

1

La capa de Transporte.

Principios del control de congestión.

Tema 3: La capa de Transporte

2

- 3.1 La capa de transporte y sus servicios
- 3.2 Multiplexación y demultiplexación
- 3.3 Transporte sin conexión: UDP
- 3.4 Principios de transferencia de datos fiable
- 3.5 Transporte orientado a conexión: TCP
- 3.6 Principios del control de congestión
 - 3.6.1 Las causas y los costes de la congestión
 - 3.6.2 Métodos para controlar la congestión
- 3.7 Mecanismo de control de congestión de TCP

Principios de control de la congestión

3

- **Congestión de la red.**
 - Consecuencia del aumento del tráfico en la red.
 - Primero se manifiesta con el aumento del retardo.
 - Luego, cuando se hace más severa, desbordamiento de los *buffers* de los *routers*.
 - Descarte/pérdida de paquetes en los *routers* → TCP responde retransmitiendo.
 - Reducción del rendimiento ofrecido a las aplicaciones.
 - Retransmisión de paquetes como mecanismo de fiabilidad.
 - Se ocupa de un síntoma de la congestión, la pérdida de paquetes.
 - "A veces puede ser peor el remedio que la enfermedad".
 - No se ocupa de la causa de la congestión.
 - Muchos emisores generando tráfico a velocidades mayores que las recomendables en situaciones concretas.

Principios de control de la congestión

4

- **Retos ante la congestión de la red.**
 - Evitación de la congestión.
 - Procedimientos adecuados para evitar la congestión.
 - Solución de la congestión.
 - Reacción adecuada ante la congestión.
 - Procedimientos apropiados para regular el flujo de datos cuando aparezcan síntomas de congestión.
- **Control de la congestión: evitación y solución.**
 - Es uno de los problemas más importantes de las redes de computadores junto a la transferencia fiable.

Causas de la congestión

5

- Causas de la congestión.
 - Emisión de datos a velocidades inapropiadas.
 - Por situaciones circunstanciales o no.
 - Mecanismos de fiabilidad inapropiados.
 - Conducen a retransmisiones innecesarias.
 - Compartición de los enlaces de transmisión.
 - *Buffers* de capacidad limitada en los *routers*.
 - Rutas multi-saltos solapadas.
 - Diferentes rutas origen-destino comparten *routers* y enlaces de transmisión.
- A continuación:
 - Tres escenarios donde se produce congestión cuando el tráfico alcanza ciertos valores.
 - Complejidad creciente en cada escenario considerado.

Consecuencias de la congestión

6

□ Escenario 1

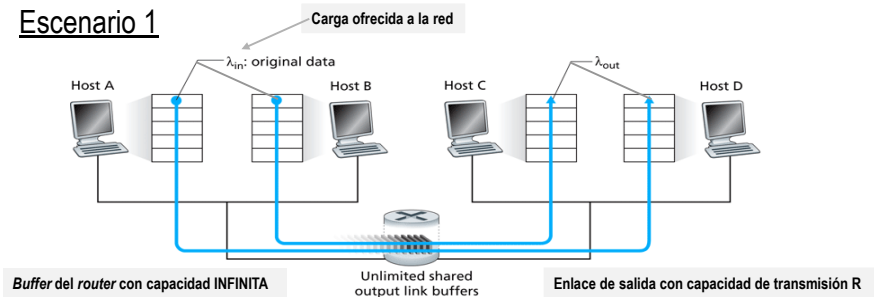


Figure 3.43 ♦ Congestion scenario 1: Two connections sharing a single hop with infinite buffers

- Sólo datos originales.
- Protocolo de transporte simple:
 - NO control de errores (NO retransmisiones), NO control de flujo, NO control de congestión.
- Se ignora sobrecarga de cabeceras $\rightarrow \lambda_{in}$ Bps = velocidad con que las aplicaciones entregan datos al *router*.
- Capacidad del enlace de salida del *router* compartida entre los *hosts* A y B.

Consecuencias de la congestión

7

Escenario 1

Rendimiento del Escenario 1

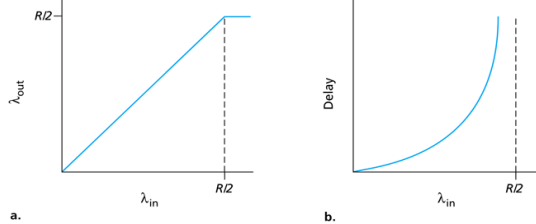


Figure 3.44 ♦ Congestion scenario 1: Throughput and delay as a function of host sending rate

Operar a una tasa de transferencia agregada $\cong R$ puede parecer ideal desde el punto de vista de λ_{out} , pero...



Grandes retardos de cola cuando la tasa de llegada de paquetes se aproxima a la capacidad del enlace R.

Tasa de transferencia por conexión

- $\lambda_{out} = f(\lambda_{in})$
- Si $\lambda_{in} < R/2 \rightarrow \lambda_{out} = \lambda_{in}$
 - Receptor recibe todo lo que envía el emisor
- Si $\lambda_{in} > R/2 \rightarrow \lambda_{out} = R/2$
 - Resultado de la comparación de la capacidad del enlace
- Hosts C y D nunca dispondrán de una $\lambda_{out} > R/2$

Retardo por conexión

- Retardo = $f(\lambda_{in})$
- $\lambda_{in} \rightarrow R/2 \rightarrow$ Retardo crece considerablemente
- Si $\lambda_{in} > R/2 \rightarrow$ Retardo infinito
 - Suponiendo buffers de capacidad INFINITA y transmitiendo durante un periodo de tiempo INFINITO
- Operar a una tasa agregada próxima a R \rightarrow Retardos INFINITOS

Consecuencias de la congestión

8

Escenario 2

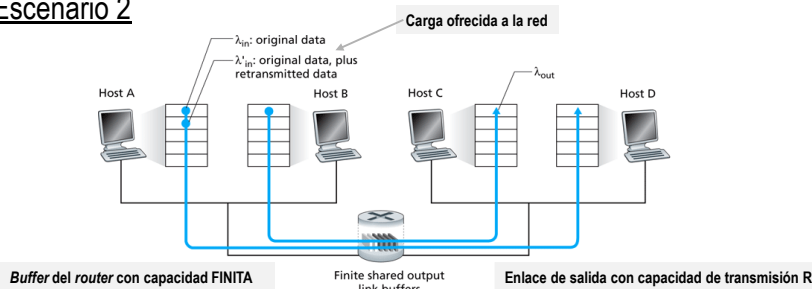


Figure 3.45 ♦ Scenario 2: Two hosts (with retransmissions) and a router with finite buffers

- Datos originales y retransmisiones.
- Buffer FINITO \rightarrow Descarte de paquetes.
- Protocolo de transporte FIABLE, gestión de errores con retransmisiones \rightarrow Rendimiento según cómo se hagan las retransmisiones
- λ_{in} = Velocidad a la que la aplicación envía datos originales al socket (en Bps).
- λ'_{in} = Velocidad a la que la capa de transporte envía segmentos a la red (originales y retransmitidos, en Bps).
- Capacidad del enlace de salida del router compartida entre los hosts A y B.

Consecuencias de la congestión

9

Escenario 2

Rendimiento del Escenario 2

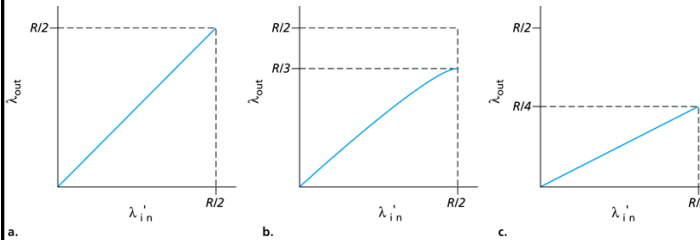


Figure 3.46 ♦ Scenario 2 performance with finite buffers

Caso a), NO realista

- Host A conoce el estado del *buffer* (lleno o vacío). → Envía datos SÓLO cuando el *buffer* está vacío.
- No se producen pérdidas de paquetes (no retransmisión).
- $\lambda_{in} = \lambda'_{in} = \lambda_{out}$ (tasa de transferencia) → Rendimiento IDEAL (nada se pierde)
- $\lambda_{in} \leq R/2$, ya que NO se pierden paquetes.



Caso b), algo más realista que el caso a)

- Emisor **retransmite** SÓLO cuando tiene CERTEZA que un paquete se ha perdido (p.e, *Timeout* muy grande)
- Si $\lambda'_{in} = R/2 \rightarrow \lambda_{out} = R/3 \rightarrow$ De $R/2$ unidades de datos enviadas, en media:
 - 0,333 R son datos originales ($\cong 66,6\%$)
 - 0,166 R son datos retransmitidos ($\cong 33,3\%$)

Caso c), más realista aún que el caso b)

- Si emisor retransmite cuando el *timeout* expira y el paquete NO se ha perdido, sino sólo demorado → **Retransmisión PREMATURA**
- Paquete original y duplicado llegan al receptor → **Descarte del duplicado**
 - Se desaprovechada la capacidad de transmisión del *router*.
 - Si el *router* envía, en media, cada paquete dos veces:
 - $\lambda_{out} \rightarrow R/4$ cuando $\lambda_{in} \rightarrow R/2$

Tema 3: La capa de transporte

Consecuencias de la congestión

10

Escenario 3

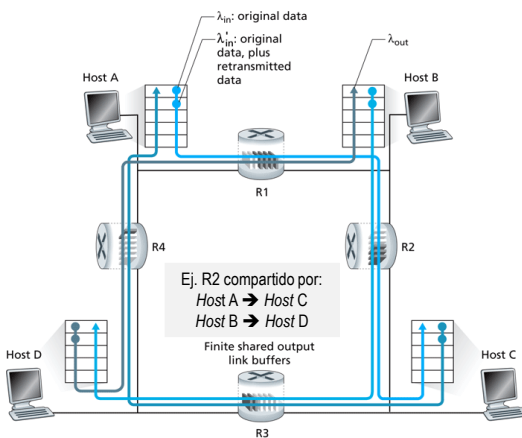


Figure 3.47 ♦ Four senders, routers with finite buffers, and multihop paths



Rutas multi saltos solapadas: diferentes rutas origen - destino que comparten routers y enlaces de transmisión.

- Cada *host* emite datos originales y retransmisiones.
 - λ_{in} = Velocidad a la que la aplicación envía datos originales al *socket* (en Bps).
 - λ'_{in} = Velocidad a la que la capa de transporte envía segmentos a la red (originales y retransmitidos, en Bps).
- Protocolo de transporte FIABLE → Gestión de errores con *timeout* y retransmisiones.
 - Rendimiento según las retransmisiones.
- Routers con *buffers* FINITOS.
 - Descarte de paquetes al llenarse los *buffers*.
- Capacidad compartida de los enlaces: R Bps
- Rutas multi saltos solapadas, dos saltos por ruta:
 - Host A → Host C, a través de R1 y R2
 - Host C → Host A, a través de R3 y R4
 - Host B → Host D, a través de R2 y R3
 - Host D → Host B, a través de R4 y R1

Tema 3: La capa de transporte

Consecuencias de la congestión

11

Escenario 3

Resultado de las rutas solapadas multisaltos

Para valores muy pequeños de λ_{in} :

- Los buffers no se desbordan $\rightarrow \lambda'_{in} = \lambda_{in}$
- Tasa de transferencia (λ_{out}) \equiv Carga ofrecida (λ_{in})
- Incremento de $\lambda_{in} \rightarrow$ Incremento de λ_{out}

Para valores muy grandes de λ_{in} (y por tanto también de λ'_{in}):

- Reducción drástica de la tasa de transferencia (λ_{out})
- Descarte de paquetes "avanzados" en la ruta \rightarrow Desaprovechamiento de los recursos.

Ejemplo para el caso del router R2 y las conexiones A-C y B-D ($\lambda_{A-C}, \lambda_{B-D}$):

- En R2 la λ_{A-C} está limitada a R debido al paso previo por R1.
- Si $\lambda_{B-D} \gg \lambda_{A-C} \rightarrow$ La carga de tráfico A-C que logre rebasar R2 será cada vez menor a medida que aumente la carga de tráfico B-D, ya que λ_{B-D} es más inmediata a R2 que λ_{A-C}
 $\rightarrow \lambda_{A-C}$ a la salida de R2 tiende a cero $\rightarrow \lambda_{A-C}$ terminal a terminal tiende a cero.

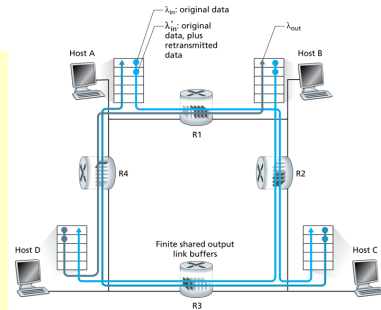


Figure 3.47 • Four senders, routers with finite buffers, and multihop paths

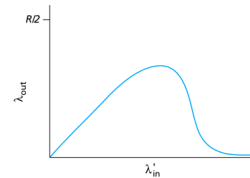


Figure 3.48 • Scenario 3 performance with finite buffers and multihop paths

Consecuencias de la congestión: resumen

12

- Costes debidos a la congestión de la red.
 - Grandes retardos de cola.
 - Cuando la tasa de llegada de paquetes se aproxima a la capacidad del enlace (caso 1).
 - Retransmisiones.
 - Emisor tiene que retransmitir para compensar la pérdida de paquetes por buffers desbordados (casos 2 y 3).
 - Desperdicio de recursos (ancho de banda y capacidad en los buffers)
 - Debido a retransmisiones innecesarias por grandes retardos (casos 2 y 3), reenvío de copias innecesarias.
 - Debido fundamentalmente al descarte de paquetes "avanzados" en una ruta (caso 3).
- La congestión de la red \rightarrow Reducción del rendimiento que perciben las aplicaciones.

Métodos de control de congestión

13

- En general las técnicas para control de congestión se pueden clasificar atendiendo a dos casos:
 - La capa de red NO interviene.
 - Control de congestión “terminal a terminal”.
 - La capa de red SÍ interviene.
 - Control de congestión “asistido por la red”.
- Control de congestión “terminal a terminal”.
 - La capa de red no provee soporte explícito a la capa de transporte(p.e, caso de TCP).
 - La congestión tiene que ser inferida por los hosts a partir sólo del comportamiento observado de la red, p.e:
 - Retardos crecientes.
 - Pérdida de segmentos TCP por vencimiento del *timeout* o tres ACKs duplicados.

Métodos de control de congestión

14

- Control de congestión “asistido por la red”
 - Los *routers* (capa de red) proporcionan notificación explícita al emisor, indicando el estado de congestión de la red.
 - Mediante algún procedimiento de realimentación al emisor.
 - Se utiliza, p.e:
 - En redes ATM, en el mecanismo de control de congestión ABR (*Available Bit-Rate*).
 - Propuesto también para redes TCP/IP.
 - Bits ECN (*Explicit Congestion Notification*) de la cabecera IP (en la práctica no utilizado).
 - Mecanismo de señalización a nivel IP para indicar congestión en la red.
 - Posibilita notificación de la congestión *end-to-end* sin descarte de paquetes.
 - Conlleva a “marcar” paquetes en vez de descartarlos cuando se presenta congestión.

Métodos de control de congestión

15

- Control de congestión “asistido por la red”.

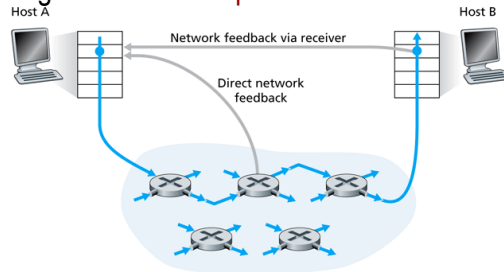


Figure 3.49 ♦ Two feedback pathways for network-induced congestion information

Dos posibles formas de notificación explícita de la congestión (ECN) al emisor

- Notificación directa al emisor desde un *router* del trayecto, o
- Notificación indirecta al emisor desde un *router* del trayecto (a través del receptor).
 - ▣ *Router* marca/actualiza un campo de cabecera del paquete de datos que va de emisor a receptor.
 - ▣ Receptor notifica al emisor → Tiempo de notificación > RTT entre los sistemas finales.