

# Tecnología Digital IV: Redes de Computadoras

## Clase 11: Nivel de Transporte - Parte 5

Lucio Santi & Emmanuel Iarussi

Licenciatura en Tecnología Digital  
Universidad Torcuato Di Tella

10 de abril de 2025

# Agenda

- Servicios del nivel de transporte
- Multiplexación y demultiplexación
- Transporte no orientado a conexión: UDP
- Transferencia de datos confiable
- Transporte orientado a conexión: TCP
- **Protocolos modernos: QUIC**
- **Introducción al control de congestión**

# El protocolo QUIC

# Evolución del nivel de transporte

- TCP y UDP han sido los principales protocolos de transporte durante 40 años
- En el transcurso, se desarrollaron distintas variantes de TCP para abordar escenarios específicos, e.g.:

Escenario	Características
Redes inalámbricas	Pérdidas por enlaces inalámbricos “ruidosos” (TCP lo interpreta como congestión) <b>[TCP/NC]</b>
Caminos con capacidad y RTTs grandes	Ventanas de congestión muy grandes <b>[HSTCP]</b>
Redes de datacenters	Sensibilidad a delays <b>[DCTCP]</b>
Flujos de tráfico en <i>background</i>	Flujos TCP de baja prioridad <b>[TCP Nice]</b>

- Más recientemente, se optó por trasladar funcionalidad del nivel de transporte al nivel de aplicación (sobre UDP)
  - **HTTP/3: QUIC**

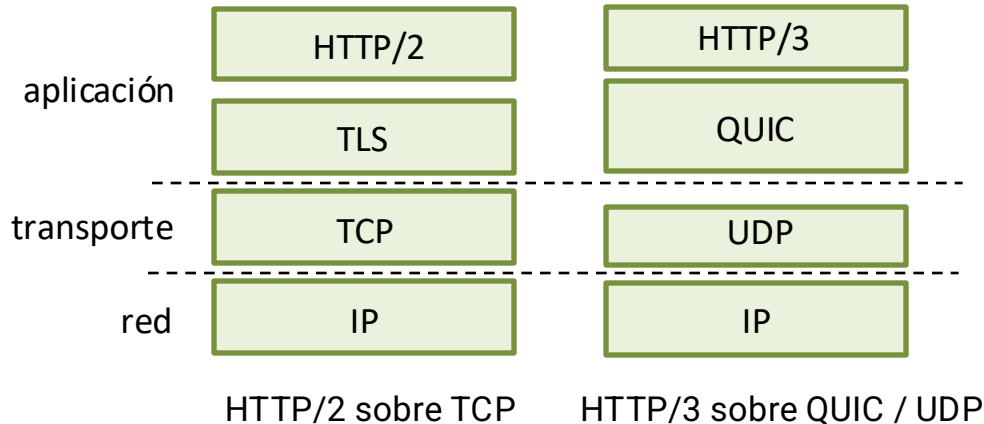
# QUIC: *¿Qué es?*

**QUIC** (Quick UDP Internet Connections) es un **protocolo de transporte confiable y seguro**, construido sobre **UDP**, que:

- Ofrece **bajo retardo en el establecimiento de conexión**.
- Integra **seguridad de extremo a extremo (como TLS 1.3)** directamente en la capa de transporte.
- Soporta **múltiples flujos simultáneos** sin interferencia entre ellos (multiplexación).
- Está optimizado para **aplicaciones web modernas**.

# QUIC: *Quick UDP Internet Connections*

- **QUIC** [[RFC 9000](#)] es un protocolo de aplicación que corre sobre UDP
  - Permite aumentar el rendimiento de HTTP (HTTP/3, [RFC 9114](#))
  - Desplegado en servidores y apps de Google (e.g. Chrome, YouTube)
  - Alrededor del **8%** del tráfico de Internet corre sobre QUIC (abril 2025)



# QUIC: *Quick UDP Internet Connections*

Incorpora diversos enfoques de TCP: establecimiento de conexión, control de errores y control de congestión (que veremos luego)

- **Control de errores y congestión:**

*"Readers familiar with TCP's loss detection and congestion control will find algorithms here that parallel well-known TCP ones."*

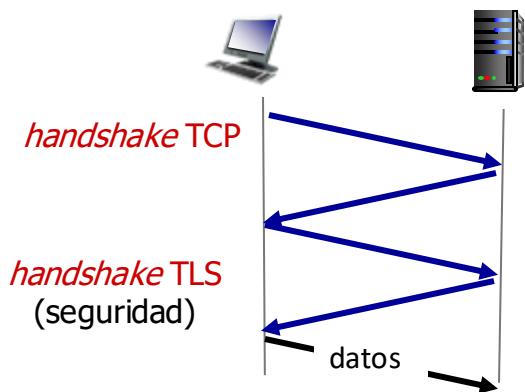
[fragmento del [RFC 9002](#)]

- **Establecimiento de conexión:** define estado para confiabilidad, control de congestión, autenticación y encriptación sólo en un RTT

- Multiplexa diversos *streams* de aplicación sobre una misma conexión

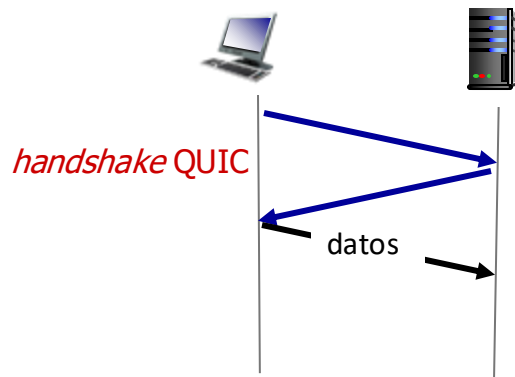
- Mantiene separado por *stream* la confiabilidad y la seguridad
- Control de congestión compartido

# QUIC: establecimiento de conexión



**TCP** (estado para confiabilidad y control de congestión) + **TLS** (estado para autenticación y encriptación)

- Dos *handshakes* en serie

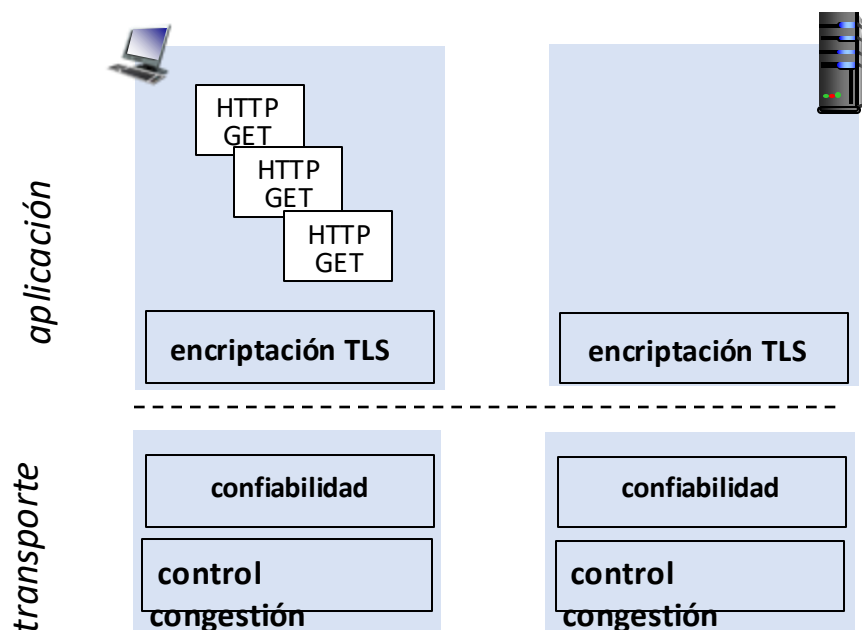


**QUIC:** estado para confiabilidad, control de congestión, autenticación y encriptación

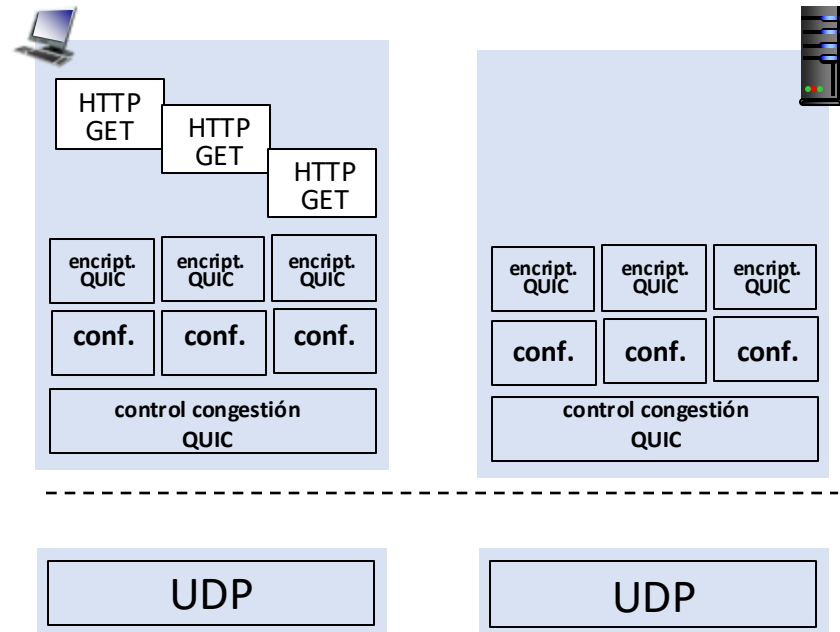
- Un *handshake*



# Streams en QUIC



HTTP/1.1



HTTP/3 + QUIC  
(sin *HOL blocking*)

# ¿Porqué usar QUIC y no TCP?

Problema con TCP	Cómo lo soluciona QUIC
Handshake lento (3 pasos + TLS)	Handshake rápido: 1 solo round-trip con cifrado incluido
Bloqueo por cabeza de línea	Múltiples flujos independientes (multiplexación)
Retransmisiones lentas y congestión	Mejor manejo de pérdidas y control de congestión
Cambios lentos en el estándar TCP	Va sobre UDP, permite evolución rápida sin cambiar el SO

# Intro al Control de Congestión

# Demoras en una red

Es el tiempo que tarda un paquete en viajar desde su origen hasta su destino.

Este retardo está compuesto por:

- Demora de procesamiento en cada router
- Demora de encolamiento (tiempo **en la cola** del router)
- Demora de transmisión (tiempo necesario para colocar todos los bits del paquete en el canal)
- Demora de propagación (tiempo que tardan las señales en recorrer el medio físico).

En redes congestionadas, la demora de encolamiento suele ser la que más varía y contribuye a un retardo mayor.

# Principios del control de congestión

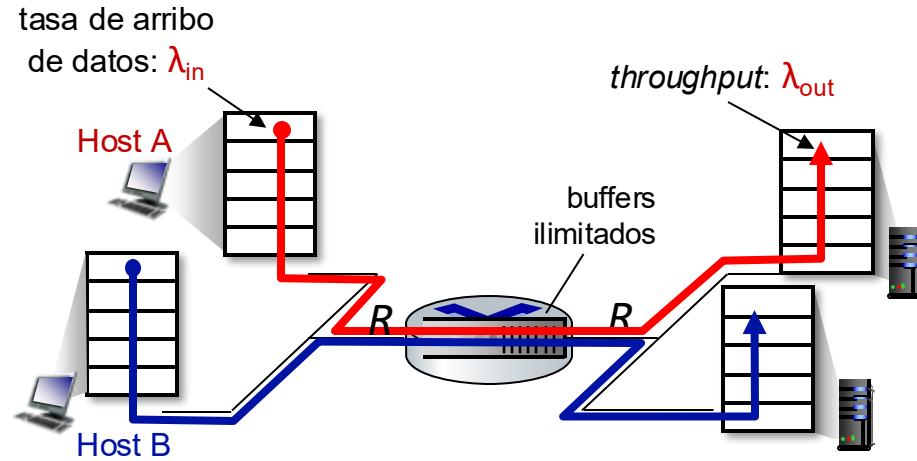
## Congestión

- En términos informales, *múltiples emisores enviando muchos datos, colapsando los recursos de la red*
- Síntomas:
  - Delays prolongados (encolamiento en routers)
  - Pérdida de paquetes (*overflow* de buffers en routers)
- Distinto a control de flujo:
  - Control de flujo: un emisor saturando a un receptor
- Uno de los problemas más **trascendentales** en redes de computadoras

# Causas y costos de la congestión: caso 1

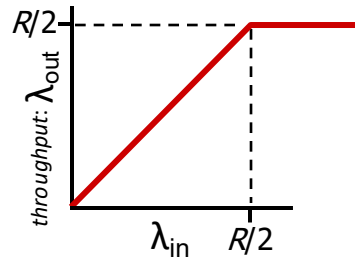
## Escenario simplista:

- Un router, buffers ilimitados
- Capacidad de los enlaces:  $R$
- Dos flujos (equivalentes)
- Sin retransmisiones

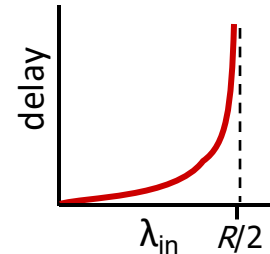


¿Qué ocurre cuando la tasa de arribo de datos  $\lambda_{in}$  se aproxima a  $R/2$ ?

$$\text{Demora total} = 1/((R/2) - \lambda_{in})$$



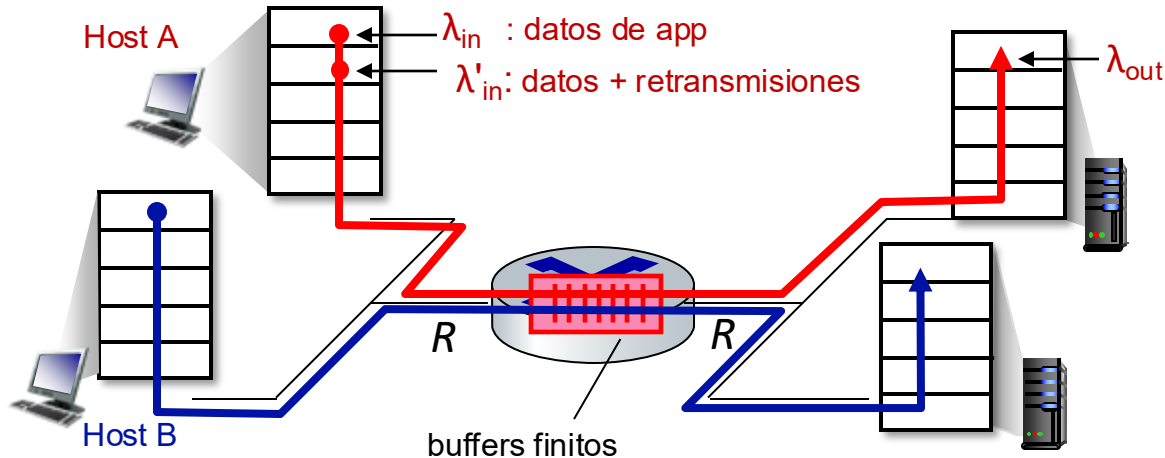
máximo throughput por conexión:  $R/2$



**delays prolongados** a medida que  $\lambda_{in}$  se aproxima a la capacidad

# Causas y costos de la congestión: caso 2

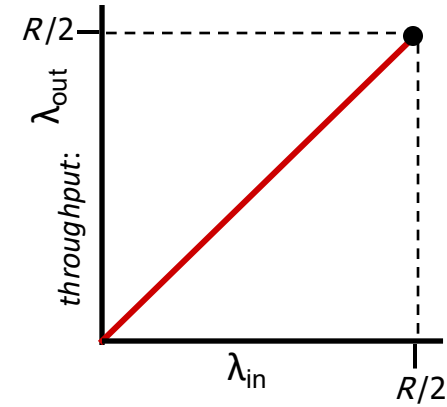
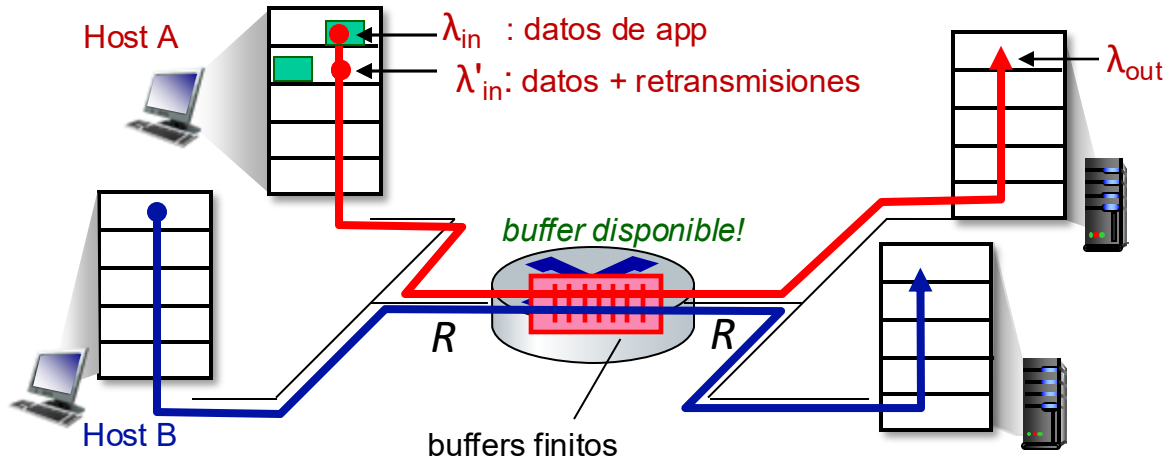
- Símil caso anterior pero con **buffers finitos**
- El emisor retransmite paquetes perdidos
  - Tasa de arribo mensajes de aplicación:  $\lambda_{in}$
  - Tasa de envío de **segmentos**:  $\lambda'_{in} \geq \lambda_{in}$



# Causas y costos de la congestión: caso 2

Suposición: *oráculo* de buffers libres

- El emisor envía segmentos sólo cuando hay buffers disponibles

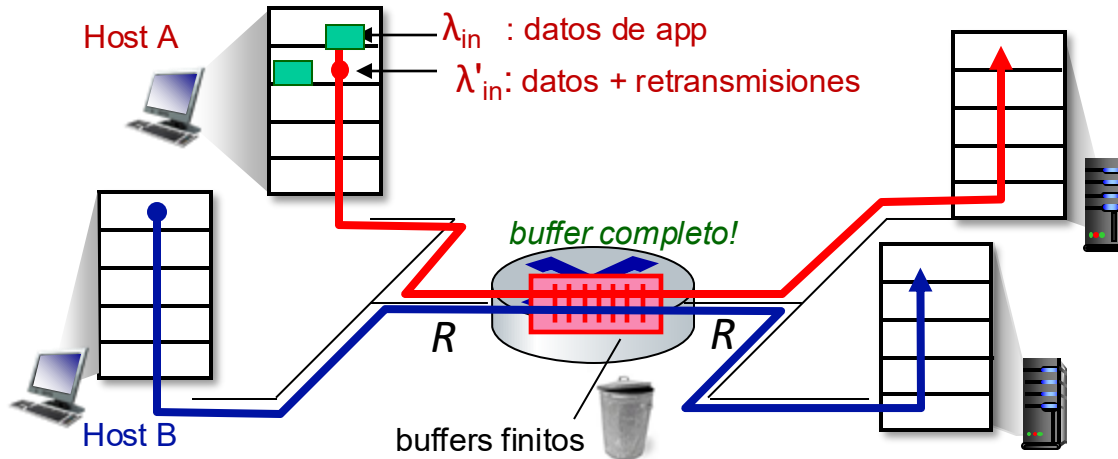




# Causas y costos de la congestión: caso 2

Suposición: *oráculo* de pérdidas

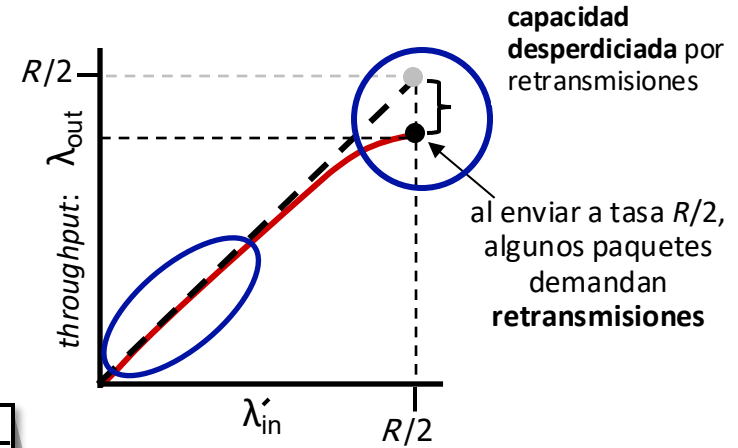
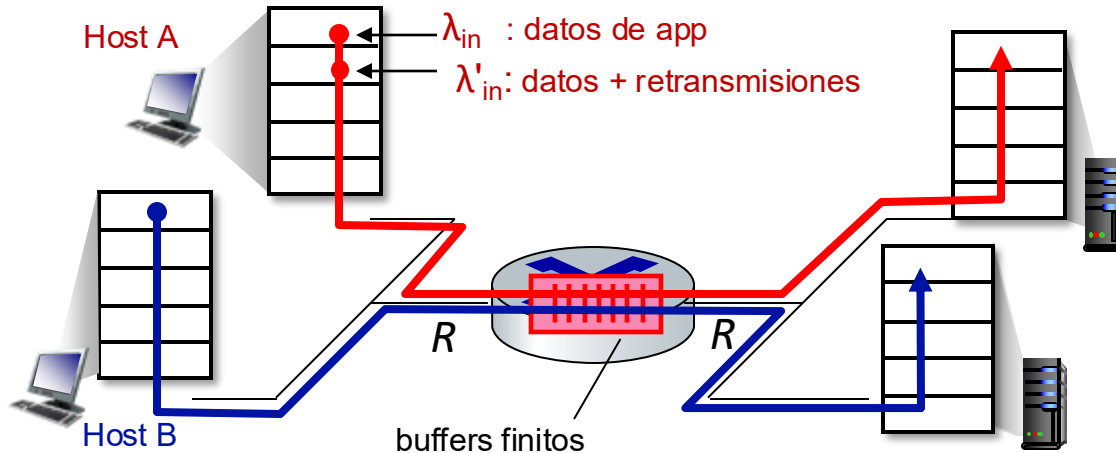
- Los paquetes pueden descartarse en el router por *overflow*
- El emisor puede saber cuándo se descartó un paquete: retransmite sólo en tal caso



# Causas y costos de la congestión: caso 2

Suposición: *oráculo* de pérdidas

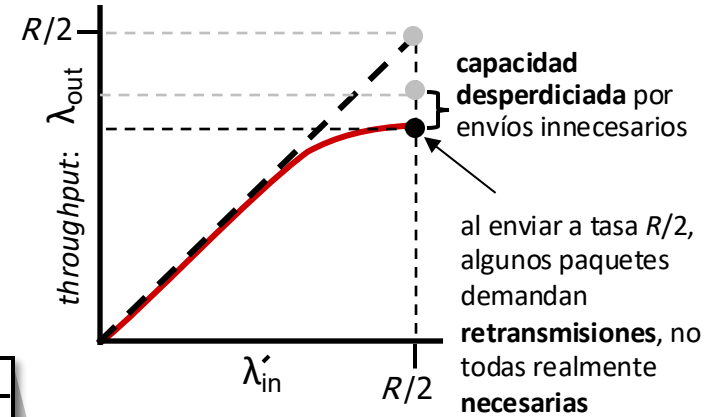
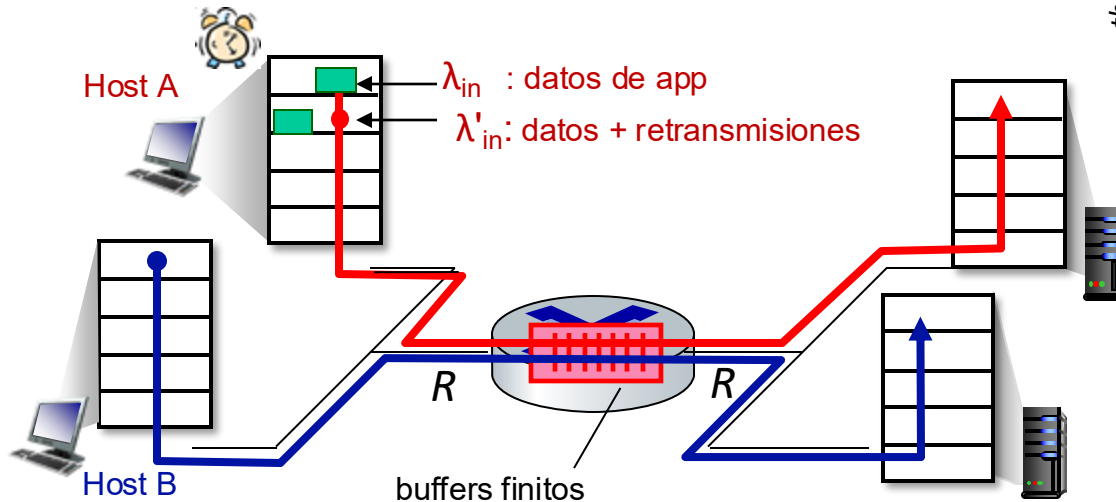
- Los paquetes pueden descartarse en el router por *overflow*
- El emisor puede saber cuándo se descartó un paquete: retransmite sólo en tal caso



# Causas y costos de la congestión: caso 2

Suposición: retransmisiones innecesarias

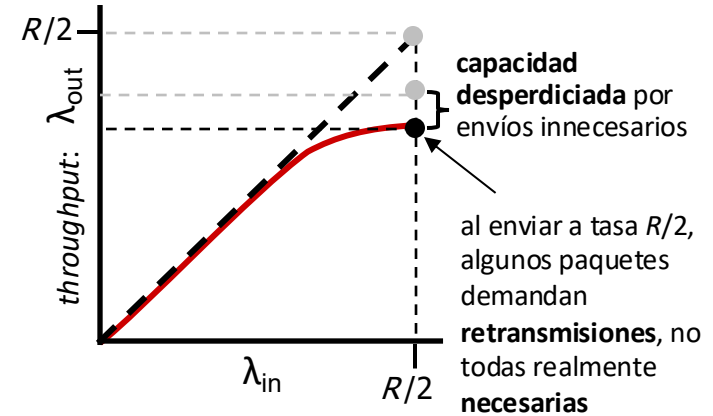
- Puede haber pérdida de paquetes por overflow en los routers y consecuentes retransmisiones
- Los *timers* del emisor pueden dispararse prematuramente, generando envíos innecesarios



# Causas y costos de la congestión: caso 2

Suposición: retransmisiones innecesarias

- Puede haber pérdida de paquetes por overflow en los routers y consecuentes retransmisiones
- Los *timers* del emisor pueden dispararse prematuramente, generando envíos innecesarios



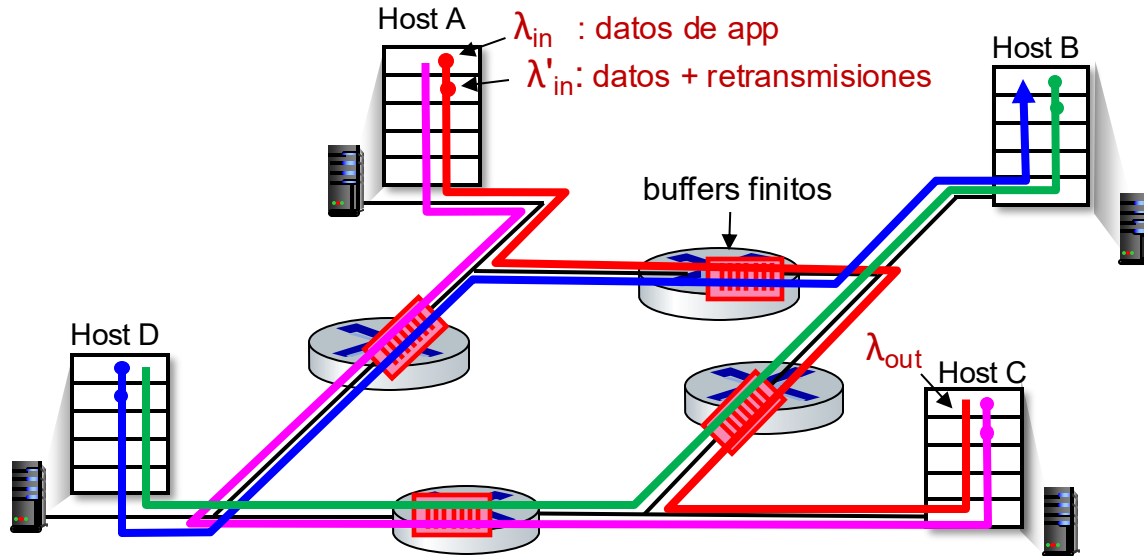
## Costos de la congestión

- Más trabajo requerido para un *throughput* dado (por **retransmisiones**)
- Retransmisiones **innecesarias**: el enlace transporta múltiples copias de los paquetes (degradando el *throughput*)

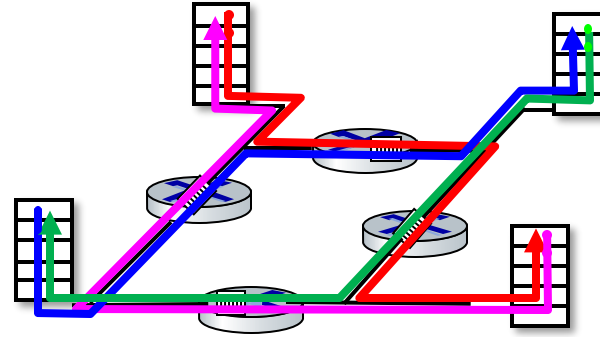
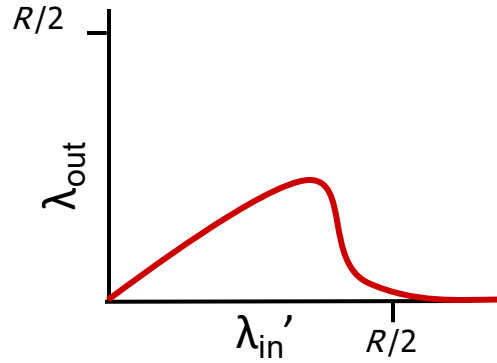
# Causas y costos de la congestión: caso 3

- Cuatro emisores
- Caminos *multi-hop*
- Timeouts y retransmisiones

Conforme aumenta  $\lambda_{in}'$ , los paquetes del flujo azul llegando al router de más arriba se descartarán:  
**throughput**  $\rightarrow 0$



# Causas y costos de la congestión: caso 3

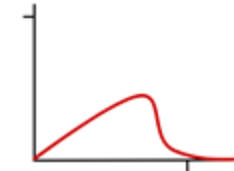
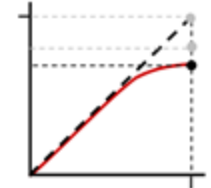
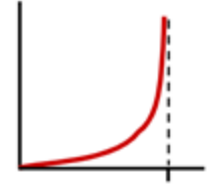


## Costo de la congestión

- Cuando se descarta un paquete, se **desperdician** los recursos de la red empleados previamente para transportar dicho paquete

# Causas y costos de la congestión

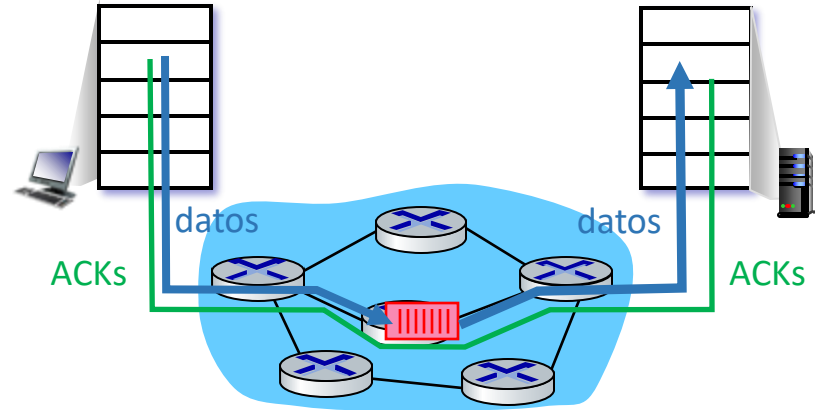
- El *throughput* no puede exceder la capacidad
- El delay aumenta a medida que nos aproximamos a la capacidad
- Las pérdidas/retransmisiones degradan el *throughput*
- Las retransmisiones innecesarias degradan el *throughput* aún más
- Se desperdician los recursos de la red utilizados para transportar paquetes eventualmente descartados



# Estrategias para el control de congestión

## Control de congestión *end-to-end*

- Sin retroalimentación explícita de la red
- La congestión se “infiera” del delay y/o las pérdidas observadas
- Enfoque adoptado por **TCP**





# Estrategias para el control de congestión

## Control de congestión asistido por la red

- Los routers proveen **feedback directo** a los hosts con flujos pasando a través de ellos
- Pueden indicar el nivel de congestión o bien definir explícitamente la tasa de envío
- Implementando e.g. en la extensión **ECN** (*Explicit Congestion Notification*) de TCP

