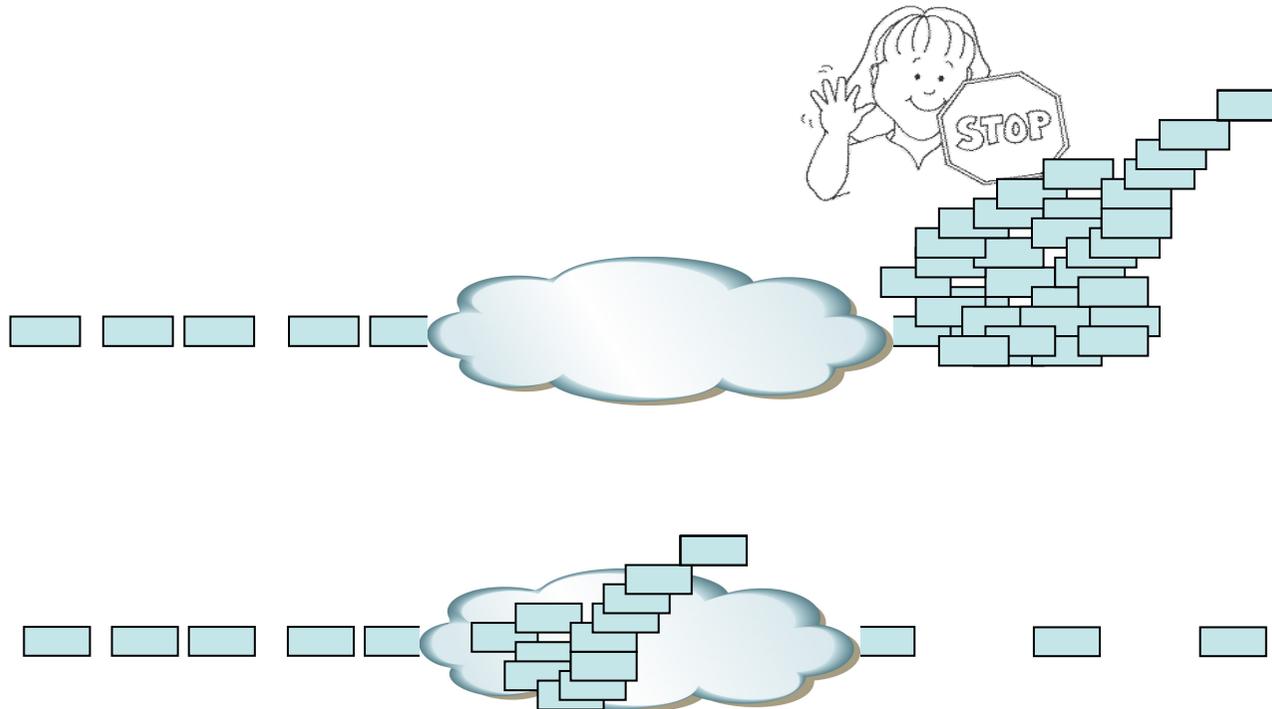
Congestion collapse

- En la red cada vez menos tráfico la atraviesa
- Congestión, retardos, pérdidas, retransmisiones...
- Hace falta una técnica para ajustar la tasa de envío de esos flujos



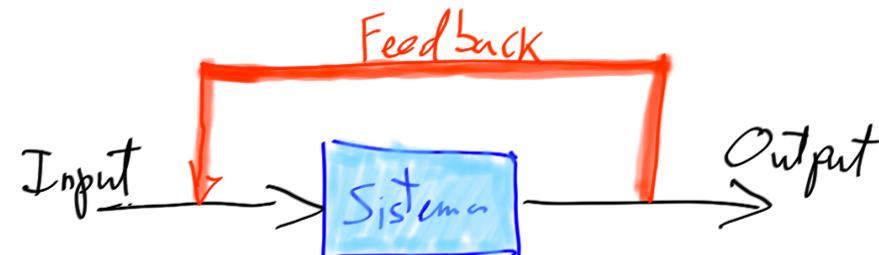
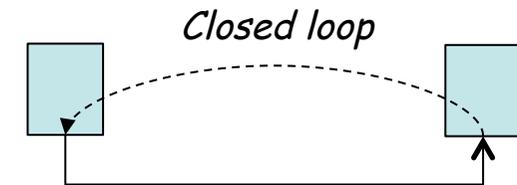
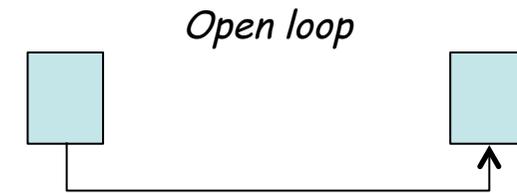
Control de flujo o de congestión

- El objetivo del control de flujo es proteger al receptor de un exceso de tráfico
- El objetivo del control de congestión es proteger a la red
- La fuente debería enviar a una tasa que sea el mínimo de lo que acepta la red y lo que acepta el receptor
- Los mecanismos acaban entremezclados



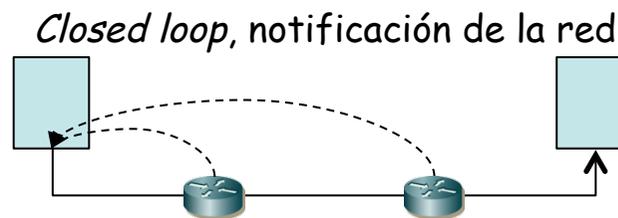
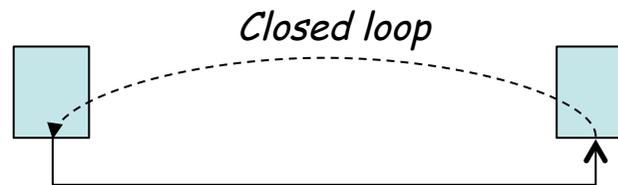
Bucle abierto o cerrado

- Necesitamos “controlar” esas tasas de envío
- El control puede ser:
 - En bucle abierto (proactivo):
 - Acciones tomadas unilateralmente por la fuente
 - Por ejemplo porque conozca la capacidad del cuello de botella
 - Suele requerir control de admisión en la red
 - En bucle cerrado (reactivo):
 - Acciones en base a realimentación
 - La información puede venir del destino o de nodos intermedios



Realimentación implícita

- ¿Realimentación explícita o implícita?
- De momento consideraremos que no hay una notificación explícita de la congestión por parte de la red



Realimentación implícita

- ¿Realimentación explícita o implícita?
- De momento consideraremos que no hay una notificación explícita de la congestión por parte de la red
- La realimentación implícita se basa en lo que pueden detectar los extremos:
 - Retardos: El retardo que sufren los paquetes, por transmisión, propagación, procesado, encolado, retransmisiones en nivel de enlace, etc.
 - Pérdidas: Que los paquetes nunca lleguen debido a cola llena, filtrado, fallo de equipos, corrupción por ruido, etc.
 - Corrupción: Que los paquetes lleguen alterados por ruido, equipo defectuoso, usuario malicioso, etc.
- Hay múltiples causas posibles para cada efecto medido
- Haremos suposiciones, por ejemplo:
 - Incremento RTT por mayor encolado, no por cambio de ruta
 - Descartes debidos a cola llena, no a corrupción por ruido
 - etc.



Realimentación binaria

- Las pérdidas de paquetes han sido la notificación implícita empleada desde el comienzo del control de congestión en TCP
- Requiere que efectivamente se deban a congestión y no por corrupción en entornos ruidosos (wireless...)



- En respuesta el emisor debe reducir su tasa de envío



- No tiene sentido solo reducir; necesita alguna regla para saber cuándo aumentarla
- Ante una señal de que sí llegan datos
- ¿Cómo hacemos el incremento/decremento?



{M|A}I{M|A}D

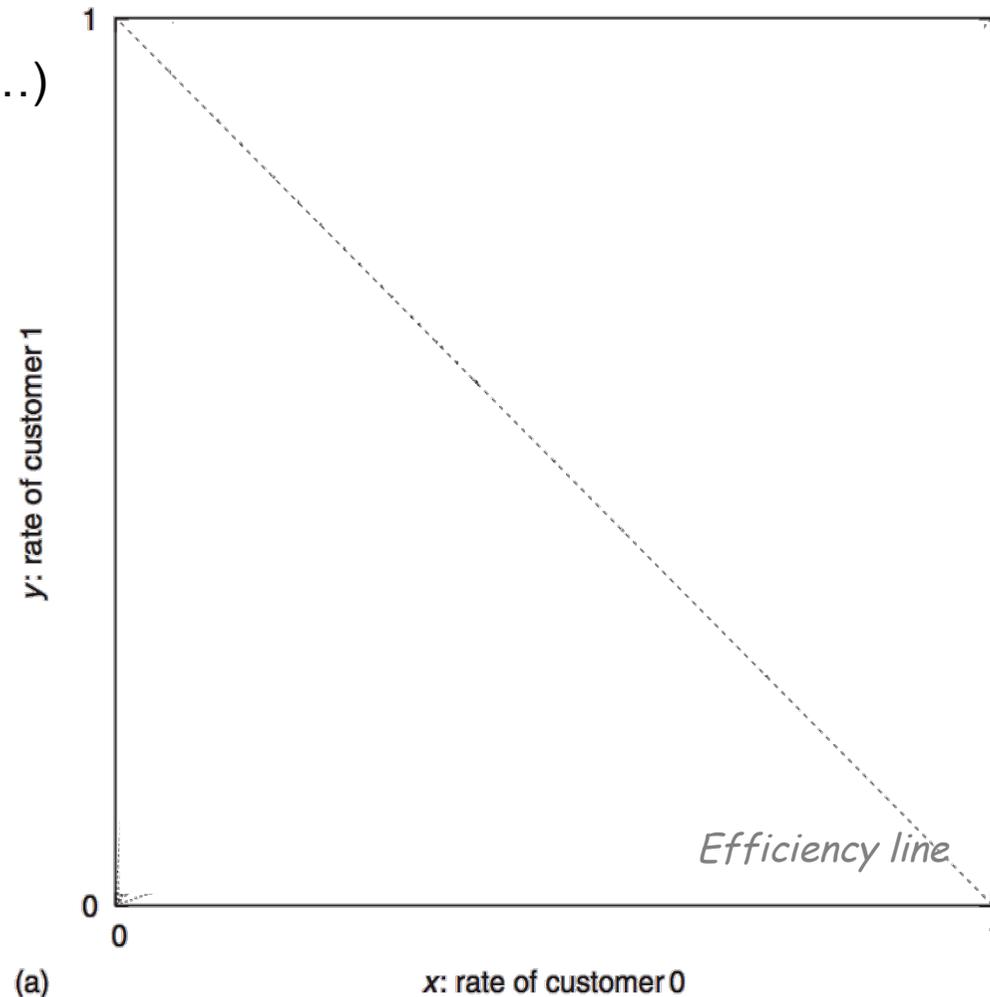
- $x(t)$: tasa de envío
- Planteamos una familia de controles para esta tasa:

$$x(t + \Delta t) = \begin{cases} a_i + b_i x(t) & \text{si no hay congestión} \\ a_d + b_d x(t) & \text{si hay congestión} \end{cases}$$

- MIMD: *Multiplicative Increase, Multiplicative Decrease*
 $a_i=0; a_d=0; b_i>1; 0<b_d<1$
- AIAD: *Additive Increase, Additive Decrease*
 $a_i>0; a_d<0; b_i=1; b_d=1$
- AIMD: *Additive Increase, Multiplicative Decrease*
 $a_i>0; a_d=0; b_i=1; 0<b_d<1$
- MIAD: *Multiplicative Increase, Additive Decrease*
 $a_i=0; a_d<0; b_i>1; b_d=1$

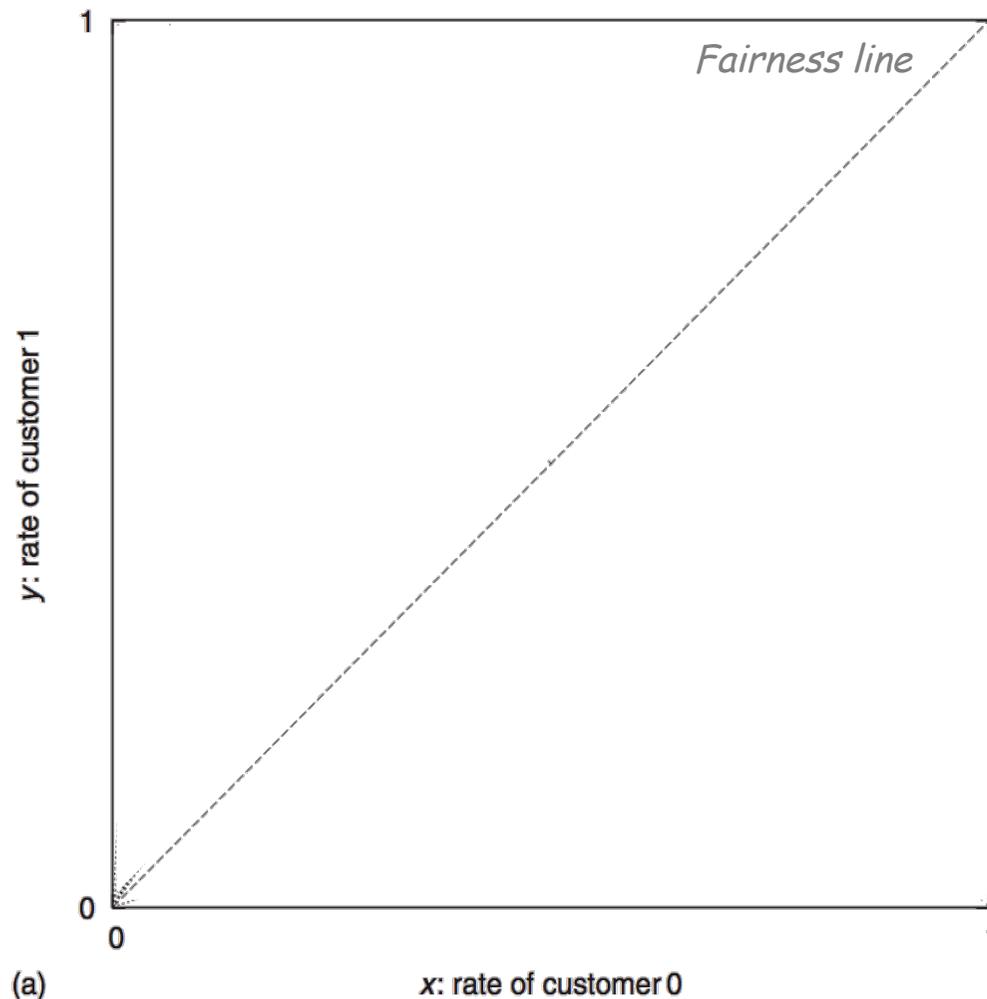
AIAD, MIMD, MIAD

- Supongamos dos fuentes saturando un enlace (capacidad unidad)
- Uso eficiente (*efficiency*)
 - Por debajo capacidad libre
 - Por encima pérdidas
- Reparto justo (*fairness*) (...)



AIAD, MIMD, MIAD

- Supongamos dos fuentes saturando un enlace (capacidad unidad)
- Uso eficiente (*efficiency*)
 - Por debajo capacidad libre
 - Por encima pérdidas
- Reparto justo (*fairness*)
- (...)

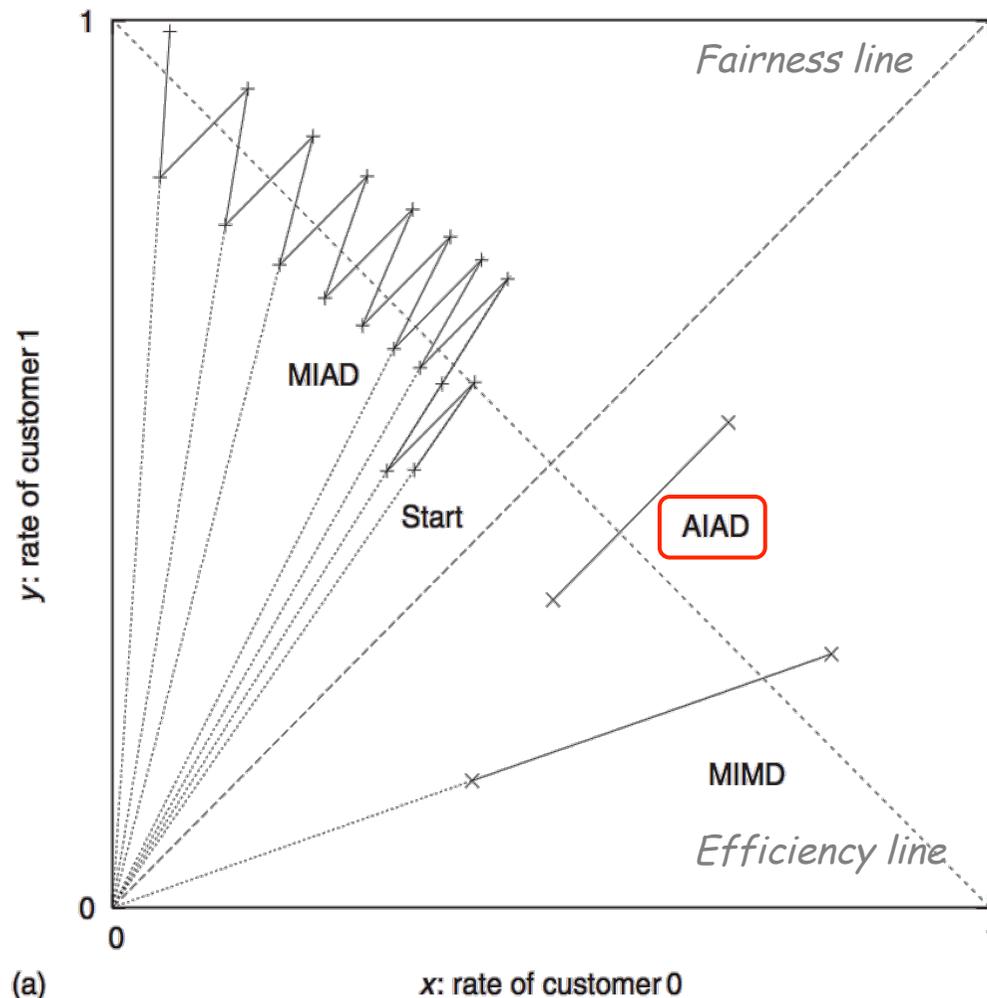


AIAD, MIMD, MIAD

- Supongamos dos fuentes saturando un enlace (capacidad unidad)
- Uso eficiente (*efficiency*)
 - Por debajo capacidad libre
 - Por encima pérdidas
- Reparto justo (*fairness*)

AIAD

- Si empiezan bajo línea de eficiencia ...
- Aumentan tasa igual ambos (paralelo a *fairness*)
- Hasta cruzar la línea de eficiencia y tener pérdidas
- Decrementa tasa igual ambos (paralelo a *fairness*)



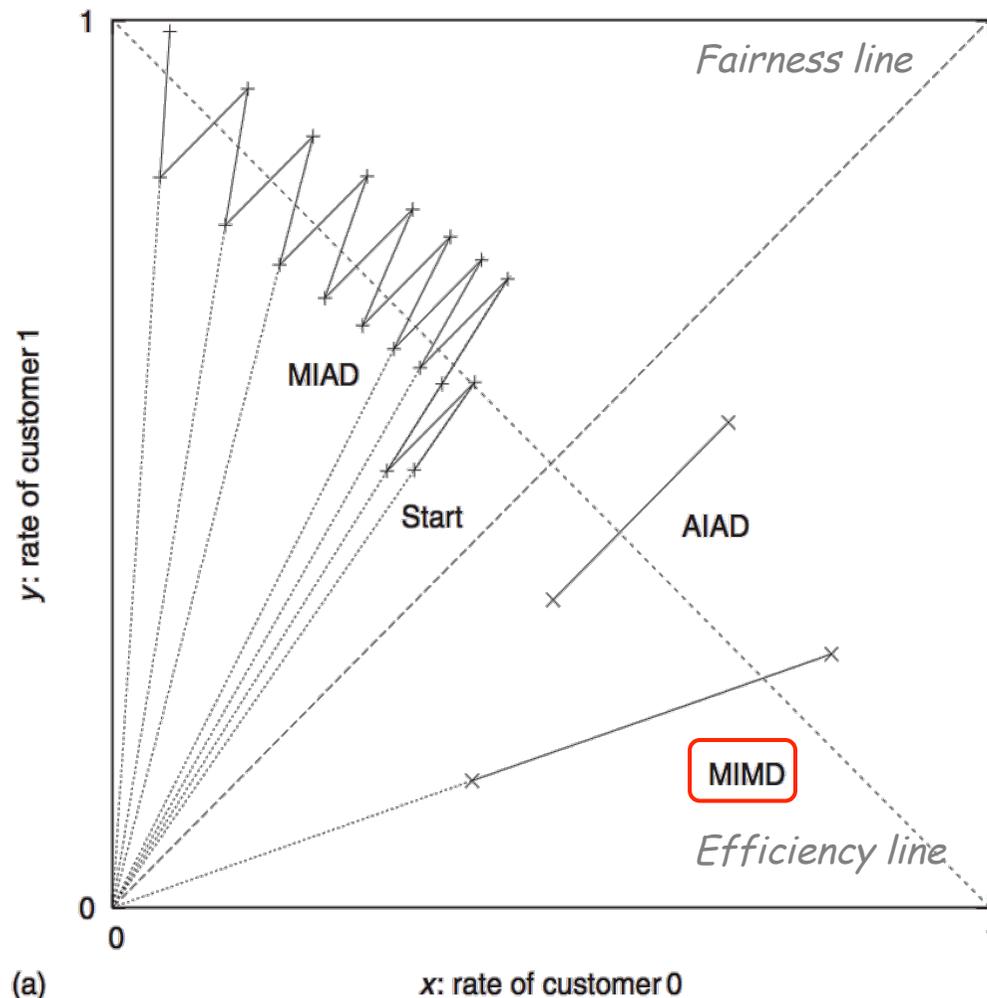
AIAD, MIMD, MIAD

- Supongamos dos fuentes saturando un enlace (capacidad unidad)
- Uso eficiente (*efficiency*)
 - Por debajo capacidad libre
 - Por encima pérdidas
- Reparto justo (*fairness*)

AIAD

MIMD

- Si empiezan bajo línea de eficiencia ...
- Aumentan tasa igual ambos en un factor (recta por origen)
- Hasta cruzar la línea de eficiencia y tener pérdidas
- Decrementan tasa igual ambos en un factor (recta por origen)



AIAD, MIMD, MIAD

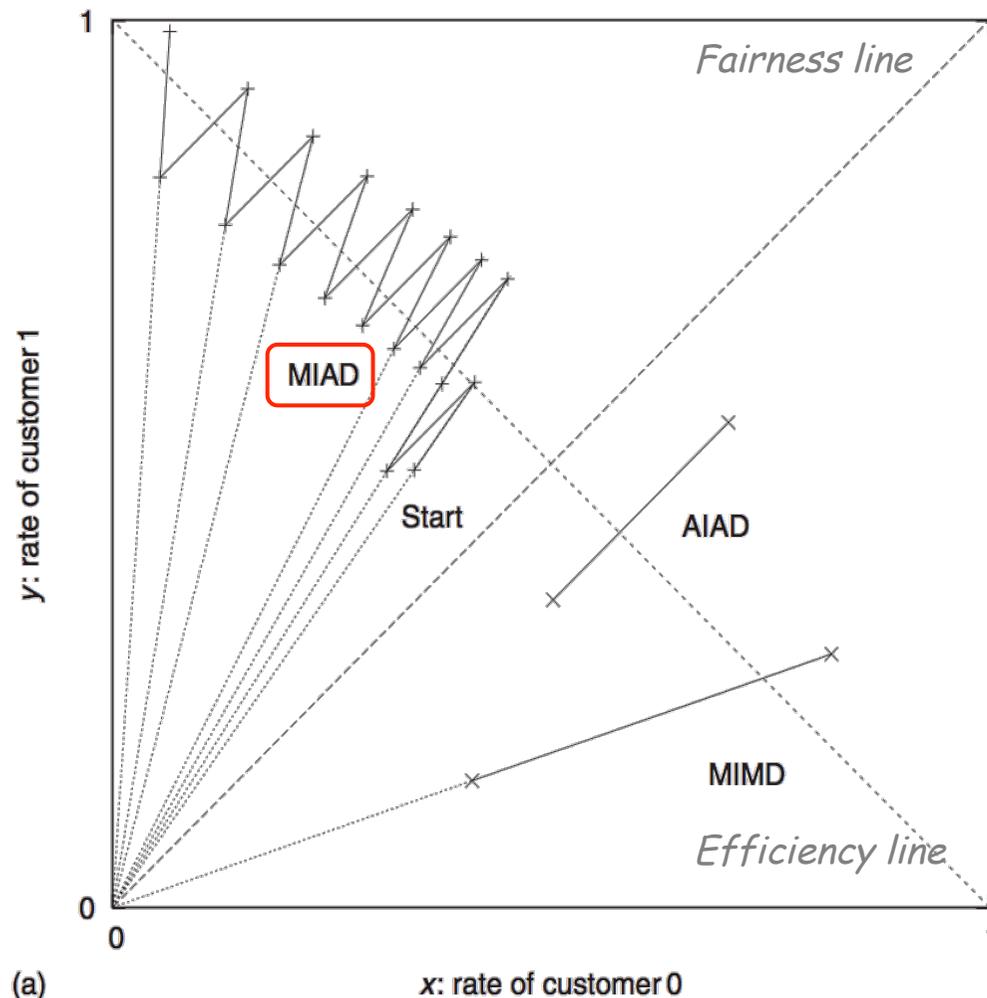
- Supongamos dos fuentes saturando un enlace (capacidad unidad)
- Uso eficiente (*efficiency*)
 - Por debajo capacidad libre
 - Por encima pérdidas
- Reparto justo (*fairness*)

AIAD

MIMD

MIAD

- Si empiezan bajo línea de eficiencia ...
- Aumentan tasa igual ambos en un factor (recta por origen)
- Hasta cruzar la línea de eficiencia y tener pérdidas
- Decrementan tasa igual ambos (paralelo a *fairness*)
- Converge a injusto
- Todo para el que empezó con más

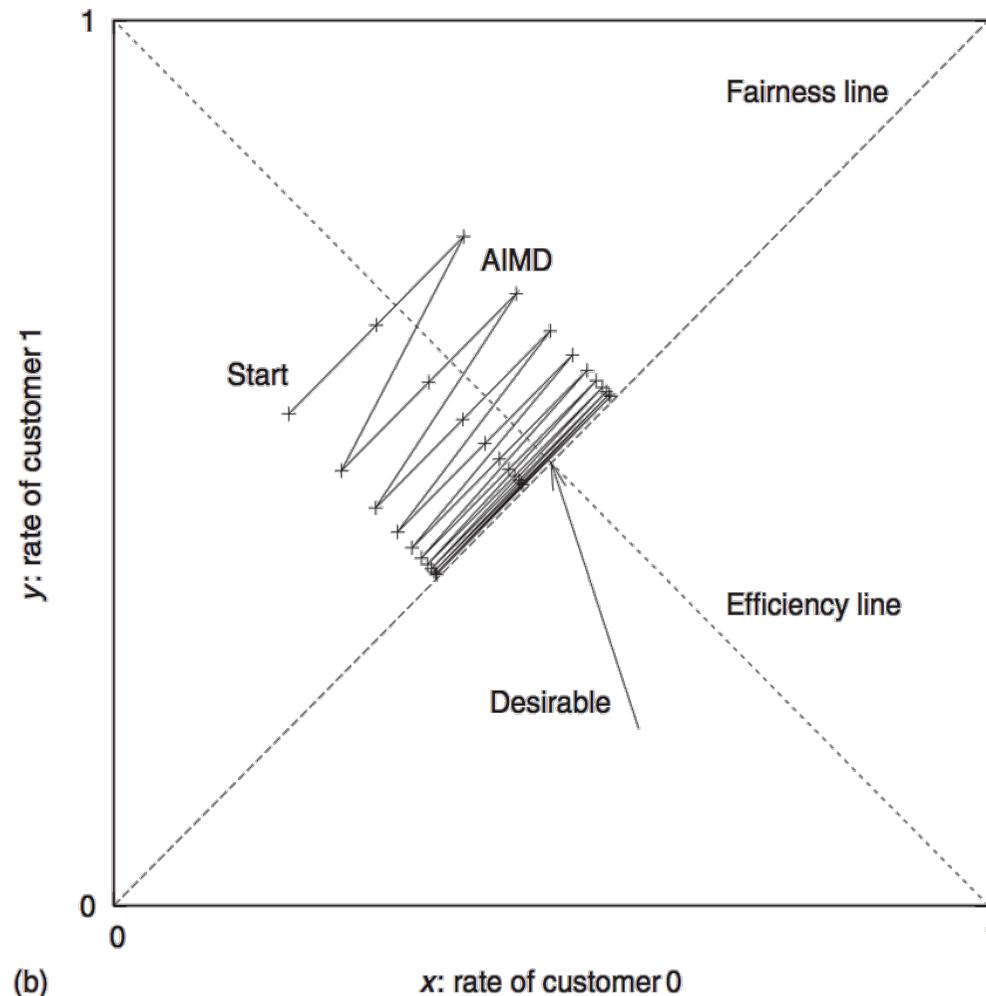


AIMD



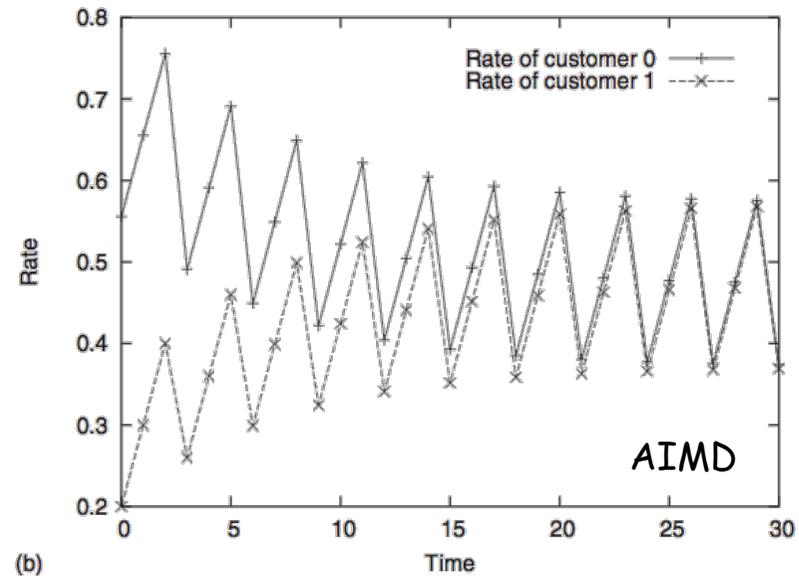
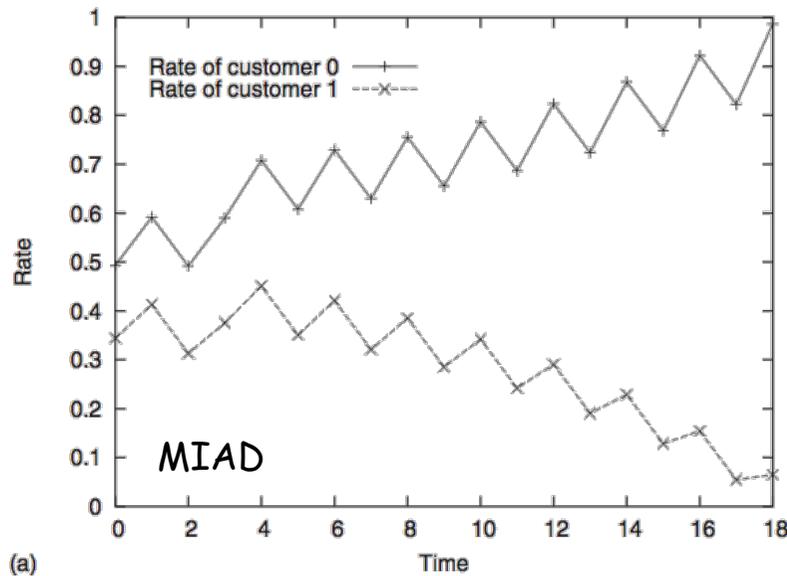
Raj Jain

- Si empiezan bajo la línea de eficiencia
- Aumentan tasa igual ambos (paralelo a *fairness*)
- Hasta cruzar la línea de eficiencia y tener pérdidas
- Decrementan tasa igual ambos en un factor (recta por origen)
- Converge a un equilibrio
- Fluctúa en torno al óptimo



MIAD y AIMD vs time

- En MIAD se pierde el reparto justo
- En AIMD se logra pero con oscilaciones
- El control de congestión de TCP es básicamente AIMD
- Si el tiempo que tarda la realimentación es diferente para cada fuente puede no dar reparto justo
- El tiempo de la realimentación se basa en timeout que se calcula en base a estimación del RTT
- ¿Cómo hacer ese control de la velocidad?



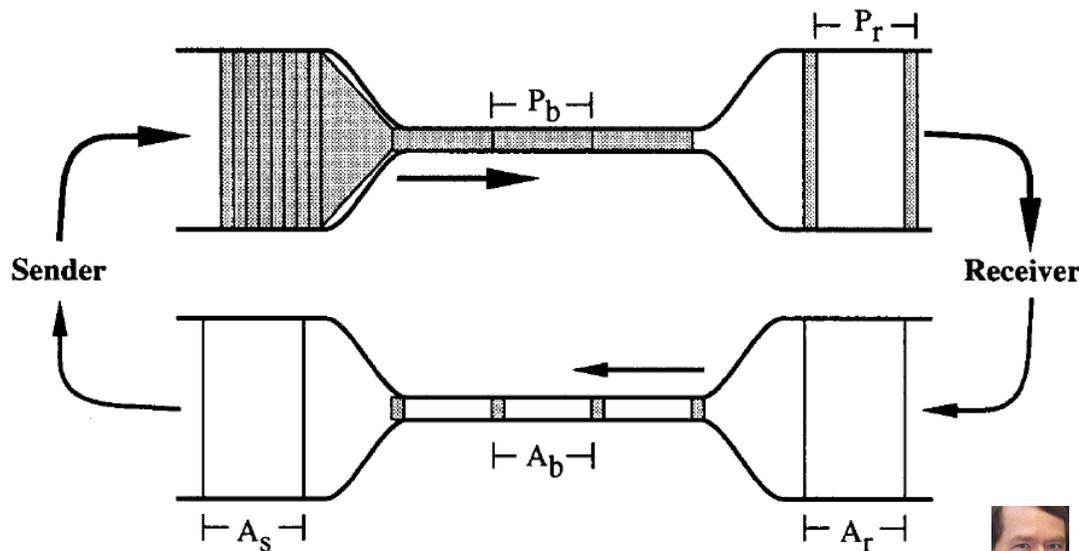
Rate- vs Window- control

- Dos métodos de control del envío datos
- Control basado en la tasa de envío
 - De alguna forma el emisor sabe a qué tasa debe enviar
 - Le informa el receptor o un equipo intermedio
 - Simple
 - Más apropiado para streaming
 - No queremos que se detenga pues la fuente no lo hace
 - Mantendría la tasa ante congestión
 - Si se le notifica de la nueva tasa adecuada podría hacer transcodificación



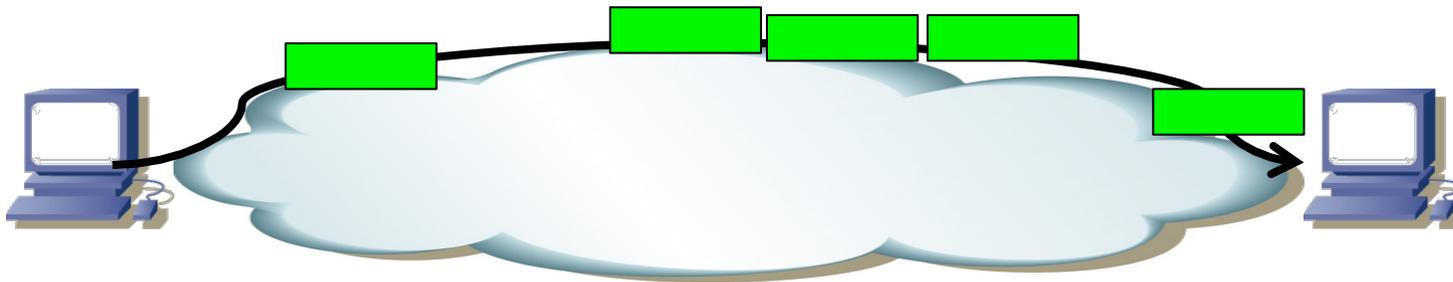
Rate- vs Window- control

- Dos métodos de control del envío datos
- **Control basado en ventana**
 - Hay un máximo de paquetes que pueden estar “en vuelo”
 - Solo se puede introducir uno cuando otro ha salido/llegado
 - Receptor notifica a emisor de que un paquete ha llegado
 - Más conservador pues deja de enviar si paquetes “no salen”
 - Self-Clocking: la tasa de envío está limitada por la de ACKs
 - Pero puede dar lugar a ráfagas (...)



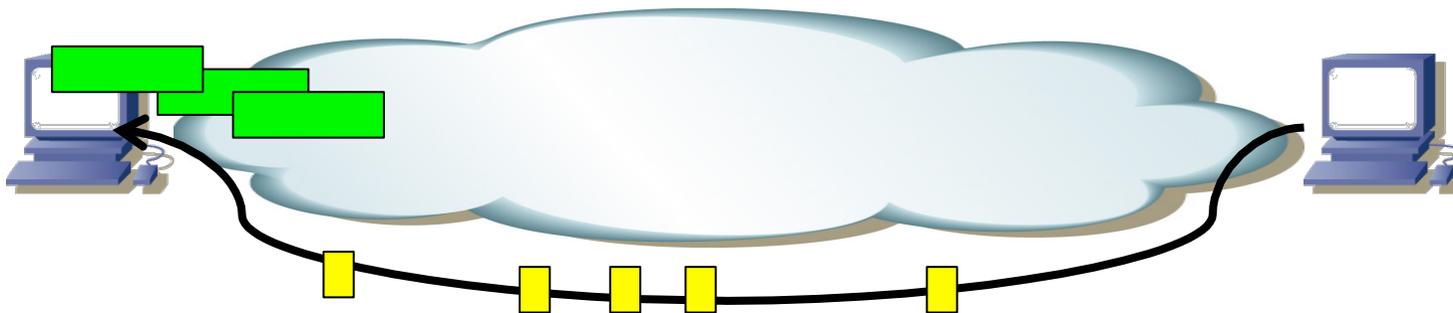
Rate- vs Window- control

- Dos métodos de control del envío datos
- Control basado en ventana
 - **Pero puede dar lugar a ráfagas**
 - Ejemplo:
 - Emisor envía y los paquetes se encolan en un puerto de salida
 - Cuando se transmiten salen juntos a la tasa de ese enlace
 - Puede ser de alta capacidad y solo haber sufrido congestión temporal
 - (...)



Rate- vs Window- control

- Dos métodos de control del envío datos
- Control basado en ventana
 - Pero puede dar lugar a ráfagas
 - Ejemplo:
 - Emisor envía y los paquetes se encolan en un puerto de salida
 - Cuando se transmiten salen juntos a la tasa de ese enlace
 - Puede ser de alta capacidad y solo haber sufrido congestión temporal
 - Los ACKs de paquetes que llegan próximos saldrán próximos
 - Y provocarán que se “abra” la ventana de golpe y el emisor pueda enviar varios paquetes a la tasa de su interfaz



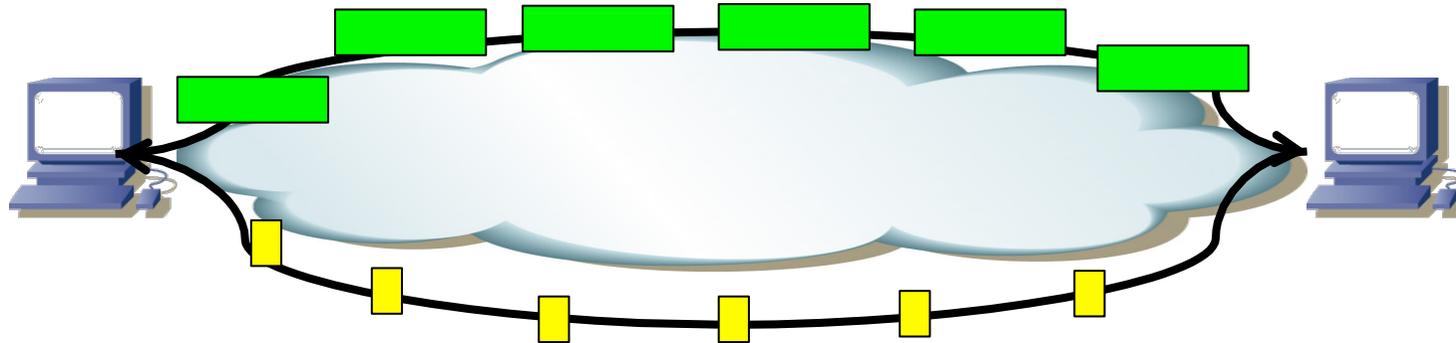
Rate- vs Window- control

- Dos métodos de control del envío datos
- Control basado en ventana
 - Pero puede dar lugar a ráfagas
 - **Otro ejemplo:**
 - Ha habido una transferencia pero ya no hay más datos a transmitir
 - Se confirma todo, la ventana queda completamente “abierta”
 - Ahora si la aplicación genera una gran cantidad de datos se podrán enviar tantos como la ventana en una ráfaga



Rate- vs Window- control

- Dos métodos de control del envío datos
- Control basado en ventana
 - La ventana debe ser lo suficientemente grande para sacar provecho a todo el canal
 - Eso es un tamaño al menos $BW \times RTT$
 - Eso requiere una buena estimación del RTT



RTT para calcular RTO

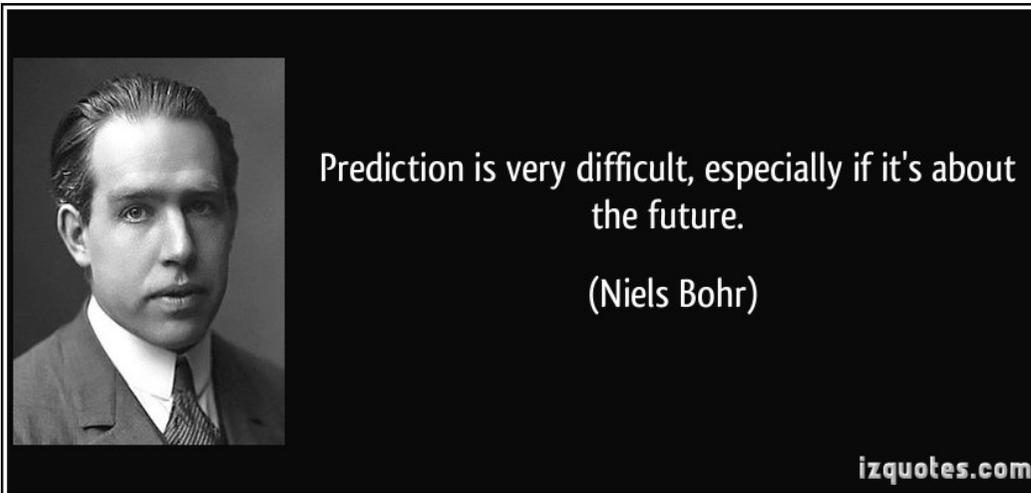
- *Retransmission Timeout*
- Esquema ARQ (Automatic Repeat Request)
- Tras un tiempo sin recibir confirmación se retransmite
- ¿Valor de ese tiempo?
- Si sobre-estimamos tarda en reconocer la pérdida
 - Tarda en retransmitir
 - Tarda en darse cuenta de que puede haber congestión
- (...)

**REPEAT YOUR
MESSAGE
FOR SUCCESS**



RTT para calcular RTO

- Si sub-estimamos retransmite sin necesidad y considera que hay congestión cuando no la hay
- Idealmente debería ser un RTT o algo relacionado
- Cuidado que el RTT medido es “viejo”
- Se hace una predicción



Resumen

- El exceso de demanda lleva a congestión
- Incluso puede colapsar la red
- Las fuentes (protocolos) controlan el envío, en general mediante técnicas de control de ventana
- Incrementan la tasa “despacio” (aditivo) y decrementar ante congestión de forma agresiva (multiplicativo) : AIMD
- Se puede enviar tanto como “quepa” en la red
- Necesitamos una estimación del RTT
- El RTO va a ponerse a partir de la estimación del RTT