

*Reti di calcolatori e Internet:
Un approccio top-down*

7^a edizione
Jim Kurose, Keith Ross

Pearson Paravia Bruno Mondadori Spa

Capitolo 3: Livello di trasporto

- ❑ 3.1 Servizi a livello di trasporto
- ❑ 3.2 Multiplexing e demultiplexing
- ❑ 3.3 Trasporto senza connessione: UDP
- ❑ 3.4 Principi del trasferimento dati affidabile
- ❑ 3.5 Trasporto orientato alla connessione: TCP
 - struttura dei segmenti
 - trasferimento dati affidabile
 - controllo di flusso
 - gestione della connessione
- ❑ 3.6 Principi del controllo di congestione
- ❑ 3.7 Controllo di congestione TCP

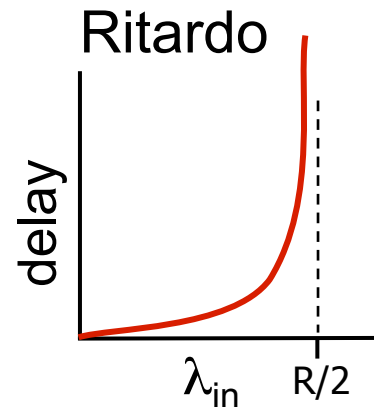
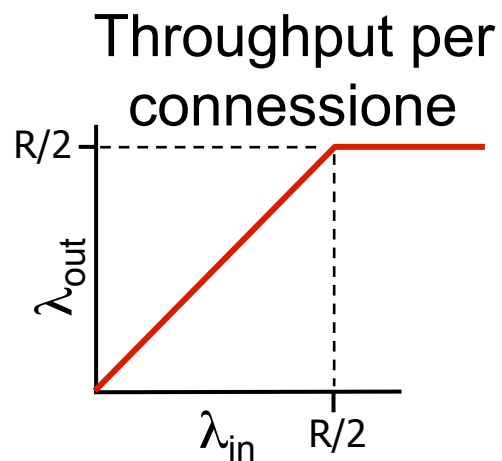
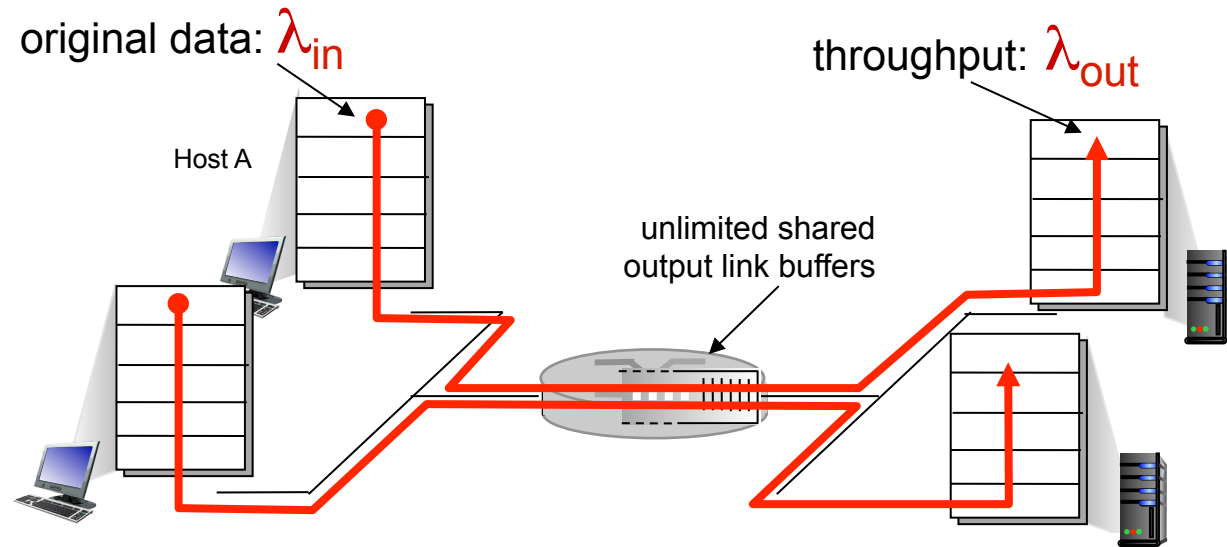
Principi del controllo di congestione

Congestione:

- ❑ informalmente: "troppe sorgenti trasmettono troppi dati, a una velocità talmente elevata che la *rete* non è in grado di gestirli"
- ❑ differisce dal controllo di flusso!
- ❑ sintomi:
 - pacchetti smarriti (overflow nei buffer dei router)
 - lunghi ritardi (accodamento nei buffer dei router)

Cause/costi della congestione: scenario 1

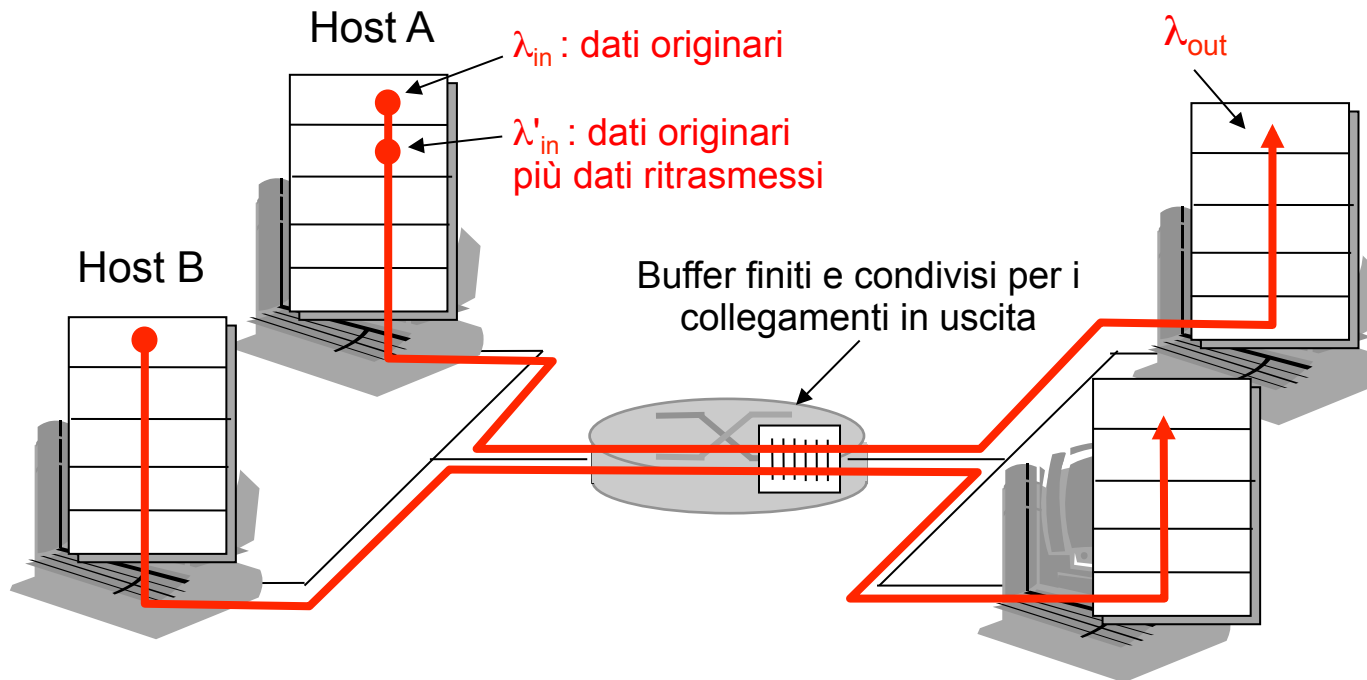
- ❑ due mittenti, due destinatari
- ❑ un router con buffer illimitati
- ❑ nessuna ritrasmissione



- ❑ grandi ritardi se congestionati
- ❑ throughput massimo

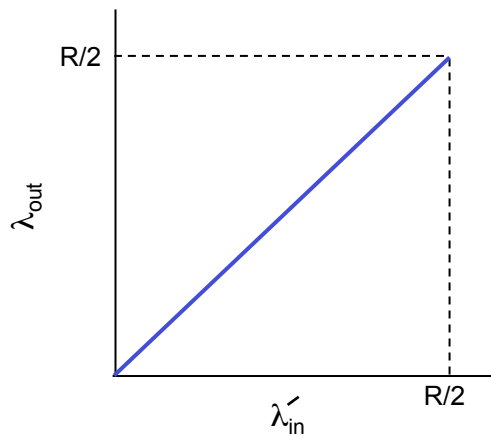
Cause/costi della congestione: scenario 2

- ❑ un router, buffer *finiti*
- ❑ il mittente ritrasmette il pacchetto perduto
 - application-layer input = application-layer output: $\lambda_{in} = \lambda_{out}$
 - transport-layer input include le *ritrasmissioni*: $\lambda'_{in} \geq \lambda_{in}$

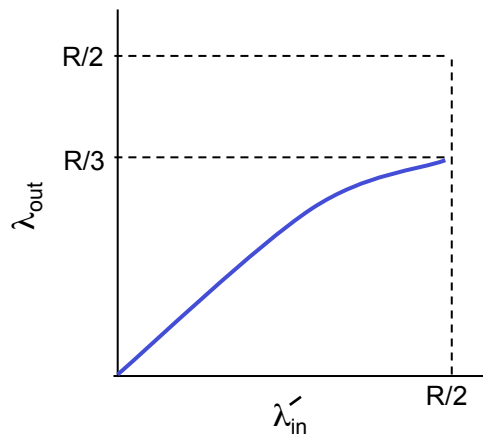


Cause/costi della congestione: scenario 2

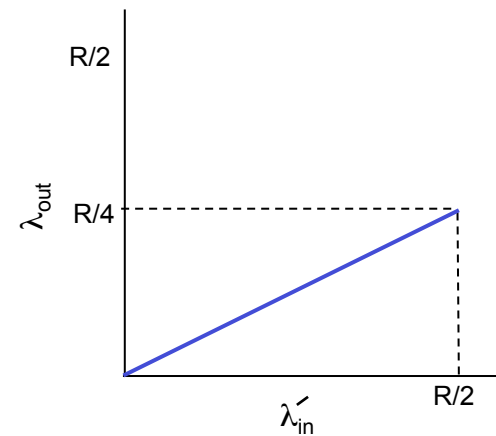
- λ_{out} è il goodput
- Ritrasmissione "perfetta" solo quando la perdita è certa: $\lambda'_{in} > \lambda_{out}$
- La ritrasmissione del pacchetto ritardato (non perduto) rende λ'_{in} più grande (rispetto al caso perfetto) per lo stesso λ_{out}



a) l'host trasmette solo quando è certo che nel buffer ci sia spazio ("magicamente")



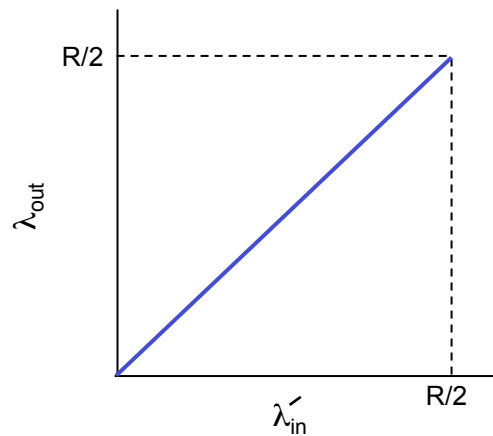
b) l'host ritrasmette solo i dati realmente perduti ("magicamente" o timeout molto grande)



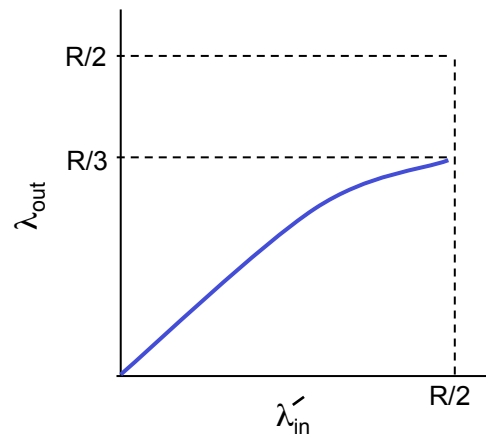
c) l'host ritrasmette anche pacchetti in ritardo

Cause/costi della congestione: scenario 2

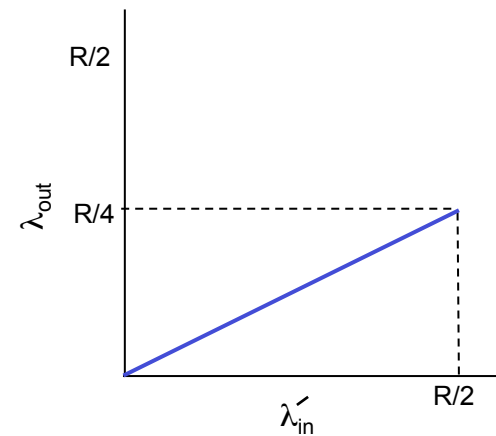
- λ_{out} è il goodput
- Ritrasmissione "perfetta" solo quando la perdita è certa: $\lambda'_{in} > \lambda_{out}$
- La ritrasmissione del pacchetto ritardato (non perduto) rende λ'_{in} più grande (rispetto al caso perfetto) per lo stesso λ_{out}



a.



b.



c.

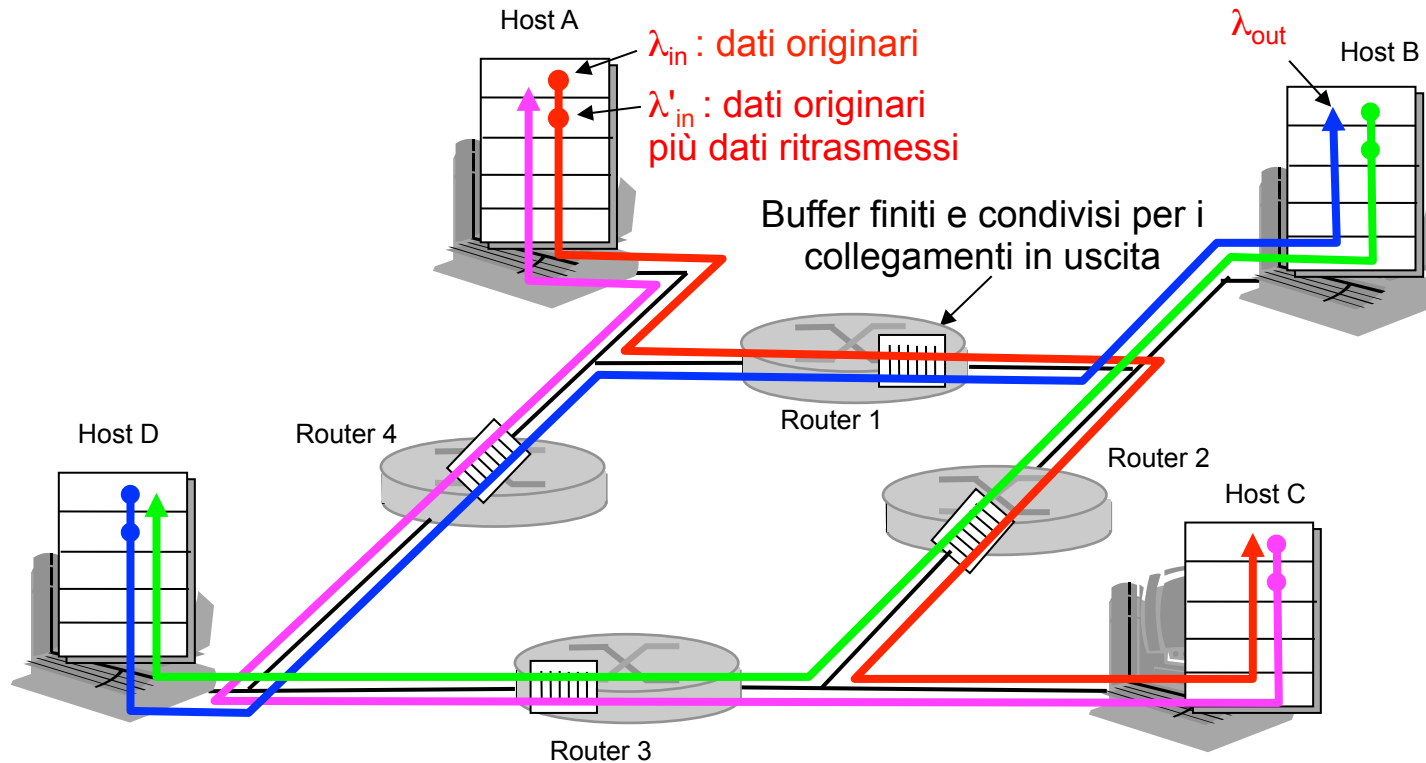
"Costi" della congestione:

- Più lavoro (ritrasmissioni) per un dato "goodput"
- Ritrasmissioni non necessarie: il collegamento trasporta più copie del pacchetto

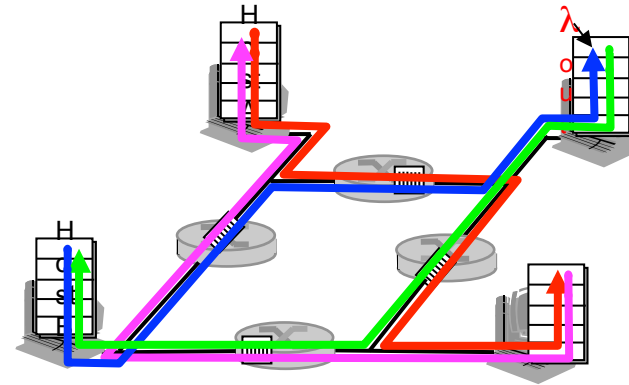
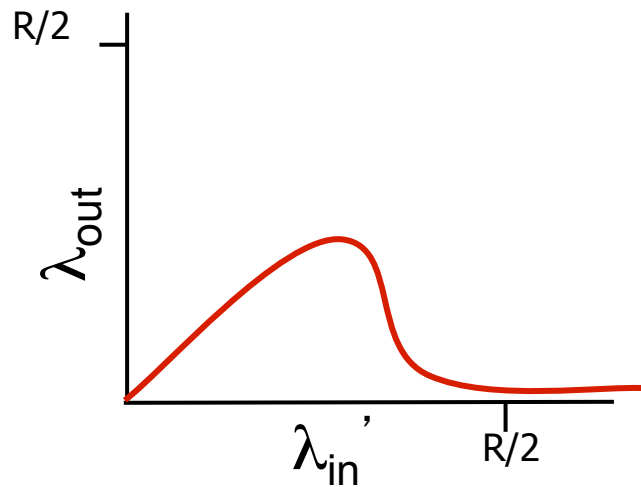
Cause/costi della congestione: scenario 3

- ❑ Quattro mittenti
- ❑ Percorsi multihop
- ❑ timeout/ritrasmissione

D: Che cosa accade quando λ_{in} e λ'_{in} aumentano?



Cause/costi della congestione: scenario 3



Un altro "costo" della congestione:

- Quando il pacchetto viene scartato, la capacità trasmissiva utilizzata sui collegamenti di upstream per instradare il pacchetto risulta sprecata!

Approcci al controllo della congestione

I due principali approcci al controllo della congestione:

Controllo di congestione punto-punto:

- ❑ nessun supporto esplicito dalla rete
- ❑ la congestione è dedotta osservando le perdite e i ritardi nei sistemi terminali
- ❑ metodo adottato da TCP

Controllo di congestione assistito dalla rete:

- ❑ i router forniscono un feedback ai sistemi terminali
 - un singolo bit per indicare la congestione
 - comunicare in modo esplicito al mittente la frequenza trasmissiva

Capitolo 3: Livello di trasporto

- ❑ 3.1 Servizi a livello di trasporto
- ❑ 3.2 Multiplexing e demultiplexing
- ❑ 3.3 Trasporto senza connessione: UDP
- ❑ 3.4 Principi del trasferimento dati affidabile
- ❑ 3.5 Trasporto orientato alla connessione: TCP
 - struttura dei segmenti
 - trasferimento dati affidabile
 - controllo di flusso
 - gestione della connessione
- ❑ 3.6 Principi del controllo di congestione
- ❑ 3.7 Controllo di congestione TCP

Controllo di congestione TCP

- ❑ Il mittente limita la trasmissione tramite una "finestra di congestione" (CongWin)
- ❑ Approssimativamente:

$$\text{Frequenza d'invio} = \frac{\text{CongWin}}{\text{RTT}} \text{ byte/sec}$$

- ❑ CongWin è una funzione dinamica della congestione percepita

In che modo il mittente percepisce la congestione?

- ❑ Evento di perdita = timeout o ricezione di 3 ACK duplicati
- ❑ Il mittente TCP riduce la frequenza d'invio (CongWin) dopo un evento di perdita

tre meccanismi:

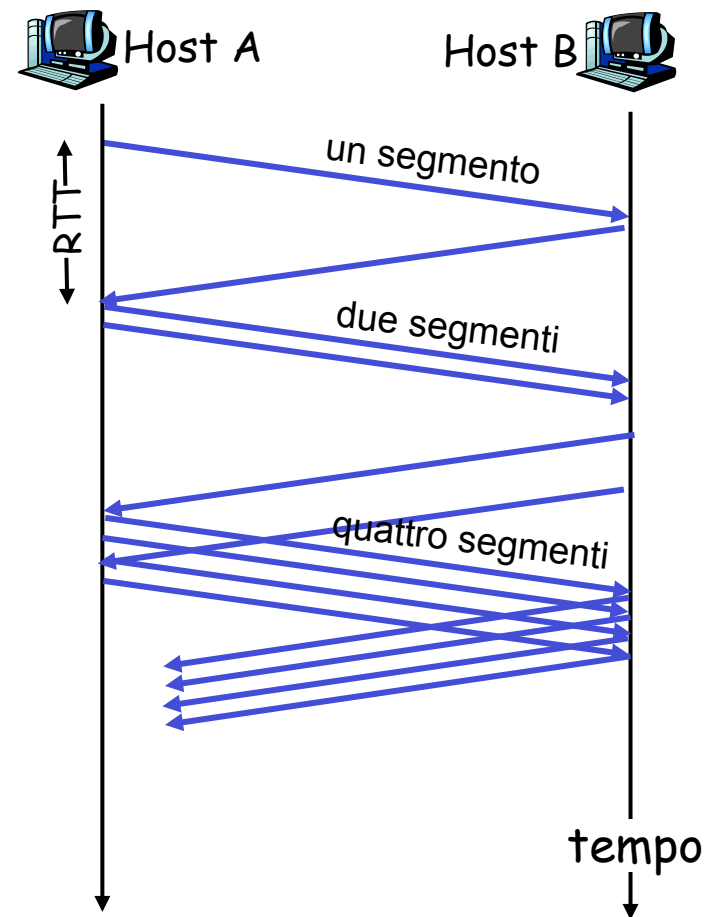
- Partenza lenta
- AIMD (*additive-increase multiplicative-decrease*)
- Reazione agli eventi di perdita

Partenza lenta

- ❑ Quando si stabilisce una connessione,
 $\text{CongWin} = 1 \text{ MSS}$
 - Esempio: $\text{MSS} = 500 \text{ byte}$
 $\text{RTT} = 200 \text{ msec}$
 - Frequenza iniziale = 20 kbps
- ❑ La larghezza di banda disponibile potrebbe essere $\gg \text{MSS}/\text{RTT}$
 - Consente di raggiungere rapidamente una frequenza d'invio significativa
- ❑ Quando inizia la connessione, la frequenza aumenta in modo esponenziale, fino a quando non si verifica un evento di perdita
 - ❑ CongWin aumenta di 1MSS alla ricezione di ogni Ack
 - ❑ In un RTT la dimensione di CongWin raddoppia

Partenza lenta (continua)

- ❑ Quando inizia la connessione, la frequenza aumenta in modo esponenziale, fino a quando non si verifica un evento di perdita:
 - raddoppia CongWin a ogni RTT
 - ciò avviene incrementando CongWin per ogni ACK ricevuto
- ❑ **Riassunto:** la frequenza iniziale è lenta, ma poi cresce in modo esponenziale

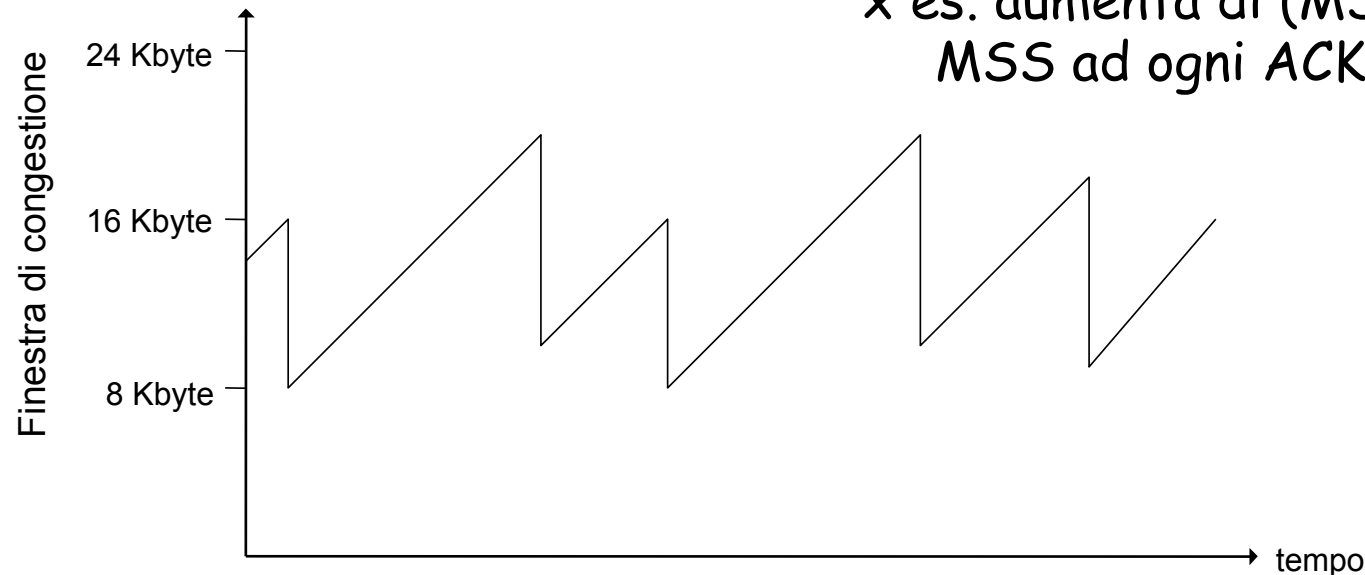


Controllo di congestione TCP: incremento additivo e decremento moltiplicativo (AIMD)

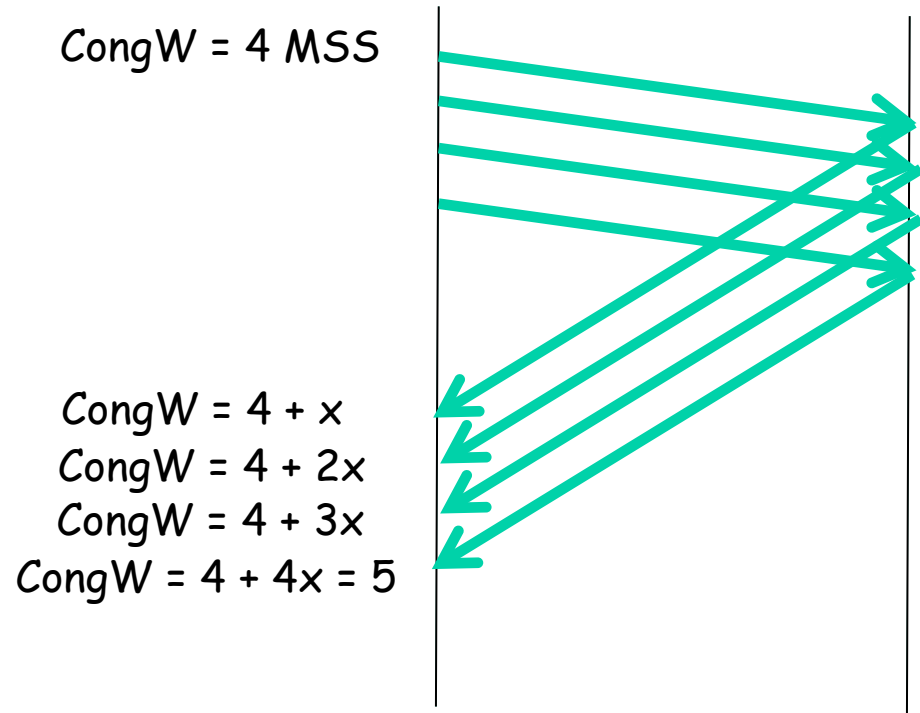
Approccio: aumenta il tasso trasmissivo sondando la rete, fino a quando non si verifica una perdita

Decremento moltiplicativo:
riduce a metà CongWin dopo un evento di perdita

Incremento additivo: aumenta CongWin di 1 MSS a ogni RTT in assenza di eventi di perdita: *sondaggio*
x es. aumenta di $(MSS/CongWin)$ MSS ad ogni ACK



Controllo di congestione AIMD



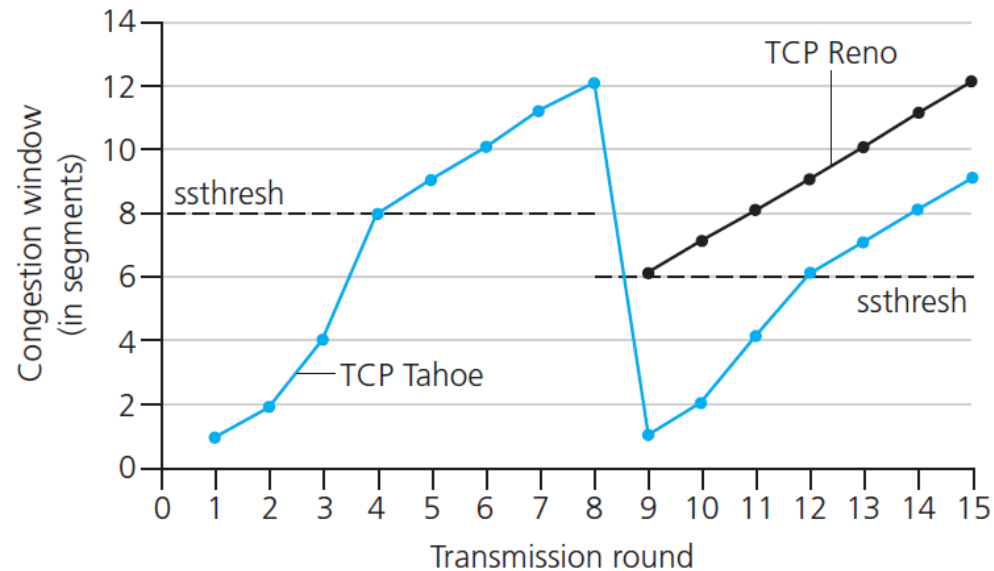
TCP: reazione ad un evento di perdita

- ❑ Perdita indicata da timeout:
 - **CongWin** impostata ad 1 MSS;
 - Crescita esponenziale (come nella fase di slow start) fino alla soglia, poi passaggio a AIMD
- ❑ Perdita indicata da 3 ACK duplicati (TCP RENO) => Fast Recovery
 - **CongWin** viene dimezzata e si prosegue con AIMD
 - (TCP Tahoe) invece imposta sempre **CongWin** a 1

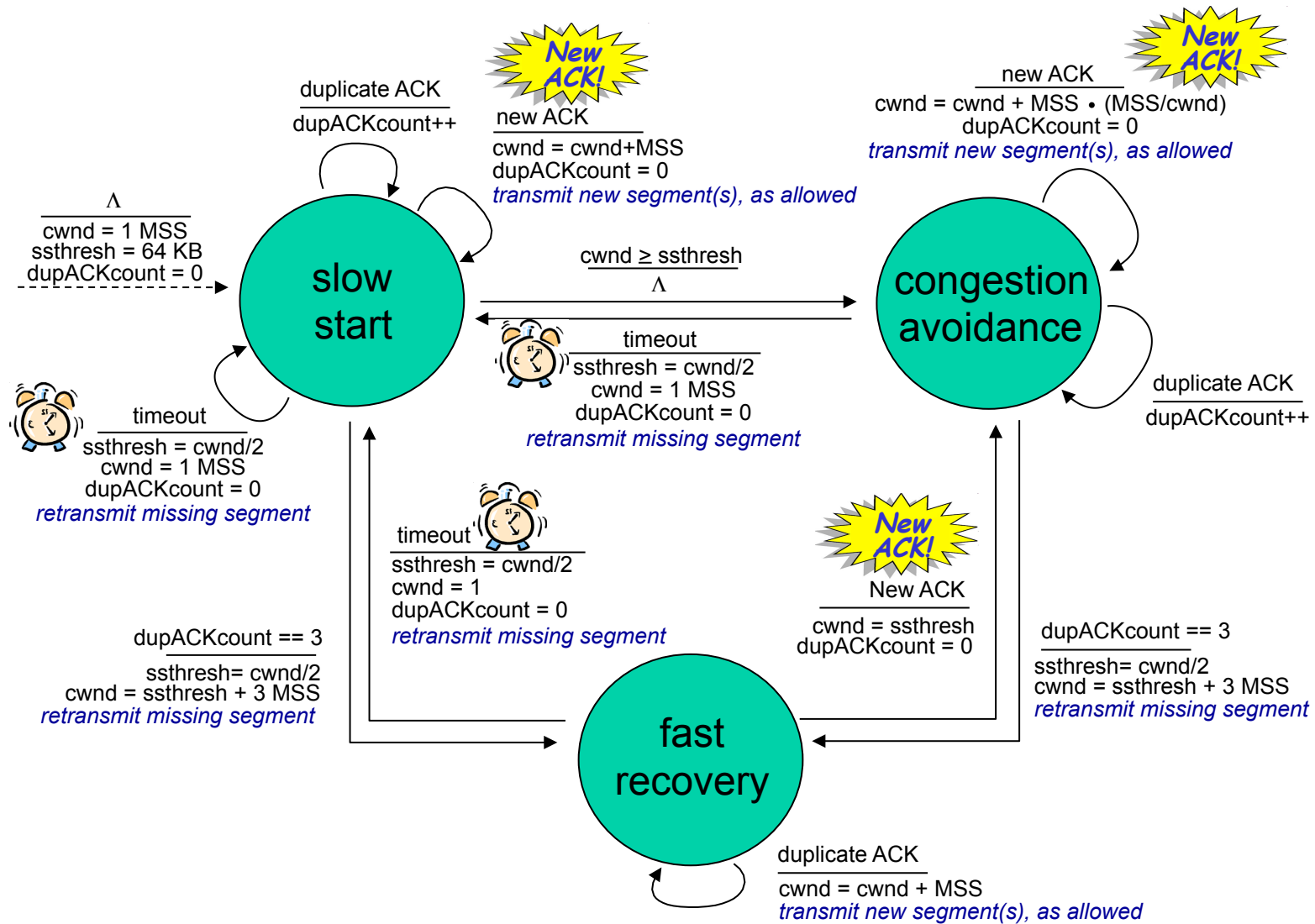
TCP: passaggio tra slow start e congestion avoidance

Valore Soglia:

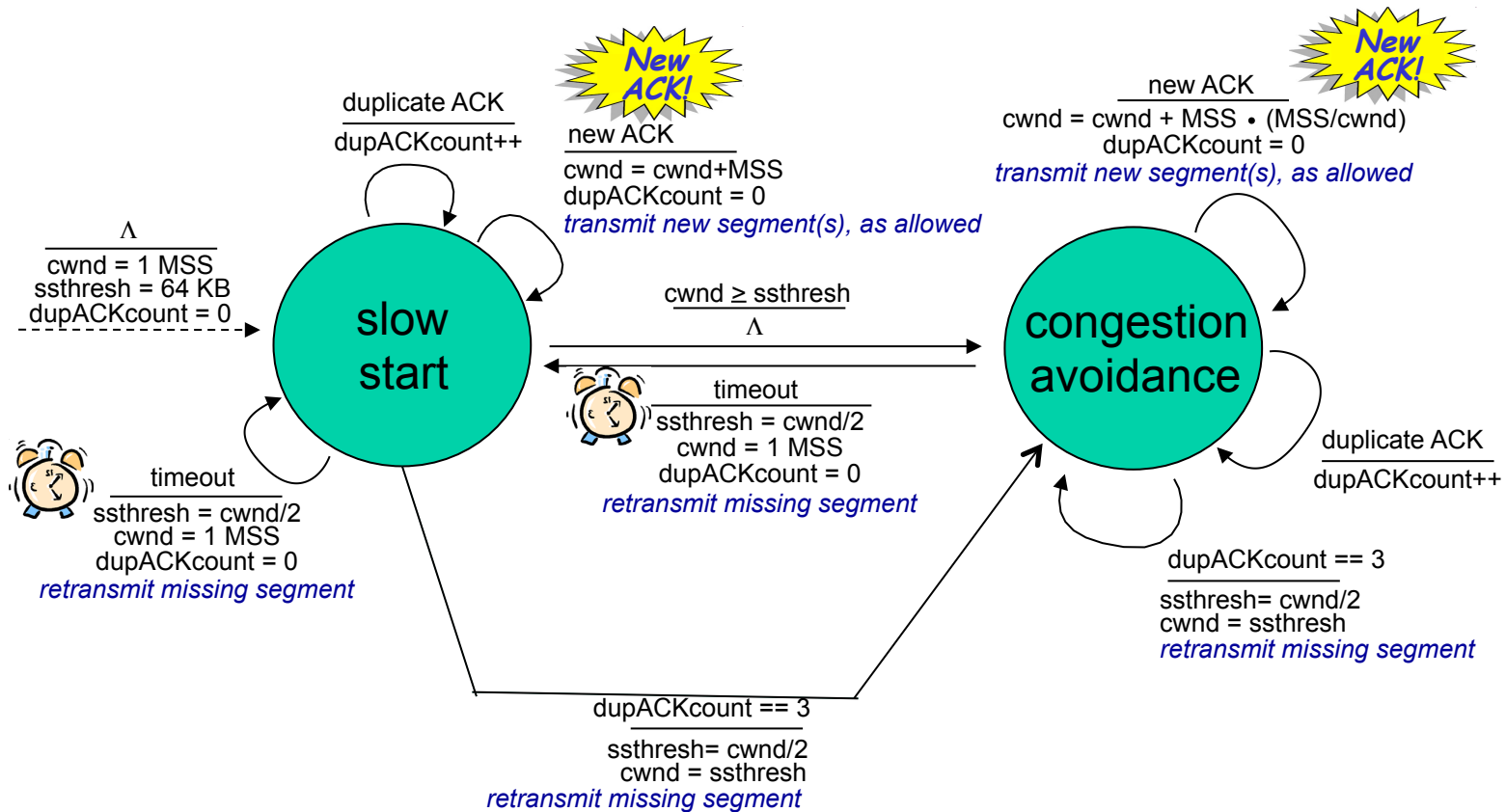
- ❑ **Soglia** variabile
- ❑ Quando si rileva una perdita, la soglia si imposta a $\frac{1}{2}$ **CongWin**



Riepilogo: Controllo di Congestione del TCP



Riepilogo: Controllo di Congestione del TCP



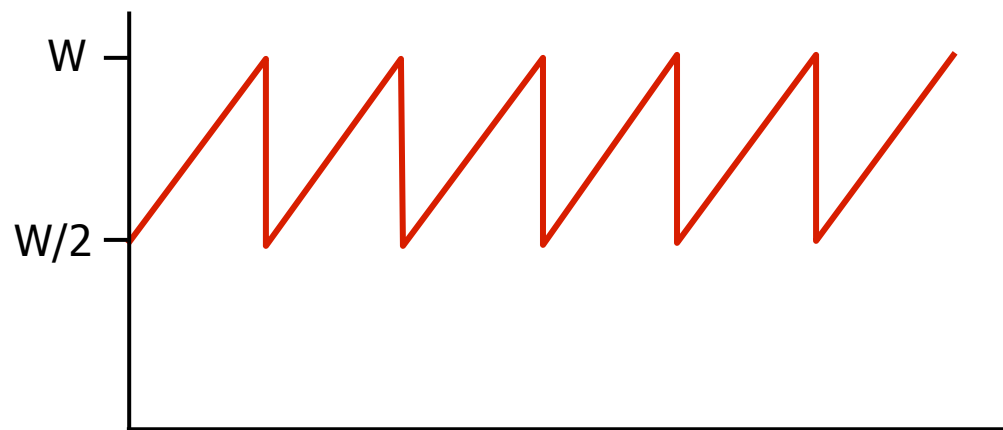
Versione semplificata senza il Fast Recovery

Riassunto: il controllo della congestione TCP

- ❑ Quando CongWin è sotto la soglia (Threshold), il mittente è nella fase di **partenza lenta**; la finestra cresce in modo esponenziale.
- ❑ Quando CongWin è sopra la soglia, il mittente è nella fase di **congestion avoidance**; la finestra cresce in modo lineare.
- ❑ Quando si verificano **tre ACK duplicati**, il valore di Threshold e CongWin vengono impostati a CongWin/2
- ❑ Quando si verifica un **timeout**, il valore di Threshold viene impostato a CongWin/2 e CongWin è impostata a 1 MSS.

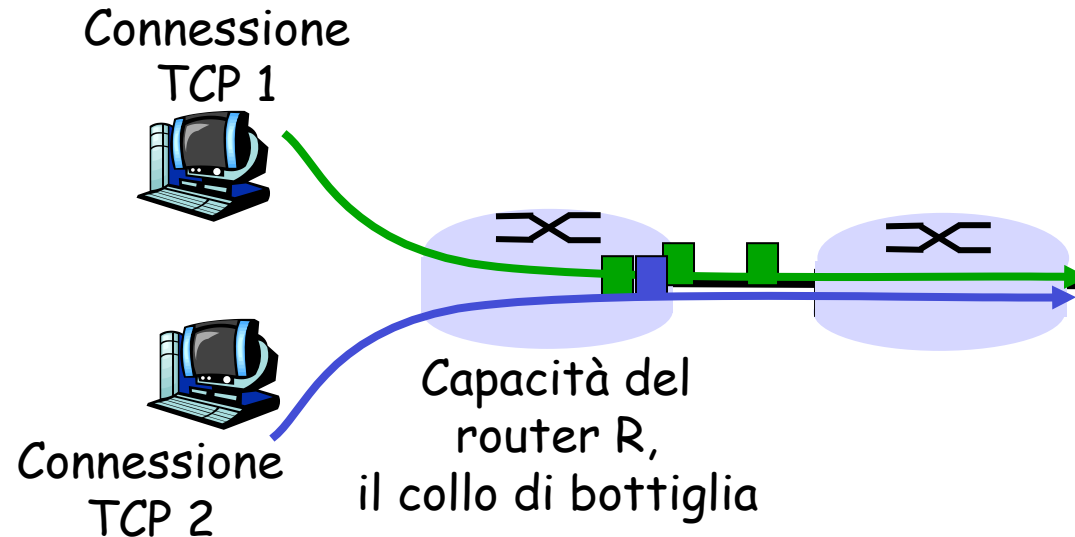
Throughput TCP

- ❑ Qual è il throughput medio di TCP in funzione della dimensione della finestra e di RTT?
 - Ignoriamo le fasi di partenza lenta
- ❑ Sia W la dimensione della finestra quando si verifica una perdita.
- ❑ Quando la finestra è W , il throughput è W/RTT
- ❑ Subito dopo la perdita, la finestra si riduce a $W/2$, il throughput a $W/2RTT$.
- ❑ Throughput medio: $0,75 W/RTT$



Equità di TCP

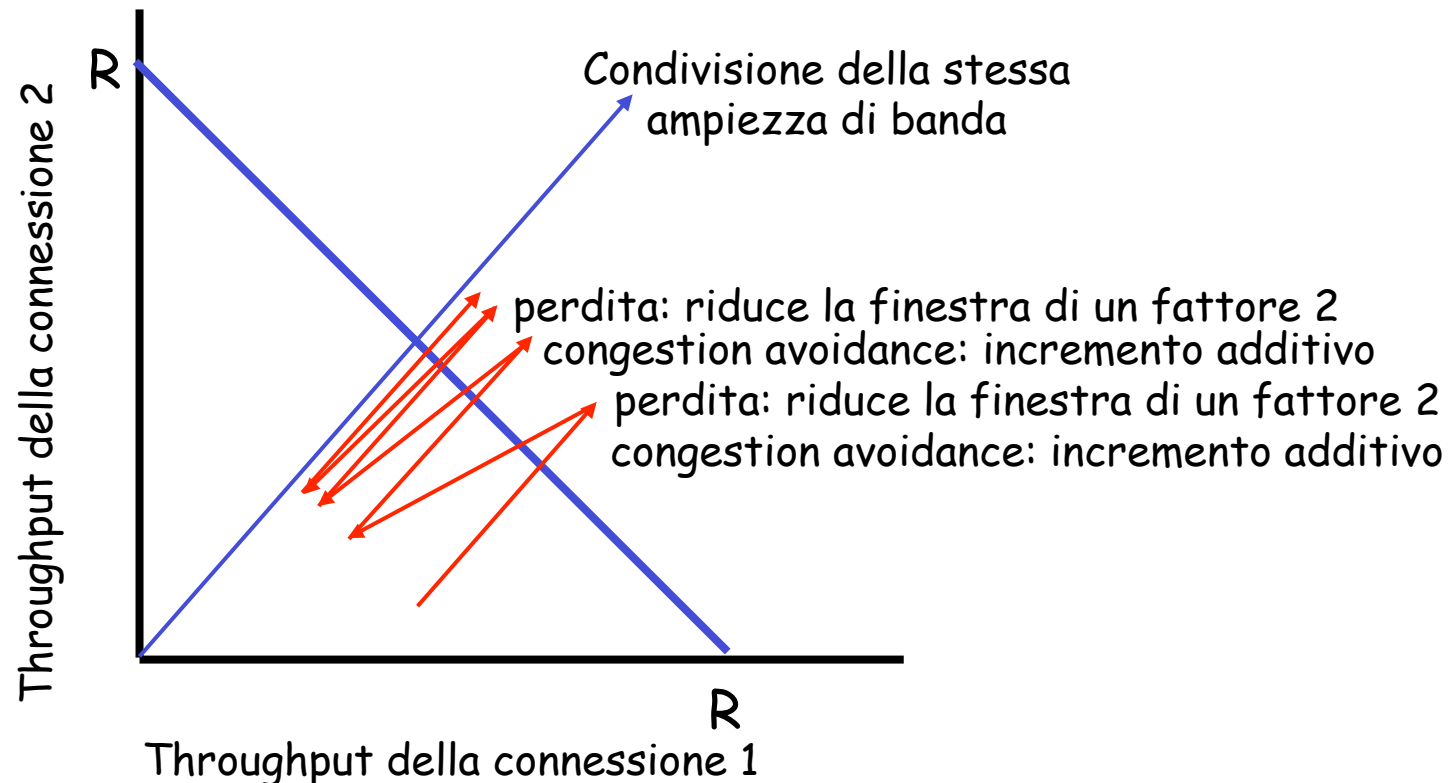
Equità: se K sessioni TCP condividono lo stesso collegamento con ampiezza di banda R , che è un collo di bottiglia per il sistema, ogni sessione dovrà avere una frequenza trasmissiva media pari a R/K .



Perché TCP è equo?

Due connessioni in concorrenza tra loro:

- ❑ L'incremento additivo determina una pendenza pari a 1, all'aumentare del throughput (ci si muove in avanti su una retta a 45°)
- ❑ Il decremento moltiplicativo riduce il throughput in modo proporzionale (ci si muove all'indietro sulla congiungente con l'origine)



Equità

Equità e UDP

- ❑ Le applicazioni multimediali spesso non usano TCP
 - non vogliono che il loro tasso trasmissivo venga ridotto dal controllo di congestione
- ❑ Utilizzano UDP:
 - immettono audio/video a frequenza costante, tollerano la perdita di pacchetti
- ❑ Area di ricerca: evitare che il traffico UDP estrometta il traffico TCP

Equità e connessioni TCP in parallelo

- ❑ Nulla può impedire a un'applicazione di aprire connessioni in parallelo tra 2 host
- ❑ I browser web lo fanno
- ❑ Esempio: un collegamento di frequenza R che supporta 9 connessioni;
 - Se una nuova applicazione chiede una connessione TCP, ottiene una frequenza trasmissiva pari a $R/10$
 - Se la nuova applicazione chiede 11 connessioni TCP, ottiene una frequenza trasmissiva pari a $R/2$!

Explicit Congestion Notification (ECN)

Controllo della congestione assistito dalla rete:

- Due bit nell'header IP (campo Tipo di Servizio) impostati dai *router* per indicare la congestione
- Il destinatario viene informato dai router della presenza di congestione
- Il destinatario imposta il bit ECE (echo) in un segmento di ack verso il mittente
- Il mittente dimezza la finestra di congestione

