

Controllo della congestione

AIMD

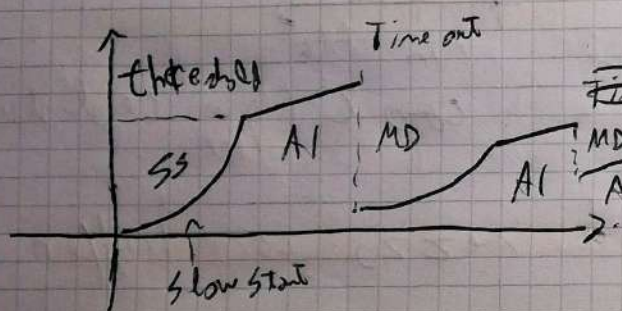
Incremento additivo - Decremento moltiplicativo

Non b SR e va bene, non b SR ...

Se a un certo punto c'è timeout faccio decremento di molti K in modo improvviso

In realtà nella prima fase cerchiamo di andare più velocemente (a velocità quadratica) per poi procedere a velocità lineare.

In caso di collo (grasso) ritorno a velocità quadratica per poi passare a lineare dopo un po' (In realtà la crescita è esponenziale, non quadratica)



3 acks

← Qui non siamo un collo stretto, in caso di 3 ack duplicati fast retransmit (nel Reno)

TCP-Tahoe

- Slow Start
- Congestion avoidance
- Fast retransmit algorithm

In caso di timeout
riferito a 1 (cwnd)
 $threshold = cwnd / 2$

In realtà la crescita non è lineare, è esponenziale curva

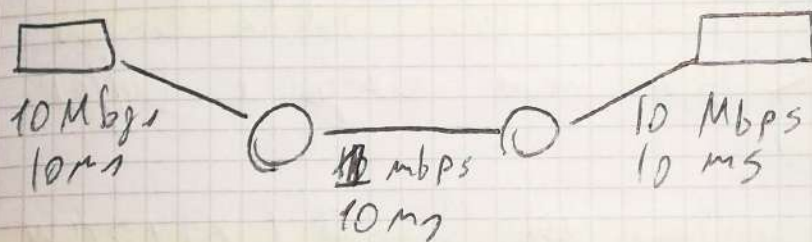
Tahoe riparte sempre da 1 (in timeout da 3 ACK)
Non riparte da 1 in timeout, la fast recovery in 3 ack

TCP-Reno

- Slow-start
- Congestion avoidance
- Fast retransmit
- Fast recovery

In caso di timeout. Riparto dalla soglia ~~recovery~~ calcolata ($threshold = cwnd / 2$)
 $cwnd = 1$
 $threshold = cwnd / 2$
se 3 ACK

Controllo flusso e congestione



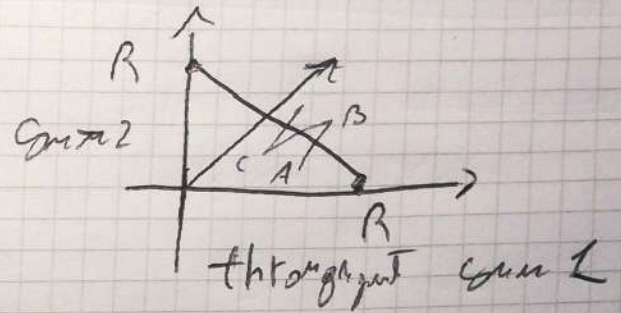
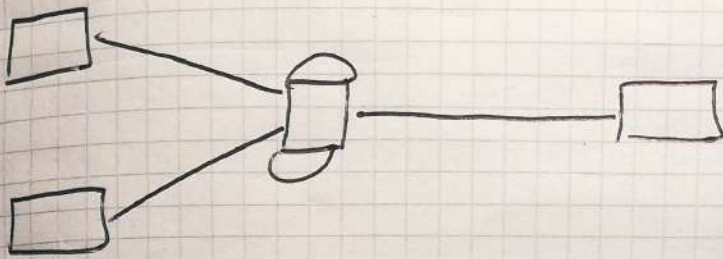
① Canale senza perdite
Ritardo costante
Grafici in stile
prestazioni una simile

② Canale con perdite ma senza perdite
Grafici
Prestazioni di Reno leggermente superiori

③ Canale con perdite, lunghezza di buffer variabile e
ritardo variabile
Grafici ~~simulazioni~~
Prestazioni di Reno leggermente superiori

④ 3 flussi concorrenti
Canale con perdite, bu variabile, ritardo variabile
Reno prende il sopravvento

Le connessioni TCP hanno la proprietà Fairness
ovvero condivisione della rete / banda



La retta \nearrow ha $x=y$ che è
il nostro obiettivo

Desidero voremo stare in $(R/2, R/2)$, ma è stato
impossibile.

Concludo di stare sulla bisettrice

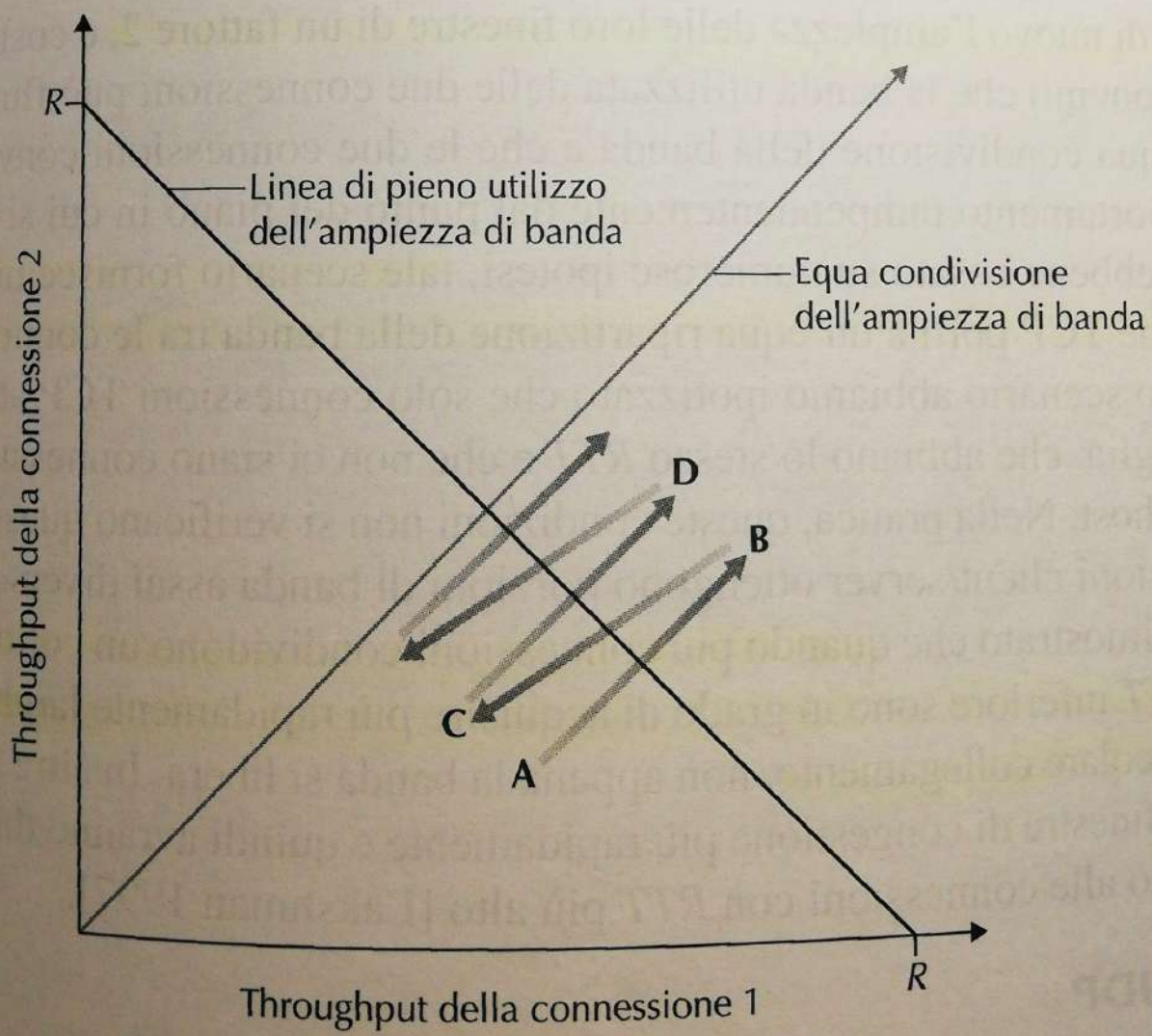


Figura 3.55 Throughput delle connessioni TCP 1 e 2.

C comincia alla bissettice con un movimento a zigzag
selezionato.

Da A partono a B facendo una sola linea
(fino che $x+y < BW$ non riempie i buffer ed
entro i time out)

Arrivano in B e avviene un timeout ($x+y > BW$)

~~si~~ da $x, y \rightarrow \frac{x}{2}, \frac{y}{2}$ e arrivano da C.

Ricomincia il ciclo e cominciamo regole di più
alla bissettice.

E' difficile prevedere la fairness nel caso ci siano
sul canale una TCP "greedy" (Reno) e connection
"elusive". Reno prenderebbe tutto.