Università degli Studi di Roma "Tor Vergata"
Laurea in Informatica

Sistemi Operativi e Reti
(modulo Reti)
a.a. 2023/2024

Esercitazione: CSMA/CD e Controllo della Congestione TCP

dr. Manuel Fiorelli

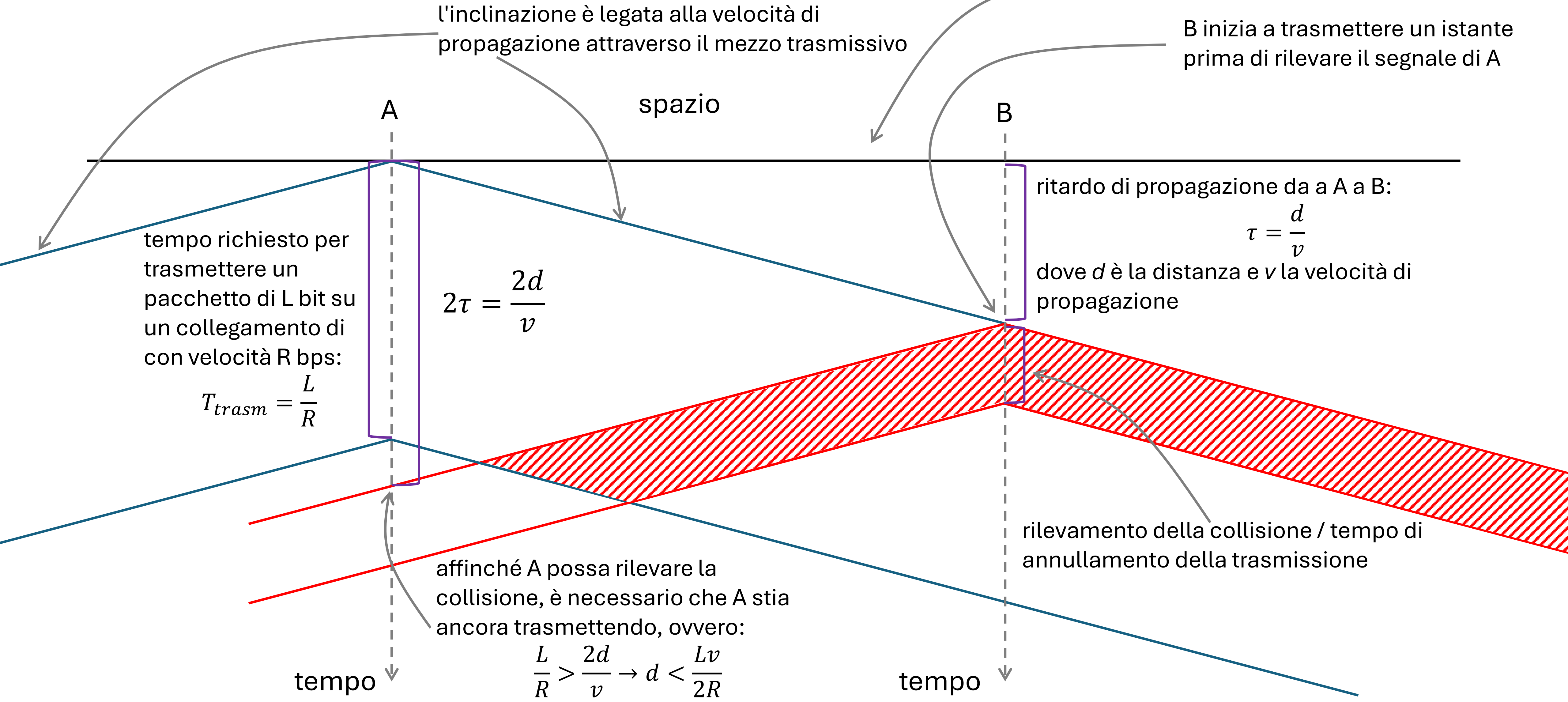
manuel.fiorelli@uniroma2.it

https://art.uniroma2.it/fiorelli

Esercizio 1

Facendo riferimento allo schema CSMA/CD, determinare che vincolo c'è tra lunghezza minima dei pacchetti e distanza massima tra due interfacce.

Esercizio 1 (soluzione)



Esercizio 1 (soluzione)

Partendo dalla relazione appena trovata:

$$\frac{L}{R} > \frac{2d}{v} \rightarrow d < \frac{Lv}{2R}$$

Supponiamo:

- lunghezza minima di un pacchetto $L = 64\ B = 64\ bit$
- $R = 10\ Mbps$
- $v = \frac{2}{3}c = 2 \cdot 10^8 m/s$

Inserendo questi valori deriviamo:

$$d < \frac{512\ * 2 \cdot 10^8}{2\ * 10 \cdot 10^6}\ \frac{bit \cdot m/s}{bit/s} = 5120\ m$$

Nota: al crescere di R, *d* diminuisce: per esempio, a 1 Gbps *d* dovrebbe essere inferiore a circa 51 m. Pertanto, occorre aumentare il valore di *L*. Abbiamo visto come evoluzioni di Ethernet permettano (sotto certe condizioni) di evitare l'uso di CSMA/CD.

Esercizio 1 (soluzione)

Nota:

Il calcolo appena mostrato si riferisce a CSMA/CD in generale.

Specifici standard come Ethernet devono ovviamente rispettare questi limiti superiori, ma possono dettare vincoli più specifici tenendo in considerazione molti altri fattori.

Il limite calcolato si riferisce all'intero dominio di collisione. Inoltre, altri fattori possono determinare vincoli sulla lunghezza dei tratti individuali di cavo.

Esercizio 2 (vedi esercizio originale in Cap 3, problema P47)

Derivare la formula di alto **livello del throughput** medio di una connessione TCP in funzione del **tasso di perdita di pacchetti** *L*. Si trascurino le altre fasi del controllo della congestione, assumendo che TCP sia sempre nella fase di congestion avoidance, assumendo nello specifico che la finestra di congestione esibisca un andamento a dente di sega tra W/2 e W.

$$\text{TCP throughput medio} = \frac{1.22 \cdot \text{MSS}}{\text{RTT}\sqrt{L}}$$

NOTA: il tasso di perdita di pacchetti si riferisce alla frequenza degli eventi di perdita (timeout o triplice ACK duplicato), qualunque sia la loro causa (es. overflow dei buffer o altri problemi di trasmissione, inclusi errori sui bit).

Esercizio 2 (soluzione)

La periodicità dell'andamento a dente di sega ci permette di calcolarne il throughput medio su un singolo periodo (di crescita lineare di cwnd).

Per semplicità:

- esprimiamo l'ampiezza della finestra di congestione in segmenti (di pari lunghezza MSS) anziché in byte
- assumiamo che TCP operi per cicli di trasmissione (di durata pari a un RTT): all'inizio del ciclo invia tutti i segmenti (di pari lunghezza MSS) e, trascurando il ritardo di trasmissione dei segmenti, riceve tutti gli ACK prima dell'inizio del ciclo successivo, nel quale la finestra di congestione è incrementata di 1 (MSS)
- trascuriamo qualsiasi overhead dovuto alle intestazioni

Esercizio 2 (soluzione)

In un singolo periodo (che consta di $\frac{W}{2} + 1$ cicli di trasmissione), trasmetteremo dunque il seguente numero di pacchetti (contenenti ciascuno un segmento):

$$\frac{W}{2} + \left(\frac{W}{2} + 1\right) + \cdots + \left(\frac{W}{2} + \frac{W}{2}\right) = \frac{W}{2}\left(\frac{W}{2} + 1\right) + \sum_{i=1}^{\frac{W}{2}} i = \frac{W}{2}\left(\frac{W}{2} + 1\right) + \frac{\frac{W}{2}\left(\frac{W}{2} + 1\right)}{2} = \frac{W}{2}\left(\frac{W}{2} + 1\right)\left(1 + \frac{1}{2}\right) = \frac{3}{2}\frac{W}{2}\left(\frac{W}{2} + 1\right)$$

$$= \frac{3}{8}W^2 + \frac{3}{4}W$$

Ipotizzando che si verifichi una sola perdita alla fine dell'ultimo ciclo, il tasso di perdita (cioè pacchetti persi / pacchetti trasmessi):

$$L = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

Esercizio 2 (soluzione)

Per W grande, il termine quadratico è dominante, per cui possiamo derivare la seguente approssimazione:

$$L = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W} \approx \frac{1}{\frac{3}{8}W^2} \rightarrow W \approx \sqrt{\frac{8}{3}\frac{1}{\sqrt{L}}}$$

Questo valore di W è espresso in segmenti: lo moltiplichiamo per MSS per ottenerne la dimensione in bit (o multipli):

$$W' = \sqrt{\frac{8}{3}\frac{1}{\sqrt{L}}}MSS$$

Inseriamo questo valore nell'equazione del throughput medio (si veda sul libro come viene derivata questa formula e su quali assunzioni si poggia!):

$$throughput\ medio = 0.75\frac{W'}{RTT} = 0.75\frac{\sqrt{\frac{8}{3}\frac{1}{\sqrt{L}}}MSS}{RTT} = 0.75\sqrt{\frac{8}{3}}\frac{MSS}{RTT\sqrt{L}} = \frac{1.22 \cdot MSS}{RTT\ \sqrt{L}}$$

Esercizio 3 (vedi esercizio originale in Cap 3, problema P48)

Si consideri che una **singola connessione** TCP Reno utilizzi un collegamento da R = 10 Mbps e si supponga che esso sia l'unico collegamento congestionato lungo il percorso tra il mittente e il destinatario.

Si assuma che:

- la connessione sia sempre in congestion avoidance (andamento a dente di sega)
- RTT = 150 ms
- tutti i segmenti abbiamo dimensione 1500 byte
- il mittente abbia sempre dati da trasmettere
- la finestra di ricezione sia molto più grande della finestra di congestione del mittente

Determinare:

- dimensione massima della finestra, espressa in segmenti
- throughput medio, espresso in bps
- tempo impiegato per raggiungere nuovamente la dimensione massima della finestra dopo il dimezzamento dovuto alla perdita di un pacchetto

Esercizio 3 (soluzione)

La dimensione della finestra W determina il tasso di trasmissione del mittente:

$$tasso\ di\ trasmissione = \frac{W \cdot MSS}{RTT}$$

Quando il tasso di trasmissione eccede la capacità del collegamento R, i pacchetti tendono a accumularsi nel buffer (davanti al collegamento), finché pacchetti non sono scartati. Possiamo quindi approssimare l'ampiezza massima della finestra come segue:

$$\frac{W \cdot MSS}{RTT} \leq R \ \rightarrow W \leq \frac{R \cdot RTT}{MSS} \rightarrow W_{max} \approx \frac{R \cdot RTT}{MSS} = \frac{10 \cdot 10^6 * 150 \cdot 10^{-3}}{1500 \cdot 8} \frac{\frac{bit}{s} \cdot s}{bit}$$

$$= \frac{1}{8} \cdot 10^3 = 0.125 \cdot 10^3 = 125$$

Esercizio 3 (soluzione)

Avendo determinato che $W_{max} \approx 125\ segmenti$, applicando la formula vista a lezione otteniamo che il throughput medio è:

$$troughput\ medio = \frac{3}{4}\frac{W_{max} \cdot MSS}{RTT} = \frac{3}{4}\frac{125 * 1500 \cdot 8}{150 \cdot 10^{-3}}\frac{bit}{s}$$
$$= 750 \cdot 10^{4} bps = 7.5\ Mbps$$

(nelle soluzioni del libro viene suggerito di calcolare prima il *ceiling* di $\frac{3}{4}W_{max}$, che è l'ampiezza media della finestra in segmenti, ottenendo 94, convertire dunque il valore in bit e dividerlo per RTT, ottenendo un valore leggermente maggiore *7.52 Mbps*)

Esercizio 3 (soluzione)

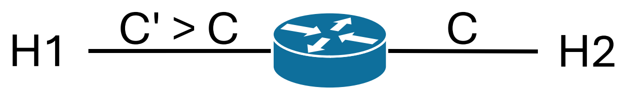
La finestra viene dimezzata da 125 a 62 quando viene rilevata una perdita, dopo di che viene incrementata di 1 ad ogni RTT (stiamo ignorando slow start e fast recovery!)

Per tornare alla dimensione massima della finestra occorrono pertanto:

$$(125 - 62)\ RTT = 63\ RTT = 63 \cdot 150\ ms = 9.45\ s$$

Esercizio 4 (vedi esercizio originale in Cap 3, problema P49)

Si consideri il caso in cui una **singola connessione TCP Reno** (pertanto il controllo di congestione è basato sulle perdite) attraversa due collegamenti, di capacità C' e C tale che C' > C, come in figura.



Dimostrare che se la dimensione *B* del buffer davanti al secondo collegamento è tale che $B \geq C * (2 * \mathit{ritardo\ di\ propagazione\ tra\ mittente\ e\ destinatario})$, allora il collegamento collo di bottiglia sarà sempre occupato a trasmettere.

Esercizio 4 (soluzione)

Assumiamo per semplicità che:

- TCP invii segmenti (interi, di lunghezza 1 MSS ciascuno) in cicli (round) di trasmissione, di durata pari a RTT, incrementando di 1 MSS (quindi di un segmento) la dimensione della finestra di congestione all'inizio del ciclo (round) successivo
- C sia espresso in pacchetti (segmenti) al secondo

Assumendo che C' > C, il mittente continuerà a aumentare l'ampiezza della finestra e quindi la velocità di trasmissione, fino a eccedere la capacità del collegamento collo di bottiglia, riempendo il suo buffer

-> *takeaway*: per quanto possano essere grandi i buffer sul collegamento collo di bottiglia, TCP li riempirà causando una perdita!

Esercizio 4 (soluzione)

Sia W la dimensione della finestra quando viene rilevata la perdita (quindi la dimensione massima della finestra), TCP dimezza la finestra a W/2 (ignoriamo slow start e fast recovery!).

Siccome ci sono W segmenti in transito non ancora riscontrati, TCP non può trasmettere segmenti finché non riceve W/2 ACK.

Se il collegamento collo di bottiglia non è mai a corto di pacchetti, vuol dire che trasmette a un tasso costante C, quindi i riscontri arrivano al mittente con la stessa frequenza C, che dunque dovrà attendere per $\frac{\frac{W}{2}}{C}$ cioè $\frac{W}{2C}$.

Durante questo periodo il collegamento avrà prelevato dal buffer $\frac{W}{2}$ pacchetti pertanto il buffer deve essere $B \geq \frac{W}{2}$

Esercizio 4 (soluzione)

Quando il mittente riprende a trasmettere, il buffer è vuoto, quindi per evitare che il collegamento collo di bottiglia sia inutilizzato occorre che il tasso di invio del mittente $\frac{\frac{W}{2}}{RTT} = \frac{W}{2RTT}$ sia uguale a C.

In altre parole,

$$\frac{W}{2RTT} = C \;\rightarrow W = 2 \cdot RTT \cdot C$$

Sostituendo questa espressione nella disuguaglianza di prima, otteniamo:

$$B \geq RTT \cdot C$$

Quando il mittente riprende a trasmettere il buffer è vuoto, quindi non c'è ritardo di accodamento e trascurando le altre componenti del ritardo (si ripassi quali sono!) possiamo approssimare RTT come il doppio del ritardo di propagazione tra mittente e destinatario, dimostrando come richiesto che:

$$B \geq C * (2 * \; ritardo\ di\ propagazione\ tra\ mittente\ e\ destinatario)$$

Esercizio 4 (soluzione)

Si noti che:

- Se il buffer fosse più piccolo, il collegamento andrebbe a corto di pacchetti durante la pausa del mittente. Ne seguirebbe che il collegamento non potrebbe essere occupato tutto il tempo a trasmettere pacchetti, con una conseguente perdita di throughput
- Se il buffer fosse più grande, il buffer non si svuoterebbe mai (cosa che in realtà abbiamo assunto nella dimostrazione), determinando la presenza di un numero minimo di pacchetti nel buffer e quindi di un ritardo di accordamento minimo non nullo → peggioramento dell'RTT → problema per applicazioni real-time
- Alla fine, la scelta migliore è quella di dimensionare il buffer uguale alla dimensione minima trovata

Esercizio 4 (soluzione)

Note:

- Questo esercizio voleva dare un'idea della ragione della vecchia *rule of thumb* per il dimensionamento dei buffer dei router. Si noti che questa menziona in realtà l'RTT medio!!!!!
- Si ricordi che è stata recentemente proposta una nuova regola con a denominatore $\sqrt{N}$ dove N è il numero di flussi TCP indipendenti che attraversano un collegamento (si veda nelle slide e nel libro le altre specifiche assunzioni sotto cui vale)