

Controllo di congestione(TCP)

Si chiama congestione il fenomeno che si verifica quando le sorgenti tentano di utilizzare più banda di quella disponibile. Provoca lunghi ritardi e perdita di pacchetti.

TCP regola la velocità di trasmissione per adattarsi a quella della rete.

Finestra di congestione

La larghezza della finestra di invio è limitata da una finestra di congestione (cwnd), ovvero $swnd = \min(rwnd, cwnd)$.

La dimensione si misura in MSS (maximum segment size), ovvero la quantità massima di dati trasmissibili in un segmento, determinato sottraendo al MTU (maximum trasmission unit, dimensione link frame) la dimensione di header TCP e IP.

Controllo punto-punto

Modalità:

slow start si inizializza $cwnd = 1$ e si incrementa di 1 MSS per ogni ACK *non duplicato* — raddoppia ogni RTT

congestion avoidance / AIMD ad ogni riscontro non duplicato $cwnd = cwnd + \frac{1}{RTT}$, in modo da aumentare la finestra di 1 MSS per ogni RTT (*additive increment*, lineare), mentre in caso di perdita si dimezza la finestra (*multiplicative decrement*). Le implementazioni effettive non dimezzano immediatamente cwnd, scegliendo un'altra strategia sul breve termine.

Reno

La modalità è stabilita in base a una variabile *ssthresh*, inizializzata a 64 KB:

- se $cwnd < ssthresh$ incremento esponenziale (slow start),
- altrimenti incremento lineare.

In congestion avoidance, in caso di perdita si distinguono:

- 3 ACK duplicati: $ssthresh = cwnd/2$, $cwnd = ssthresh + 3$ e si va in *fast recovery* — probabile perdita di un singolo pacchetto;
- timeout: $ssthresh = cwnd/2$, $cwnd = 1$, slow start — situazione più problematica.

In slow start vengono entrambi gestiti come il timeout. In ogni caso, cambiando modalità si azzerava il contatore id ACK duplicati.

Fast recovery

- si incrementa cwnd di 1 per ogni ACK duplicato ricevuto (esponenziale);
- non appena si riceve un nuovo ACK (ovvero il destinatario ha ricevuto e riscontrato la ritrasmissione del segmento perso), $cwnd = ssthresh$ e si torna in congestion avoidance;
- se scade un RTO si va in slow start, dimezzando di nuovo cwnd.

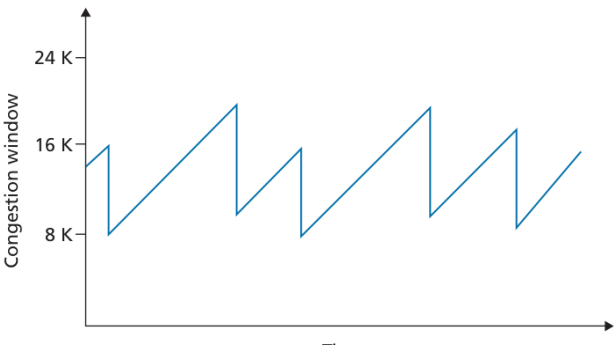
La strategia di inizializzazione e incremento di cwnd fa in modo che si continuino ad inviare altri segmenti se il destinatario li sta ricevendo. Quando il segmento originariamente perso viene riscontrato si riduce di nuovo la finestra per evitare il sovraccarico della rete.

Tahoe

Implementazione precedente a Reno, non prevede la modalità fast recovery e gestisce allo stesso modo timeout e 3 ACK duplicati.

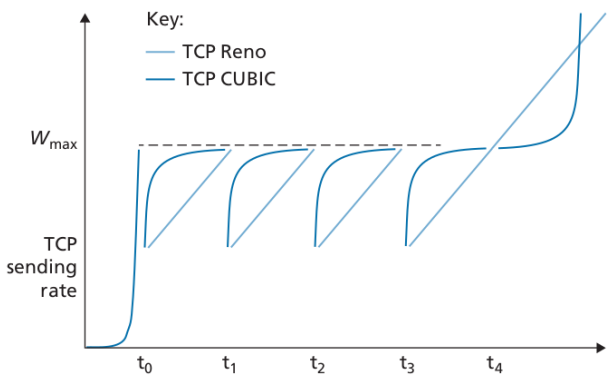
Cubic

Con gli algoritmi classici la cwnd ha un andamento a dente di sega:



Visto che la dimensione che provoca congestione (W_{max}) tende a rimanere costante per periodi di tempo apprezzabili, si può migliorare il throughput incrementando cwnd più velocemente quando è lontano da W_{max} .

TCP Cubic sostituisce AIMD con un incremento cubico rispetto alla distanza da W_{max} :



Explicit congestion notification (ECN)

- i router possono indicare il verificarsi di congestione tramite una configurazione dei due bit *type of service* (ToS) negli header IP;
- l'host che riceve l'ECN informa l'altro capo della comunicazione tramite il bit ECE (ECN Echo) nell'header TCP;
- questo reagisce come se fosse stato perso un pacchetto, e lo segnala impostando il bit CWR (Congestion Window Reduced) nell'header TCP.