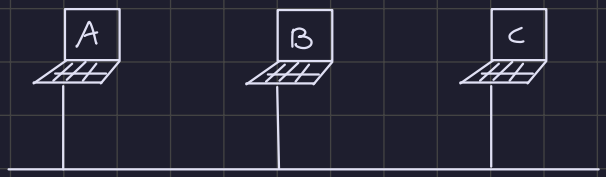
Esame 09/01/23

Domande sulla teoria (4 punti ciascuna)

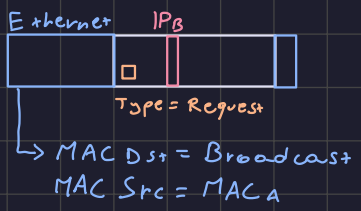
Lo studente risponda in maniera concisa, ma precisa, alle seguenti domande riguardanti la parte teorica. E' necessario che lo studente ottenga almeno 7 punti (su un totale di 12 punti a disposizione). In caso contrario, gli esercizi non verranno considerati e il voto finale sarà insufficiente.

1. Si spieghi lo scopo e il funzionamento del protocollo ARP (Address Resolution Protocol), e in particolare i messaggi trasmessi dalle stazioni, specificando gli indirizzi usati.
2. Si spieghi brevemente la funzionalità di frammentazione dei pacchetti IP, mostrando un caso in cui essa è necessaria, gli apparati coinvolti sia nella frammentazione che nel riassemblaggio e i principali campi dell'header coinvolti.
3. L'header del protocollo UDP contiene solo 4 campi: Source Port, Destination Port, Length e Checksum. Si spieghi brevemente a cosa servono tali campi.

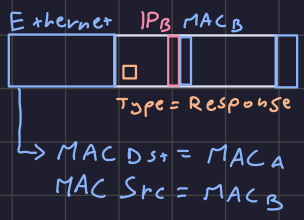
1. Il protocollo ARP serve per ottenere l'indirizzo MAC partendo dall'indirizzo IP. Una stazione richiede l'indirizzo MAC di una specifica stazione inviando la richiesta in broadcast (di livello 2) specificando l'IP della stazione di cui vuole ottenere il MAC. Una stazione che riceve il messaggio e che conosce l'indirizzo MAC associato all'IP richiesto risponde inviando, alla stazione che lo ha richiesto, l'indirizzo MAC. Tutte le risposte ARP ottenute vengono salvate all'interno di una tabella ARP e hanno un tempo di validità oltre il quale vengono eliminate. Un esempio è il seguente:



Ci sono 3 stazioni collegate sullo stesso mezzo condiviso, la stazione A vuole conoscere l'indirizzo MAC della stazione B, quindi invia un messaggio ARP in broadcast specificando l'indirizzo MAC sorgente e destinazione, l'IP di cui vuole sapere il MAC e il tipo di messaggio:



Tutte le stazioni ricevono il messaggio, ma supponiamo che soltanto B conosca il suo indirizzo MAC, quindi la stazione B prepara un messaggio di risposta diretto verso la stazione A in cui specifica il suo indirizzo MAC:

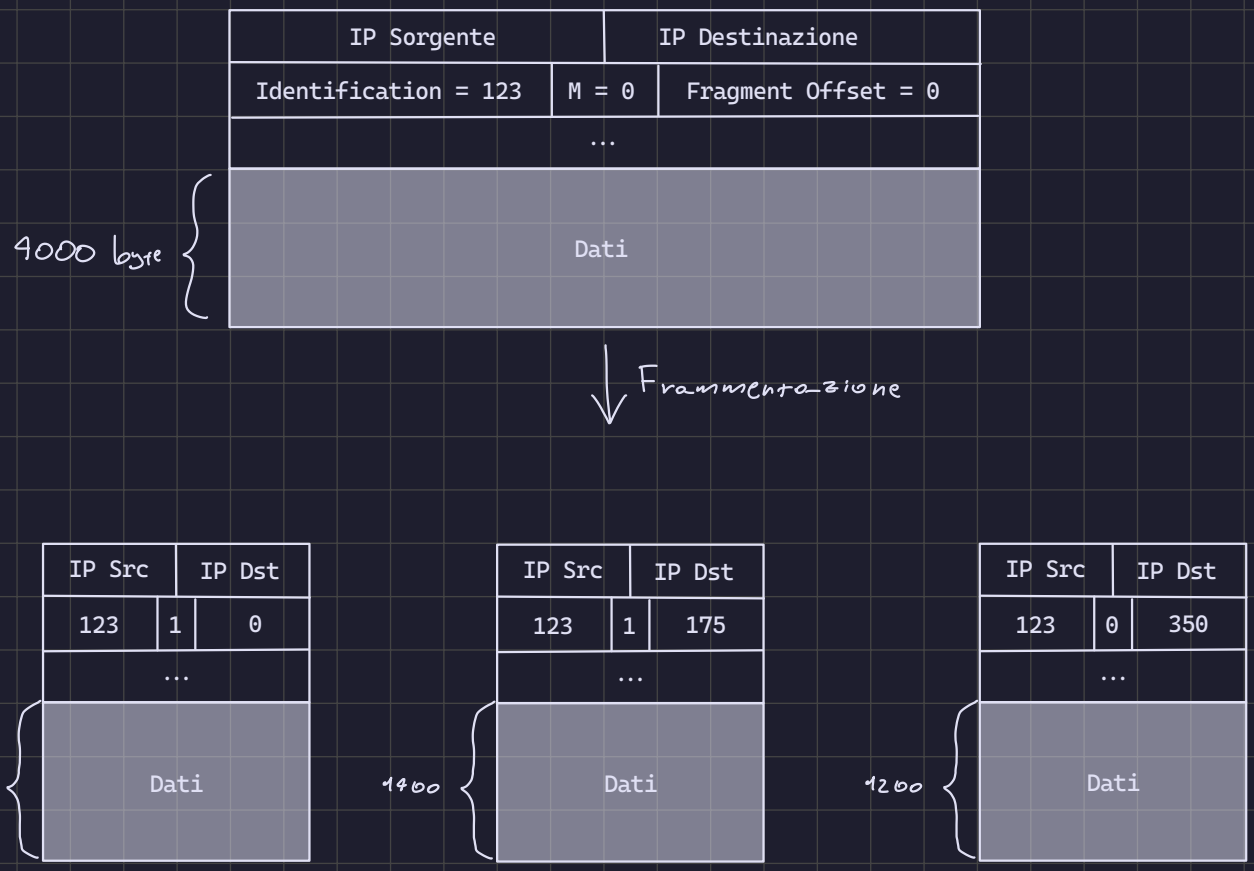


La stazione A riceverà questo messaggio e salverà il MAC della stazione B ottenuto all'interno della sua tabella ARP.

2. Un pacchetto IP potrebbe passare su un link che ha la MTU, cioè la lunghezza massima che un certo canale può trasmettere, minore della lunghezza del pacchetto, in questo caso il router su cui si trova il pacchetto deve preoccuparsi di frammentarlo, cioè separarlo in più pezzi abbastanza grandi da poter essere trasmessi sul link. I pacchetti frammentati verranno poi ricevuti dal destinatario che fa partire un timer al primo pacchetto ricevuto e se non li riceve tutti entro la scadenza scarta tutti i pacchetti e aspetta che vengano rimandati. L'unico responsabile per il riassemblaggio dei pacchetti è l'host di destinazione. I campi dell'header coinvolti nella frammentazione sono:
   - Identification: È un identificativo generato dall'host che ha creato il pacchetto e serve a identificare più frammenti che fanno parte del pacchetto originale
   - Flag M:
     - Vale 0 se il pacchetto non è stato frammentato, oppure se è l'ultimo frammento
     - Vale 1 se il pacchetto è stato frammentato, tranne se è l'ultimo frammento
   - Fragment Offset: È un offset rispetto al primo byte del pacchetto originale diviso per 8

   Un esempio di frammentazione è la seguente:

   Si ha un pacchetto da 4000 byte per il payload e 20 byte per l'header che arriva ad un router con un link che ha MTU = 1420 byte. Il router allora deve frammentare il pacchetto in 3 parti: due da 1400 byte e una da 1200 byte (1400*2 + 1200 = 4000).



Gli offset sono:

$$\text{Primo frammento} = \frac{0}{8} = 0$$

$$\text{Secondo frammento} = \frac{1400}{8} = 175$$

$$\text{Terzo frammento} = \frac{2800}{8} = 350$$

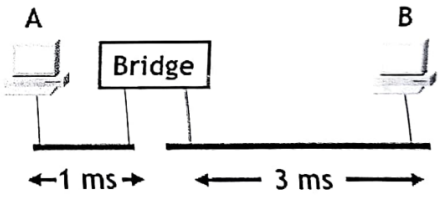
I tre frammenti hanno lo stesso numero Identification perchè fanno parte dello stesso pacchetto originale. Inoltre il primo frammento ha Fragment Offset a 0 perchè è il primo e di conseguenza ha il flag M settato a 1 per far capire che è un frammento. L'ultimo frammento, invece ha il flag M posto a 0, quindi per sapere che è un frammento si guarda il Fragment Offset, che in questo caso è diverso da 0.

3. L'header del protocollo UDP contiene i seguenti campi:
   - Source Port: È la porta sorgente, cioè quel numero che identifica un processo all'interno di un host.
     Le porte si dividono in 2 tipi:
     - Porte statiche: Vanno dalla 0 alla 1023, sono definite da uno standard e sono utilizzate lato server, quindi un client non le può utilizzare
     - Porte dimaniche: Vanno dalla 1024 alla 65535 e sono assegnate dal client quando viene aperto un socket, cioè un flusso di comunicazione bidirezionale.
   - Destination Port: È la porta destinazione
   - Length: Indica la lunghezza del messaggio, cioè la lunghezza dell'header più la lunghezza del payload
   - Checksum: È un valore univoco che serve per controllare se ci sono errori all'interno del messaggio. Il checksum si ottiene da una funzione di hash che prende in input un messaggio e da in output un valore univoco a dimensione fissa. La caratteristica di questa funzione è che per ogni input esiste un output univoco.
     Chi manda il messaggio mette in input il segmento e riceve un checksum da mettere all'interno del campo del messaggio. Chi lo riceve mette in input il segmento e controlla se il checksum che ha ottenuto è uguale a quello all'interno del campo, se non lo è scarta il segmento.

Il protocollo UDP è un protocollo non connection oriented, cioè non bisogna scambiare dei messaggi per instaurare una connessione prima di scambiare dei messaggi, ed è anche non affidabile, cioè se un messaggio non arriva a destinazione non viene ritrasmesso. Sarà deciso a livello applicativo come ordinare i pacchetti e come gestire la perdita.

Esercizio 1 (7 punti)

Si consideri la configurazione in figura, dove due segmenti di rete sono collegati da un Bridge; su ciascun segmento vi è una stazione (A e B rispettivamente). Il Bridge è un particolare tipo di stazione che memorizza ciascuna trama che arriva da un segmento di rete e, una volta ricevuta completamente, la ritrasmette sull'altro segmento di rete (tale comportamento è valido, in modo indipendente l'uno dall'altro, in entrambi i sensi); le trame restano in memoria del Bridge fino a quando la trasmissione sull'altro segmento non è andata a buon fine.



Le stazioni e il Bridge utilizzano un protocollo ***CSMA*** 1-persistent. Le caratteristiche del sistema sono:

- velocità dei segmenti: 800 kbit/s;
- lunghezza delle trame generate dalle stazioni: 1200 byte;
- ritardo di propagazione pari ad 1 ms tra la stazione A e il bridge; ritardo di propagazione pari a 3 ms tra la stazione B e il bridge;

Le stazioni generano le seguenti trame:

- stazione A: due trame (A1, A2) agli istanti t_{A1}=295 ms e t_{A2}=325 ms, entrambe dirette a B.
- stazione B: tre trame (B1, B2 e B3) agli istanti t_{B1}=300 ms, t_{B2}=339 ms e t_{B3}=356 ms, tutte dirette ad A;

In caso di collisione, si supponga che le stazioni decidono di ritrasmettere Z millisecondi dopo la fine della trasmissione della trama corrotta; il numero Z viene deciso secondo il seguente metodo:

- si attende un tempo pari a Z = Sc * N + T, dove
  - Sc = somma delle cifre che compongono l'istante di inizio trasmissione
  - N = numero di collisioni subite da quella trama
  - T tempo di trama

ad esempio, se l'istante di inizio trasmissione è 418 msec, Z = (4+1+8)*N + T

Determinare:

1. graficamente le trasmissioni delle diverse trame, indicando se avviene collisione, in quali istanti essa viene eventualmente avvertita e da quali apparati;
2. il periodo di vulnerabilità del sistema preso in considerazione.

$V = 800\ Kbit/s$

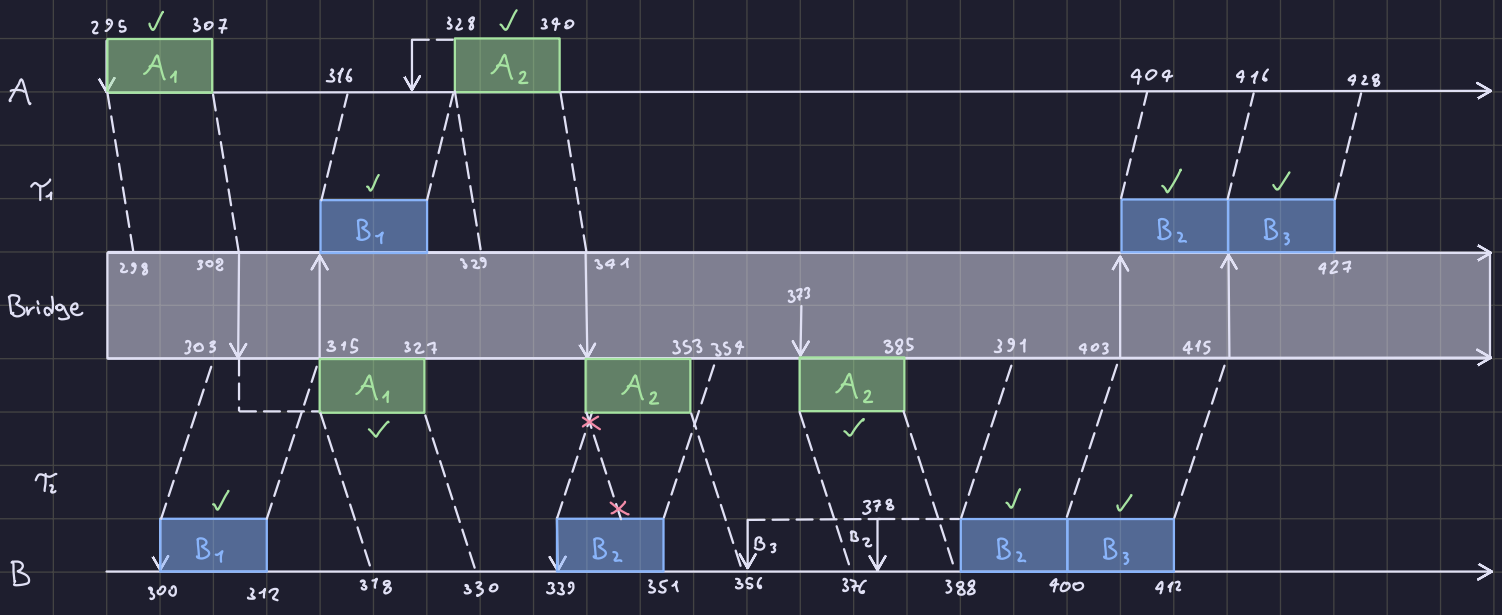
$L = 1200\ byte$

$\tau_1 = 1ms \quad \tau_2 = 3ms$

$$T = \frac{L}{V} = \frac{1200 \cdot 8}{800 \cdot 10^3} = 12\ ms$$

$$A: \begin{cases} t_{A1} = 295\ ms \rightarrow B \\ t_{A2} = 325\ ms \rightarrow B \end{cases}$$

$$B: \begin{cases} t_{B1} = 300\ ms \rightarrow A \\ t_{B2} = 339\ ms \rightarrow A \\ t_{B3} = 356\ ms \rightarrow A \end{cases}$$



Alla collisione tra A2 e B2:

$Z_{A2} = (3+4+1) \cdot 1 + 12 = 20\ ms \rightarrow 353 + 20 = 373\ ms$ A_2 ritrasmette per primo

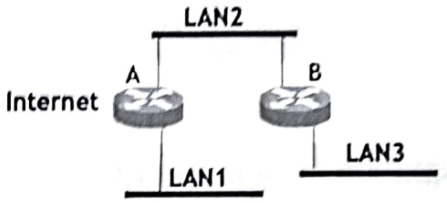
$Z_{B2} = (3+3+9) \cdot 1 + 12 = 27\ ms \rightarrow 351 + 27 = 378\ ms$

Il periodo di vulnerabilità, cioè quel periodo in cui si possono verificare delle collisioni, cambia in base al protocollo utilizzato, in questo caso per il CSMA Persistent vale 2τ. Più precisamente per il canale che va da A al Bridge il periodo è 2*1ms=2ms, mentre per il canale che va da B al Bridge il periodo è 2*3ms=6ms.

Esercizio 2 (7 punti)

La rete rappresentata nella figura a lato è composta da due router (A e B) e da tre LAN1-3, ed è interconnessa ad Internet tramite il router A. Si considerino i vincoli seguenti:

- L'indirizzo di broadcast di LAN 1 è 162.73.95.255;
- Le LAN1, LAN2 e LAN3 devono poter contenere rispettivamente almeno 500, 700 e 600 host;



In base ai suddetti vincoli:

1. Si scriva il più piccolo blocco CIDR da assegnare alla rete;
2. Partendo da tale blocco CIDR, si scriva un piano di indirizzamento per tutte le LAN;
3. Si scrivano le tabelle di routing del router B, supponendo che il router A abbia dichiarato di poter raggiungere tutti gli host su Internet con 4 hop.

Lan 1: 500 → $512 = 2^9$ host

Lan 2: 700 → $1024 = 2^{10}$ host

Lan 3: 600 → $1024 = 2^{10}$ host

1. Il blocco CIDR più piccolo da assegnare alla rete si ottiene dal blocco più piccolo in base 2 che riesce a contenere la somma di tutti i blocchi in base 2 delle singole lan:

Blocco CIDR = $1024 \cdot 2 + 512 = 2560 \rightarrow 4096 = 2^{12}$ host

Il blocco CIDR totale avrà bisogno di 12 bit di suffisso, quindi 20 bit di prefisso.
Da questo blocco si può ottenere l'indirizzo di rete partendo dall'indirizzo che è stato fornito. Un indirizzo di rete è caratterizzato da tutti i bit di suffisso posti a 0, mentre un indirizzo di broadcast è caratterizzato da tutti i bit di suffisso posti a 1.

162 . 73 . 95 . 255 /20

↓

Broadcast: 162 . 73 . 0 1 0 1 1 1 1 1 . 1 1 1 1 1 1 1 1

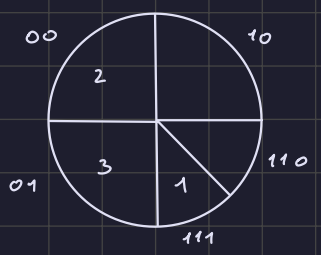
Prefisso

Rete: 162 . 73 . 0 1 0 1 0 0 0 0 . 0 0 0 0 0 0 0 0

Prefisso

↓

162 . 73 . 80 . 0 /20



2.

La LAN 1 ha bisogno di 9 bit di suffisso, quindi 23 bit di prefisso, di cui 3 fanno parte della sottorete.
Si può ricavare la sotterete della LAN 1 partendo dall'indirizzo appartenente alla LAN 1 che è stato fornito:

Rete: 162 . 73 . 0 1 0 1 1 1 1 0 . 0 0 0 0 0 0 0 0

Prefisso Sottorete

↓

162 . 73 . 94 . 0 /23

Broadcast: 162 . 73 . 0 1 0 1 1 1 1 1 . 1 1 1 1 1 1 1 1

Prefisso | Sottorete

↓

162 . 73 . 95 . 255 /23

La LAN 2 ha bisogno di 10 bit di suffisso, quindi 22 bit di prefisso, di cui 2 fanno parte della sottorete.

Rete: 162 . 73 . 0 1 0 1 0 0 0 0 . 0 0 0 0 0 0 0 0

Prefisso | Sottorete

↓

162 . 73 . 80 . 0 /22

Broadcast: 162 . 73 . 0 1 0 1 0 0 1 1 . 1 1 1 1 1 1 1 1

Prefisso | Sottorete

↓

162 . 73 . 83 . 255 /22

La LAN 3 ha bisogno di 10 bit di suffisso, quindi 22 bit di prefisso, di cui 2 fanno parte della sottorete.

Rete: 162 . 73 . 0 1 0 1 0 1 0 0 . 0 0 0 0 0 0 0 0

Prefisso | Sottorete

↓

162 . 73 . 84 . 0 /22

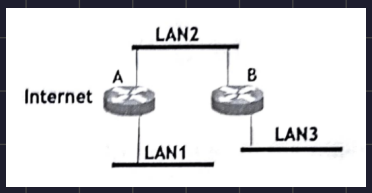
Broadcast: 162 . 73 . 0 1 0 1 0 1 1 1 . 1 1 1 1 1 1 1 1

Prefisso | Sottorete

↓

162 . 73 . 87 . 255 /22

3. La tabella di routing del router B è la seguente:



dst	next hop	cst
LAN1	A	2
LAN2	B	1
LAN3	B	1
Internet	A	5
B	B	0
A	A	1

Esercizio 3 (7 punti)

Un'applicazione A deve trasferire 73200 byte all'applicazione B utilizzando il protocollo TCP. Si supponga che la connessione tra A e B sia già stata instaurata. La trasmissione dei segmenti inizia al tempo t=0. Sono noti i seguenti parametri:

- MSS concordata pari a 1200 byte;
- RCVWND annunciata da B ad A pari a 19200 byte; a partire dal tempo t_a>6.0 la destinazione annuncia una RCVWND pari a 4800 byte; a partire dal tempo t_b>11.0 la destinazione annuncia una RCVWND pari a 14400 byte;
- SSTHRESH iniziale = RCVWND;
- CWND= 1 segmento a t=0;
- RTT pari a 1.0 secondo, costante per tutto il tempo di trasferimento;
- RTO base = 2*RTT; nel caso di perdite consecutive dello stesso segmento, i timeout seguenti raddoppiano fino ad un massimo di 4 volte il RTO base, dopodiché la connessione viene abbattuta;
- il tempo di trasmissione dei segmenti è trascurabile rispetto RTT;
- il ricevitore riscontra immediatamente i segmenti.

Inoltre si supponga che la rete vada fuori servizio nel seguente intervallo di tempo:

- da t_1=6.5s a t_2=7.5s.

Si tracci l'andamento della CWND nel tempo e si determini in particolare:

1. il valore finale di CWND (sia graficamente, sia esplicitandolo);
2. i valori assunti dalla SSTHRESH durante il trasferimento (graficamente);
3. il tempo necessario per il trasferimento dei dati (sia graficamente, sia esplicitandolo);
4. il numero di segmenti trasmessi ad ogni intervallo, specificando se ne vengono ricevuti i riscontri o meno (sia graficamente, sia esplicitando i valori).

L = 73200 byte = 61 segmenti

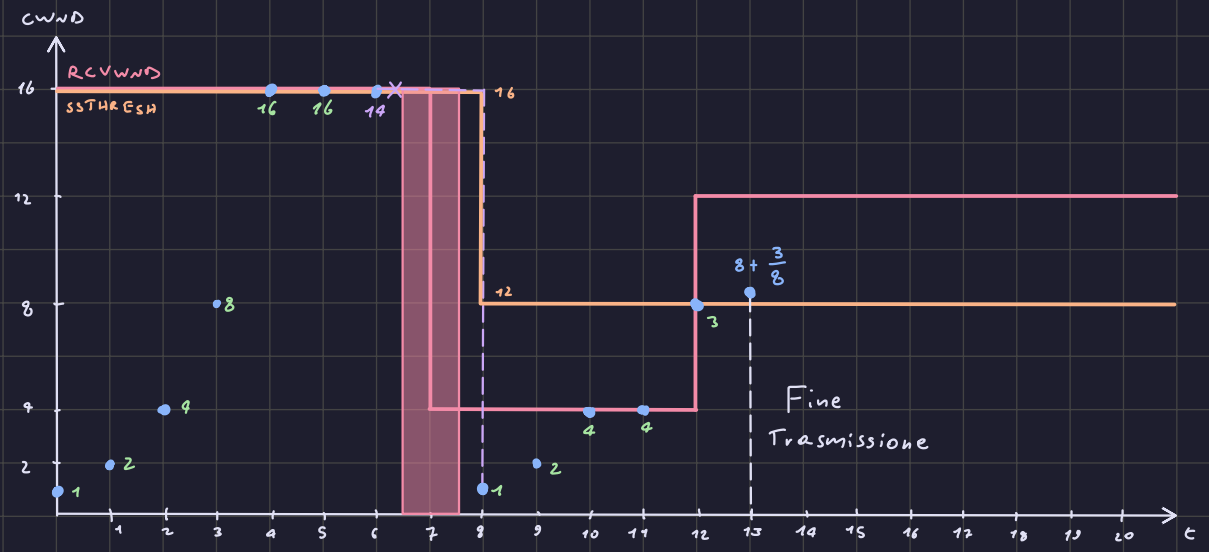
MSS = 1200 byte

SSTHRESH = RCVWND

RTT = 1s costante

$$RCVWND = \begin{cases} 19200 & t \leq 6 = 16 \text{ segmenti} \\ 4800 & t > 6 = 4 \text{ segmenti} \\ 14400 & t > 11 = 12 \text{ segmenti} \end{cases}$$

Down = (6.5, 7.5)



segmenti trasmessi = 1 + 2 + 4 + 8 + 16 + 16 + ~~14~~ 16 + ~~61~~ + 1 + 2 + 4 + 4 + 3 + 8

1 3 7 15 31 47 48 50 54 58 61

Il tempo necessario per trasmettere tutti segmenti è di 13 secondi e il valore finale della cwnd è 8+3/8 perchè a t = 12 mancano soltanto 3 segmenti da inviare per completare la trasmissione, quindi vengono ricevuti soltanto 3 riscontri a t = 13 e la cwnd viene impostata a cwnd = min(rcvwnd, cwnd_old + #ack/cwnd_old) = 8 + 3/8.