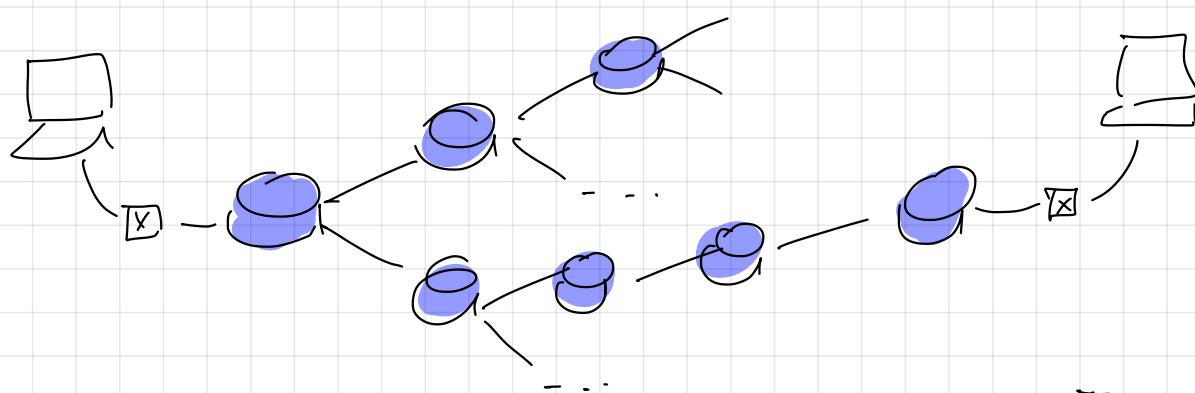


LEZIONE NEL 9/11/2021

FRAMMENTAZIONE IP (\neq SEGMENTAZIONE TRASPORTO)

MSS del livello di Transporto è posto su MTU del livello data link / fizico

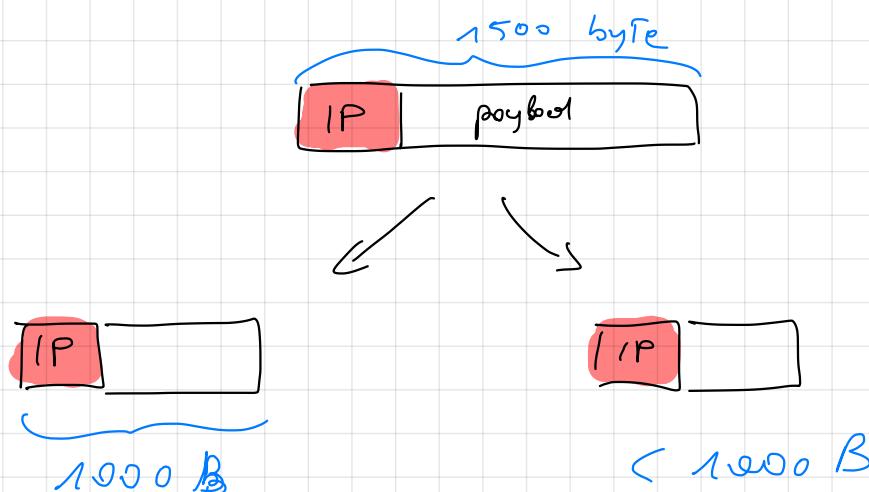
Che cosa succede se nel pacchetto che ha MTU più grande rispetto a quello degli host che lo stanno scambiando le informazioni?



Ogni tecnologia ha uno MTU diverso

Ethernet \rightarrow 1500B
SDH \rightarrow 300B

Il pacchetto viene frammentato in pacchetti con MTU corrispondente ai link di uscita



I campi dell'header comuni nella frammentazione sono:

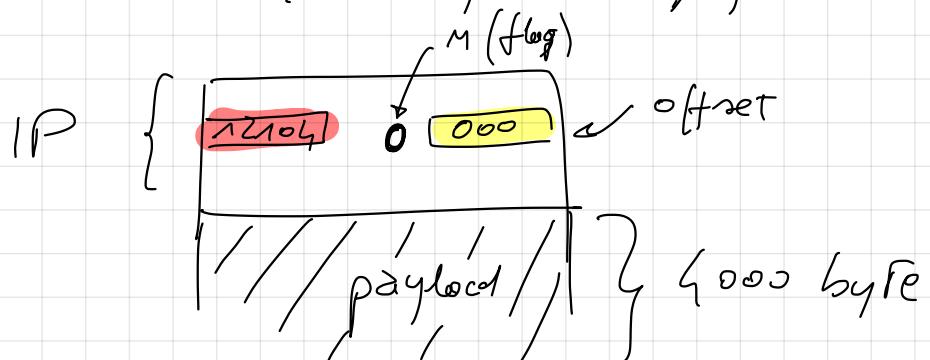
Identification → # mappare dato della sovrapposizione
nel spazio pacchetti

Offset → posizione del frammento rispetto
al pacchetto originario (diviso x P)

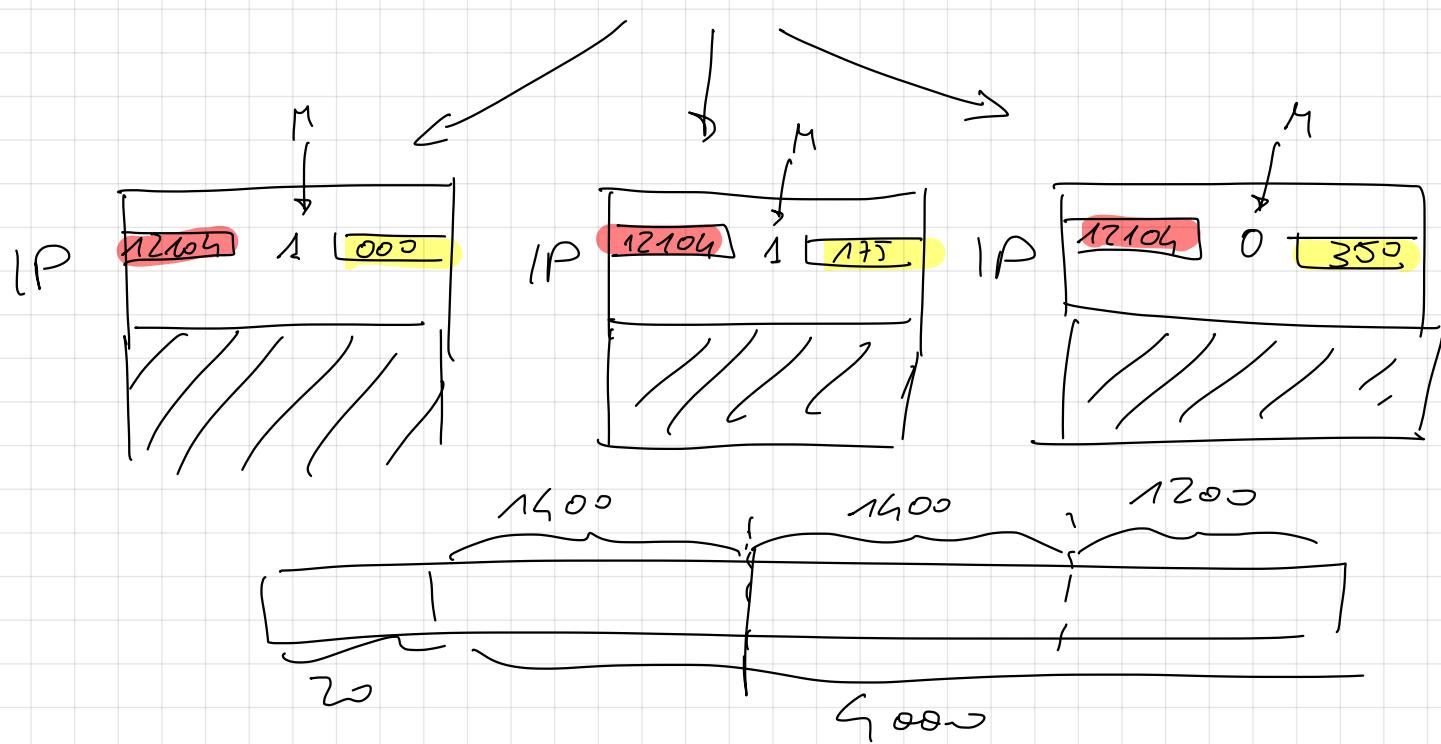
Esempio:

MTU 5000 byte

pacchetti con 4000 byte di payload e 20 byte IP



Alla parola → MTU con 1420 byte → 1600 byte payload
20 byte IP



$$\begin{array}{l} \text{Offset II}^{\circ} \text{ frammento } 1400 / 8 = 175 \\ \text{..} \quad \text{III}^{\circ} \quad \text{..} \quad 2800 / 8 = 350 \end{array}$$

i flag (3 bit) DM

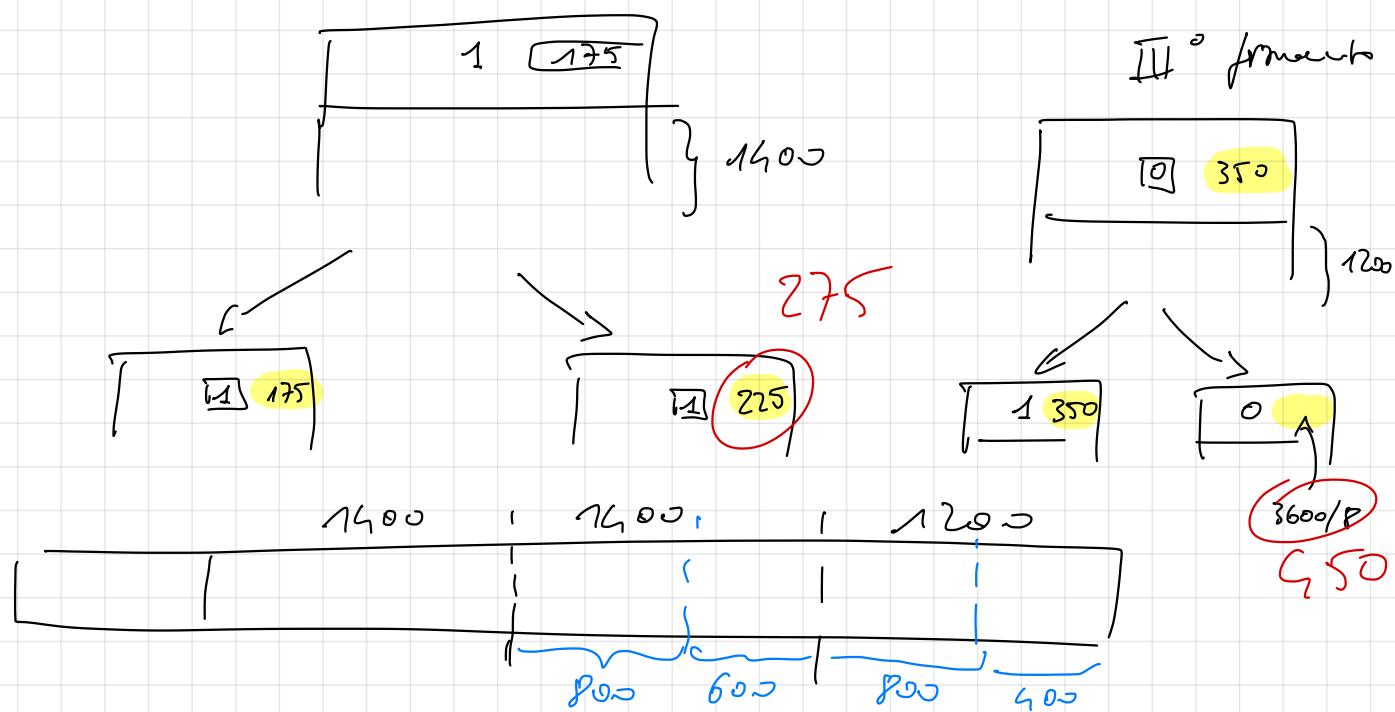
D \rightarrow Do not fragment

M \rightarrow More fragment

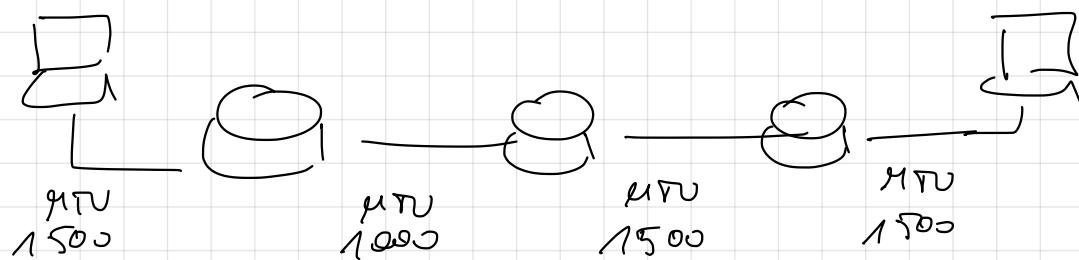
\emptyset perché
non è frammento,
oppure è l'ultimo
frammento

1
fragment
intermedio

Esempio \rightarrow il II frammento passa attraverso router
con MTU 820 byte



? Chi riassembra i frammenti?



Il risucchio avviene sempre sull'host di destinazione

? Cosa accade se un frammento non perde?

Quando l'end host (destinazione) riceve il minimo frammento

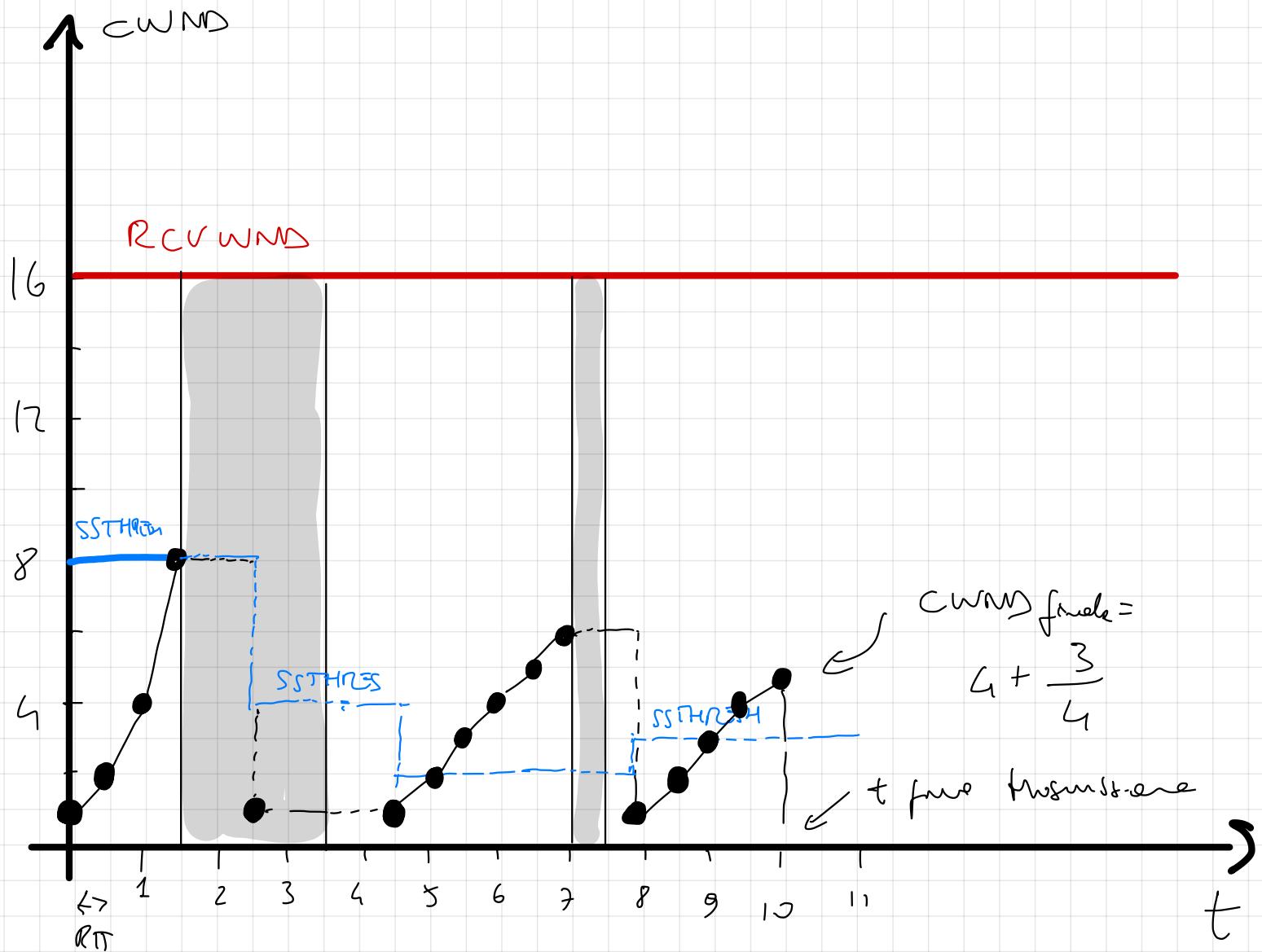
fa partire un timer (200 - 500 ms)

Se non ricevo tutti i frammenti entro lo scadere del timer, scarto tutto

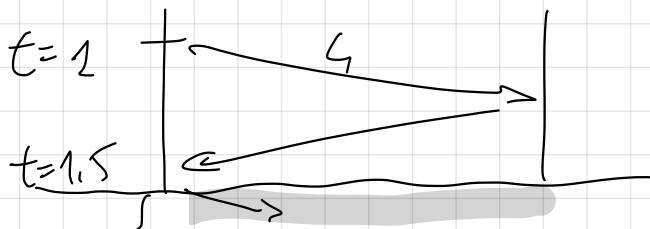
Soluzione iterativo TCP

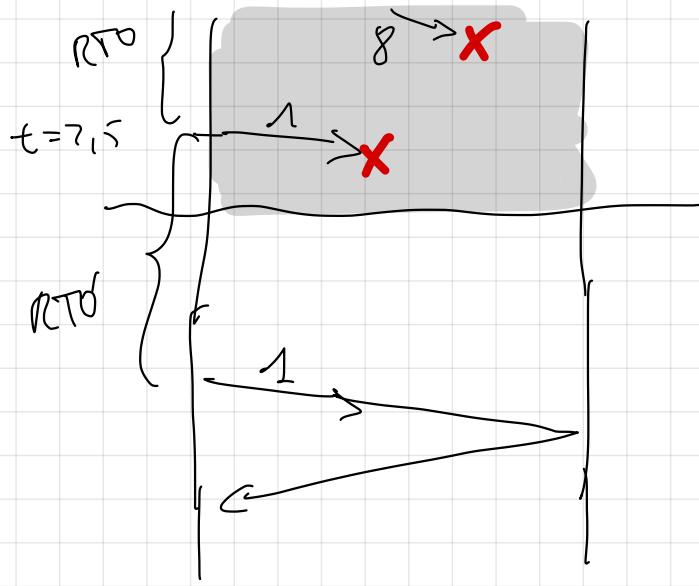
eff segmenti da trasmettere $\rightarrow 31$

$$RCVWND_{init} = 16 \text{ segm.} \quad SSTHRESH_{init} = P \text{ segm}$$

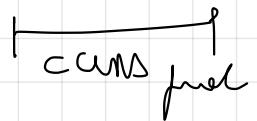
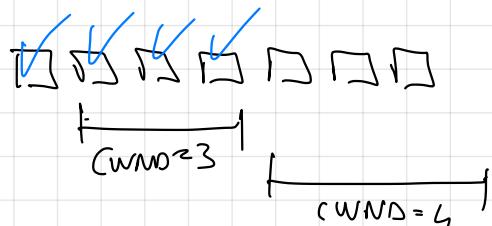


$$1 + 2 + 4 + 8 + 1 + 1 + 2 + 3 + 4 + 5 + 6 + 1 + 2 + 3 + \underbrace{\overline{4}}_{(28)}$$





$$SSTHRESH_{new} = \max\left(\frac{CWND}{2}, 2\right)$$



Ejercicio TCP (3)

Aplicación A \rightarrow B 104 000 Byte

MSS = 1200 Byte

RCVWND_{initial} = 24 000 Byte SSTHRESH_{initial} = RCVWND_{initial}

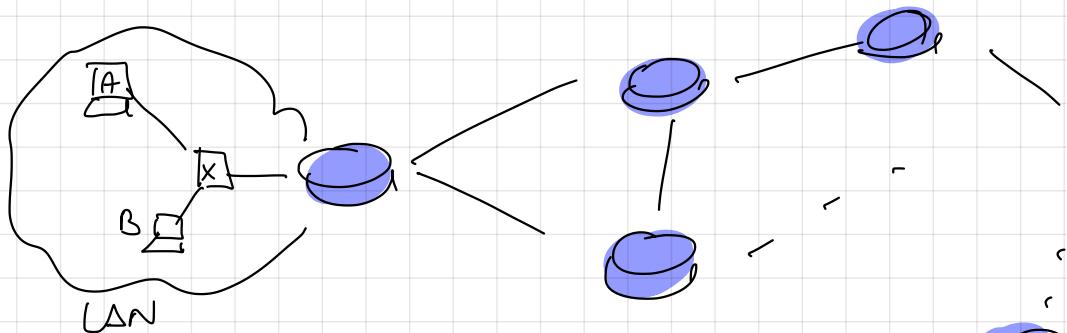
RTT = 0,5 constante, RTO = 2RTT, retrasos finos per ritrasmissione

Retrasos finos: 3,5 \rightarrow 4
6,5 \rightarrow 10,5

LEZIONE DEL 10/11/2021

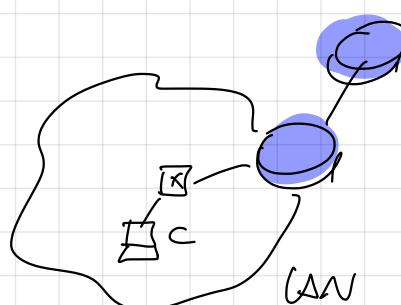
ROUTING / INSTRUMENTO

Consegna diretta / indiretta



A e B appartengono alla stessa rete

→ devono avere lo stesso prefisso



Se A vuole inviare un pacchetto all'host B

→ CONSEGNA diretta

L'host A controlla l'indirizzo di destinazione, controlla la propria maschera, confronta i bit del ^{no} prefisso e i bit del prefisso di B → se sono uguali → allora A e B appartengono alla stessa rete

Se sono diversi (ad esempio, host C) → CONSEGNA indiretta
Il pacchetto viene consegnato al router

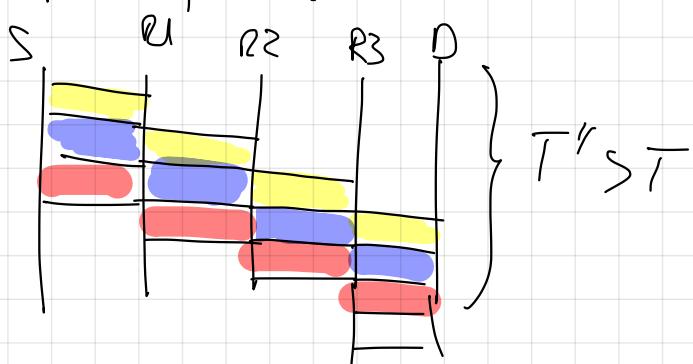
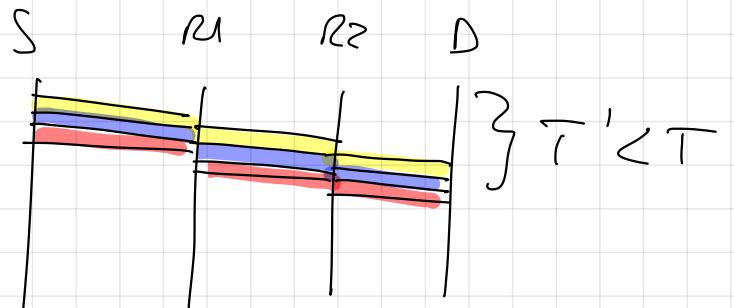
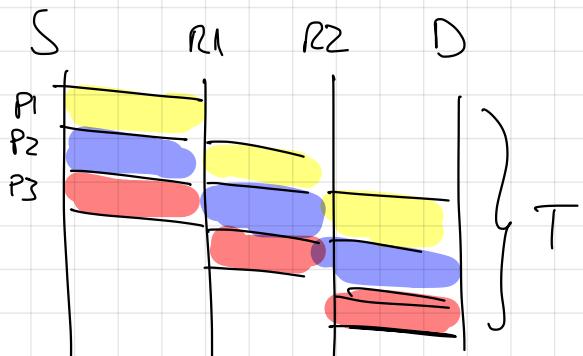
? Come fa il router a sapere come seguire le destinazioni?

ROUTING → processo di scelta del cammino "migliore"
Se uno sceglie o tutte le possibili destinazioni

"migliore" → esistono diversi criteri

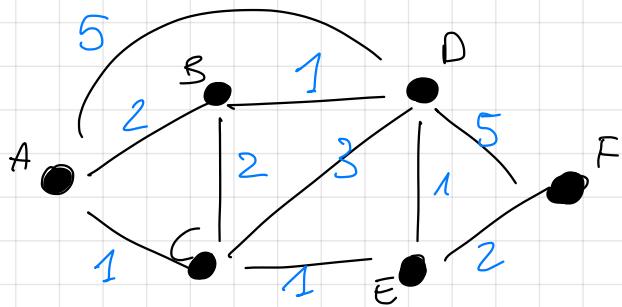
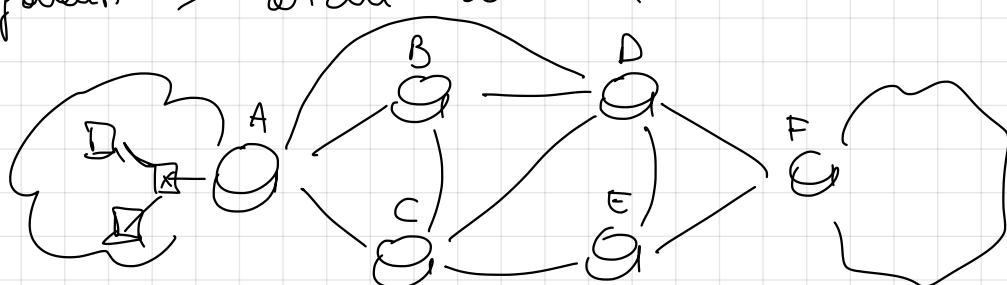
Distanza (cammino più corto)
 ? distanza in framme di? \Rightarrow come
 # di frame
 ottimale

D velocità di trasmissione



→ ASTRAZIONE CON GRAFI

Traffico → nodi di un grafo
 collegamenti → archi del " "



A più archi posso trovare un percorso che caratterizza l'arco

Esempio → Ho mi definisco minima, ha il # di hop

- peso è = 1 per tutti gli ordini
- se mi interessa la velocità di trasmissione
- peso è proporzionale all'area della banda
(se la banda è grande → peso è piccolo
→ comunque minimo è il comm.
con i pesi piccoli → banda maggiore)

Dato un prof (topologia, nodi e ordini)

Dato un criterio (\rightarrow associazione di pesi agli ordini)

Routing \rightarrow problema del calcolo del cammino minimo non da prof

Definisco costo di un zw / non pesi

Costo dell'arco del nodo i al nodo j $\rightarrow c(i,j) = w_i$

Costo di un cammino è la somma dei costi degli ordini del cammino quel cammino

Costo del cammino da nodo i al nodo k

$$\rightarrow D(i, k) = \sum_{\substack{l, m \in \\ \text{cammino}}} c(l, m)$$

Esistono 2 categorie di algoritmi di routing

\rightarrow Vettore Distanza (Distance vector)

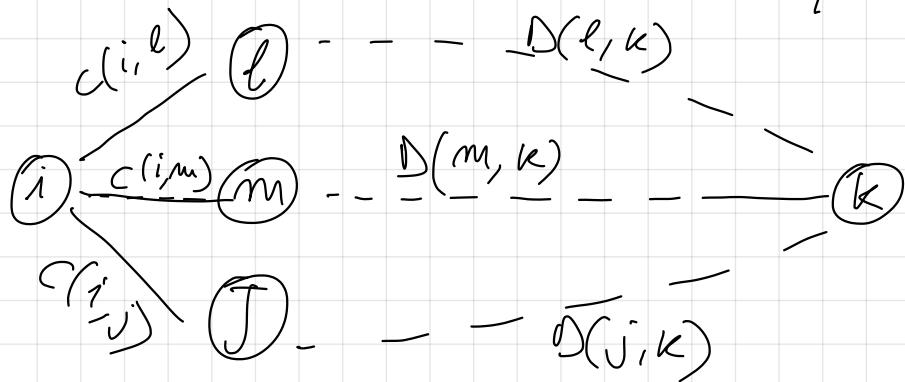
\rightarrow Stato dei collegamenti (Link state)

(esiste una 3^a categoria \rightarrow datapath \rightarrow SDN)

Algoritmi distance vector

\rightarrow Criterio di convergenza: uno percorso di cammino

minimo è sede il cammino minimo fra i nodi dei determinanti sole percorso



Se $D(i, k)$ è minimo ponendo allo m , allora lo $D(m, k)$ è minimo

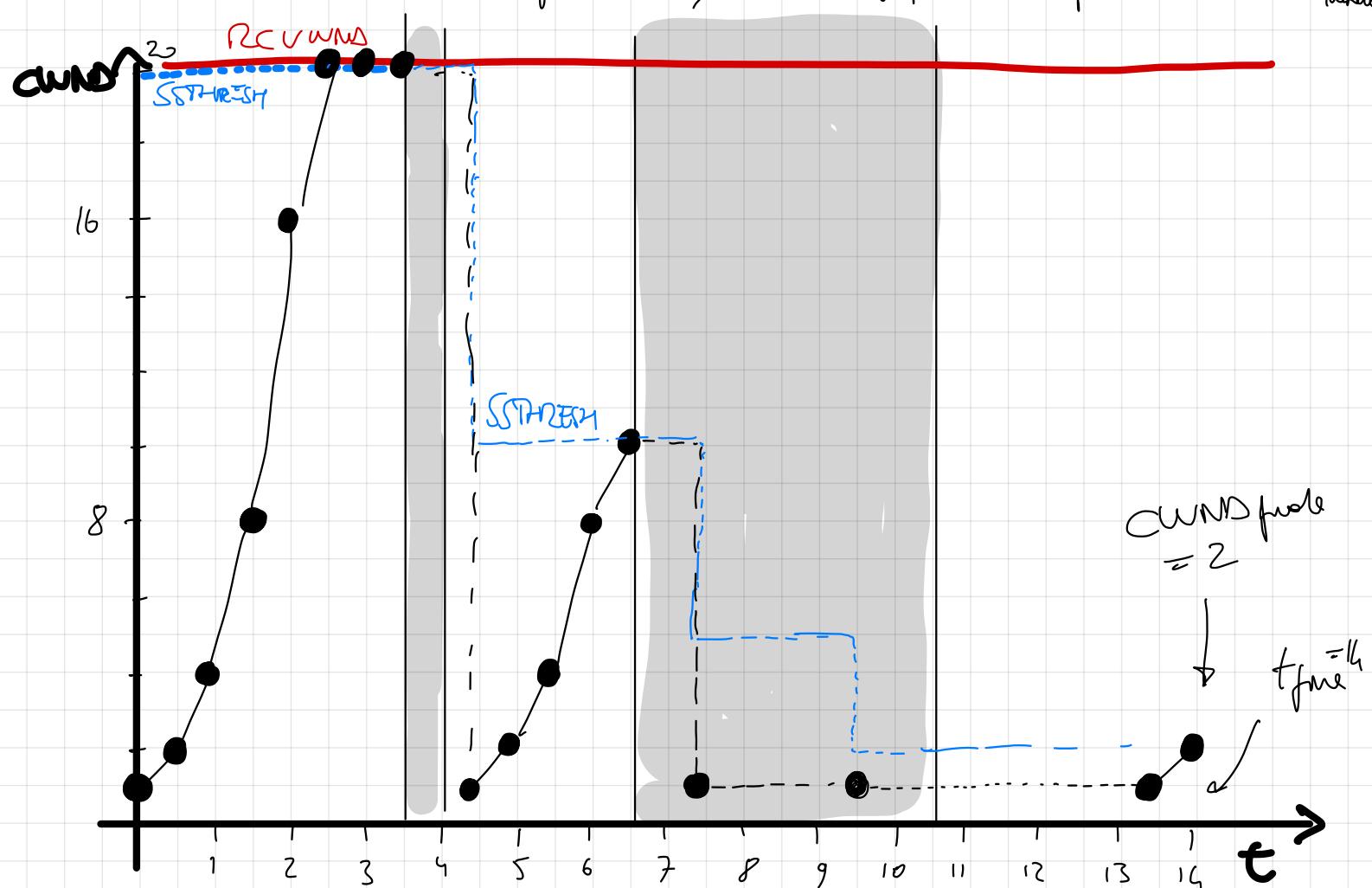
il modo di calcolare le distanze verso il nodo k
considerando tutti i minimi

$$D(i, k) = \left\{ \begin{array}{l} c(i, l) + D(l, k) \\ c(i, m) + D(m, k) \\ c(i, j) + D(j, k) \end{array} \right\}_{\text{min}}$$

$$D(i, k) = \min_{v \in \text{viani di } i} \left(c(i, v) + D(v, k) \right)$$

Solutions ES TCP

do i diversi 8+ segnali, $RCVWND_{out} = 20$ segnali = $SSTHRESH_{middle}$



$$1 + 2 + 4 + 8 + (6 + 20 + 20 + \cancel{16}) + 1 + 2 + 4 + 8 + \cancel{16}$$

$\frac{(71)}{20} + \frac{(86)}{10}$

~~1~~ + ~~1~~ + 1

TE 26/09/2019, es. 3

A applicat. A do devo trasferire 104 400 byte \rightarrow B

$$\mu SS = 1200$$

$RCVWND_{middle} = 9600$ byte

A partire dal tempo $t_2 > 4.0$ la destinazione annuncia una $RCVWND = 16.400$ byte

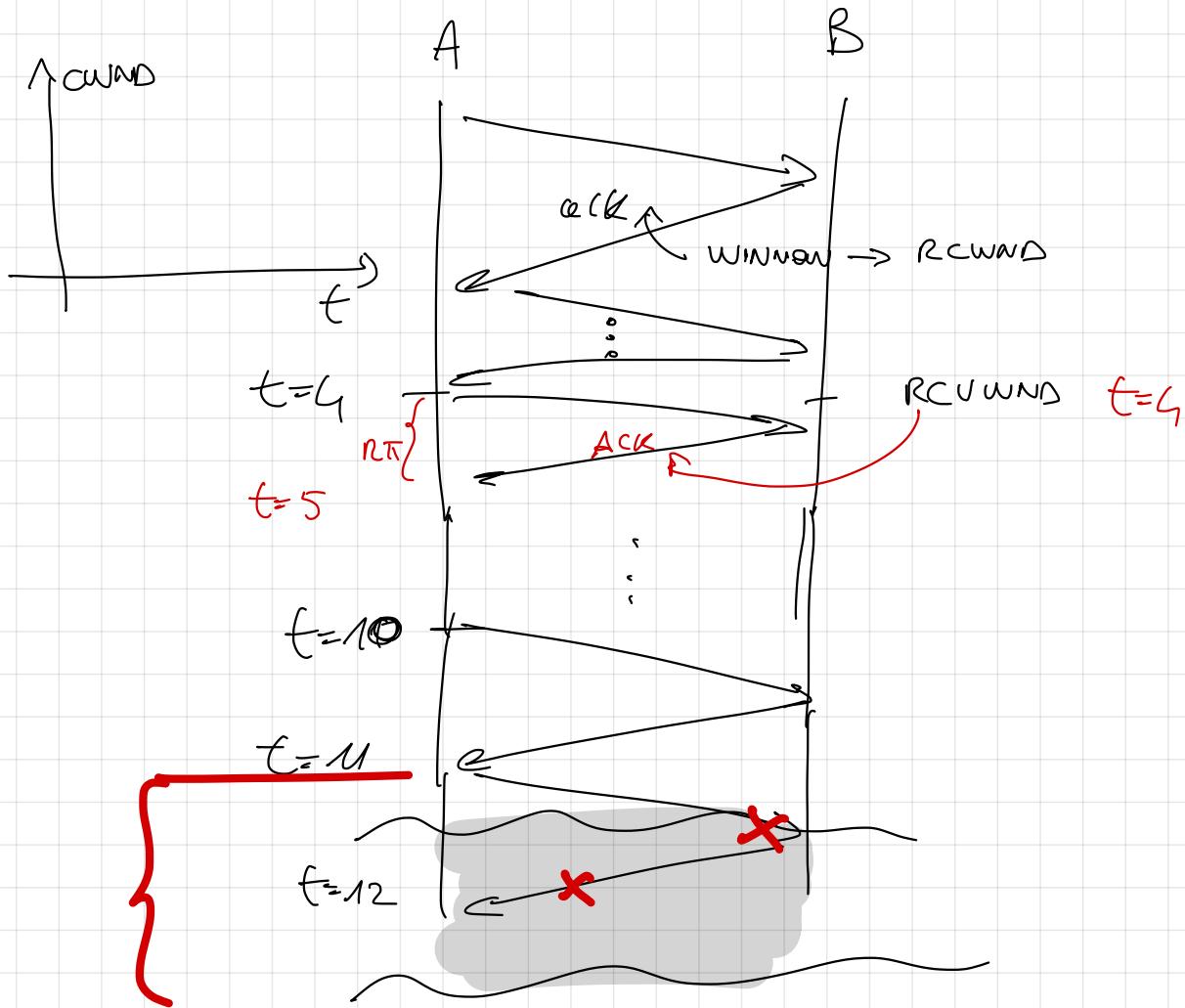
A partire dal tempo $t_5 > 9.0$ la destinazione annuncia una $RCVWND = 7200$ byte

$$SSTHRESH_{\text{node}} = RCVWNDS_{\text{node}}$$

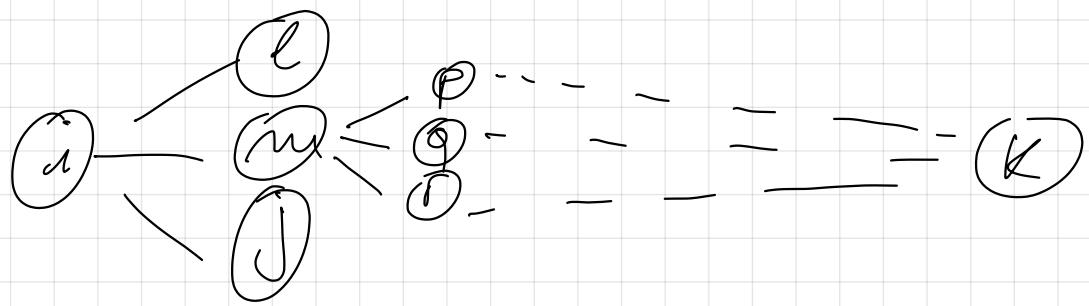
$$CWND_{t=0} = 1$$

$RTT = 1 \text{ sec}$, constant per file if files

Ref. from up 11, 5 \rightarrow 12, 5



LEZIONE DELL' 11/11/2021



$$D(i, k) = \min_{\substack{\text{st} \in \text{vicini} \\ d(i)}} (c(i, st) + D(st, k))$$

$$D(m, k) = \min_{\substack{\text{st} \in \text{vicini} \\ d(m)}} (c(m, st) + D(st, k))$$

Algoritmo basato su DV (distance vector)

→ tabella di routing → il destinazione, il next hop
e il costo per raggiungere tale destinazione

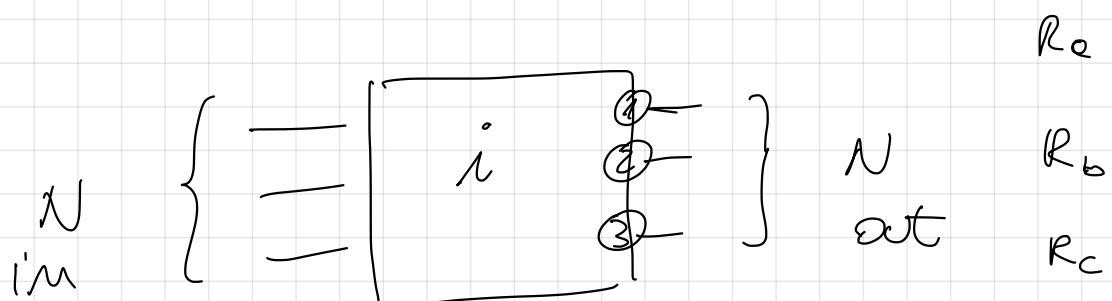
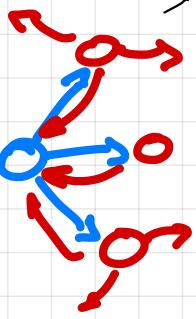


Tabella di Routing

dest	next hop	costo
K	R _a	D(i, k)
W	R _c	D(i, w)

Inizializzazione

- se destinazione, se si trova direttamente allora $D(i, v) = c(i, v)$
altrimenti $D(i, v) = \infty$



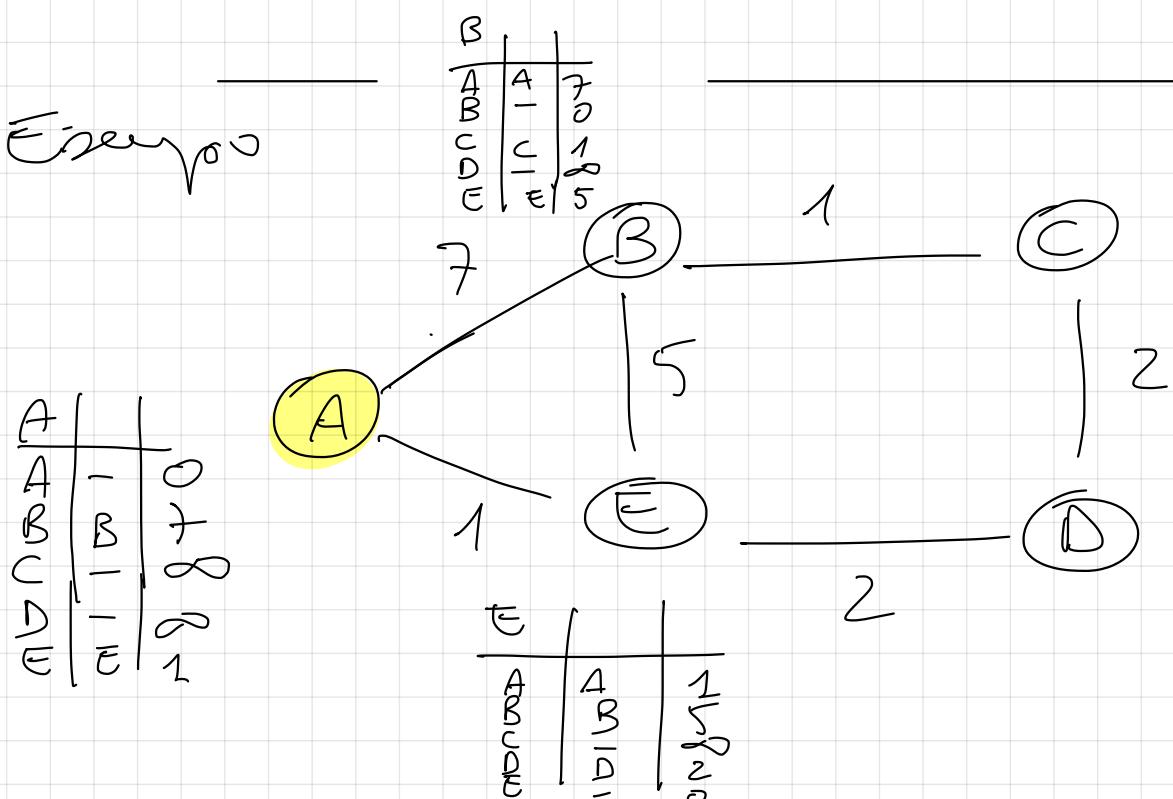
Periodi concreti (ad es. ogni 3 minuti)

- il modo mento si muovi vicini il più DV
- con i DV vicini dei vicini, aggiorno la mia tabella di routing

$$D(i, K) = \min_{\forall k} \left(c(i, k) + D(k, K) \right)$$

\neq vicini
diretti di i

Sì può dimostrare che, se il graf è completamente connesso e i pesi sono > 0, l'algoritmo DV converge alla soluzione ottima
(# iterazioni \leq dimensione del grafo)



Il modo A ricava dei valori vicini i DV

da B ricava

	B
A	7
B	0
C	1
D	∞
E	5

e che E ricava

	E
A	4
B	5
C	∞
D	2
E	0

da A e B

$$D(A, B) = C(A, \frac{B}{E}) + D(\frac{B}{E}, B)$$

$$\begin{aligned} \rightarrow B & \quad 7 & + & \quad 0 & = 7 \\ \rightarrow E & \quad 1 & + & \quad 5 & = 6 \end{aligned}$$

quindi A efforza la riga congiunta a B con il valore 6 e next hop E

da A e C

$$D(A, C) = C(A, \frac{C}{E}) + D(\frac{C}{E}, C)$$

$$\begin{aligned} \rightarrow B & \quad 7 & + & \quad 1 & = 8 \\ \rightarrow E & \quad 1 & + & \quad \infty & = \infty \end{aligned}$$

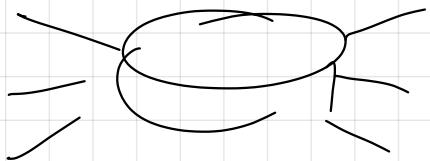
A efforza la riga congiunta alle dest C con il valore 8 e next hop B

Algoritmi LNK STRE

→ algoritmo centrizzato

Ogni nodo ha a disposizione l'intero topologio della rete (grafico con i pezzi agli ordini) e calcola i cammini minimi automaticamente

→ algoritmo DIJKSTRA



Calcolo i costi minimi

distane

→ popola la tabella di
Dijkstra

dist	next hop	costo