

4.2.3.8 DIVISIONE EQUA RISORSE

Si può dimostrare che in condizioni ideali il TCP è in grado di limitare la congestione e distribuire in modo equo la capacità del link.

Le condizioni ideali sono alterate da diversi RTT per diversi flussi e da buffer nei nodi minori del prodotto banda-ritardo.

4.2.3.9 FAST RETRANSMIT/RECOVERY

Il TCP, mantenendo i pacchetti fuori sequenza, è in grado di richiedere un pacchetto mancante per evitare lo scadere del timeout. Il meccanismo viene chiamato dagli ACK duplicati: per ogni pacchetto fuori sequenza viene inviato un ACK con AV pari al pacchetto mancante. Gli algoritmi di fast retransmit e recovery rinviavano i pacchetti mancanti.

Il meccanismo di fast * non riduce la CWND, in quanto la congestione non è severa. Questo meccanismo funziona solo se non ci sono perdite multiple nella stessa finestra.

Funzionamento:

- 1) al 2° ACK duplicato si pone $SSTHRESH = \max\left(\frac{FLIGHT-SIZE}{2}, 2 MSS\right)$
- 2) viene ritrasmesso il pacchetto perso.
- 3) si pone $CWND = SSTHRESH + 3 MSS$
- 4) per ogni ACK duplicato la CWND viene incrementata di 1
- 5) vengono trasmessi nuovi segmenti se CWND e RCVWND lo permettono
- 6) appena arriva un ACK che riscontra i nuovi dati si esce dalla fase di recovery e si pone $CWND = SSTHRESH = \max\left(\frac{FLIGHT-SIZE}{2}, 2 MSS\right)$

La versione che implementa i meccanismi sopra è la TCP RENO (la prima è TCP Tahoe)