

# Capítulo 3: Capa Transporte - IV

## ELO322: Redes de Computadores Agustín J. González

Este material está basado en:

- Material de apoyo al texto *Computer Networking: A Top Down Approach Featuring the Internet*. 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:

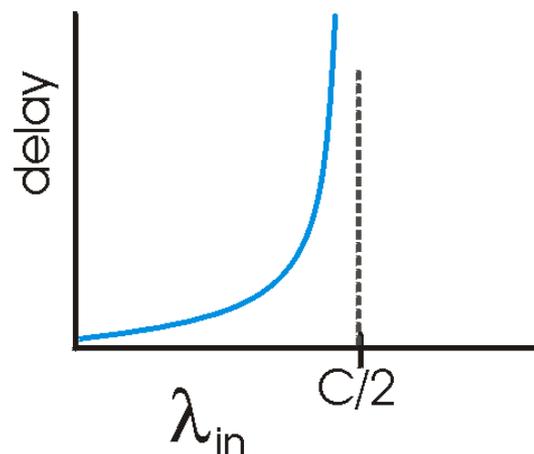
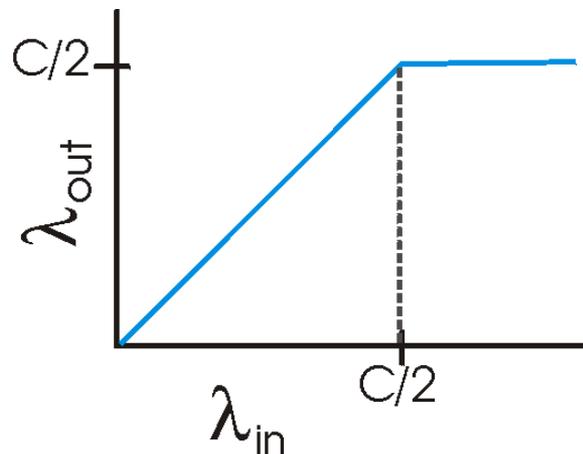
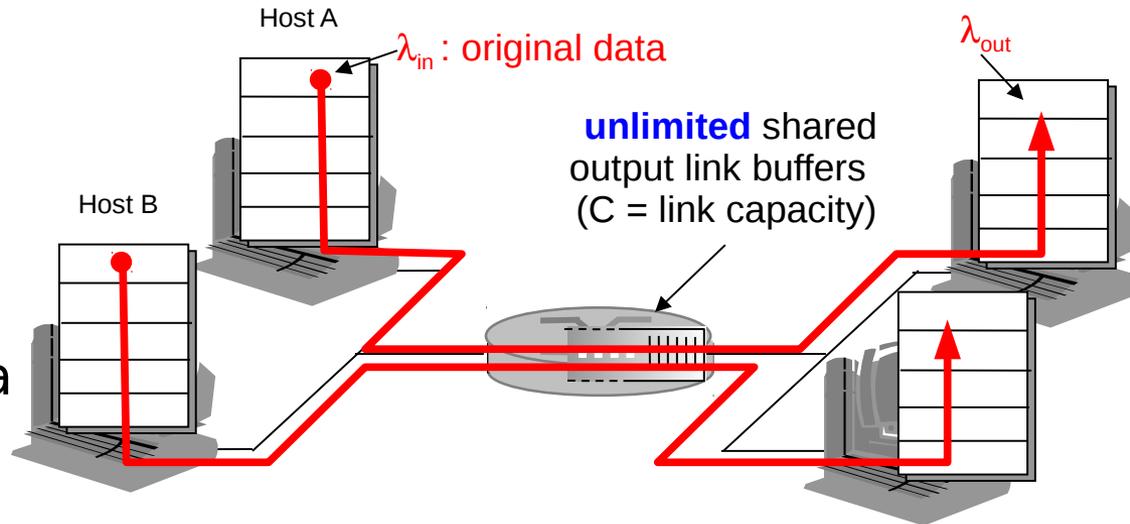
- ❑ Informalmente: “demasiadas fuentes enviando muchos datos muy rápido para que la red lo maneje”
- ❑ Es distinto a control de flujo, el cual ocurre entre dos procesos (Tx y Rx).
- ❑ Manifestaciones:
  - Pérdidas de paquetes (buffer overflow en routers)
  - Grandes retardos (en las colas en los router)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

La capa de red no ofrece garantías de: entrega de paquetes en orden, tasas de transferencia fija, llegada confiable de datos, y retardo acotado desde transmisión hasta recepción. ¿Cuáles de estos requerimientos son posibles de garantizar vía una programación adecuada de la capa de transporte?

- Entrega de paquetes en orden y llegada confiable de datos

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

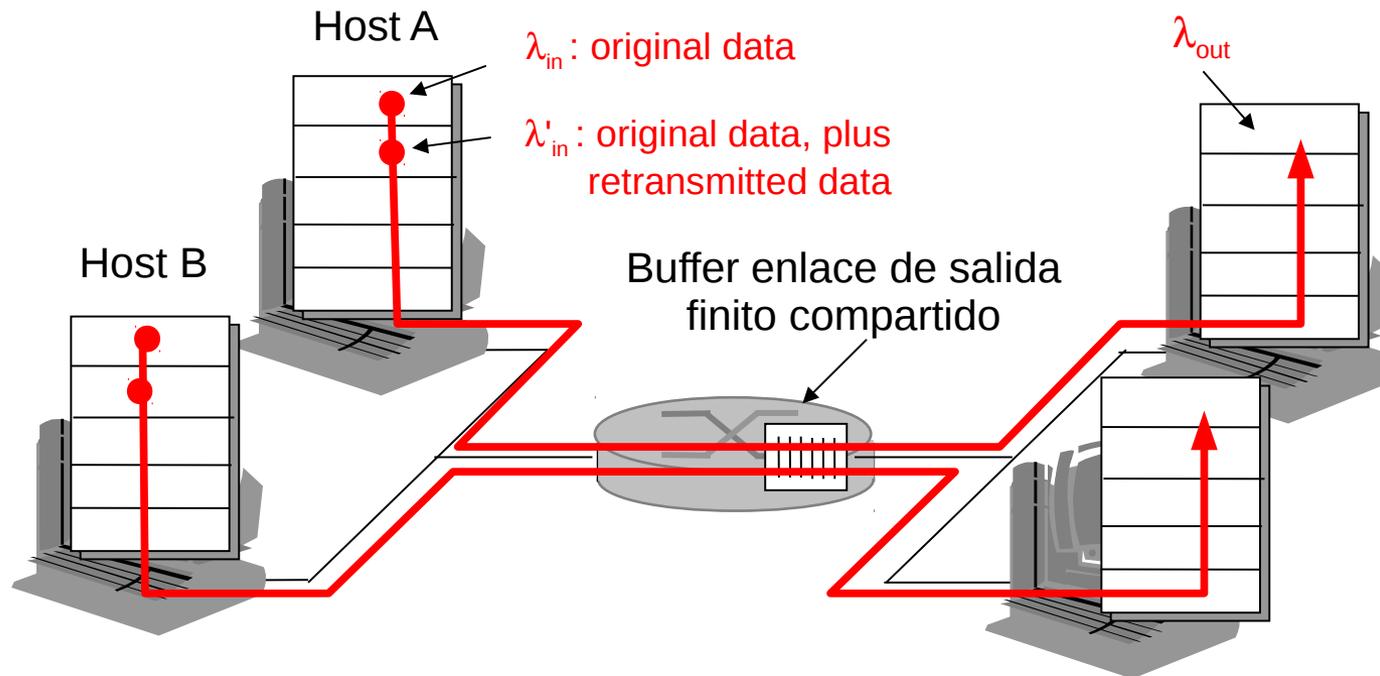
- dos transmisores, dos receptores
- un router, buffer tamaño **infinito**
- **sin retransmisión**
- $\lambda_{in}$  datos enviados por la aplicación (bytes/sec)
- $\lambda_{out}$  datos recibidos por la aplicación (bytes/sec)



- grandes retardos en estado de congestión (muchos paquetes esperando en cola)
- máximo flujo posible (throughput =  $C$ ) de datos

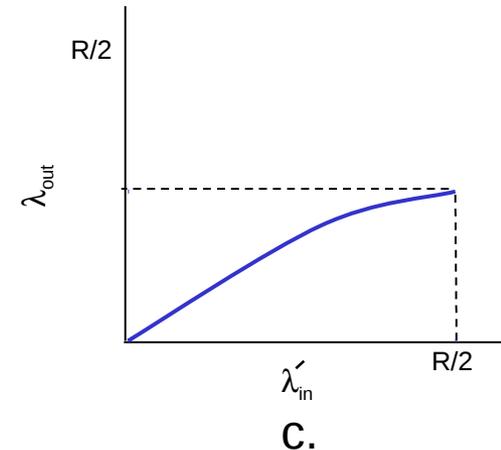
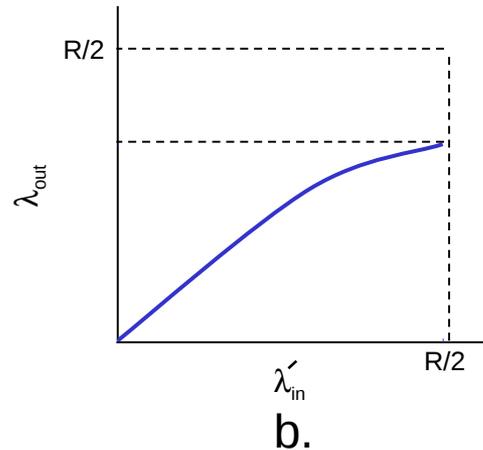
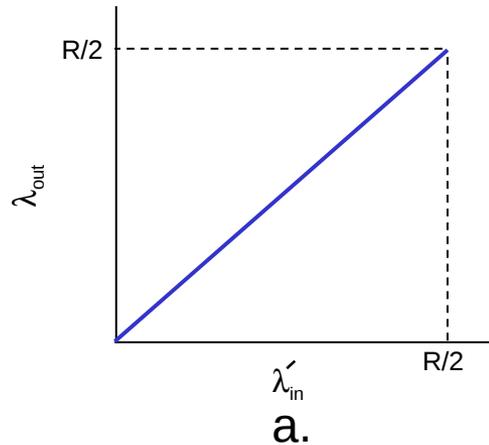
# Causas de congestión: escenario 2 (buffer no $\infty$ , con re-envío )

- ❑ un router, buffer *finito*
- ❑ 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 (no por timeout prematuro):  $\lambda'_{in} > \lambda_{out}$ . Retransmisión de paquetes hace que  $\lambda'_{in}$  sea más grande (que el caso perfecto) para el mismo  $\lambda_{out}$ .
- (c) además de b, retransmisiones innecesarias (timeout prematuros) : enlaces envían paquetes de datos duplicados.



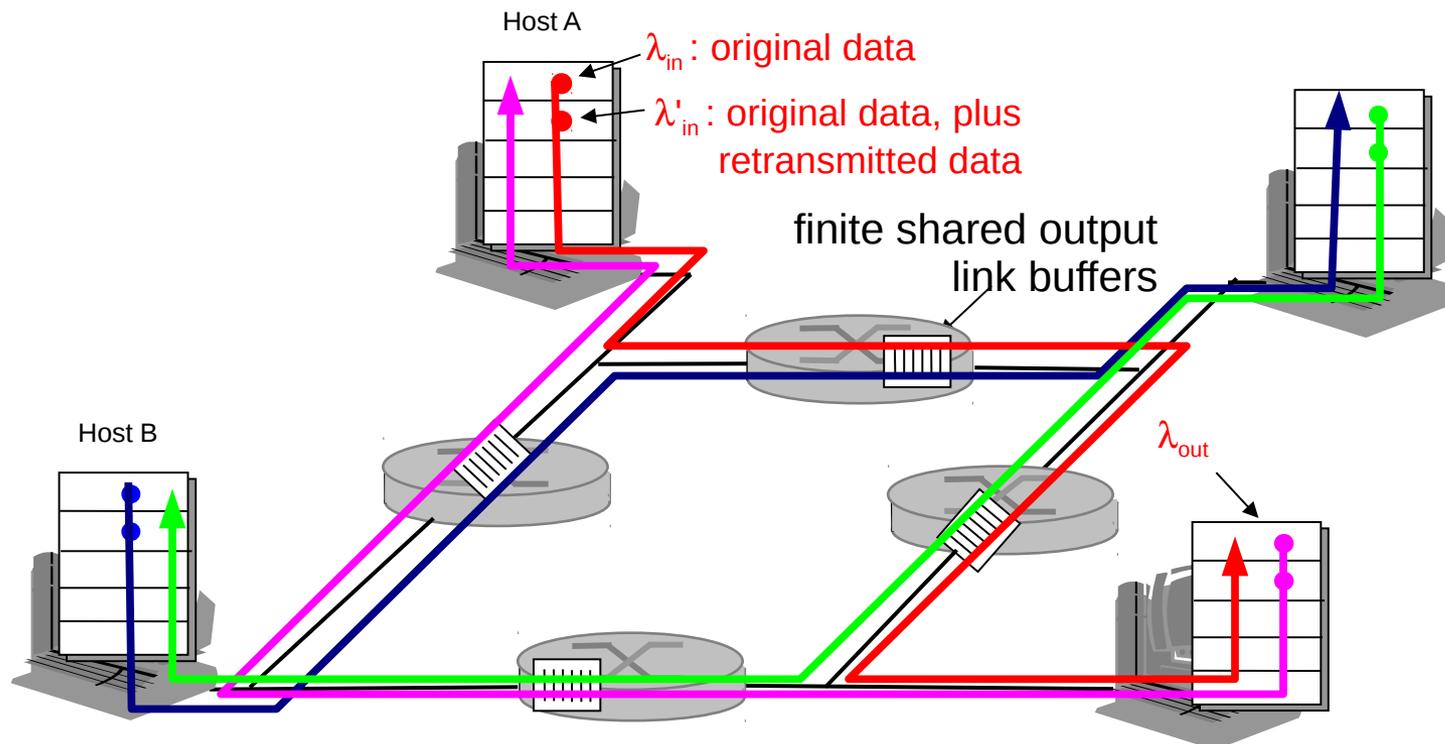
## “costos” de congestión:

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

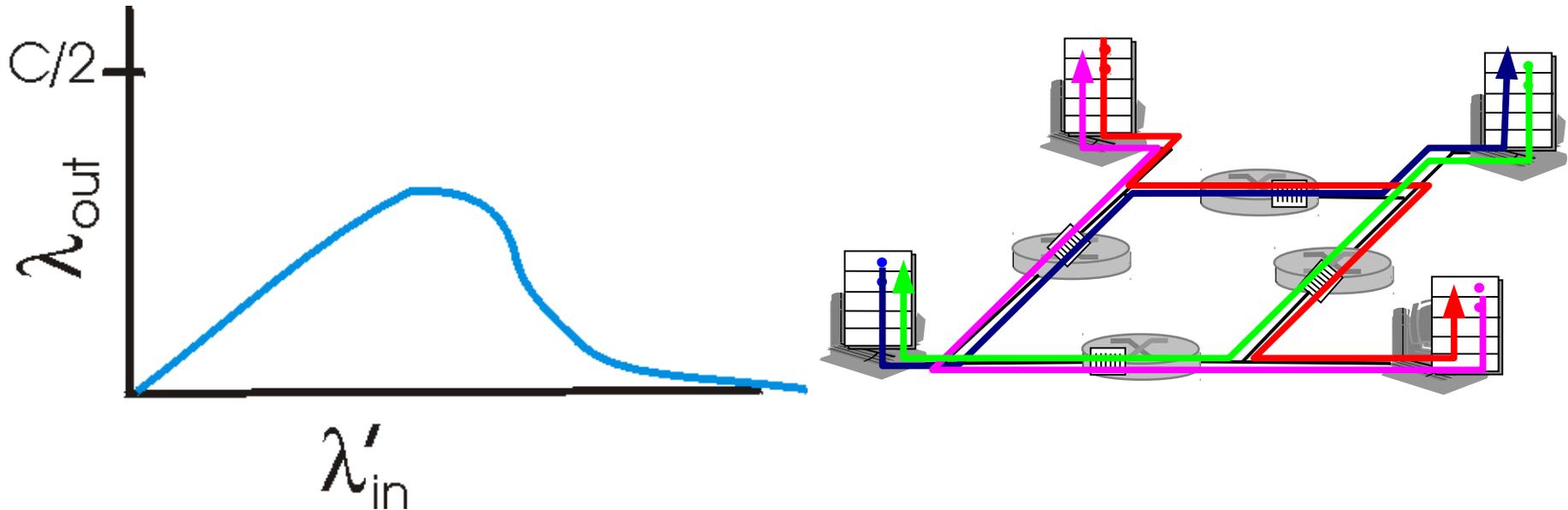
# Causas/costos de congestión: escenario 3

- ❑ 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!

# Estrategias para control de congestión

Los podemos clasificar en dos grupos amplios:

```
graph TD; A[Los podemos clasificar en dos grupos amplios:] --> B[Control de congestión extremo a extremo:]; A --> C[Control de congestión asistido por la red:];
```

## Control de congestión extremo a extremo:

- No hay información de realimentación explícita de la red
- La congestión es inferida desde las pérdidas y retardos observados por terminales en los extremos
- **Es la estrategia usada por TCP**

## Control de congestión asistido por la red:

- routers proveen realimentación a sistemas extremos
  - Un Bit único indicando congestión (e.g. SNA, DECbit, TCP/IP ECN, ATM)
  - Explícitamente se informa al Tx la tasa que el router puede soportar
- Estrategia redes ATM (Asynchronous Transfer Mode)

# Caso de estudio: Control de congestión en ATM ABR (tecnología de red capa 3 y menores)

## **ABR: Available Bit Rate:**

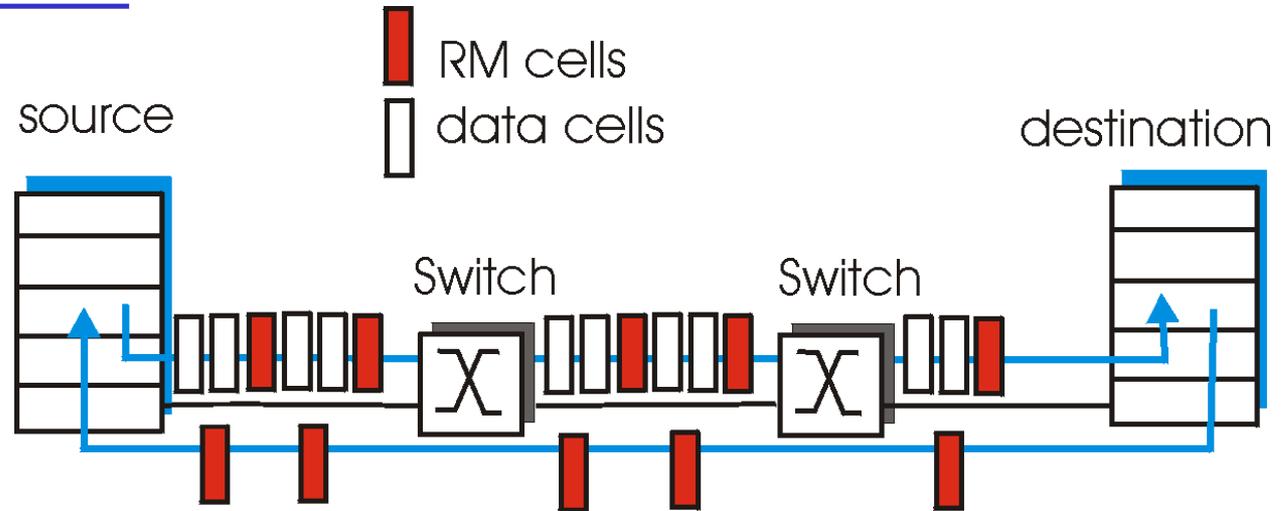
- ❑ Es un servicio “elástico” o flexible
- ❑ Si camino del Tx tiene poca carga,
  - Tx debería usar ancho de banda disponible
- ❑ Si camino de Tx a Rx está congestionado,
  - Tx reduce a un mínimo la tasa garantizada

## **Celdas RM (Resource Management):**

- ❑ Enviadas a intervalos por Tx entre celdas de datos
- ❑ bits en celda RM son modificados por switches
  - **Bit NI:** no incrementar tasa (= congestión moderada)
  - **Bit CI:** Congestion Indication
- ❑ Celdas RM son retornadas al Tx por el Rx con bits intactos

ATM: Asynchronous Transfer Mode

# Caso de estudio: Control de congestión en ATM ABR



- En celda RM hay campo ER (explicit rate) de dos bytes:
  - Un Switch congestionado puede bajar valor de ER en la celda
  - Tasa de envío del Tx se ajusta a la tasa mínima soportable en el camino entre fuente y destino (la del switch más crítico)
- En celda de datos hay Bit EFCI (explicit forward congestion indicator): éste es fijado en 1 por switch congestionado
  - Si celda de datos precedente a celda RM tiene el EFCI marcado, el destino marca bit CI en celda RM retornada.