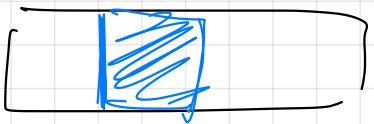
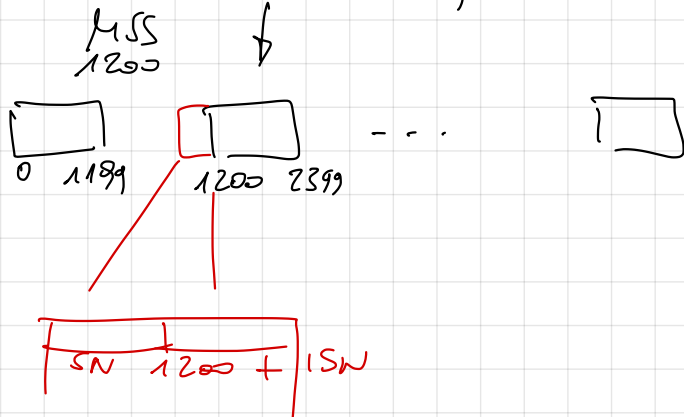
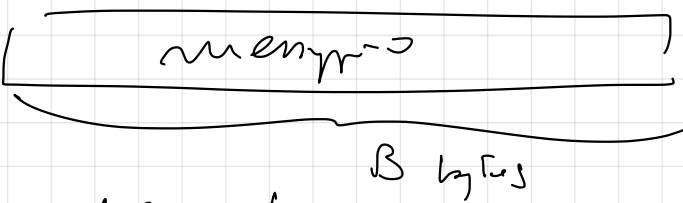
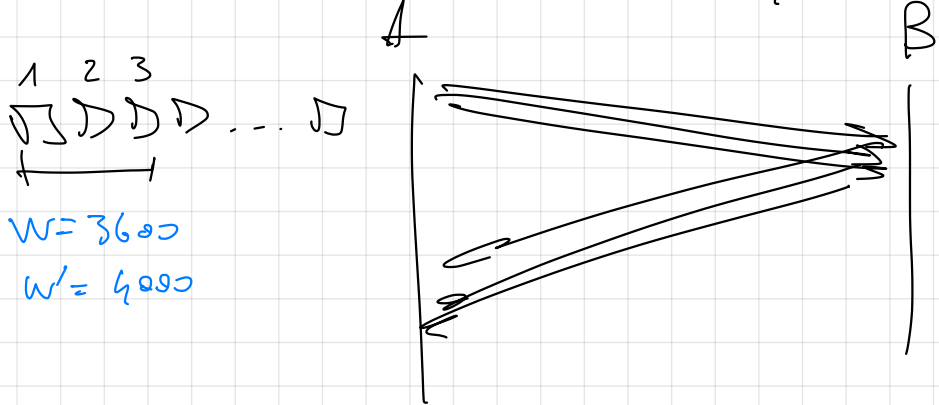


LEZIONE DEL 2/11/2021

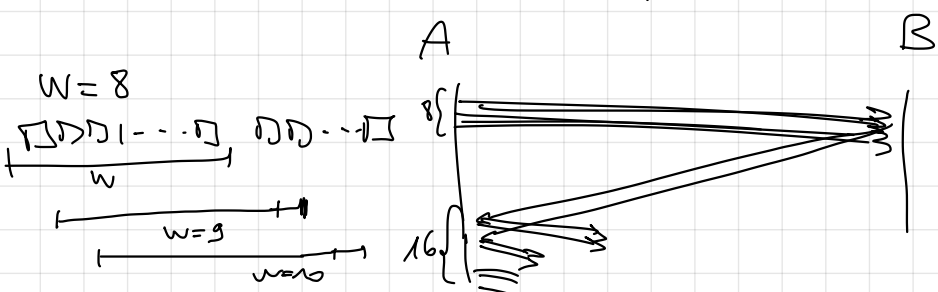
Nota su Sequence Number / Acknowledge Number / Window

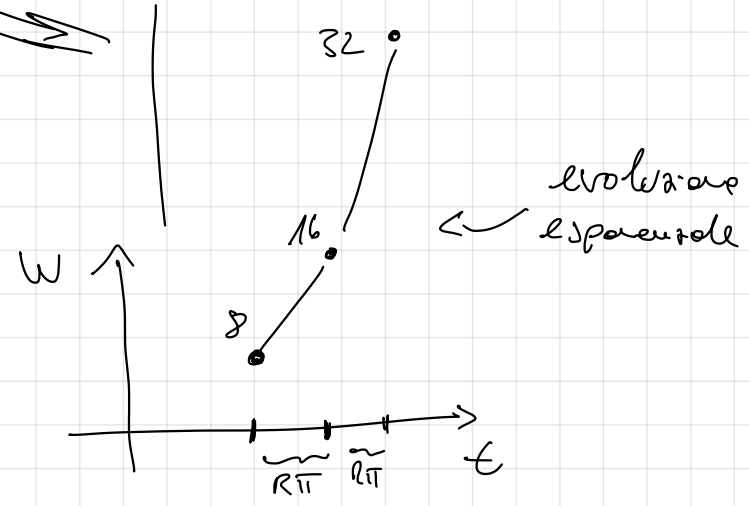
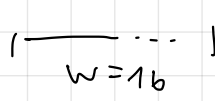


Nel momento, l'ACK Number contiene il primo byte che ci si aspetta di ricevere

AUMENTO DELLA FINestra → SLOW START

Per ogni ack ricevuto, si aumenta la finestra di un segmento (oltre a far scorrere lo spazio a destra di un segmento)



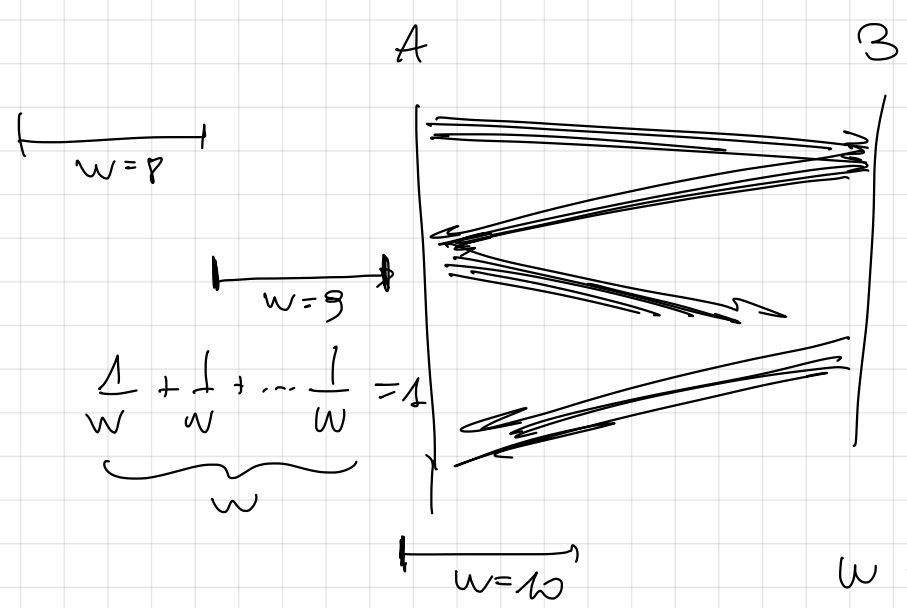


Rappresentazione alternativa

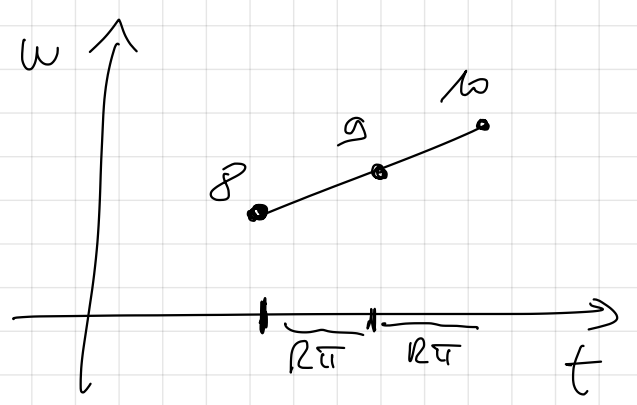
Algoritmo

CONGESTION AVOIDANCE

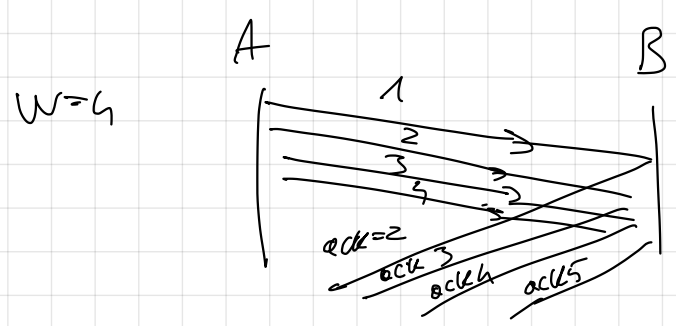
Per ogni ack ricevuto, aumento lo finestra di $\frac{1}{w}$



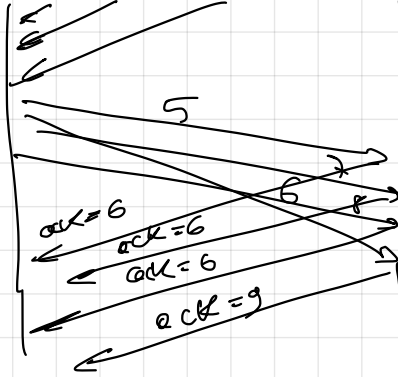
Andiamo
lineare
↓



Rappresentazione nel tempo



$$W=5$$



1 2 3 4 5 6 7 8
☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Diminuzione \rightarrow VANILLA
 \rightarrow FR/FR

Se non arrivo un riscontro, si aspetta lo scadere del RTO, si pone lo finetto $W=1$ (VANILLA) $W = \frac{W'}{Z}$ al momento della perdita

e si ricalcolano i sequenti per i (FR/FR)
 VANILLA \rightarrow GO-Back-N
 FR/FR \rightarrow selective repeat

SOLUZIONE Esercizio con 3 LAN di 300/40/90 host

I° passo → tradurre i vincoli in blocchi di base potere di 2

L1 → 300 host → 512 indirizzi
L2 → 40 " → 64 "
L3 → 90 " → 128 "

II° passo → Sommare le dimensioni dei blocchi $704 \leftarrow$ # di indirizzi minimo per soddisfare le esigenze del problema

III° passo → trovare la potenza di 2 immediatamente superiore
→ $1024 \rightarrow 2^{10}$

Il blocco CIDR totale avrà $32 - 10 = 22$ bit per il prefisso
→ /22

IV° passo → traduzione in binario

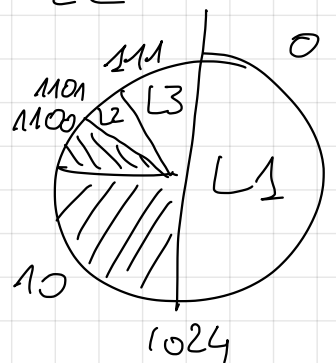
1001 0100 0000 1100 0100 1111 1111 1111

Blocco CIDR totale

" " 01 00 11 00 0000 0000

→ notazione decimale puntata

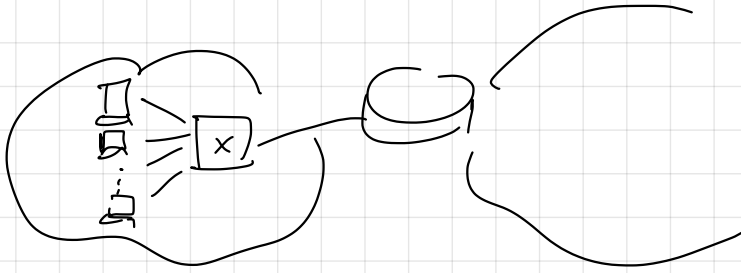
148 . 12 . 76 . 0 / 22



L1 " " 01 00 11 0 0 0000 0000
148 . 12 . 76 . 0 / 23

$\frac{1}{1000000} \quad \frac{1}{100000} \quad \frac{1}{10000} \quad \frac{1}{1000} \quad \frac{1}{100} \quad \frac{1}{10} \quad \frac{1}{1} \quad \frac{1}{10} \quad \frac{1}{100} \quad \frac{1}{1000} \quad \frac{1}{10000} \quad \frac{1}{100000} \quad \frac{1}{1000000}$
 148 . 12 . 79 . 128 / 25

" " 01 0011 11 01 00 0000
 |-----|
 148 . 12 . 79 . 64 / 26



LEZIONE DEL 3/11/2021

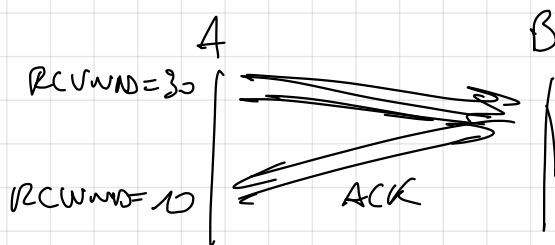
ALGORITMO DI CONTROLLO DELLA CONGESTIONE DEL TCP

- CWND \rightarrow Congestion Window \rightarrow dimensione della finestra di trasmissione
- RTT \rightarrow calcolato dinamicamente, basato su RTT
- RCV WND \rightarrow Receive Window \rightarrow finestra massima di ricezione
- Ssthresh \rightarrow Slow Start Threshold \rightarrow soglia per distinguere l'uso dell'algo. SS e CA

Inizializzazione

$$CWND = 1$$

RCV WND = viene comunicato dalla destinazione in ogni header TCP \rightarrow campo window



$$Ssthresh_{iniziale} = RCV WND \text{ (oppure } RCV WND / 2)$$

Algoritmo

1. Invia un # sequenti per allo CWND
2. Quando arrivano i ricevuti (ACK)

if $CWND < Ssthresh$

// uso lo Slow Start

$$CWND_{new} = \min(CWND_{old} + \#ACK, Ssthresh, RCV WND)$$

else

// uso il Congestion Avoidance

$$CWND_{new} = \min(CWND_{old} + \frac{\#ACK}{CWND_{old}}, RCV WND)$$

Vedo al punto 2

3. Se non arrivano i riscontri

Aspetto che scada il RTO

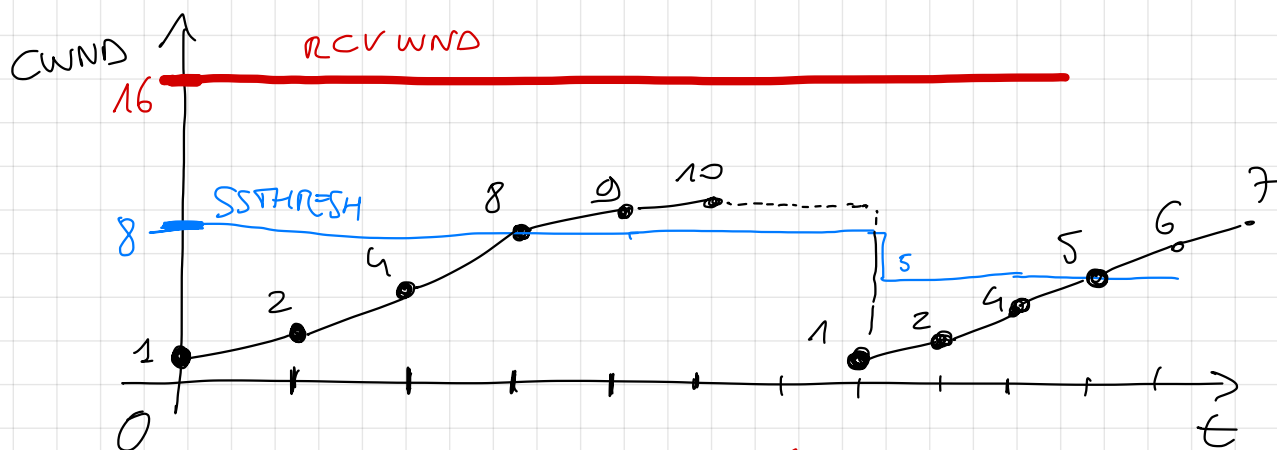
Poi $SSTHRESH = \frac{cwnd}{2}$ al momento dello perdita

Poi $cwnd = 1$ (Vanilla TCP) / $cwnd = SSTHRESH$ ($\frac{FR}{FR}$)

Per i seguenti ritrasmissioni, poi $RTO_{new} = 2 \cdot RTO$

Torno al punto 1

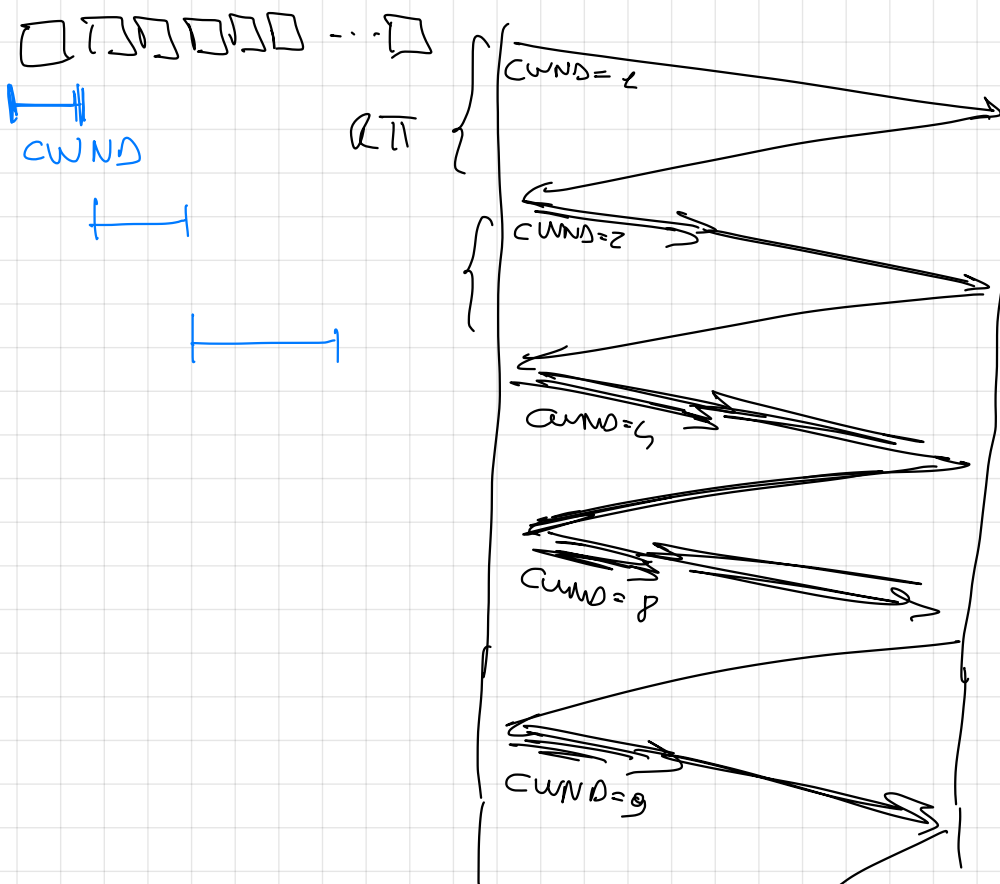
Esempio



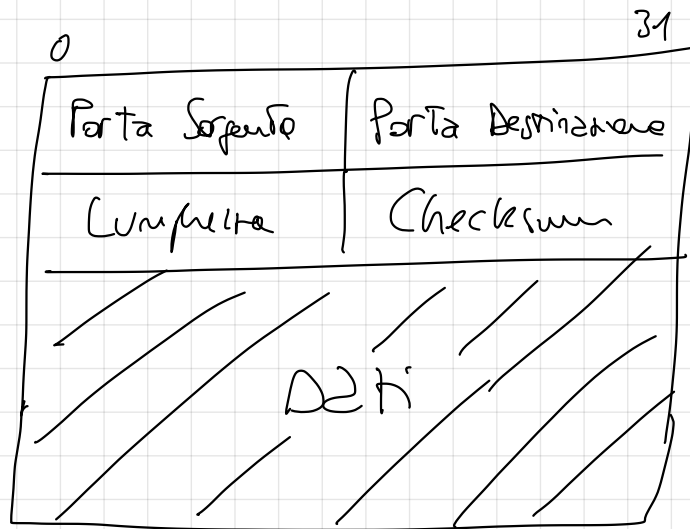
sequenti
messi

1 + 2 + 4 + 8 + 9 + ~~10~~ + 1 + 2 + 4 + 5

A B



Il protocollo UDP



Esercizio TCP

Un'applicazione A deve trasferire verso un'applicazione B
96 000 Byte

Si suppone che la connessione sia già stata instaurata

MSS = 1000 byte

RCVWND = 32000 byte, costante per tutto il tempo

$SSTHRESH_{min} = RCVWND_{min} / 2$

RTT \rightarrow costante, pari a 0,5 secondi

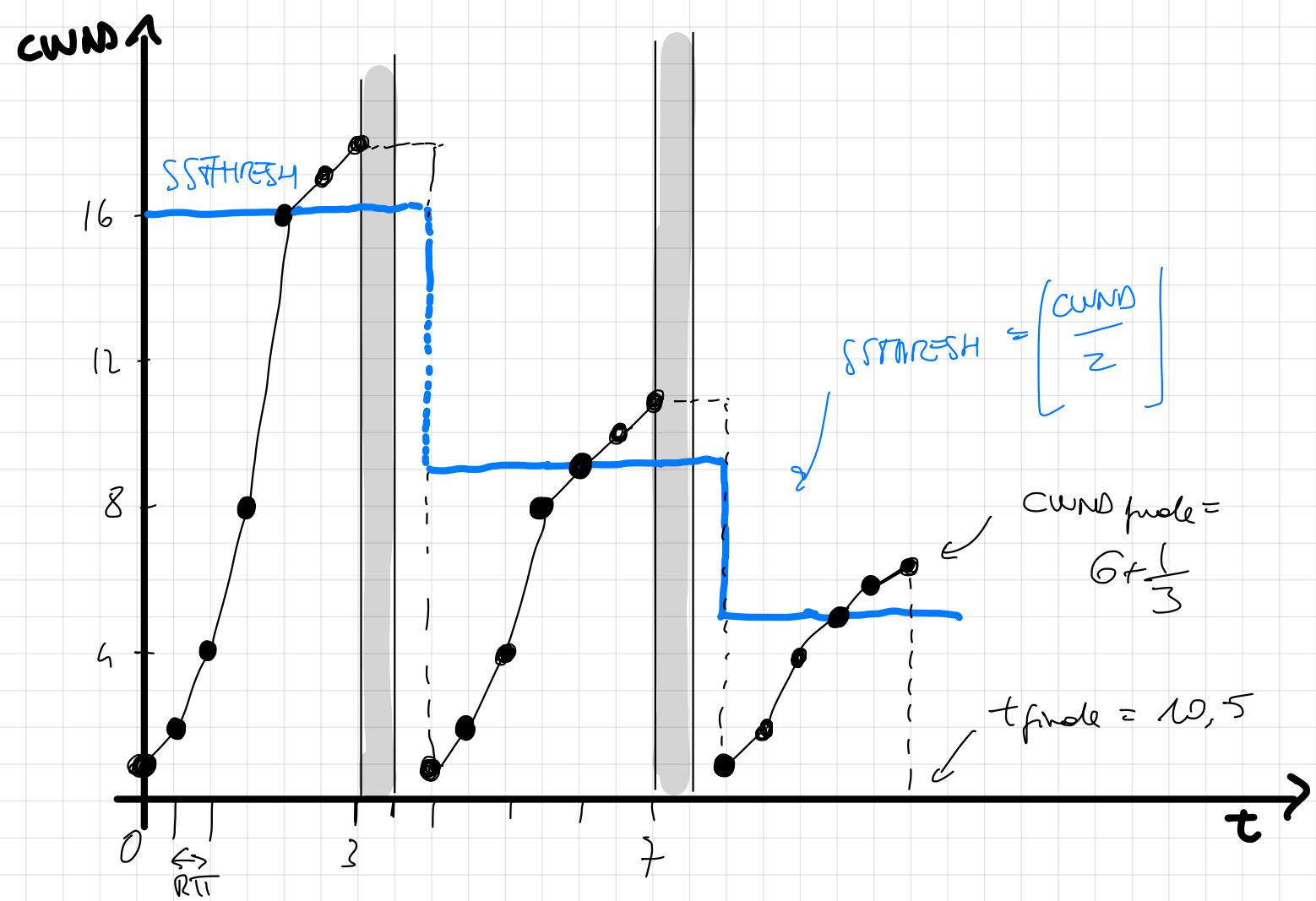
RTO = 2 RTT, raddoppio in caso di perdite frequenti

Down di rete (rete fuori uso) negli intervalli aperti $3 \rightarrow 3,5$
 $7 \rightarrow 7,5$

segmenti da trasmettere $\rightarrow 96000 / 1000 \rightarrow 96$ segmenti

RCVWND $\rightarrow 32000 / 1000 \rightarrow 32$ segmenti

~~SSTHRESH~~_{net} $\rightarrow 16$ segmenti



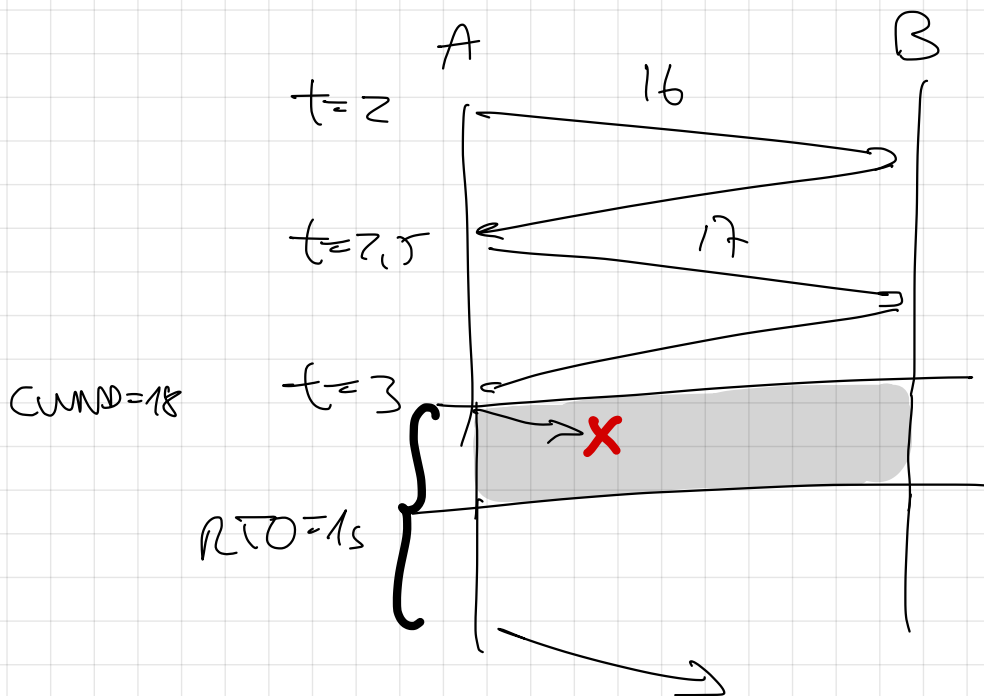
sequenti
messaggi

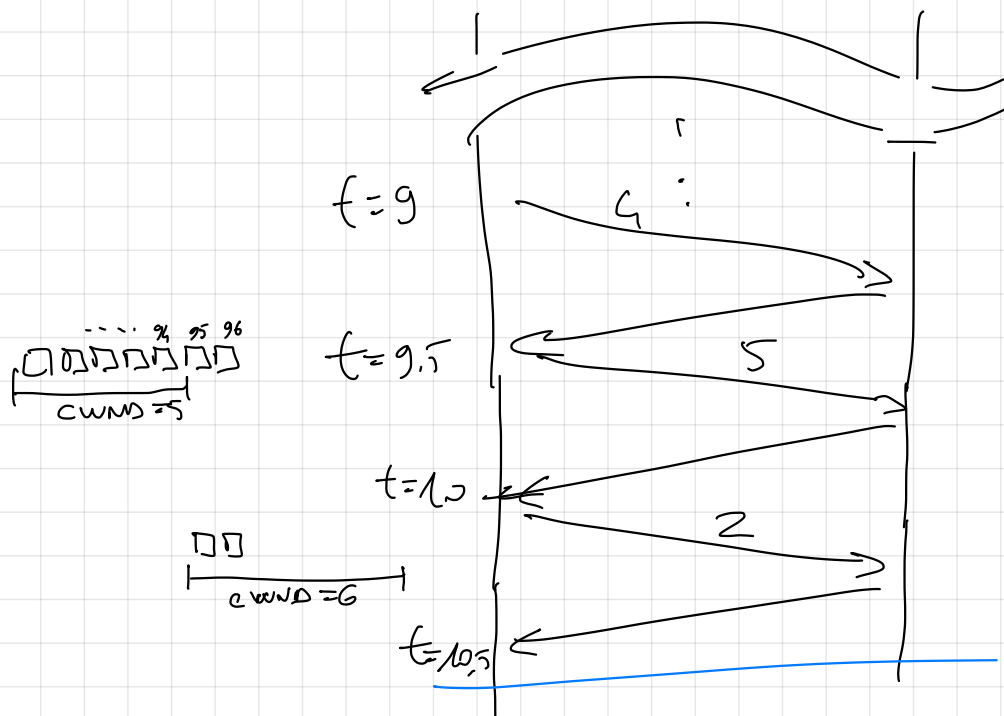
$$1 + 2 + 4 + 8 + 16 + 17 + 18 + 1 + 2 + 4 + 8 + 9$$

(48)

$$+ 10 + 11 + 1 + 2 + 4 + 5 + \left(\frac{2}{6} \right)$$

(94)





$$CWND_{new} = CWND_{old} + \frac{\#ack}{CWND_{old}}$$

$$6 + \frac{2}{6}$$

ESERCIZIO TCP

Applicazione A \rightarrow dati verso Applicazione B 46500 byte

MSS = 1500 byte

$RCVWND_{initiale} = 24000$ byte $SSTHRESH_{initiale} = \frac{RCVWND_{initiale}}{2}$

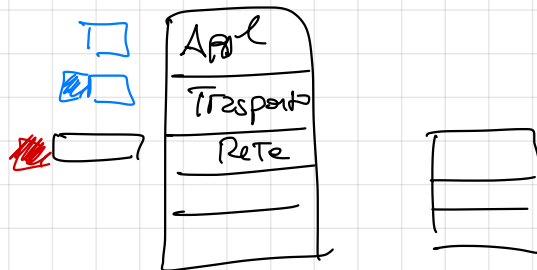
$RTT = 0,5$ costante $RTO_{base} = 2RTT$

Adattamento per perdite consecutive

ReSe from us $1,5 \rightarrow 3,5$
 $7 \rightarrow 7,5$

LEZIONE 2 EL 4/11/2021

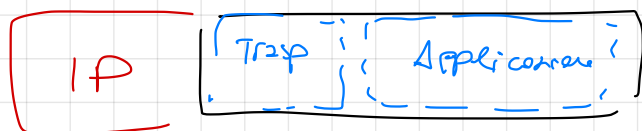
Il protocollo IP



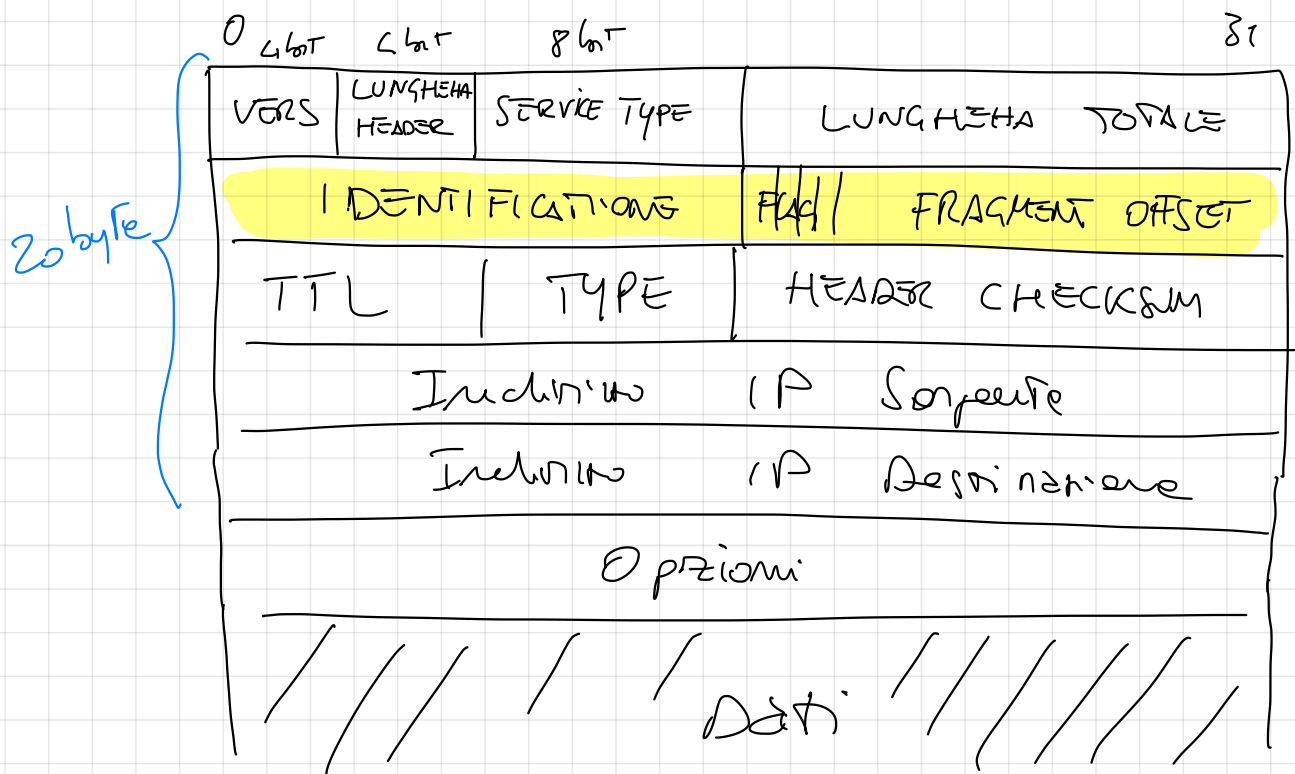
Scopo del livello di Rete:

BEST EFFORT

Dato una destinazione, fare il possibile per consegnare il pacchetto a tale destinazione



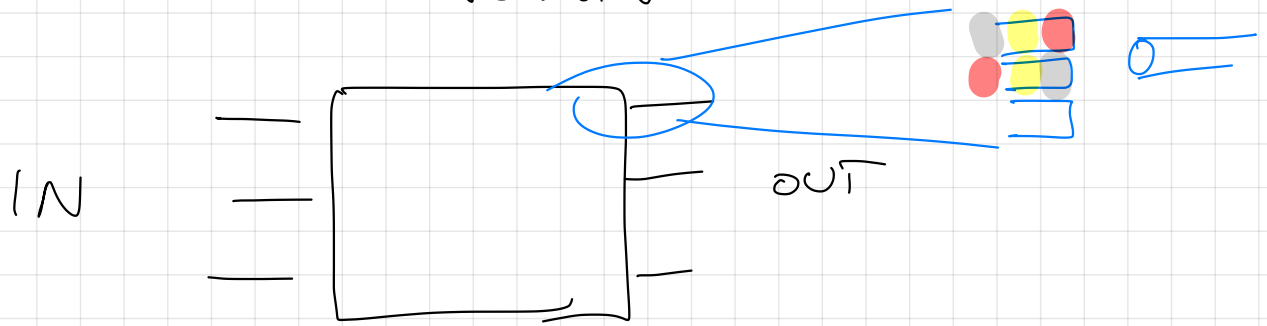
Header del protocollo IP



VERS → indica la versione del protocollo utilizzato
→ attualmente V04 (v4)
→ esiste V06 (v6)

LUNGH. HEADER → 20 se non ci sono opzioni, >20 altrimenti

Service Type \rightarrow Codice che identifica le porte di servizio



TTL \rightarrow Time To Live \rightarrow Contatore incrementato dalla sorgente (64/128) che identifica il # massimo di hop che il pacchetto può attraversare

Ogni Router decrementa tale valore di 1. Se un router riceve un pacchetto con TTL=1, e quindi per default 0 \rightarrow scarta il pacchetto e manda un messaggio di errore alla sorgente.

Type \rightarrow Codice che identifica i dati trasportati nel payload (il protocollo usato)