TUTORATO!

TUTORATO IRS
25/11/2021

COMINCIAMO CON ESERCIZI RELATIVI AL SUBNETTING

- Riportare in binario e decimale l'indirizzo di rete a cui appartiene l'host

131.175.23.1/22

→ 10000011.10101111.00010111.00000001
MASK 11111111.11111111.11111100.00000000

131. 175 . 20 . 0 /22

che sarebbe: 10000011.10101111.00010100.00000000
QUESTO È L'INDIRIZZO CHE IDENTIFICA LA RETE A CUI APPARTIENE 131.175.23.1/22

basta mettere a 0 tutto ciò che è a destra di questa linea rossa - è la linea che ho tracciato tra gli 0 e 1 della subnet mask. Alla sinistra, ciò che c'è nella prima riga rimane invariato.

- Data una LAN con un solo ROUTER e data la network 192.168.0.0/24 si vuole partizionare la LAN in sottoreti per avere un numero massimo di host per sottorete pari a 62, utilizzando un'appropriata netmask.

i) calcolare il NUMERO MASSIMO DI SOTTORETI possibili;

ii) calcolare indirizzi di BROADCAST e RETE di TUTTE le nuove sottoreti, inoltre mostrare gli indirizzi in forma binaria.

i) allora, premessa: ricordare sempre che quando dobbiamo fare una subnet con N host, al numero di indirizzi che devo riservargli ne devo aggiungere sempre 2 in più: BROADCAST E RETE → 62+1+1 = 64!

mask originale: 11111111.11111111.11111111.00000000 (/24)

(256)10

256/64 indirizzi = 4 subnet!

la nuova mask sarà una /26, riserverà gli ultimi 6 bit agli indirizzi e 7°-8° bit dell'ultimo byte rappresentano le subnet (00, 01, 10, 11) → uso gli indirizzi per SN, finiti quelli cambio SN.

Rete 1: R: 192.168.0.0 (!= quella a mask 24) | hosts: 192.168.0.1 ÷ 192.168.0.62 | Broadcast: 192.168.0.63] mask /26!

Rete 2: R: 192.168.0.64 | host: 192.168.0.65 ÷ 126* | BC: 192.168.0.127

this answers question 2.

Rete 3: R: 192.168.0.128 | host: 192.168.0.129 ÷ 190 | BC: 192.168.0.191] /26

Rete 4: R: 192.168.0.192 | host: 192.168.0.193 ÷ 254 | BC: 192.168.0.255] /26

*specificate SEMPRE L'IP COMPLETO, non così.

per il binario divertitevi

- DIRE SE QUESTI INDIRIZZI INDIVIDUANO UNA RETE O UN HOST.

192.168.72.0/18 → host! perché:

L'host in binario è

11000000.10101000.01001000.00000000

La mask a barra 18 è:

111111111.11111111.11000000.00000000

Il fatto che abbia almeno un "1" alla destra della famosa barra rossa ci basta per dire che non identifica una rete. Lo stesso indirizzo ma con /21 sarebbe stato una rete.

- Sia data una rete con 31 host. Determinare la netmask minima.

Occhio a questo tipo di domanda, qui ci viene dato il nº HOST.
Potremmo pensare "beh basta una netmask a /27, che lascia spazio a 32 indirizzi quando a noi ne servono 31... EXCEPT! che a noi 31 indirizzi NON bastano! mettete in conto anche RETE e BROADCAST e avremo 33 indirizzi necessari. 31 sono solo gli host! come da consegna.
Quindi scattiamo a 64, 32 non bastano → la SN minima è a mask /26.

- Dato l'indirizzo e la netmask, questi indirizzi sono di rete? Y/N.

- 192.168.5.0/24 ____
- 192.168.4.23/24 ____
- 192.168.2.36/30 ____
- 192.168.2.36/29 ____
- 192.168.2.32/28 ____
- 192.168.3.32/27 ____
- 192.168.3.0/23 ____
- 192.168.2.0/31 ____
- 192.168.2.0/23 ____
- 192.168.16.0/21 ____
- 192.168.2.0/21 ____

Soluzione premio e antefatto

1. • 192.168.5.0/24 — SI
2. • 192.168.4.23/24 — NO
3. • 192.168.2.36/30 — SI
4. • 192.168.2.36/29 — NO
5. • 192.168.2.32/28 — SI
6. • 192.168.3.32/27 — SI
7. • 192.168.3.0/23 — SI
8. • 192.168.2.0/31 — SI
9. • 192.168.2.0/23 — SI
10. • 192.168.16.0/21 — SI
11. • 192.168.2.0/4 — NO

perché:

i) 192 . 168 . 5 . 0
11000000.10101000.00000101.00000000 - IP address
11111111.11111111.11111111.00000000 - mask

1) 192 . 168 . 4 . 23
11000000.10101000.00000100.00010111 - IP address
11111111.11111111.11111111.00000000 - mask

3) 192 . 168 . 2 . 36
11000000.10101000.00000010.00100100 - IP address
11111111.11111111.11111111.11111100 - mask

4) 192 . 168 . 2 . 36
11000000.10101000.00000010.00100100 - IP address
11111111.11111111.11111111.11111000 - mask

...Etc. Honorable mention l'indirizzo 8, che ha /31: è una rete che può indirizzare soltanto Rete e broadcast, quindi non ha granché senso d'esistere. ma sì, è valido per ind. di rete.

TUTORATO IReS 02/12/2021

- Sia dato l'IP 103.2.10.64 e la maschera di sottorete 255.255.255.224 (/27)

  Indicare l'indirizzo di Network.

add: 01100111.00000010.00001010.01000000 →

mask: 11111111.11111111.11111111.11100000

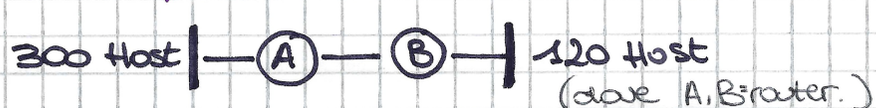
→ Network add: 103.2.10.64

- ora qui non sapendo il #host, tirerei il broadcast che sarà 01100111.00000010.00001010.01011111, io credo, ovvero 103.2.10.(31+64) → 103.2.10.95!
- Riguardo gli host, sono compresi fra 103.2.10.65 e 103.2.10.94

  103.2.10.65 ÷ 103.2.10.94 = 30 Host.

II) ESERCIZI RELATIVI al NETWORKING - PROGETTARE LA RETE

- Supponendo che la rete abbia indirizzo 192.168.0.0/22, stabilisci se ci sono abbastanza indirizzi per una network a /22;
  - a) se sì, assegnali; ↓ così fatta



(dove A, B router.)

Qui se vedo bene abbiamo bisogno di indirizzi per:
300 + 120 + 1 + 1 host = 422.

Una mask a /22 significa che abbiamo 32 - 22 = 10 bit a disposizione;
quindi 2^{10} = 1024 indirizzi disponibili [non ci stiamo larghissimi]
Appurato che sì, sono abbastanza, li assegnamo:

- a) Ora qui dobbiamo considerare le reti da 300 e da 120 H come subnet distinte come da schema

  I° subnet: 300 H (+1 del nodo A!) = 301 H

  Rete: 192.168.0.0/23

  H: 192.168.0.1 ÷ 254

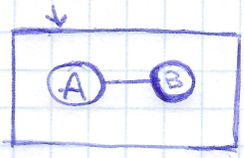
  Broadcast: 192.168.0.255

FAQ: perché a /23?
Usare una mask a /23 ci lascia liberi 9 bit per host. Possiamo assegnare 2^9 = 512 host. Ce ne servono 301, perché riservarne 1024 quando ne bastano 512? :)
Dice l'RFC 1925 al punto 12: "PERFECTION HAS BEEN REACHED NOT WHEN THERE IS NOTHING LEFT TO ADD, BUT WHEN THERE IS NOTHING LEFT TO TAKE AWAY."

II^a subnet: 120 Hosts + nodo B = 121

- Rete: 192.168.2.0 /25
- H: 192.168.2.1 ÷ 192.168.2.254 /25
- Broadcast: 192.168.2.255 /25

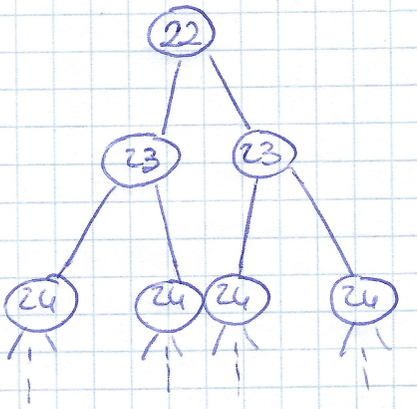
- C'è una 3^a subnet, ed è quella che comprende nodo A e B. ma solo 2 host, appunto nodo A e nodo B



- Rete: 192.168.2.128 /30 ← per 2 host basta
- H: 192.168.2.129 ÷ 192.168.2.130
- Broadcast: 192.168.2.131

TUTORATO IRES 07 DICEMBRE 2021

Con una Rete a /22 possiamo fare 2 reti /23, o 4 reti a /24. o 1 a /23 e 2 a /24 etc. La progressione segue l'andamento di un ALBERO BINARIO, tipo:



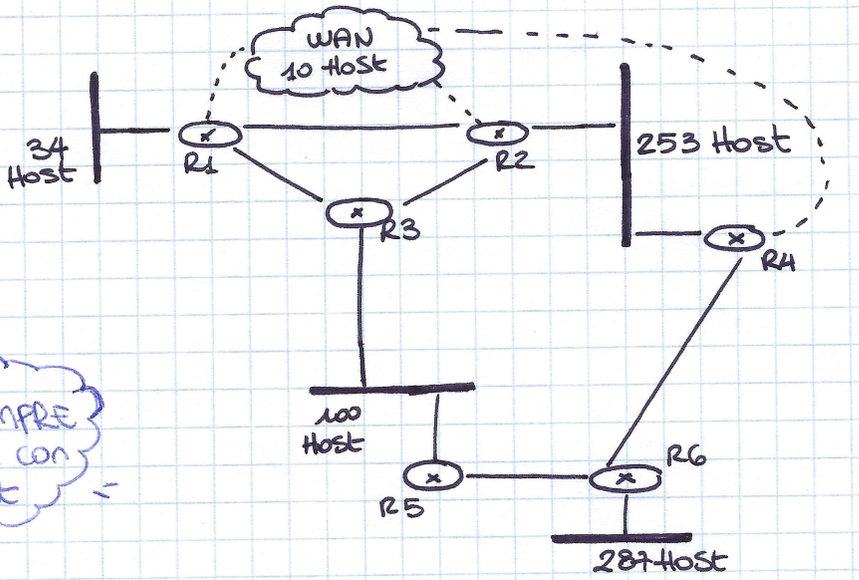
così via fino alla Barra 30.

ad esempio, se in questo contesto usassimo uno dei /23 "figli" di /22, ci rimarrebbero disponibili soltanto i nodi figli dell'altro /23 (lui compreso).

Crearsi un albero delle maschere del genere è utile quando si fa subnetting a mask variabile, io credo.

Detto ciò:

1.) Si consideri la Rete 192.168.0.0/21 in "figura": determinare ed assegnare IP agli host delle sottoreti - tutti adiacenti fra loro.



partiamo SEMPRE dalla Rete con PIÙ host

LAN 1: 287 Host + R6 → 288 → mask 23

indirizzi:

R: 192.168.0.0

H: 192.168.0.1 ÷ 192.168.1.254

B: 192.168.1.255

] /23 naturalmente

LAN2: 253 Host + R2, R4 = 255 Host → /23

indirizzi:

R: 192.168.2.0

H: 192.168.2.1 ÷ 192.168.3.254

B: 192.168.3.255

LAN3: 100 Host + R3, R5 = 102 → /25 (128 host)

indirizzi:

R: 192.168.4.0

H: 192.168.4.1 ÷ 192.168.4.126

B: 192.168.4.127

LAN4: 34 Host + R1 = 35 → /26

indirizzi:

R: 192.168.4.128

H: 192.168.4.129 ÷ 192.168.4.190

B: 192.168.4.191

LAN5 (WAN) 10 Host + R1, R2, R4 = 13 Host → /28 (16 fino a host)

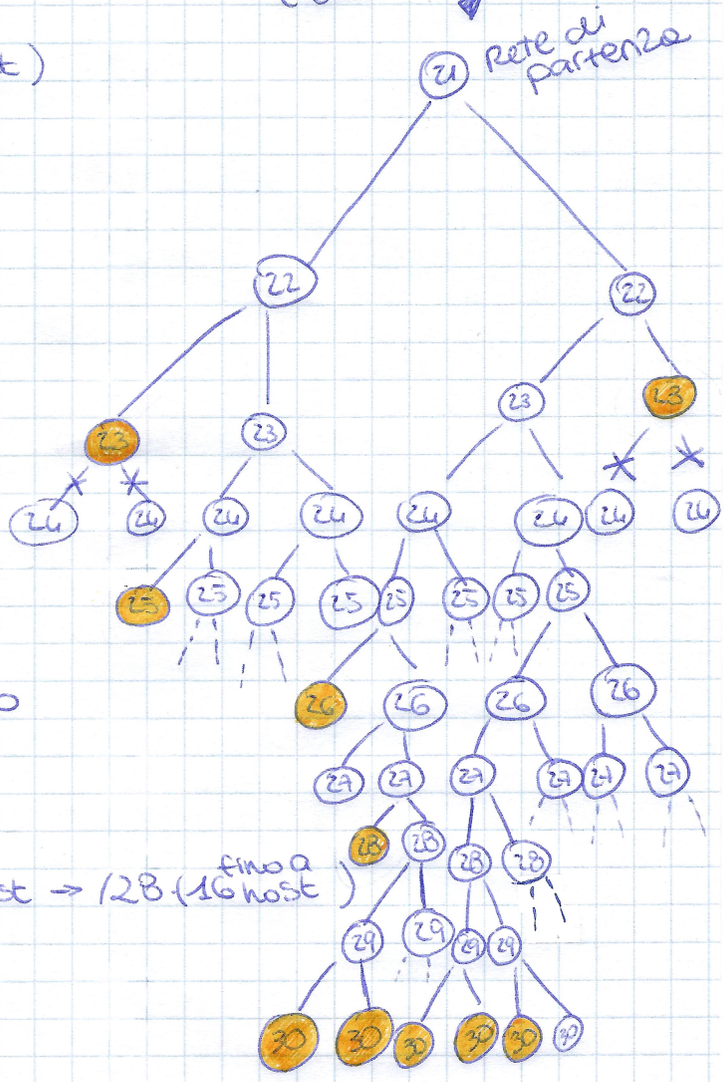
indirizzi:

R: 192.168.4.192

H: 192.168.4.193 ÷ 192.168.4.206

B: 192.168.4.207

TIPO in questo esercizio abbiamo occupato i nodi dell'albero così:



Mancano 5 subnet: i collegamenti tra Router (SSSI, sono subnet. 2 Host).

LAN6: Rete: 192.168.4.208
Host: 192.168.4.209, 192.168.4.210
Broadcast: 192.168.4.211

LAN8: Rete: 192.168.4.216
Host: 192.168.4.217, 218
Broadcast: 192.168.4.219

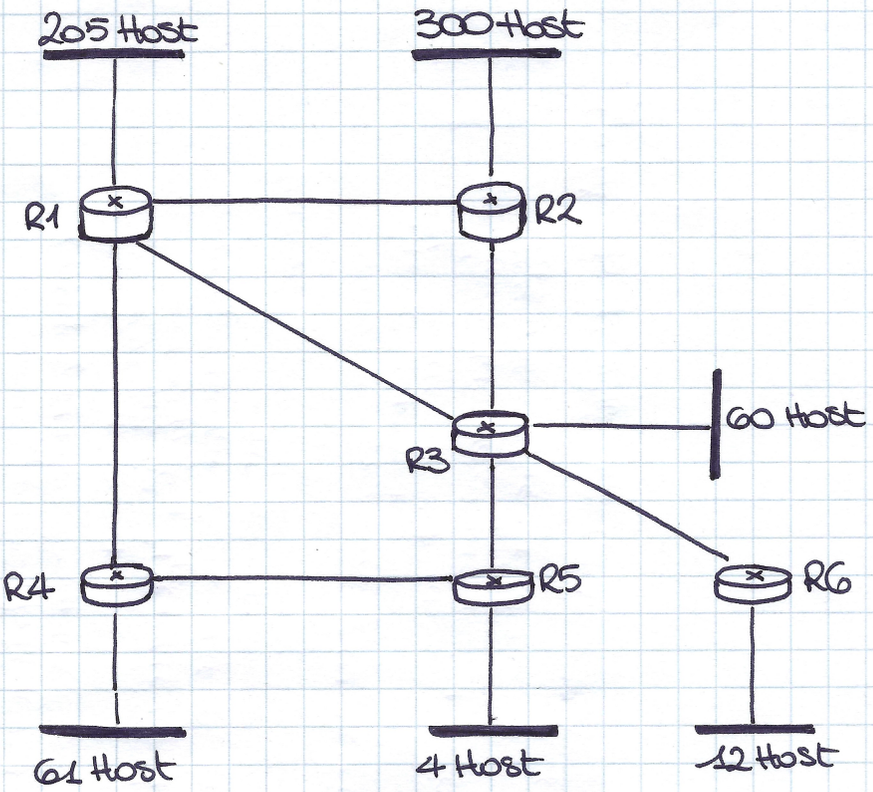
TUTTE /30

LAN7: Rete: 192.168.4.212
Host: 192.168.4.213, 192.168.4.214
Broadcast: 192.168.4.215

LAN9: Rete: 192.168.4.220
Host: 192.168.4.221, 222
Broadcast: 192.168.4.223

LAN10 → R: 192.168.4.224
H: 192.168.4.225, 192.168.4.226
B: 192.168.4.227

11) Data questa Rete e l'indirizzo di partenza 192.168.0.0/22



Realizzare un piano di indirizzamento a mask VARIABILE, stesse condizioni della consegna precedente, host contigui + tenete in conto che NON sono previste espansioni delle reti.

→ Quindi se abbiamo 63 Host, ad esempio, non serve usare /24 e riservarne 128 "in caso", bastano 64 (/25)! Questo implicava.

LAN1: 300H + R_2 → 301 indirizzi → 512 da riservare → /23

Rete: 192.168.0.0
H: 192.168.0.1 ÷ 192.168.1.254
BC: 192.168.1.255
} /23

LAN2: 205H + R_1 = 206 indirizzi → 256 → /24

Rete: 192.168.2.0
H: 192.168.2.1 ÷ 192.168.2.254
BC: 192.168.2.255
} /24

LAN3: 61H + R_4 = 62 → 64 → /26 (Ricordiamo, no espansione)

Rete: 192.168.3.0
H: 192.168.3.1 ÷ 192.168.3.62
BC: 192.168.3.63
} /26

LAN4: 60H + R3 = 61 → 64 → /26

Rete: 192.168.3.64
H: 192.168.3.65 ÷ 192.168.3.126 } /26
BC: 192.168.3.127

LAN5: 12H + R6 = 13 → 16 → /28

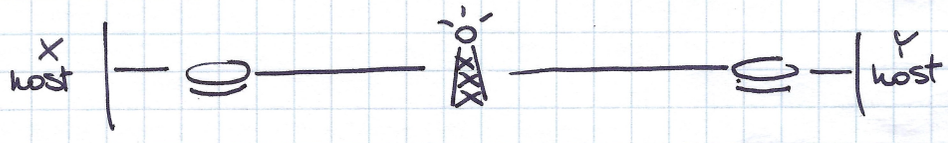
Rete: 192.168.3.128
H: 192.168.3.129 ÷ 192.168.3.142 } /28
BC: 192.168.3.143

LAN6: 4H + R5 = 5 → 8 → /29

Rete: 192.168.3.144
H: 192.168.3.145 ÷ 192.168.3.150 } /29
BC: 192.168.3.151

Ci sono 7 coll. tra router → 7 Reti a /30, che all'esame dovremo scrivere una per una, come al punto 1).

NOTA: Se ci trovassimo con un FONTE RADIO, ovvero

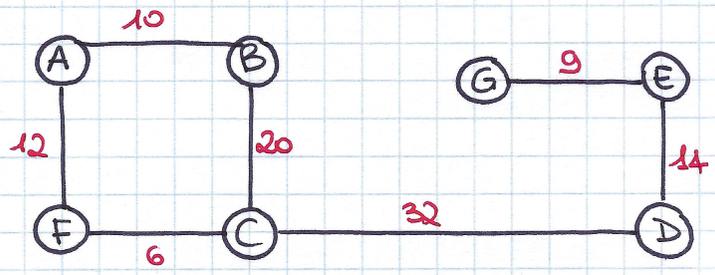


queste non sono da considerare reti separate, ma un'unica rete estesa con X+Y host

Lezione 4 9 Dicembre 2021

Network discovery - Link State / DistanceVector (RIP)

1) RIP



Qui per ogni nodo ci interessa quanti possibili cammini possiamo percorrere verso il resto della rete, interrogando via via i nodi a N+1 hop di distanza (1..2..3..)

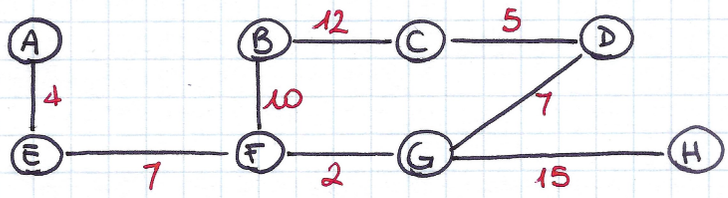
A	B	C	D	E	F	G
10, 1 12, 1	10, 1 20, 1	6, 1 20, 1 32, 1	32, 1 14, 1	9, 1 14, 1	6, 1 12, 1	9, 1
6, 2, F 20, 2, B	6, 2, C 32, 2, C	12, 2, F 10, 2, B 14, 2, D	6, 2, C 9, 2, E 20, 2, C	32, 2, D	10, 2, A 32, 2, C	14, 2, E
32, 3, F	14, 3, C	9, 3, D	12, 3, C 10, 3, C	20, 3, D 6, 3, D	14, 3, C	32, 3, E
14, 4, B	9, 4, C			10, 4, D 12, 4, D	9, 4, C	10, 4, E 6, 4, E
9, 5, F						10, 5, E 12, 5, E

Questa tupla X-Y-Z rappresenta:

- X -> il costo del segmento scoperto
- Y -> n° hop del nodo di partenza
- Z -> nodo vicino che conosco (da cui poi mi arriva questa informazione)

Ad ogni "ciclo" N, e per ogni casella che devo riempire, vado al ciclo precedente a vedere, nelle celle dei nodi che ho scoperto in quel ciclo; se contengono dei cammini che non conosco, li aggiungo, altrimenti no. Dovrebbe essere questo l'"algoritmo" ufficiale.

altro RIP



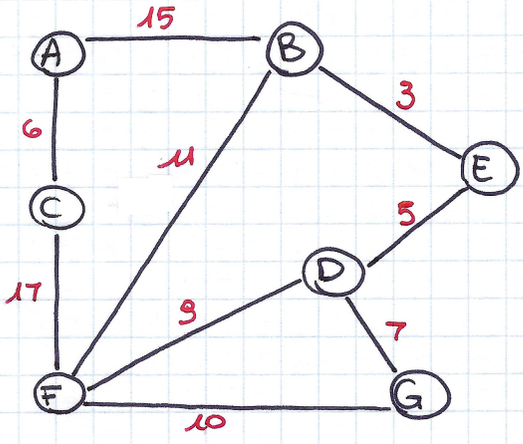
A	B	C	D	E	F	G	H
4-1	12-1 10-1	12-1 5-1	5-1 7-1	4-1 7-1	10-1 C-1 7-1	2-1 7-1 15-1	15-1
7-2-E	5-2-C 2-2-F 1-2-F	10-2-B 7-2-C	2-2-G 12-2-G 15-2-C	10-2-F 2-2-F	7-2-G 15-2-G 4-2-E 12-2-B	5-2-D 10-2-F 7-2-F	7-2-G 2-2-G
10,3,E 2,3,E	7,3,F 4,3,F 15,3,F	2,3,B 7,3,B 15,3,D	10,3,C 7,3,G	12,3,F 7,3,F 15,3,F	5,3,G	4,3,F 12,3,D	5,3,G 7,3,G 10,3,G
12,4,E 7,4,E 15,4,E	/	4,4,B	4,4,B	5,4,F	/	/	4,4,G 12,4,G
5,5,E							

1) 7,3,F o 7,3,C, è uguale!
2) 2,3,B o 2,3,D, è uguale
3) 10,3,C o 10,3,G è uguale
4) 5,3,G o 5,3,B è uguale
5) 12,3,D o 12,3,F è uguale

~~[illegible]~~ eps

NOTA: non so se sia sempre vero, ma finora si è sempre verificato - a fine scoperte, sotto ogni nodo compaiono N tuple, ogni volta (dove N = numero degli "archi" o percorsi o collegamenti o che so)

III)

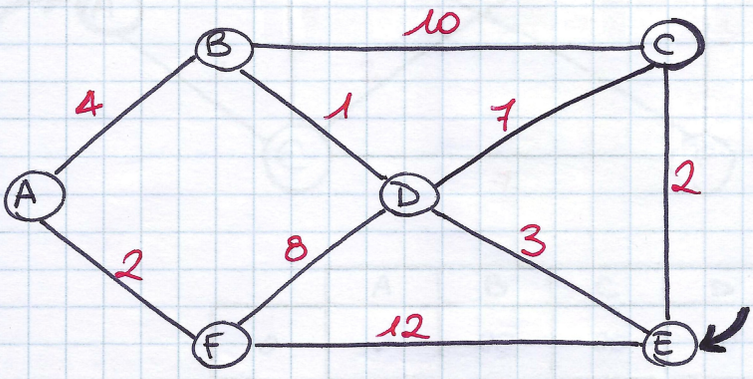


A	B	C	D	E	F	G
6-1	15-1	6-1	5-1	3-1	9-1	7-1
15-1	11-1	17-1	7-1	5-1	10-1	10-1
17-2-C	3-1	11-2-F	9-1	7-2-D	11-1	9-2-D
11-2-B	6-2-A	9-2-F	10-2-F	9-2-D	17-1	5-2-D
3-2-B	17-2-B	10-2-F	3-2-E	15-2-B	3-2-B	11-2-F
10-3-F	9-2-F	15-2-A	11-2-F	11-2-B	7-2-G	17-2-F
9-3-F	10-2-F	3-3-F	17-2-F	10-3-D	15-2-B	6-3-F
5-3-B	5-2-E	5-3-F	6-3-F	6-3-B	6-2-C	3-3-D
7-4-B	7-3-B	7-3-F	15-3-E	17-3-D	5-2-D	15-3-F

A/N: qui per ragioni logistiche ho compattato tutto usando un colore diverso per ogni #hop (come facciamo a lezione, in effetti)

ESERCIZI SU ALGORITMI LINK STATE

1)



Nota: a differenza di RIP, qui ci interessa, dato un NODO DI PARTENZA, quali sono i percorsi più brevi (in termini di COSTO, non di n° hop) per raggiungere gli altri nodi della rete.

Passo \ nodo	A	B	C	D	E	F
0	∞	∞	∞	∞	✓	∞
1	∞	∞	2, E	3, E	-	12, E
2	∞	12, C	✓	3, E	-	12, E
3	∞	4, D	-	✓	-	11, D
4	8, B	✓	-	-	-	11, D
5	✓	-	-	-	-	10, A

- al passo 0, conosco ovviamente solo il nodo di partenza
- alla fine di ogni passo, seleziono quello con peso minore
- qui mi domando, nel caso di C:
  - c'è un percorso più "breve" (per la precisione, meno costoso) di quello che ho per raggiungere C?
    - sì -> lo sostituisco;
    - no -> lo confermo;

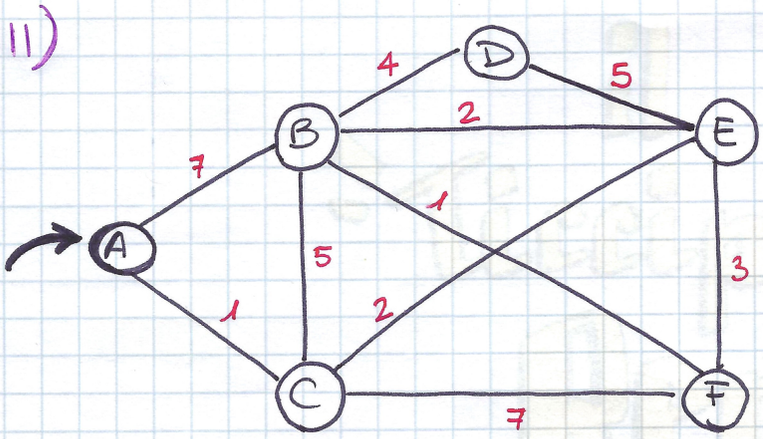
tipo questo lo abbiamo trovato facendo 3+1+4 ovvero E→D→B→A !

Nota 2: la lettera accanto al costo in tabella, nel caso di Link State, non si riferisce al nodo adiacente a quello di partenza, ma al nodo adiacente al NUOVO nodo SCOPERTO IN RIFERIMENTO AL NUOVO PERCORSO TROVATO
Cioè detta in maniera diversa: qui vediamo il penultimo passo del percorso (il più vicino alla fine), in RIP il 2° (il più vicino all'inizio).

Nota 3 (importante): perché l'esercizio si consideri concluso, è necessario fornire una TABELLA DI ROUTING coi percorsi calcolati sopra:

Router	NextHop	Costo
A	B	8
B	D	4
C	E	2
D	E	3
E	-	-
F	A	10

11)



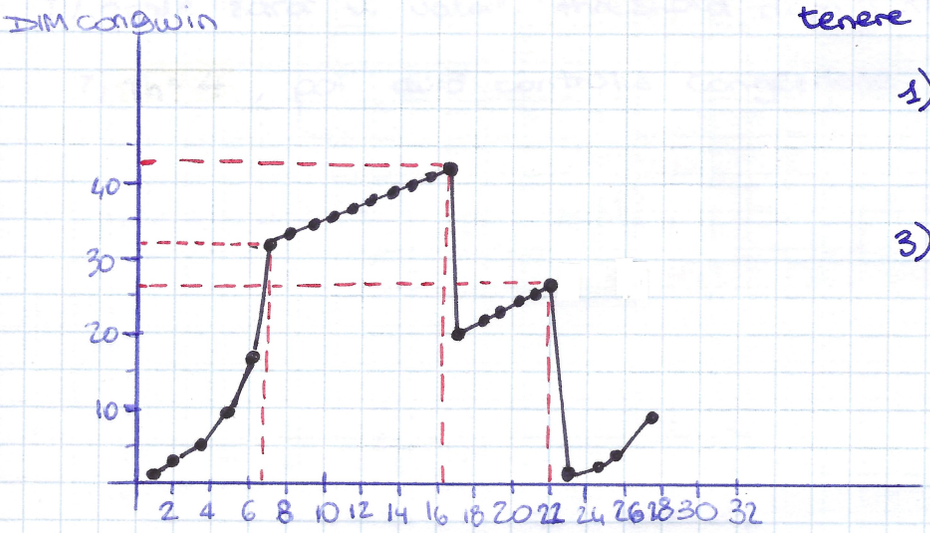
	A	B	C	D	E	F
0	✓	∞	∞	∞	∞	∞
1	-	7,A	1,A	∞	∞	∞
2	-	6,C	✓	∞	3,C	8,C
3	-	5,E	-	8,E	✓	6,E
4	-	✓	-	8,E	-	6,E
5	-	-	-	8,E	-	✓

tabella di routing:

Nodo	Nx Hop	costo
A	-	-
B	E	5
C	A	1
D	E	8
E	C	3
F	E	6

il Ticcippi Reno

- Gli esercizi sul TCP Reno (menzionato nel corso) si svolgono così:



- il TCP Reno ha 4 cose importanti da tenere a mente:

  1) AVVIO LENTO;

  2) CONTROLLO CONGESTIONE;

  3) TRIPLO ACK DUPLICATO;

  4) TIMEOUT;

Domande più gettonate:

1) Individuare gli intervalli dove c'è un **avvio lento** TCP:

   1) sono quelli simili a delle CURVE ESPONENZIALI, quindi:

      - **intervallo 1÷6**
      - **intervallo 23÷26**

2) L'intervallo/i di CONTROLLO CONGESTIONE:

   2) le RETTE crescenti, quindi:

      - **intervallo 6÷16**
      - **intervallo 17÷22**

3) Dopo l'istante 16, si verifica un TIMEOUT o un 3ACK duplicato?

3) 3ACK duplicato. Se la dimensione della finestra torna a (1) dopo la perdita, allora si tratta di TIMEOUT. Se invece si DIMEZZA allora è 3ACK d.;

4) Qual è il valore di threshold al round 1?

4) Th=32, ovvero dove si ferma lo slow start;

5) Valore threshold al [redacted] round 18?

5) Th=21, ovvero 42 (valore al round 16), diviso 2.

Ogni volta che c'è una perdita, la threshold si DIMEZZA. Sia in caso di triplo ack che di timeout;

6) Valore threshold al round 24?

6) Th=13, ovvero 26/2;

7) quale sarà il valore threshold dopo il round 26?

7) Th=4, poi avrò controllo congestione;

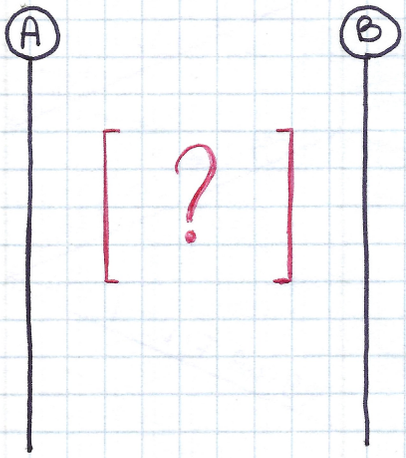
also!
Nel caso di TIMEOUT si riparte con slow start, nel caso di 3ACK col controllo congestione.

TUTORATO IReS 13/01/2022

✦ handshakes ✦

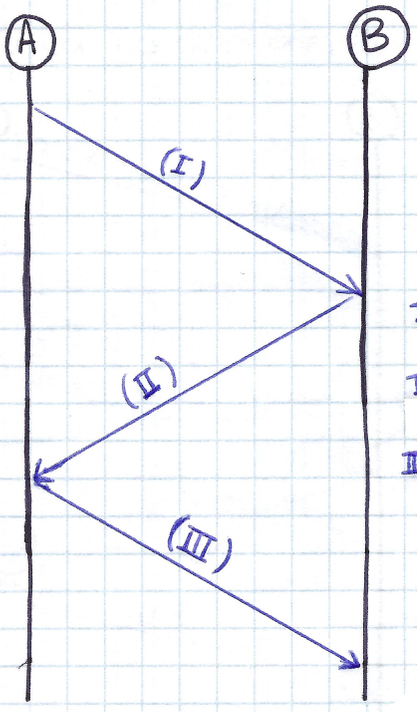
- ci viene dato un nº di sequenza e un nº di ACK. Non sono numeri significativi, sono "casuali", un riferimento da cui partire.

1) Realizzare una CONNESSIONE tra A e B (HANDSHAKE).



Con:
SeqNo = 1
AckNo = 80

1) Vogliamo fare un 3-way handshake tipico del TCP.
↳ Il famoso SYN - SYNACK - ACK



SeqNo = 1, AckNo = 80

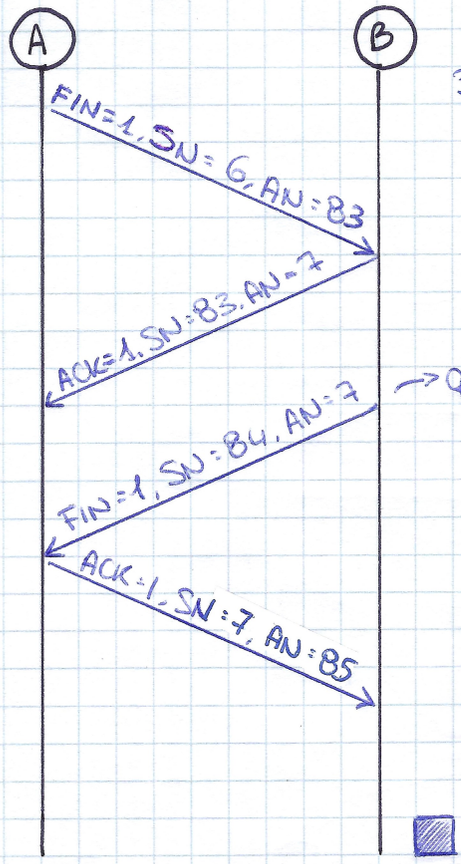
I) SYN = 1, SeqNo = 1, AckNo = 80
II) SYN = 1, ACK = 1, SeqNo = 80, AckNo = 2
III) ACK = 1, SeqNo = 2, AckNo = 81

COME MAI QUESTO SCAMBIO DI CIFRE TRA SEQ ed ACK?

È una cosa che avviene ad ogni "CAMBIO DI DIREZIONE" (da A→B a B→A). Non è altro che una scelta di design del protocollo, fatta a fini di sicurezza: quando il messaggio cambia verso, SeqNo ed AckNo vengono scambiati di valore, poi il "nuovo" AckNo viene aumentato di 1.

II) ora invece vogliamo CHIUDERE la connessione.
ovvero A chiede a B di chiudere la connessione.

II)



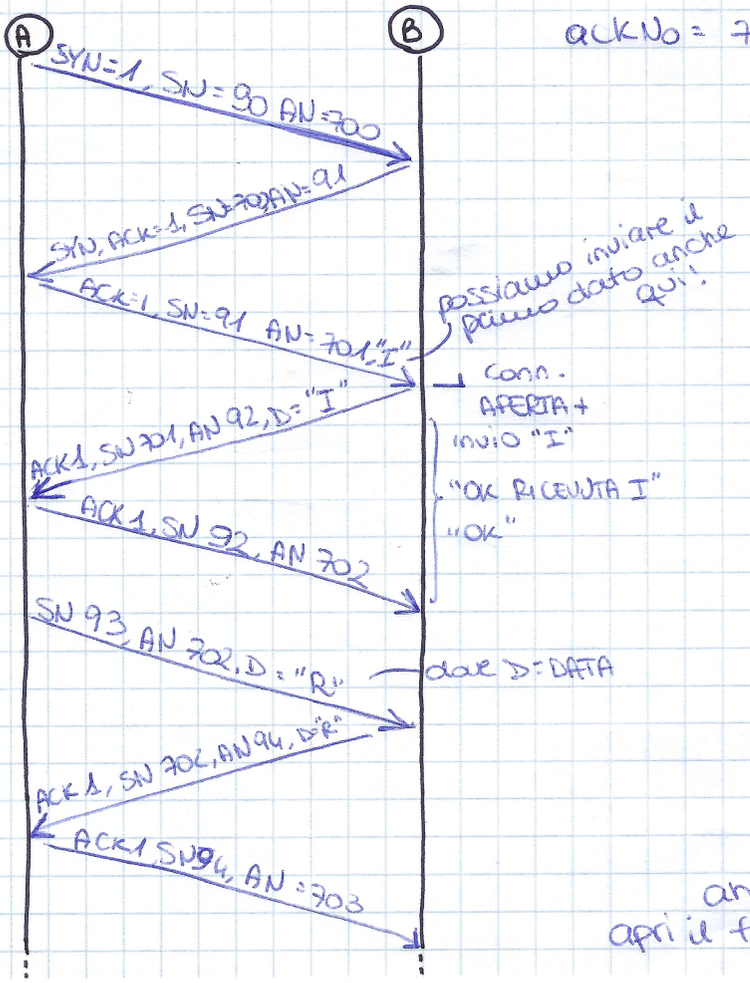
SeqNo = 6 AckNo = 83
a mio avviso potete usare la notazione che preferite per scrivere questi dati, "seqNo", "sequence_number", "SN", "SequenceNumber", "#seq", boh!

→ qui non c'è un cambio di direzione, aumenta il sequence number.

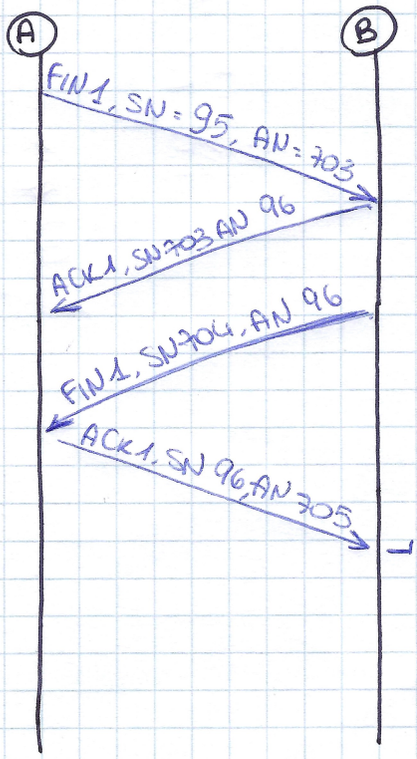
☺ lo sapete il vecchio scherzone del frigorifero? Dice: COME SI FA A CHIUDERE UN ELEFANTE IN UN FRIGORIFERO?

III) APRI la connessione, invia "I", invia "R", CHIUDI la connessione.

III)



ackNo = 700 - seqNo = 90



ah la risposta allo scherzone era:
apri il frigorifero, metti l'elefante, chiudi il frigorifero.
MI DISPIACE TANTO