

# Capítulo 3: Capa Transporte: Principios del control de congestión

ELO322: Redes de Computadores  
Agustín J. González

Este material está basado en:

- Material de apoyo al texto *Computer Networking: A Top Down Approach*. Jim Kurose, Keith Ross.

# Capítulo 3: Continuación

- ❑ 3.1 Servicios de la capa transporte
- ❑ 3.2 Multiplexing y demultiplexing
- ❑ 3.3 Transporte sin conexión: UDP
- ❑ 3.4 Principios de transferencia confiable de datos
- ❑ 3.5 Transporte orientado a la conexión: TCP
  - Estructura de un segmento
  - Transferencia confiable de datos
  - Control de flujo
  - Administración de conexión
- ❑ 3.6 Principios del control de congestión
- ❑ 3.7 Control de congestión en TCP

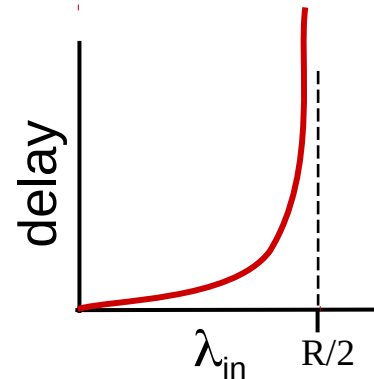
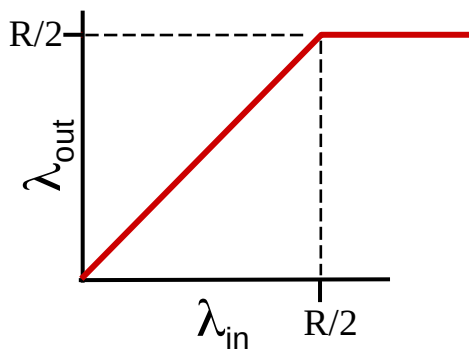
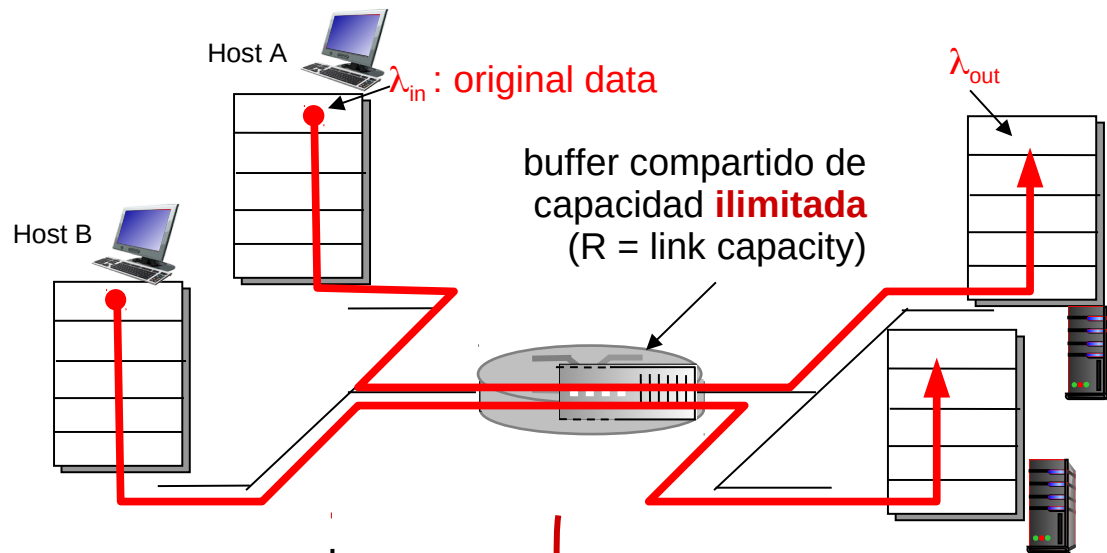
# Principios del control de congestión

## Congestión:

- ❑ Informalmente: “demasiadas fuentes enviando muchos datos muy rápido para que la red lo maneje”
- ❑ **Es distinto a control de flujo**, el cual ocurre entre dos procesos (Tx y Rx).
- ❑ Manifestaciones:
  - Pérdidas de paquetes (buffer overflow en routers)
  - Grandes retardos (en las colas en los router)
- ❑ Uno de los problemas top-10! en redes de computadores.

# Causas/costos de congestión: escenario 1 (buffer $\infty$ , sin re-envío)

- dos transmisores, dos receptores
- un router, buffer **infinito** (utópico, no real)
- **sin retransmisión**
- $\lambda_{in}$  datos enviados por cada aplicación (bytes/s)
- $\lambda_{out}$  datos recibidos por la aplicación (bytes/s)

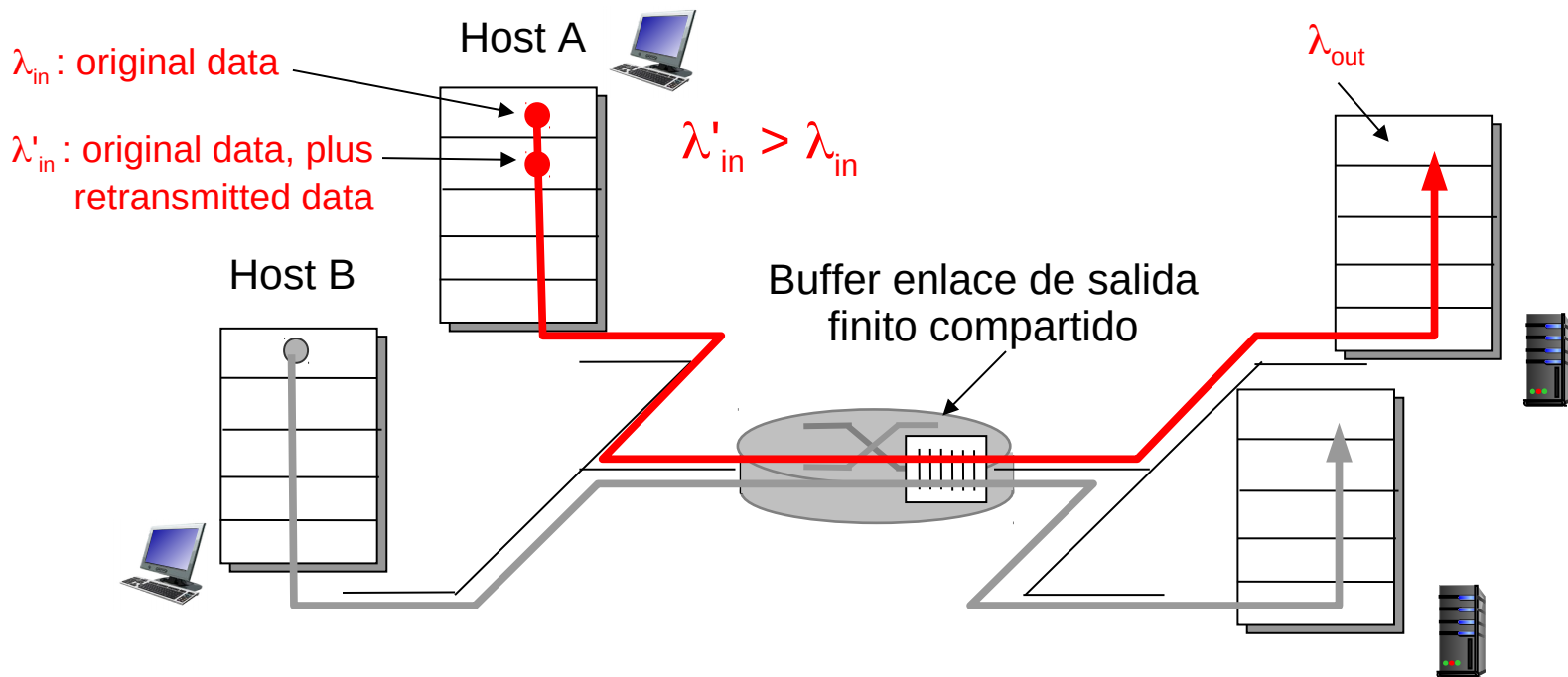


- Throughput máximo por conexión:  $R/2$

- Se incrementa el retardo cuando la tasa de llegada,  $\lambda_{in}$ , se acerca a la capacidad

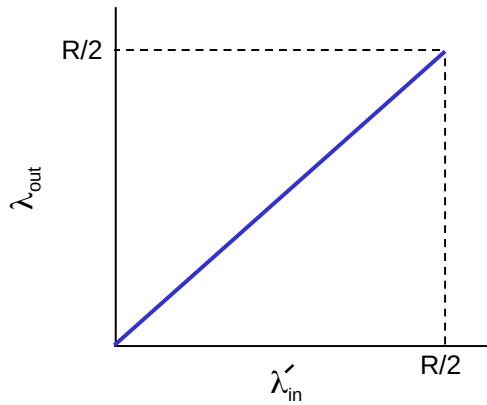
# Causas de congestión: escenario 2 (buffer no $\infty$ , hay re-envío de paquetes)

- ❑ un router, buffer *finito (caso real)*
- ❑ Se **pierden** paquetes que no tienen espacio en buffer.
- ❑ transmisor **retransmite** paquetes perdidos

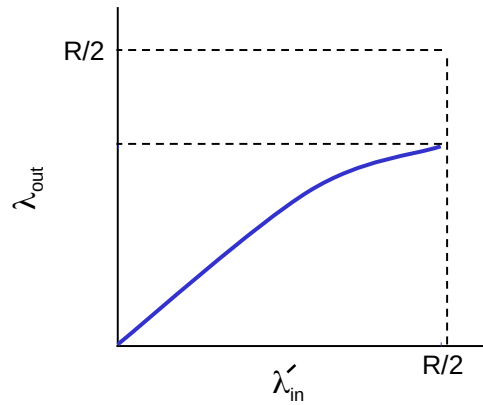


# Causas/costos de congestión: escenario 2

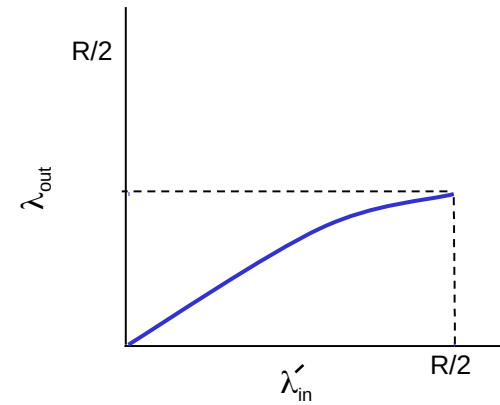
- (a) caso perfecto, sin pérdidas ni retransmisiones:  $\lambda_{in} = \lambda'_{in}$
- (b) retransmisión sólo en caso de pérdida de segmentos:  $\lambda'_{in} > \lambda_{in}$ .  
Retransmisión de paquetes hace que  $\lambda'_{in}$  sea más grande que  $\lambda_{in}$ .
- (c) además de b, puede haber retransmisiones innecesarias (timeout prematuros): enlaces envían paquetes de datos duplicados.



a.



b.



c.

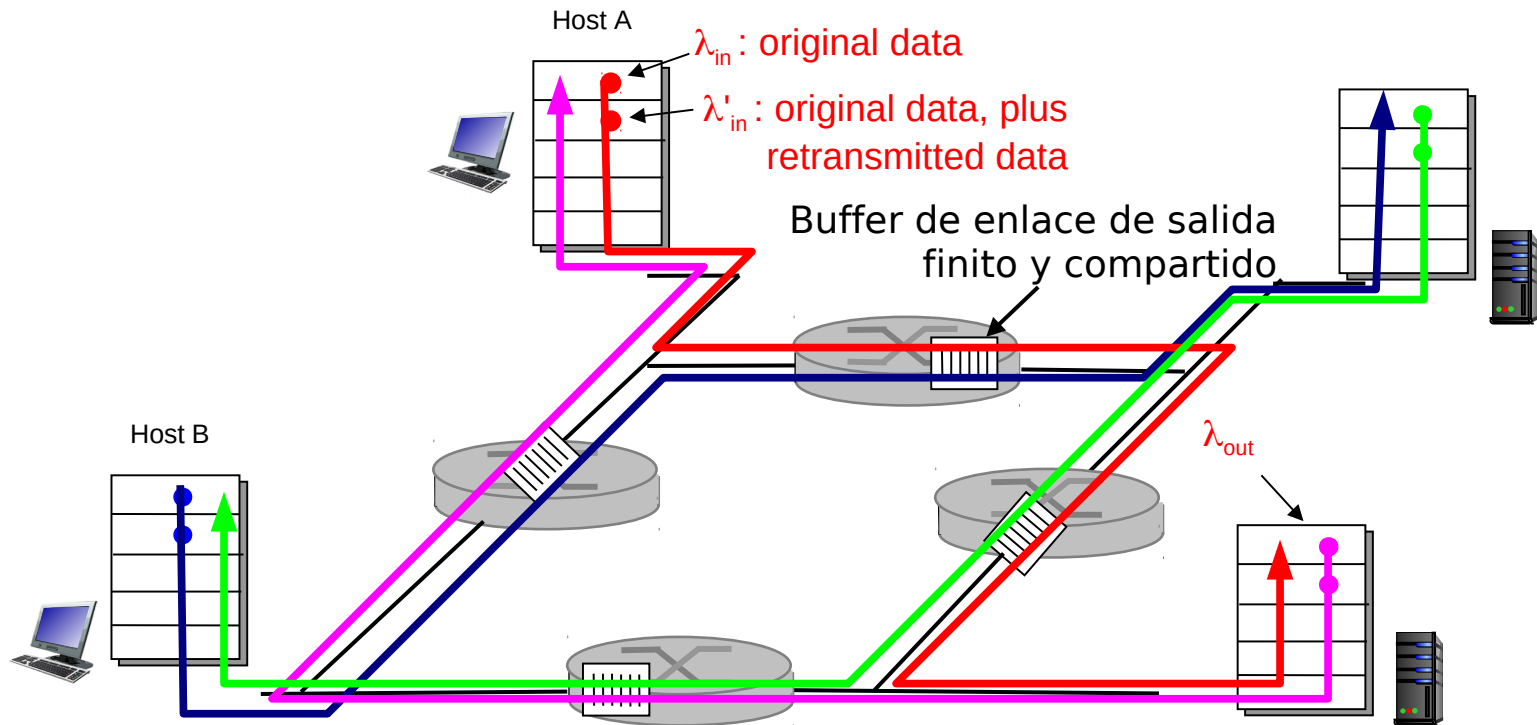
## “costos” de congestión:

- más trabajo (retransmisión) para lograr el transporte de datos  
“goodput”

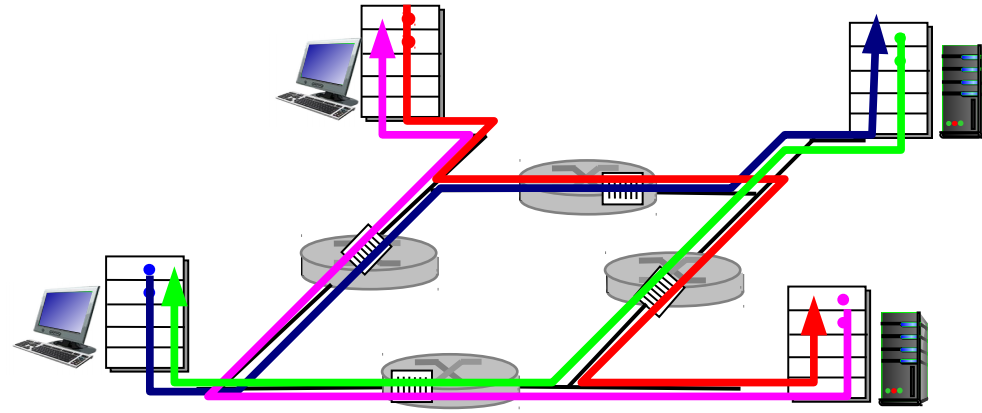
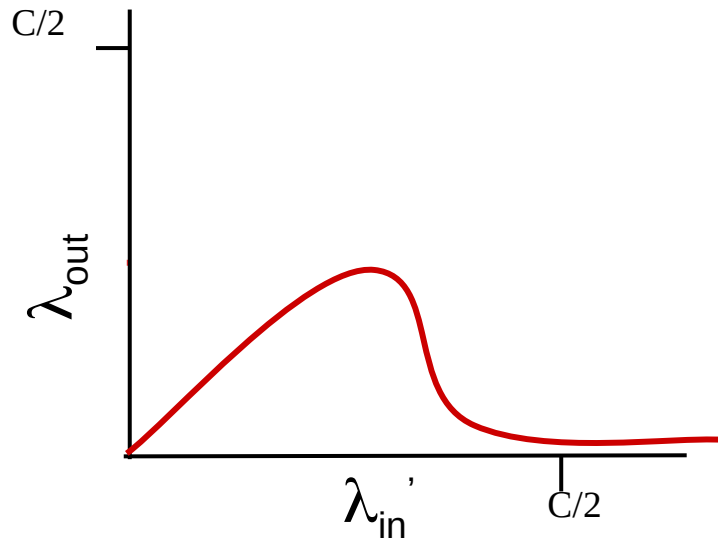
# Causas/costos de congestión: escenario 3

- ❑ cuatro transmisores
- ❑ rutas con multihops
- ❑ timeout/retransmisiones

**Q:** ¿Qué pasa cuando  $\lambda_{in}$  se incrementa y  $\lambda'_{in}$  crece?



# Causas/costos de congestión: escenario 3



## Otro "costo" de congestión:

- cuando se descartan paquetes, cualquier capacidad (de router) usada anteriormente pasa a ser un recurso desperdiciado!