

# Esercizi Capitolo 1

## Esercizio 1.1

Si considerino due host A e B distanti 10.000 Km connessi da un collegamento di  $R=1\text{Mbps}$  e con velocità di propagazione di  $2,5 \cdot 10^8 \text{ m/s}$ .

1. Consideriamo l'invio di un file di 400.000 bit dall'host A all'host B in un unico grande messaggio. Quale è il massimo numero di bit che si troveranno nel collegamento in un dato istante?

Tempo di propagazione (attraversamento del canale):  $10 \cdot 10^6 \text{ metri} / 2,5 \cdot 10^8 \text{ metri/secondo} = 4 \cdot 10^{-2} \text{ secondi} = 0,04 \text{ secondi}$

Numero di bit trasmessi in 0,04 secondi:  $1 \cdot 10^6 \text{ bps} \cdot 4 \cdot 10^{-2} \text{ s} = 4 \cdot 10^4 \text{ bit} = 40.000 \text{ bit}$

Dato che  $4 \cdot 10^5 > 4 \cdot 10^4$ , il numero di bit presenti contemporaneamente nel canale è 40.000 bit

2. Quanto vale l'ampiezza (in metri) di un bit nel collegamento?

Un bit è trasmesso in  $1/10^6 \text{ secondi} = 10^{-6} \text{ sec}$

Alla velocità di  $2,5 \cdot 10^8 \text{ m/s}$ , un bit occupa  $2,5 \cdot 10^2 \text{ metri} = 250 \text{ metri}$

Quindi nel canale si trovano:

$10.000.000 \text{ metri} / 250 \text{ metri} = 40.000 \text{ bit}$

## Esercizio 1.2/3.3

Si consideri il problema 1 del capitolo 1 del libro (4' edizione). Si chiede di rappresentare il protocollo tramite un automa a stati finiti esteso.

## Esercizio 1.3

Supponiamo che più utenti condividano un collegamento da 5Mbps. Ogni utente trasmette dati per il 10% del suo tempo, e quando trasmette necessita di una banda pari a 200Kbps. Si chiede:

1. Se nel collegamento si utilizza la commutazione di circuito, quanti utenti si possono supportare?
2. Determinare la probabilità che un utente stia trasmettendo (nell'ipotesi che si usi la commutazione di circuito e che il numero di utenti sia quello massimo ammesso)
3. Supponiamo che sia usata la commutazione di pacchetto e che siano presenti 40 utenti. Determinare la probabilità che esattamente  $n$  utenti siano attivi in trasmissione contemporaneamente (suggerimento: usare la distribuzione binomiale)
4. Calcolare la probabilità di avere almeno 26 utenti attivi contemporaneamente in fase di trasmissione

## Soluzione

1. Con la commutazione di circuito, il numero di utenti massimo è pari a  $5\text{Mbps} / 200 \text{ Kbps} = 25$
2. La probabilità che uno specifico utente stia trasmettendo è  $1/10$   
Considerando 25 utenti, la probabilità che 1 utente stia trasmettendo è:  
 $\text{Binomiale}(25,1) \cdot 0,1^1 \cdot 0,9^{(25-1)} = 25 \cdot 0,1 \cdot 0,9^{24} = (\text{circa}) 0,2$

3. La probabilità che, su 40 utenti  $n$  stiano trasmettendo è:  
Binomiale(40,n)\*0,1<sup>n</sup>\*0,9<sup>(40-n)</sup>
4. La probabilità che almeno 26 utenti siano attivi contemporaneamente è:  
SOMMA\_(n=26,40) Binomiale(40,n)\*0,1<sup>n</sup>\*0,9<sup>(40-n)</sup> = (circa) 5,6\*10<sup>-17</sup>

## Esercizi Capitolo 2

### Esercizio 2.1

1. Si consideri una pagina web (di dimensione di 9 Kbyte) contenente 5 oggetti collegati, di dimensione rispettivamente 10 KByte, 34 KByte, 26 KByte, 8 Kbyte e 41 KByte. Stimare il tempo di caricamento della pagina usando connessioni non persistenti (HTTP1.0) e ipotizzando che il RTT (Round Trip Time) tra il client e il server web sia di 25 millisecondi, che il tempo di accodamento e elaborazione sia trascurabile, e che il canale abbia un tasso di trasmissione di 256 Kbps.
2. Rifare l'esercizio del punto precedente nell'ipotesi che siano utilizzate connessioni persistenti (HTTP 1.1) e supponendo che le richieste di tutti gli oggetti collegati vengano fatte immediatamente dopo la ricezione della pagina.

#### Soluzione

1. La trasmissione di ogni oggetto comporta 1 RTT per stabilire la connessione, 1RTT per richiederlo e L/256Kbps per trasmetterlo. In totale: 12 RTT + 128Kbyte / 256 Kbps = 12\*25 msec + 4 sec = 4,3 sec
2. La trasmissione della pagina comporta 2 RTT + 9KB/256Kbps, poi la trasmissione degli altri 5 oggetti richiede 1 RTT + il tempo di trasmissione. Quindi il tempo totale è: 3RTT + 128Kbyte/ 256 Kbps = 3\*25 msec + 4 sec = 4,075 sec

### Esercizio 2.2

Si consideri un DNS locale che utilizza la modalità ricorsiva per la traduzione di un nome. Supponiamo che un cliente richieda la traduzione di un nome non presente nella cache di alcun DNS in tutta la gerarchia. Calcolare il tempo di completamento della richiesta (il tempo intercorso tra la richiesta al DNS locale e la risposta) ipotizzando:

- Tempi di elaborazione dei DNS trascurabili
- RTT tra ogni coppia di DNS pari a 4 msec
- RTT tra cliente e DNS locale pari a 0,1 msec

#### Soluzione

dal DNS Locale la query interroga nell'ordine: un DNS root, un DNS TLD, un DNS di competenza. Questo comporta un ritardo pari a 3\*4 msec = 12 msec.

L'interazione tra client e DNS Locale comporta un ulteriore ritardo di 0,1 msec. Quindi il tempo totale è 12,1 msec.

## Esercizi Capitolo 3

### Esercizio 3.1:

Disegnare l'automa a stati finiti del mittente e destinatario di un protocollo stop&wait modificato, nel quale il destinatario invia un ACK solo ogni due pacchetti dati ricevuti. Si assuma che il canale consegni i pacchetti nello stesso ordine nel quale sono stati inviati, ma che possa corrompere o perdere i pacchetti.

Suggerimento: usare numeri di sequenza e aumentare il numero degli stati, sia al mittente che al destinatario

### Esercizio 3.2:

Si consideri la seguente sequenza di pacchetti trasmessi tramite il protocollo stop & wait, con i rispettivi tempi di trasmissione (in millisecondi):

Pacchetto	p0	P1	P2	P3	P4
Tempo	0	5	6	12	50

Supponiamo che il RTT sia di 8 millisecondi e che i tempi di trasmissione, accodamento e di elaborazione siano trascurabili.

a) Mostrare l'evoluzione nel tempo dello stato del mittente e destinatario tramite la seguente tabella.

T	Evento	Stato mittente	Stato destinatario
0	Trasmesso p0	Wait for ACK 0	Wait for 0 from below
...	...	...	...

b) Ipotizzando ora che il tempo di trasmissione sia identico per tutti i pacchetti e pari a 1ms, calcolare l'utilizzazione del canale da parte del mittente.

c) Supponiamo ora che, durante la trasmissione, vengano persi o corrotti i pacchetti ACK del pacchetto P1, P2, P4. Mostrare tramite la seguente tabella come evolve nel tempo lo stato del destinatario e del mittente ipotizzando che il timeout sia di 5 ms.

T	Evento	Stato mittente	Stato mittente
0	Trasmesso p0	Wait for ACK 0	Wait for 0 from below
...	...	...	...

## Soluzione

a)

T	Evento	Stato mittente	Stato destinatario
0	Trasmesso p0	Wait for ACK 0	Wait for 0 from below
4	Ricevuto P0, spedito A0	Wait for ACK 0	Wait for 1 from below
5	Arriva P1, non viene trasmesso per ora	Wait for ACK 0	Wait for 1 from below
6	Arriva P2, non viene trasmesso per ora	Wait for ACK 0	Wait for 1 from below
8	Arriva A0, trasmetto P1	Wait for ACK 1	Wait for 1 from below
12	Arriva P3, non viene trasmesso per ora; arriva P1 al ricevente, trasmesso A1	Wait for ACK 1	Wait for 0 from below
16	Arriva A1, trasmesso P2	Wait for ACK 0	Wait for 0 from below
20	Arriva P2, trasmesso A0	Wait for ACK 0	Wait for 1 from below
24	Arriva A0, trasmesso P3	Wait for ACK 1	Wait for 1 from below
28	Arriva P3, trasmesso A1	Wait for ACK 1	Wait for 0 from below
32	Arriva A1	Wait for 0 from above	Wait for 0 from below
50	trasmesso P4	Wait for ACK 0	Wait for 0 from below
54	Arriva P4, spedito A0	Wait for ACK 0	Wait for 1 from below
58	Arriva A0	Wait for 1 from above	Wait for 1 from below

### Esercizio 3.3

Si considerino due host A (mittente) e B (destinatario) che comunicano con un protocollo di trasporto Selective Repeat. La finestra utilizzata dal protocollo ha dimensione  $N=3$ , il timeout di ritrasmissione del mittente è di 20 millisecondi, il ritardo di propagazione dei pacchetti è di 5 millisecondi (e di conseguenza il Round Trip Time (RTT) è di 10 millisecondi) e ipotizziamo che i tempi di trasmissione, di elaborazione e di accodamento siano trascurabili. Per semplicità si assuma che, in caso di ritrasmissione di più pacchetti, questi vengano ricevuti contemporaneamente dal destinatario.

Al tempo  $t=0$  la base della finestra del protocollo al mittente ha valore  $base=0$  e il prossimo numero di sequenza utilizzabile è  $nextseq=0$ , mentre il ricevente attende un pacchetto con numero di sequenza  $seqatteso=0$ .



10	Ricevuto a0	1	3	2	p1,p2	-	p0,p1
11	Spedito p3	1	4	2	p1,p2,p3	p3	p0,p1
16	Ricevuto p3, spedito a3	1	4	2	p1,p2,p3	a3	p0,p1,p3
20	p4 in attesa di spedizione	1	4	2	p1,p2,p3	a3	p0,p1,p3
21	Ricevuto a3, Scatta timeout, rispedito p1	1	4	2	p1,p2	p1	p0,p1,p3
26	ricevuto p1, spedito a1	1	4	2	p1,p2	a1	p0,p1,p3
28	Scatta timeout, rispedito p2	1	4	2	p1,p2	a1,p2	p0,p1,p3
31	Ricevuto a1, spedito p4	2	5	2	p2,p4	p2,p4	p0,p1,p3
33	Ricevuto p2, spedito a2	2	5	4	p2,p4	a2,p4	p0,p2,p1,p3
36	Ricevuto p4, spedito a4	2	5	5	p2,p4	a2,a4	p0,p2,p1,p3,p4
38	Ricevuto a2	4	5	5	p4	a4	p0,p2,p1,p3,p4
41	Ricevuto a4	5	5	5			p0,p2,p1,p3,p4

Tempo di completamento della trasmissione: 41

### Esercizio 3.4:

Si consideri una connessione TCP tra due host A e B. Al tempo  $t=0$  i valori di Estimated RTT e di DevRTT dell'host A sono rispettivamente 0,120 s e 0,033 s.

Supponiamo che, a partire dal tempo  $t$  A invii la seguente sequenza di segmenti lungo la connessione TCP all'host B e che riceva i seguenti segmenti di ACK:

Segmenti inviati (in secondi)		Segmenti ricevuti (in secondi)	
S1	$t=0,010$	ACK S1	$t=0,100$
S2	$t=0,015$	ACK S2	$t=0,210$
S3	$t=0,140$	ACK S3	$t=0,330$
S4	$t=0,190$	ACK S4	$t=0,350$
S5	$t=0,200$	ACK S5	$t=0,355$
S6	$t=0,210$	ACK S6	$t=0,360$

Si chiede di calcolare i valori di Estimated RTT e di DevRTT all'host A per ogni segmento di ACK ricevuto, assumendo che il parametro  $\alpha$  usato per calcolare la media esponenziale ponderata Estimated RTT sia  $\alpha=0,125$ , e che il parametro  $\beta$  usato per calcolare la media esponenziale ponderata DevRTT sia  $\beta=0,250$ .

ACK ricevuto	RTT campionato	Estimated RTT	Deviazione campionata	DevRTT
ACK S1				
ACK S2				
ACK S3				
ACK S4				
ACK S5				
ACK S6				

### Soluzione

Estimated RTT e DevRTT si calcolano tramite le formule:

$$\text{Estimated RTT} = (1 - \alpha) \text{ EstimatedRTT} + \alpha * \text{RTT Campionato}$$

$$\text{DevRTT} = (1 - \beta) \text{ DevRTT} + \beta (\text{RTT Campionato} - \text{Estimated RTT})$$

ACK ricevuto	RTT campionato	Estimated RTT	Deviazione campionata	DevRTT
ACK S1	0,090	0,116	0,026	0,031
ACK S2	0,195	0,126	0,069	0,041
ACK S3	0,190	0,134	0,056	0,045
ACK S4	0,160	0,137	0,023	0,039
ACK S5	0,155	0,140	0,015	0,033
ACK S6	0,150	0,141	0,009	0,027

### Esercizio 3.5

Due host comunicano tramite una rete di calcolatori inaffidabile, che può ritardare o perdere i pacchetti a causa di fenomeni di congestione) ma che non li può corrompere. Per comunicare i due host utilizzano un protocollo di tipo Stop & Wait che, oltre al normale funzionamento, prevede un ulteriore meccanismo per mitigare gli effetti della congestione.

Questo meccanismo opera come segue:

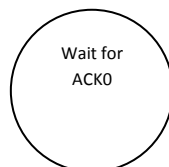
- Per ogni pacchetto trasmesso, il mittente misura il tempo *estimated-RTT* che intercorso tra l'istante di spedizione e la ricezione dell'ACK. Se *estimated-RTT* > *sogliaRTT* (dove *sogliaRTT* è un parametro preconfigurato del mittente) allora il mittente non accetta ulteriori pacchetti da spedire per un tempo pari a *Idle* (anche questo è un parametro preconfigurato del protocollo).
- Inoltre, quando scatta un timeout che segnala la mancata ricezione di un ACK, il pacchetto viene ritrasmesso solo dopo un tempo pari a *TRitasm* (nel frattempo non accetta altri pacchetti da trasmettere dal livello applicazione).

Scrivere l'automa a stati finiti per il mittente del protocollo descritto sopra. Per semplicità si riportano i soli stati Wait for call 0 from Above e Wait for ACK0. Si chiede di aggiungere gli stati necessari con le relative transizioni e di ignorare le transizioni in uscita dagli stati wait for ACK1 e Wait for call 1 from above.

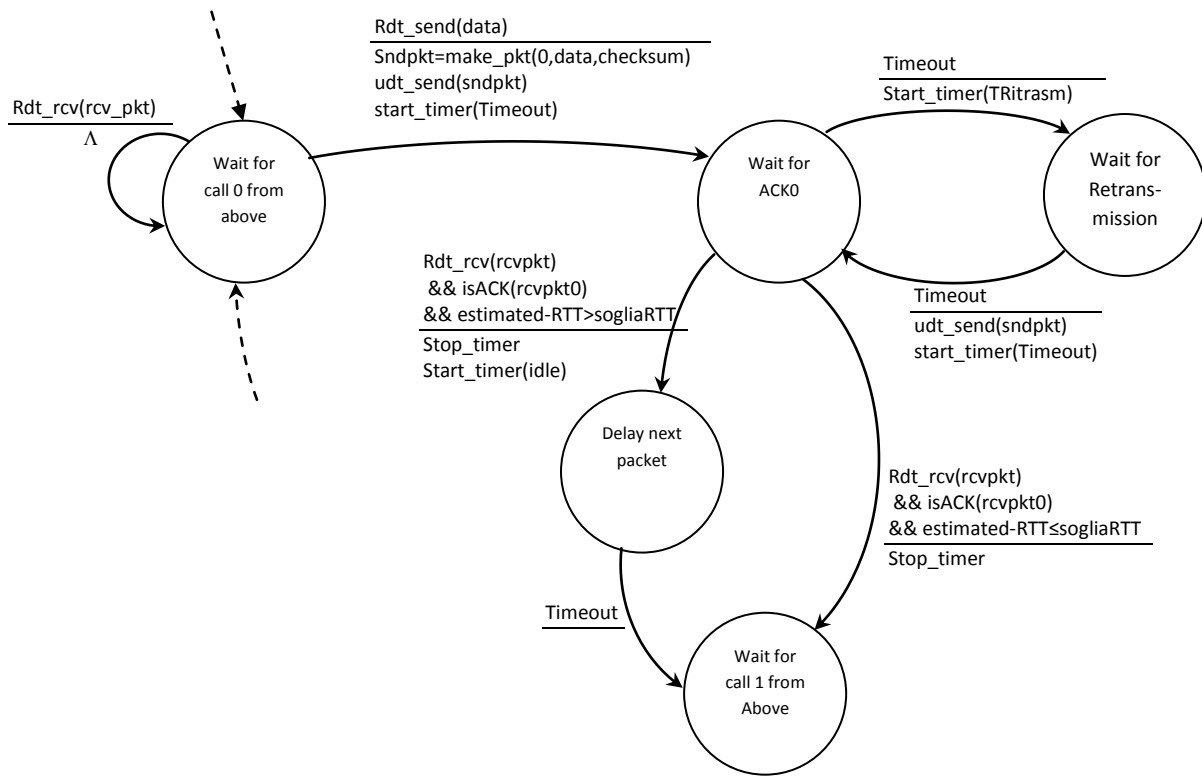
Suggerimento – utilizzare le funzioni:

- Rdt\_rcv(pkt) – per ricevere un pacchetto dal livello di rete
- Udt\_send(pkt) – per spedire il pacchetto pkt
- Rdt\_send(data) – per ricevere un dato da trasferire dal livello applicazione
- Start\_timer(x) – per impostare il timer al tempo x
- Stop\_timer – per fermare il timer

### Soluzione



# Soluzione









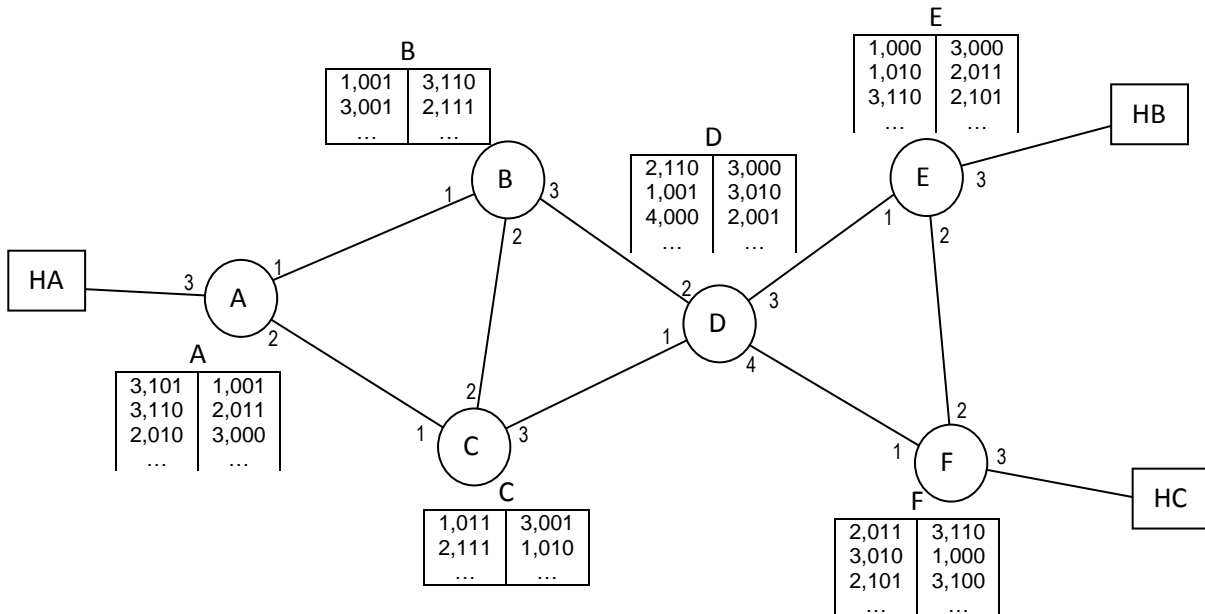
	e segmento ACK					
40	A: ricevuto segmento ACK	9.001	12.921	12.921	14.380	3
40	A: ricevuto segmento ACK	9.001	14.381	14.381	14.380	4
40	A: spediti 620 bytes	14.381	9.001*	14.381	15.001	4
50	B: Ricevuto segmento e spedito segmento ACK	9.001	15.001	14.381	15.001	4
60	A: ricevuto segmento ACK	9.001	15.001	15.001	15.001	5

\* Irrilevante: il bit A indica che il campo è non significativo

## Esercizi Capitolo 4

### Esercizio 4.1 (Tabelle di routing in reti a circuito virtuale)

Si consideri la seguente rete basata su circuiti virtuali. Ogni router è connesso, tramite porte bidirezionali, ad un certo numero di collegamenti (accanto ad ogni collegamento è mostrato il numero di porta corrispondente). Inoltre ogni router ha una tabella di routing che associa due coppie: <porta di ingresso, numero di un circuito virtuale> e <porta di uscita, numero di circuito virtuale>.



Si chiede di calcolare la destinazione e il percorso dei seguenti datagrammi:

Host sorgente	Numero di circuito virtuale	Host destinazione	Percorso
HA	101		
HB	110		
HC	010		
HA	110		

### Soluzione

Host sorgente	Numero di circuito virtuale	Host destinazione	Percorso
HA	101	HB	A,B,D,E
HB	110	HC	E,F
HC	010	HA	F,D,B,C,A
HA	110	HC	A,C,D,E,F

### Esercizio 4.2 (inoltre IP)

Un router IP con 4 porte di ingresso e 4 porte in uscita contiene la seguente tabella di routing:

prefisso dell'indirizzo di destinazione	Porta in uscita
1010 0110	3
1010 0110 1110	4
1010 0110 1110 11	2
1010 0110 1111	1
1000	1



	<b>L</b>	<b>ID</b>	<b>F</b>	<b>O</b>
P1,1	1020	300	1	0
P1,2	1020	300	1	125
P1,3	1020	300	1	250
P1,4	900	300	0	375
P2,1	1020	301	0	0
P3,1	900	302	0	0
P4,1	1020	303	1	0
P4,2	700	303	0	125

**Esercizio 4.4** (sottoreti)

Per ognuno dei seguenti indirizzi IP, indicare l'indirizzo della sottorete corrispondente:

- I1: 131.111.99.87/22
- I2: 230.88.101.230/24
- I3: 78.240.10.10/18
- I4: 120.120.64.9/20

**Soluzione**

- I1: \_\_\_\_\_
- I2: \_\_\_\_\_
- I3: \_\_\_\_\_
- I4: \_\_\_\_\_

**Soluzione**

- I1: 131.111.96-99.0
- I2: 230.88.101.0
- I3: 78.240. 0-63.0
- I4: 120.120.64-79.0

**Esercizio 4.5** (NAT)

Una rete locale è connessa ad internet tramite un NAT (Network Address Translation). L'indirizzo della sottorete locale è 192.168.1.0/24, la porta collegata ad internet del gateway ha indirizzo 136.38.243.99, e nella sottorete locale il gateway ha indirizzo 192.168.1.1.

Nella rete locale sono presenti due host di indirizzo 192.168.1.10 e 192.168.1.20 che, a partire da un certo tempo t, hanno stabilito le seguenti connessioni TCP (le connessioni sono elencate nello stesso ordine col quale sono state create):

<b>IP sorgente</b>	<b>IP Destinazione</b>	<b>Porta sorgente</b>	<b>Porta destinazione</b>
192.168.1.10	192.168.1.20	199	80
192.168.1.10	201.201.201.90	400	80
192.168.1.10	130.240.98.10	401	80
192.168.1.20	130.240.98.10	401	3100
192.168.1.20	192.168.1.10	1900	9880
192.168.1.20	201.201.201.90	900	1200

Si chiede di riempire la tabella di traduzione del NAT, assumendo che al tempo t la tabella del NAT fosse vuota e che il NAT utilizzi per la sua traduzione le porte a partire dalla 3000.

**Soluzione**

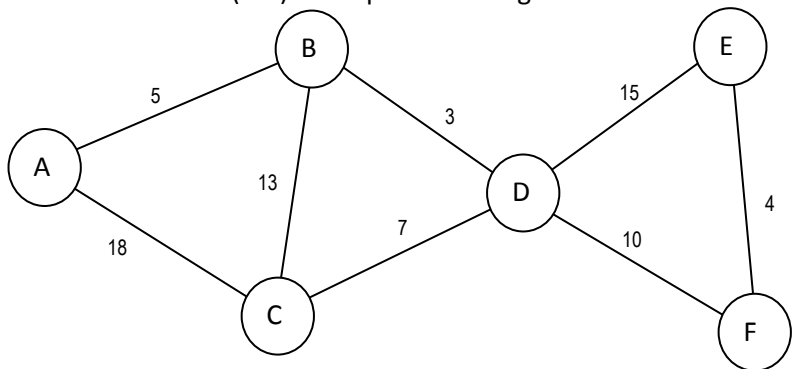
IP internet	Porta	IP LAN	Porta

**Soluzione**

IP internet	Porta	IP LAN	Porta
201.201.201.90	3000	192.168.1.10	400
130.240.98.10	3001	192.168.1.10	401
130.240.98.10	3002	192.168.1.20	401
201.201.201.90	3003	192.168.1.20	900

**Esercizio 4.6** (cammini minimi con Dijkstra)

Si consideri la seguente rete con 6 host (A-F) e con pesi sui collegamenti come indicato in figura.



Si utilizzi la seguente tabella per applicare l’algoritmo di Dijkstra per calcolare il contenuto della tabella di routing del router A. Nella tabella N’ è l’insieme dei router raggiunti, D(x) indica il costo per raggiungere x da A, e p(x) indica il predecessore del router x nell’albero dei cammini minimi.

passo	N’	D(B), p(B)	D(C), p(C)	D(D), p(D)	D(E), p(E)	D(F), p(F)
0						
1						
2						
3						
4						

5						
6						
7						

Quindi la tabella di routing del router A è:

Destinazione	Direzione
B	
C	
D	
E	
F	

**Soluzione**

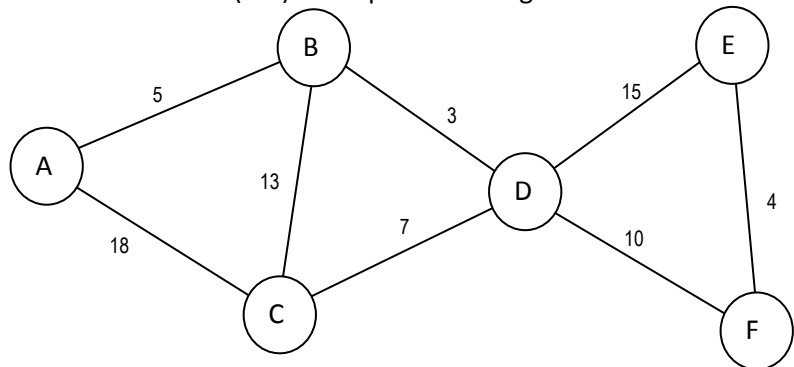
passo	N'	D(B), p(B)	D(C), p(C)	D(D), p(D)	D(E), p(E)	D(F), p(F)
0	A	5,A	18,A	-	-	-
1	A B	5,A	18,A	8,B	-	-
2	A B D	5,A	15,D	8,B	23,E	18,D
3	A B D C	5,A	15,D	8,B	23,E	18,D
4	A B D C F	5,A	15,D	8,B	22,F	18,D
5	A B D C F E	5,A	15,D	8,B	22,F	18,D

Quindi la tabella di routing del router A è:

Destinazione	Direzione
B	B
C	B
D	B
E	B
F	B

**Esercizio 4.7 (Distance Vector)**

Si consideri la seguente rete con 6 host (A-F) e con pesi sui collegamenti come indicato in figura.



All'istante t=0 ogni router conosce solo il costo per comunicare con i propri vicini ed inizia ad eseguire l'algoritmo Bellman-Ford per la determinazione della propria tabella di routing. Inoltre, le comunicazioni tra



router sono perfettamente sincrone e l'esecuzione dell'algoritmo di routing avviene per fasi: ad ogni fase (che si conclude in meno di 1 secondo, ogni router invia il proprio vettore di distanza aggiornato a tutti i router adiacenti (e di conseguenza riceve il vettore di distanza di tutti i router adiacenti) e calcola il suo nuovo vettore di distanza.

Al tempo  $t=0$  i vettori di distanza dei router sono come indicato in tabella. Ricordo che il vettore di distanza del router X è  $D_x = \langle d_x(A), d_x(B), d_x(C), d_x(D), d_x(E), d_x(F) \rangle$ , dove  $d_x(Y)$  indica il costo necessario per raggiungere Y da X.

	$D_A$	$D_B$	$D_C$	$D_D$	$D_E$	$D_F$
$t=0$	0,5,18,-,-,-	5,0,13,3,-,-	18,13,0,7,-,-	-,3,7,0,15,10	-,-,-,15,0,4	-,-,-,10,4,0
$t=1$						
$t=2$						
$t=3$						

### Variante 1

Si supponga che, immediatamente dopo il calcolo del vettore di distanza al tempo 2, il costo del collegamento tra D e F diventi 12. Utilizzare la tabella sottostante per calcolare le iterazioni necessarie all'algoritmo per convergere verso i nuovi vettori di distanza.

	$D_A$	$D_B$	$D_C$	$D_D$	$D_E$	$D_F$
$t=2$						
$t=3$						
$t=4$						
$t=5$						
$t=6$						
$t=7$						

### Variante 2

Si supponga che i router abbiano già raggiunto la convergenza verso i cammini minimi, per cui i vettori di distanza sono:

	$D_A$	$D_B$	$D_C$	$D_D$	$D_E$	$D_F$
$t=0$	0,5,15,8,22,18	5,0,10,3,17,13	15,10,0,7,21,17	8,3,7,0,14,10	22,17,21,14,0,4	18,13,17,10,4,0

Ipotizzando che in questo istante il costo del collegamento tra D e F diventi 30, riapplicare l'algoritmo Distance Vector con la tecnica dell'inversione avvelenata e ricalcolare le tabelle di routing.

### Soluzione

	$D_A$	$D_B$	$D_C$	$D_D$	$D_E$	$D_F$
$t=0$	0,5,18,-,-,-	5,0,13,3,-,-	18,13,0,7,-,-	-,3,7,0,15,10	-,-,-,15,0,4	-,-,-,10,4,0
$t=1$	0,5,18,8,-,-	5,0,10,3,18,13	18,10,0,7,22,17	8,3,7,0,14,10	-,18,22,14,0,4	-,13,17,10,4,0
$t=2$	0,5,15,8,23,18	5,0,10,3,17,13	15,10,0,7,21,17	8,3,7,0,14,10	22,17,21,14,0,4	18,13,17,10,4,0
$t=3$	0,5,15,8,22,18	"	"	"	"	"

### Soluzione variante 1

Evoluzione del sistema se al tempo due il costo del collegamento tra D e F diventa 12:

	$D_A$	$D_B$	$D_C$	$D_D$	$D_E$	$D_F$
$t=2$	0,5,15,8,23,18	5,0,10,3,17,13	15,10,0,7,21,17	8,3,7,0,14,10	22,17,21,14,0,4	18,13,17,10,4,0
$t=3$	0,5,15,8,22,18	"	"	8,3,7,0,15,12	"	20,15,19,12,4,0
$t=4$	"	5,0,10,3,18,15	15,10,0,7,22,19	8,3,7,0,15,12	23,18,22,15,0,4	20,15,19,12,4,0
$t=5$	0,5,15,8,23,20	5,0,10,3,18,15	15,10,0,7,22,19	8,3,7,0,15,12	23,18,22,15,0,4	20,15,19,12,4,0

### Soluzione variante 2

Evoluzione del sistema se al tempo due il costo del collegamento tra D e F diventa 30 con inversione avvelenata.

Al tempo  $t=0$  le tabelle di routing dei vari router sono:

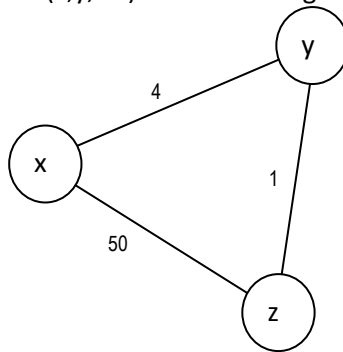
- A->B per ogni destinazione
- B->A; B->D per ogni altra destinazione
- D->B (per A,B); D->C; D->F (per E,F)
- C->D per ogni destinazione
- E->F per ogni destinazione
- F->E; F->D per ogni altra destinazione

	$D_A$	$D_B$	$D_C$	$D_D$	$D_E$	$D_F$
$t=0$	0,5,15,8,22,18	5,0,10,3,17,13	15,10,0,7,21,17	8,3,7,0,14,10	22,17,21,14,0,4	18,13,17,10,4,0
$t=1$	0,5,15,8,22,18	5,0,10,3,17,13	15,10,0,7,21,17	8,3,7,0,15,19	22,17,21,14,0,4	38,33,37,30,4,0
$t=2$	0,5,15,8,22,18	5,0,10,3,18,22	15,10,0,7,22,26	"	23,18,22,15,0,4	38,33,37,30,4,0
$t=3$	0,5,15,8,23,27	"	"	"	"	27,22,26,19,4,0
$t=4$	"	"	"	"	"	"

Nota: si consiglia di ricalcolare le tabelle di routing ad ogni passo.

#### Esercizio 4.8 (Distance Vector, count to infinity)

Si consideri la rete composta da tre router (x,y,e z) mostrata in figura.



Al tempo  $T$ , il costo del collegamento tra  $x$  e  $y$  passa improvvisamente da 4 a 60. Si chiede di eseguire le prime 10 iterazioni dell'algoritmo distance vector eseguite dai router  $x$ ,  $y$  e  $z$  assumendo che i router siano sincronizzati e scambino i vettori di distanza contemporaneamente ogni secondo.

Si inoltre assuma che al tempo  $t$  i vettori di distanza dei tre router fossero:

- $D_x = \langle x,0 \rangle, \langle y,4 \rangle, \langle z,5 \rangle$
- $D_y = \langle x,4 \rangle, \langle y,0 \rangle, \langle z,1 \rangle$
- $D_z = \langle x,5 \rangle, \langle y,1 \rangle, \langle z,0 \rangle$

	$D_x$	$D_y$	$D_z$
$T$	$\langle x,0 \rangle, \langle y,4 \rangle, \langle z,5 \rangle$	$\langle x,4 \rangle, \langle y,0 \rangle, \langle z,1 \rangle$	$\langle x,5 \rangle, \langle y,1 \rangle, \langle z,0 \rangle$
$T+1$			
$T+2$			
$T+3$			
$T+4$			
$T+5$			
$T+6$			
$T+7$			
$T+8$			
$T+9$			
$T+10$			

#### Soluzione

	$D_x$	$D_y$	$D_z$

T	<x,0>, <y,4>, <z,5>	<x,4>, <y,0>, <z,1>	<x,5>, <y,1>, <z,0>
T+1	<x,0>, <y,51>, <z,50>	<x,6>, <y,0>, <z,1>	<x,5>, <y,1>, <z,0>
T+2	<x,0>, <y,51>, <z,50>	<x,6>, <y,0>, <z,1>	<x,7>, <y,1>, <z,0>
T+3	"	<x,8>, <y,0>, <z,1>	<x,7>, <y,1>, <z,0>
T+4	"	<x,8>, <y,0>, <z,1>	<x,9>, <y,1>, <z,0>
T+5	"	<x,10>, <y,0>, <z,1>	<x,9>, <y,1>, <z,0>
T+6	"	<x,10>, <y,0>, <z,1>	<x,11>, <y,1>, <z,0>
T+7	"	<x,12>, <y,0>, <z,1>	<x,11>, <y,1>, <z,0>
T+8	"	<x,12>, <y,0>, <z,1>	<x,13>, <y,1>, <z,0>
T+9	"	<x,14>, <y,0>, <z,1>	<x,13>, <y,1>, <z,0>
T+10	"	<x,14>, <y,0>, <z,1>	<x,15>, <y,1>, <z,0>