

IL TCP DI BERKELEY

La prima proposta di protocollo **TCP** che affrontava il controllo della congestione fu fatta a **Berkeley** nel 1986 introducendo gli algoritmi **Slow Start** e **Congestion Avoidance** e applicandoli congiuntamente.

In questa proposta il protocollo **TCP** non analizza la rete cercando di capirne lo stato ma agisce sul ritmo dell'invio dei dati quando questo raggiunge un particolare valore, la **Slow Start Thershold (SSTHRESH)**, viene posta inizialmente a 64KB.

La finestra di congestione **CWND** viene determinata nel seguente modo: viene posta inizialmente pari alla dimensione del segmento massimo usato sulla connessione **MSS** e viene successivamente aggiornata con due diverse modalità:

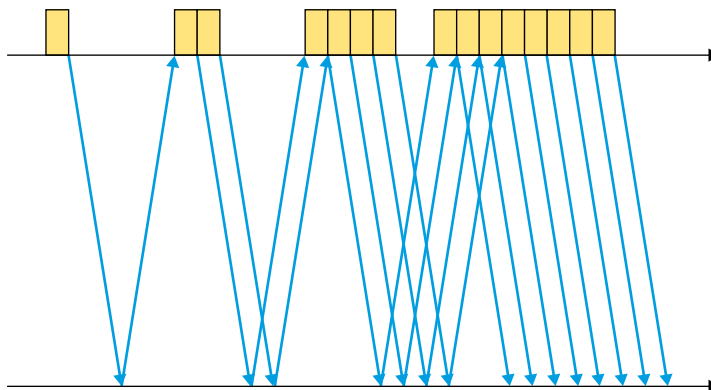
- se siamo nella situazione in cui **CWND < SSTHRESH** si utilizza l'algoritmo **Slow Start**;
- se siamo nella situazione in cui **CWND > SSTHRESH** si utilizza l'algoritmo **Congestion Avoidance**.

Descriviamo il funzionamento, a partire dall'accensione.

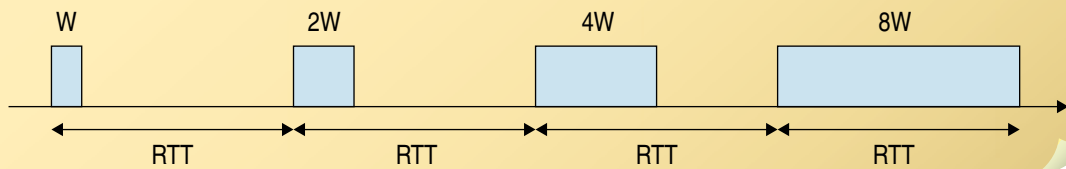
Slow Start

All'inizio il trasmettitore pone la **CWND** a 1 segmento **MSS** e la **SSTHRESH** a un valore di default molto elevato, generalmente 64KB: essendo **CWND < SSTHRESH** siamo nella situazione di **Slow Start** e si procede raddoppiando la **CWND** per ogni **ACK** ricevuto.

Si invia quindi un segmento e, se dopo **RTT** si riceve l'**ACK**, si pone **CWND** a 2 e si inviano 2 segmenti: se si ricevono 2 **ACK**, si pone **CWND** a 4 e si inviano 4 segmenti... e via di seguito fino a raggiungere il valore di **SSTHRESH** oppure fino a che si verifica una congestione.



Osserviamo che il nome **Slow Start** trae in inganno in quanto l'incremento della finestra avviene in modo esponenziale (raddoppia ogni **RTT**) che è quindi sicuramente non lento!



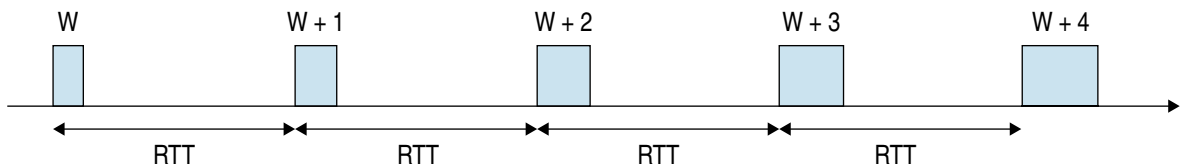
Con l'aumentare della finestra di trasmissione aumenta il **rate** di trasmissione che può essere stimato come:

$$R = \frac{CWND}{RTT} \text{ [bit/s]}$$

dove la **CWND** è espressa in bit e il **RTT** in secondi, sempre nella ipotesi che **RCVWND** > **CWND**.

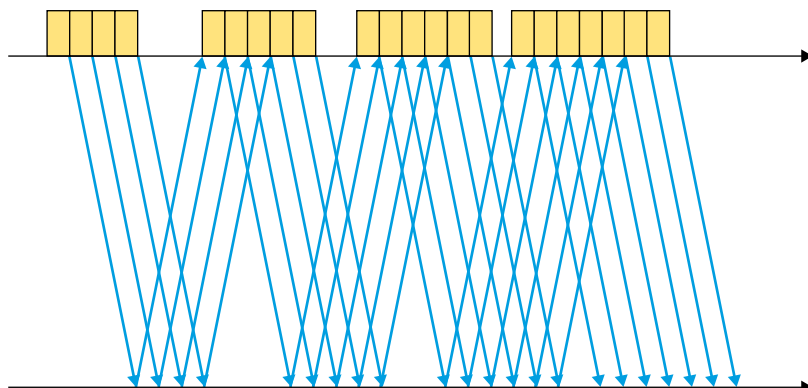
Congestion Avoidance

Abbiamo detto che lo **slow start** continua fino a che **CWND** supera la dimensione del **SSTHRESH**: oltre quel valore inizia la fase di **Congestion Avoidance** nella quale si incrementa la **CWND** di 1 **MSS** ogni **RTT** a ogni **ACK** ricevuto.



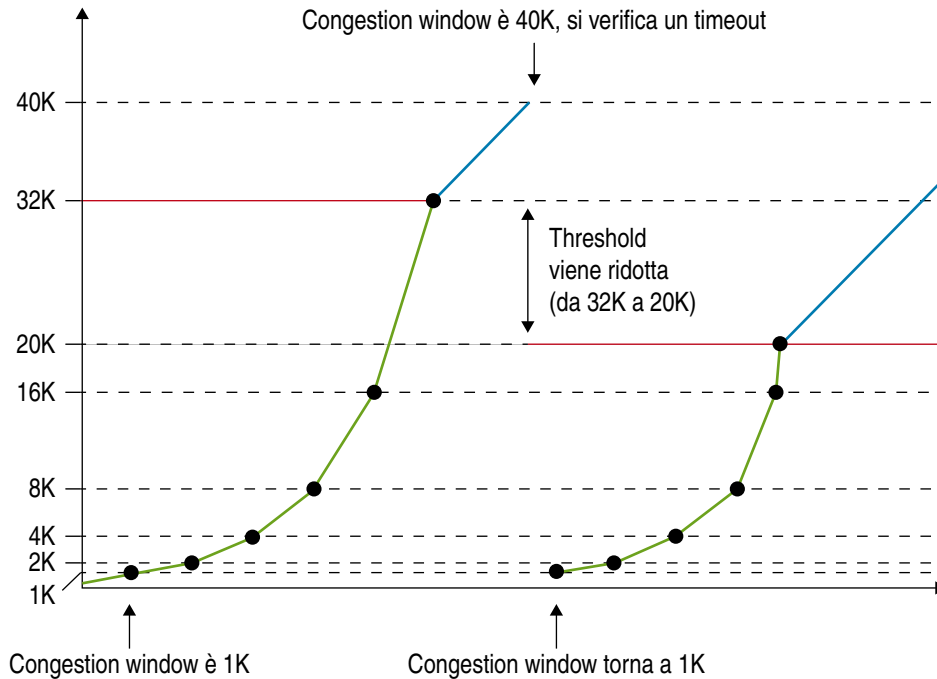
Si passa da un incremento esponenziale a un incremento lineare: dopo aver raggiunto **SSTHRESH** la finestra continua ad aumentare molto più lentamente che nella fase precedente.

In pratica se la **CWND** consente di trasmettere N segmenti, la ricezione degli **ACK** relativi a tutti gli N segmenti porta la **CWND** ad aumentare di 1 segmento.



Se si verifica la perdita di un segmento si ripete l'algoritmo da capo, dopo aver impostato il valore di **threshold** alla metà della dimensione della **congestion window**.

Nella seguente tabella riportiamo un esempio con segmenti di dimensione 1 Kbyte e prima threshold a 32 Kbyte.



Si può osservare la crescita esponenziale (in verde) fino al valore al quale è stata impostata la **threshold**, cioè 32 K, e la successiva crescita lineare (in blu) fino a che si verifica una congestione, per esempio quando raggiunge il valore di 40K.

Ora è necessario resettare l'algoritmo dopo però aver modificato il valore di **threshold** e averlo impostato alla metà del valore che precedentemente ha causato la congestione, quindi a 20K.

Quindi fino a 20K si utilizzerà lo **Slow Start** e successivamente il **Congestion Avoidance**, e via di seguito.