Reti di Calcolatori

Esercizi su routing

Convenzioni utilizzate

- Ogni nodo invia gli update periodicamente ogni T secondi
- Tutti i nodi sono sincronizzati e iniziano a scambiarsi i distance vector (DV) a partire dal tempo t=0;
  - i successivi update vengono inviati dai diversi nodi esattamente nello stesso istante;
- Se il costo di un link cambia (ad es. se un link si guasta), il nodo aspetta il successivo invio degli update
  - non notifica immediatamente il cambiamento ai vicini
  - esempio: se il link si guasta al tempo t = 3T + T/2, l'update viene inviato al tempo t = 4T;
    - semplificazione rispetto al caso generale, dove invece si invia subito un update;
- Se un update ricevuto dai vicini fa cambiare la tabella di routing di un nodo, il nodo aspetta il successivo invio degli update per notificare tale cambiamento
  - non notifica immediatamente il cambiamento ai vicini)
  - semplificazione rispetto al caso generale, dove invece si invia subito un update;

Convenzioni utilizzate

- I nodi utilizzano gli update dei vicini per aggiornare la tabella di routing, e poi scartano l'update ricevuto
  - non tengono memoria del precedente DV ricevuto;
- Se un link si guasta, tutte le destinazioni che hanno come next-hop il nodo coinvolto vengono poste come irraggiungibili
- In definitiva:
  1. ogni nodo invia il proprio DV all'istante T, 2T, 3T, …
  2. ogni nodo riceve il DV dei vicini una frazione di tempo successiva all'instante T, 2T, 3T, …
  3. con i DV ricevuti ogni nodo aggiorna la propria tabella di routing e torna al punto 1;

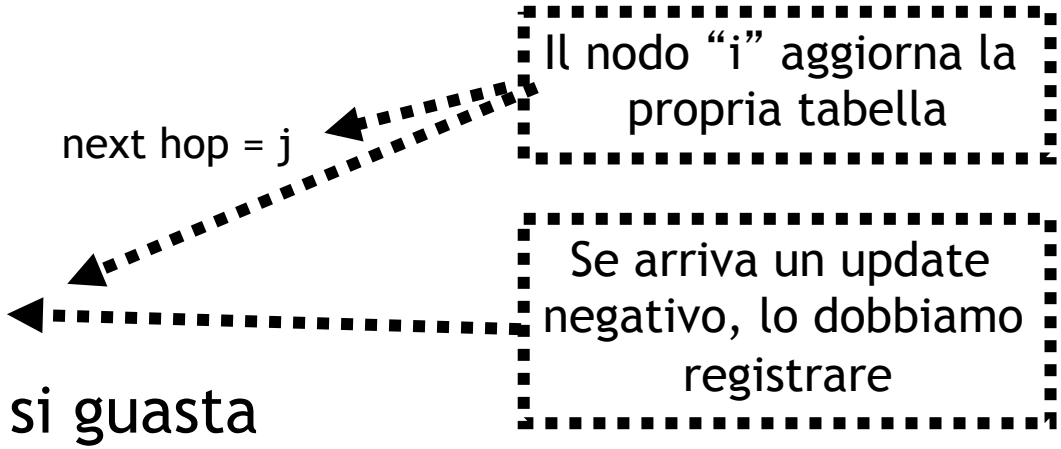
Algoritmo di aggiornamento delle tabelle di routing

- Convenzione
  - c(i,j) e' il costo del link diretto tra il nodo "i" e il suo vicino "j"
  - D(i,k) e' il costo del cammino tra il nodo "i" e il nodo "k"

- Al generico nodo "i"
  - Inizializzazione
    - D(i,i) = 0 e Next-hop(i) = "i";
    - D(i,j) = c(i,j) e Next-hop(i) = "j" se "j" e' un vicino
    - D(i,k) = inf. e Next-hop(i) = - per tutti gli altri

Aggiornamento delle tabelle di routing

- Per ogni distance vector (DV) ricevuto dal nodo "j"
  - per ogni destinazione "k" contenuta nel DV
    - il nodo calcola c(i,j)+D(j,k) e lo confronta con D(i,k) della propria tabella di routing;
    - se c(i,j)+D(j,k) < D(i,k)
      - D(i,k) = c(i,j)+D(j,k) e next hop = j
    - altrimenti, se next hop == j
      - D(i,k) = c(i,j)+D(j,k)



- Se il link verso il nodo "q" si guasta
  - per ogni destinazione "k" contenuta nella tabella di routing
    - altrimenti, se next hop == q
      - D(i,q) = inf.

Notazione utilizzata

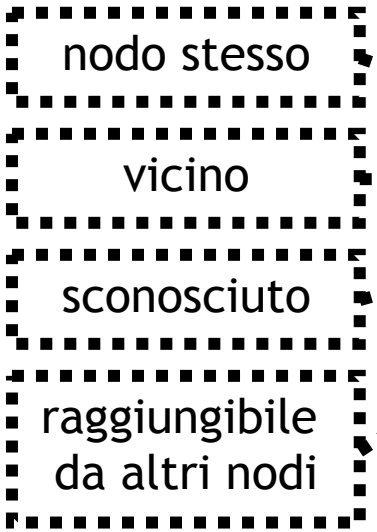


Tabella di routing
(ad es. del nodo A)

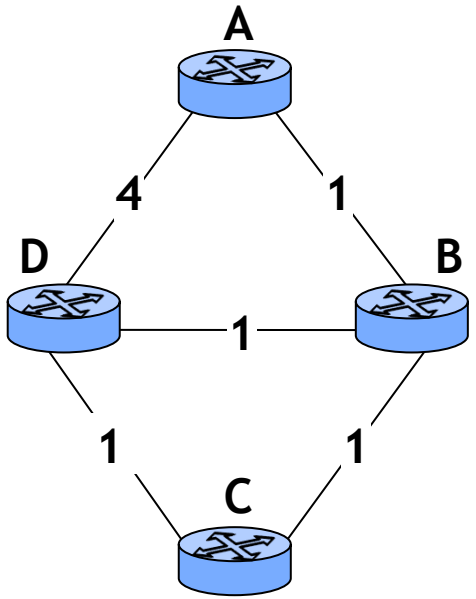
da A	dist	next
A	0	A
B	5	B
C	-	-
D	7	B

Distance Vector
(ad es. inviato da A)

da A	dist
A	1
B	5
C	-
D	2

Esercizio 1

- Con riferimento alla rete in figura, ove e' utilizzato l'algoritmo Distributed Bellman-Ford (DBF) classico senza alcun meccanismo aggiuntivo
  - Si indichi quale sarà la tabella di routing dei diversi nodi a regime
  - Si mostrino i messaggi scambiati nel caso in cui il link tra A e D si guasti
  - Si mostrino i messaggi scambiati nel caso in cui il link tra A e B si guasti
  - Nel caso in cui l'algoritmo implementi split-horizon con poison-reverse, si mostrino i messaggi scambiati nel caso in cui il link tra A e B si guasti



Esercizio 1 - Soluzione

Tabelle a regime

da A	dist	next
A	0	A
B	1	B
C	2	B
D	2	B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	1	D

dopo il guasto del link A-D ➔ nessun cambiamento

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

da D	dist	next
A	2	B
B	1	B
C	1	C
D	0	D

Tabelle subito dopo il guasto del link A-B

da A	dist	next
A	0	A
B	inf	-
C	inf	-
D	inf	-

da B	dist	next
A	inf	-
B	0	B
C	1	C
D	1	D

Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	inf	-
C	inf	-
D	inf	-

da D	dist
A	2
B	1
C	1
D	0

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

B

da B	dist	next
A	inf	-
B	0	B
C	1	C
D	1	D

da C	dist
A	2
B	1
C	0
D	1

da D	dist
A	2
B	1
C	1
D	0

da B	dist	next
A	3	C
B	0	B
C	1	C
D	1	D

poteva scegliere anche D

C

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

da B	dist
A	inf
B	0
C	1
D	1

da D	dist
A	2
B	1
C	1
D	0

da C	dist	next
A	3	D
B	1	B
C	0	C
D	1	D

D

da D	dist	next
A	2	B
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	inf
C	inf
D	inf

da B	dist
A	inf
B	0
C	1
D	1

da C	dist
A	2
B	1
C	0
D	1

da D	dist	next
A	3	C
B	1	B
C	1	C
D	0	D

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da D	dist
A	3
B	1
C	1
D	0

B

da B	dist	next
A	3	C
B	0	B
C	1	C
D	1	D

da C	dist
A	3
B	1
C	0
D	1

da D	dist
A	3
B	1
C	1
D	0

C

da C	dist	next
A	3	D
B	1	B
C	0	C
D	1	D

da B	dist
A	3
B	0
C	1
D	1

da D	dist
A	3
B	1
C	1
D	0

D

da D	dist	next
A	3	C
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	5
C	5
D	4

da B	dist
A	3
B	0
C	1
D	1

da C	dist
A	3
B	1
C	0
D	1

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da B	dist	next
A	4	C
B	0	B
C	1	C
D	1	D

poteva scegliere anche D

da C	dist	next
A	4	D
B	1	B
C	0	C
D	1	D

poteva scegliere anche B

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

predilige il colleg. diretto

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da D	dist
A	4
B	1
C	1
D	0

B

da B	dist	next
A	4	C
B	0	B
C	1	C
D	1	D

da C	dist
A	4
B	1
C	0
D	1

da D	dist
A	4
B	1
C	1
D	0

C

da C	dist	next
A	4	D
B	1	B
C	0	C
D	1	D

da B	dist
A	4
B	0
C	1
D	1

da D	dist
A	4
B	1
C	1
D	0

D

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	5
C	5
D	4

da B	dist
A	4
B	0
C	1
D	1

da C	dist
A	4
B	1
C	0
D	1

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da B	dist	next
A	5	C
B	0	B
C	1	C
D	1	D

poteva scegliere anche D

da C	dist	next
A	5	D
B	1	B
C	0	C
D	1	D

poteva scegliere anche B

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da D	dist
A	4
B	1
C	1
D	0

B

da B	dist	next
A	5	C
B	0	B
C	1	C
D	1	D

da C	dist
A	5
B	1
C	0
D	1

da D	dist
A	4
B	1
C	1
D	0

C

da C	dist	next
A	5	D
B	1	B
C	0	C
D	1	D

da B	dist
A	5
B	0
C	1
D	1

da D	dist
A	4
B	1
C	1
D	0

D

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	5
C	5
D	4

da B	dist
A	5
B	0
C	1
D	1

da C	dist
A	5
B	1
C	0
D	1

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da B	dist	next
A	5	D
B	0	B
C	1	C
D	1	D

deve scegliere D

da C	dist	next
A	5	D
B	1	B
C	0	C
D	1	D

poteva scegliere solo D

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

Split horizon con poison reverse

- In questo caso, i Distance Vector inviati da “i” a “j” contengono esplicitamente un valore pari ad infinito nelle righe in cui “i” ha come next hop “j”
- Esempio

Tabella del nodo C

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

DV inviato da C a B

da C	dist
A	inf
B	inf
C	0
D	1

Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	inf	-
C	inf	-
D	inf	-

da D	dist
A	2
B	1
C	1
D	0

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

B

da B	dist	next
A	inf	-
B	0	B
C	1	C
D	1	D

da C	dist
A	inf
B	inf
C	0
D	1

da D	dist
A	inf
B	inf
C	1
D	0

da B	dist	next
A	inf	-
B	0	B
C	1	C
D	1	D

C

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

da B	dist
A	inf
B	0
C	inf
D	1

da D	dist
A	2
B	1
C	inf
D	0

da C	dist	next
A	3	D
B	1	B
C	0	C
D	1	D

D

da D	dist	next
A	2	B
B	1	B
C	1	C
D	0	D

da A	dist
A	0
B	inf
C	inf
D	inf

da B	dist
A	inf
B	0
C	1
D	inf

da C	dist
A	2
B	1
C	0
D	inf

da D	dist	next
A	3	C
B	1	B
C	1	C
D	0	D

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	5	-
C	5	-
D	4	-

da B	dist	next
A	inf	-
B	0	B
C	1	C
D	1	D

da C	dist	next
A	3	D
B	1	B
C	0	C
D	1	D

da D	dist	next
A	3	C
B	1	B
C	1	C
D	0	D

B

da A	dist
A	0
B	inf
C	inf
D	inf

da C	dist
A	3
B	inf
C	0
D	1

da D	dist
A	3
B	inf
C	1
D	0

C

da A	dist
A	inf
B	0
C	inf
D	1

da B	dist
A	inf
B	1
C	inf
D	0

da D	dist
A	2
B	1
C	1
D	0

D

da A	dist
A	0
B	inf
C	inf
D	inf

da B	dist
A	inf
B	0
C	1
D	inf

da C	dist
A	inf
B	1
C	0
D	inf

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da B	dist	next
A	4	C
B	0	B
C	1	C
D	1	D

da C	dist	next
A	inf	-
B	1	B
C	0	C
D	1	D

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

Distance Vector ricevuti dai vicini alla successiva iterazione

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

da B	dist	next
A	4	C
B	0	B
C	1	C
D	1	D

da C	dist	next
A	inf	-
B	1	B
C	0	C
D	1	D

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

B

da A	dist
A	2
B	1
C	1
D	0

da C	dist
A	inf
B	inf
C	0
D	1

da D	dist
A	4
B	inf
C	1
D	0

C

da B	dist
A	inf
B	0
C	inf
D	1

da D	dist
A	4
B	1
C	inf
D	0

D

da A	dist
A	0
B	inf
C	inf
D	inf

da B	dist
A	4
B	0
C	1
D	inf

da C	dist
A	inf
B	1
C	0
D	inf

da A	dist	next
A	0	A
B	5	D
C	5	D
D	4	D

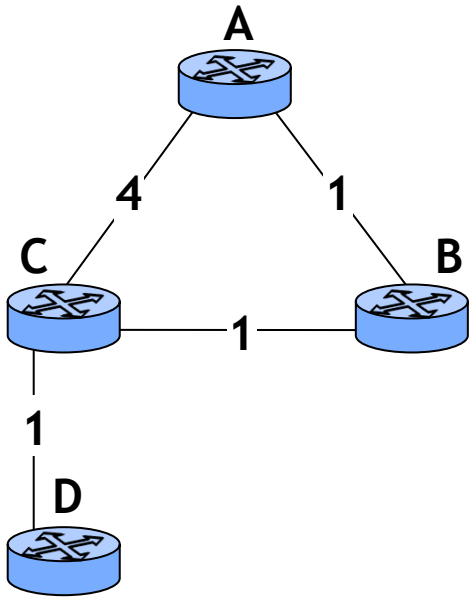
da B	dist	next
A	5	D
B	0	B
C	1	C
D	1	D

da C	dist	next
A	5	D
B	1	B
C	0	C
D	1	D

da D	dist	next
A	4	A
B	1	B
C	1	C
D	0	D

Esercizio 2

- Con riferimento alla rete in figura, ove e' utilizzato l'algoritmo Distributed Bellman-Ford (DBF) classico senza alcun meccanismo aggiuntivo
  - Si indichi quale sarà la tabella di routing dei diversi nodi a regime
  - Si mostrino i messaggi scambiati nel caso in cui il link tra A e C cambi costo, da 4 a 1
  - Si mostrino i messaggi scambiati nel caso in cui il link tra C e D si guasti (evento successivo al cambio del costo del link A-C da 4 a 1)
  - Nel caso in cui l'algoritmo implementi split-horizon con poison-reverse, si mostrino i messaggi scambiati nel caso in cui il link tra C e D si guasti (evento successivo al cambio del costo del link A-C da 4 a 1)



Esercizio 2 - Soluzione

Tabelle a regime

da A	dist	next
A	0	A
B	1	B
C	2	B
D	3	B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

da C	dist	next
A	2	B
B	1	B
C	0	C
D	1	D

da D	dist	next
A	3	C
B	2	C
C	1	C
D	0	D

Tabelle subito dopo il cambio di costo del link A-C

da A	dist	next
A	0	A
B	1	B
C	1	C
D	3	B

da C	dist	next
A	1	A
B	1	B
C	0	C
D	1	D

Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	1	B
C	1	C
D	3	B

da B	dist
A	1
B	0
C	1
D	2

da C	dist
A	1
B	1
C	0
D	1

da A	dist	next
A	0	A
B	1	B
C	1	C
D	2	C

B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

da A	dist
A	0
B	1
C	1
D	3

da C	dist
A	1
B	1
C	0
D	1

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	1	D

da A	dist
A	0
B	1
C	1
D	3

da B	dist
A	1
B	0
C	1
D	2

da C	dist	next
A	1	A
B	1	B
C	0	C
D	1	D

D

da D	dist	next
A	3	C
B	2	C
C	1	C
D	0	D

da C	dist
A	1
B	1
C	0
D	1

da D	dist	next
A	2	C
B	2	C
C	1	C
D	0	D

Dopo un' iterazione siamo gia' a regime! (le tabelle sono stabili)

Tabelle subito dopo il cambio il guasto del link C-D

da C	dist	next
A	1	A
B	1	B
C	0	C
D	inf	-

da D	dist	next
A	inf	-
B	inf	-
C	inf	-
D	0	D

Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l' iterazione

Distance Vector ricevuti dai vicini

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	1	B
C	1	C
D	2	C

da B	dist
A	1
B	0
C	1
D	2

da C	dist
A	1
B	1
C	0
D	inf

da A	dist	next
A	0	A
B	1	B
C	1	C
D	3	B

da B	dist
A	1
B	0
C	1
D	3

da C	dist
A	1
B	1
C	0
D	3

da A	dist	next
A	0	A
B	1	B
C	1	C
D	4	B

B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

da A	dist
A	0
B	1
C	1
D	2

da C	dist
A	1
B	1
C	0
D	inf

da B	dist	next
A	1	A
B	0	B
C	1	C
D	3	A

da A	dist
A	0
B	1
C	1
D	3

da C	dist
A	1
B	1
C	0
D	3

da B	dist	next
A	1	A
B	0	B
C	1	C
D	4	A

C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	inf	-

da A	dist
A	0
B	1
C	1
D	2

da B	dist
A	1
B	0
C	1
D	2

da C	dist	next
A	1	A
B	1	B
C	0	C
D	3	A

da A	dist
A	0
B	1
C	1
D	3

da B	dist
A	1
B	0
C	1
D	3

da C	dist	next
A	1	A
B	1	B
C	0	C
D	4	A

poteva scegliere anche B

	Distance Vector ricevuti dai vicini	Tabelle dopo l' iterazione	Distance Vector ricevuti dai vicini	Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	1	B
C	1	C
D	4	B

da B	dist
A	1
B	0
C	1
D	4

da C	dist
A	1
B	1
C	0
D	4

da A	dist	next
A	0	A
B	1	B
C	1	C
D	5	B

da B	dist
A	1
B	0
C	1
D	5

da C	dist
A	1
B	1
C	0
D	4

da A	dist	next
A	0	A
B	1	B
C	1	C
D	6	B

B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	4	A

da A	dist
A	0
B	1
C	1
D	4

da C	dist
A	1
B	1
C	0
D	4

da B	dist	next
A	1	A
B	0	B
C	1	C
D	5	A

da A	dist
A	0
B	1
C	1
D	5

da C	dist
A	1
B	1
C	0
D	4

da B	dist	next
A	1	A
B	0	B
C	1	C
D	6	A

C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	4	A

da A	dist
A	0
B	1
C	1
D	4

da B	dist
A	1
B	0
C	1
D	4

da C	dist	next
A	1	A
B	1	B
C	0	C
D	5	A

da A	dist
A	0
B	1
C	1
D	5

da B	dist
A	1
B	0
C	1
D	5

da C	dist	next
A	1	A
B	1	B
C	0	C
D	6	A

...all' infinito

Soluzione con Split Horizon (+ poison reverse)

Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l' iterazione

Distance Vector ricevuti dai vicini

Tabelle dopo l' iterazione

A

da A	dist	next
A	0	A
B	1	B
C	1	C
D	2	C

da B	dist
A	inf
B	0
C	1
D	2

da C	dist
A	inf
B	1
C	0
D	inf

da A	dist	next
A	0	A
B	1	B
C	1	C
D	3	B

da B	dist
A	inf
B	0
C	1
D	inf

da C	dist
A	1
B	1
C	0
D	inf

da A	dist	next
A	0	A
B	1	B
C	1	C
D	inf	-

B

da B	dist	next
A	1	A
B	0	B
C	1	C
D	2	C

da A	dist
A	0
B	inf
C	1
D	2

da C	dist
A	1
B	inf
C	0
D	inf

da B	dist	next
A	1	A
B	0	B
C	1	C
D	3	A

da A	dist
A	0
B	inf
C	1
D	inf

da C	dist
A	1
B	1
C	0
D	inf

da B	dist	next
A	1	A
B	0	B
C	1	C
D	inf	-

C

da C	dist	next
A	1	A
B	1	B
C	0	C
D	inf	-

da A	dist
A	0
B	1
C	inf
D	inf

da B	dist
A	1
B	0
C	inf
D	inf

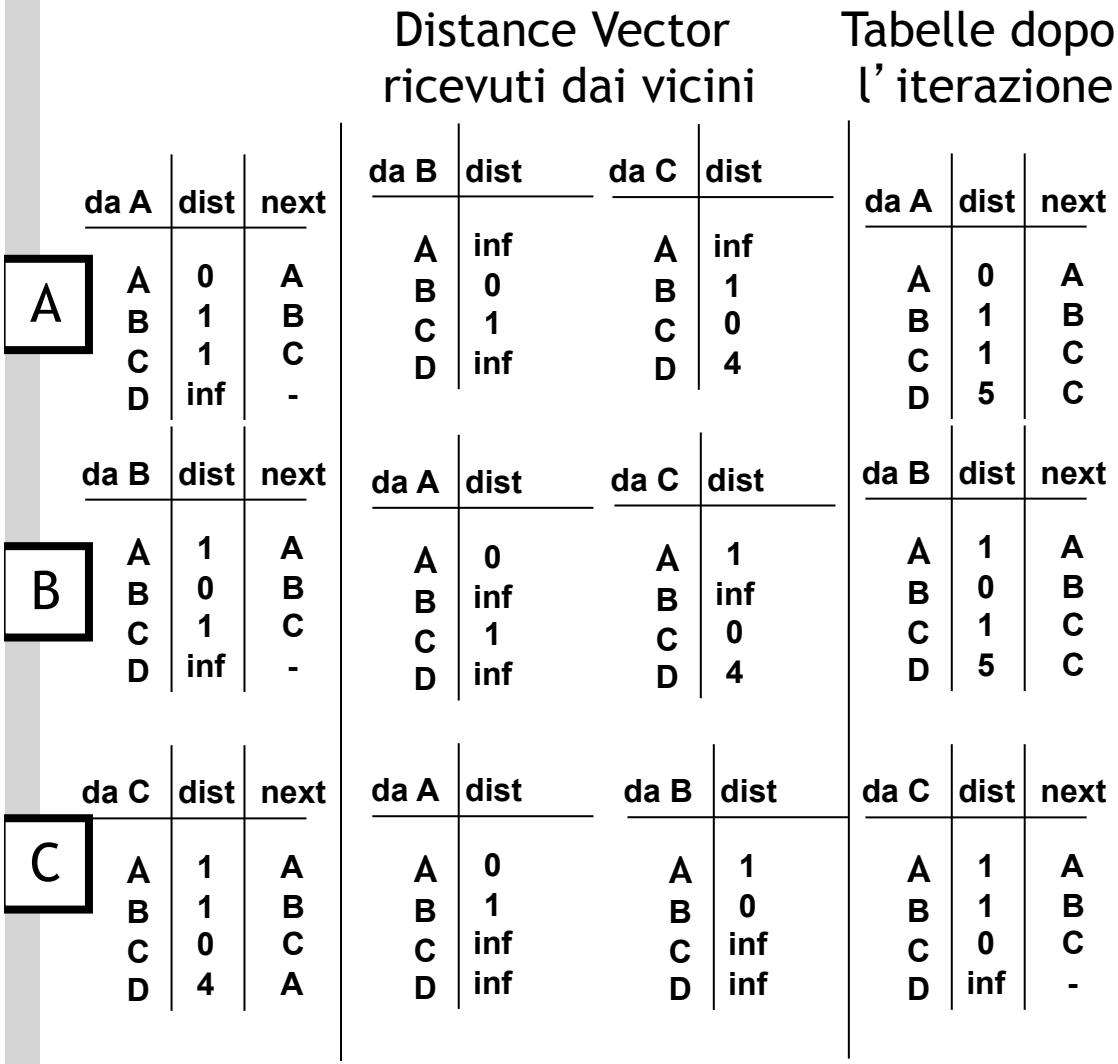
da C	dist	next
A	1	A
B	1	B
C	0	C
D	inf	-

da A	dist
A	0
B	1
C	1
D	3

da B	dist
A	1
B	0
C	1
D	3

da C	dist	next
A	1	A
B	1	B
C	0	C
D	4	A

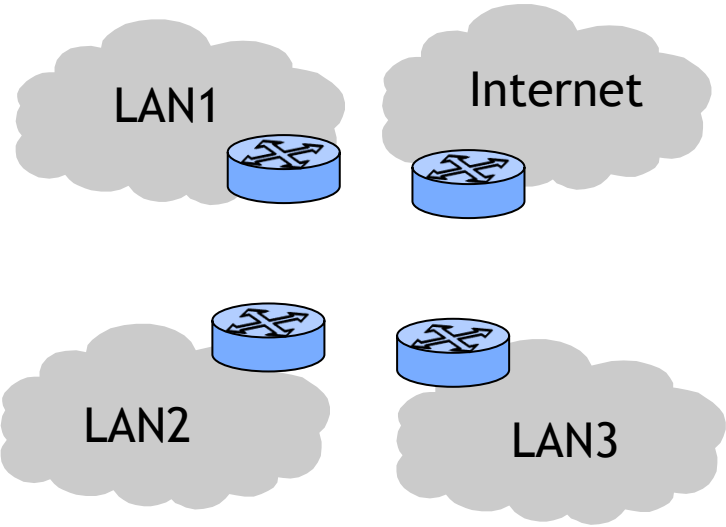
Soluzione con Split Horizon (+ poison reverse)



Siamo tornati al punto di partenza, con la distanza verso D uguale a 5 invece che uguale a 2! Anche qui l' iterazione procede all' infinito

Esercizio 3

- Si consideri la rete rappresentata in figura a lato,
  - i quattro router (RA, RB, RC e RD) sono connessi tra loro da canali punto-punto;
  - sulla rete è in funzione un protocollo di routing di tipo Distance Vector che implementa split-horizon con poison-reverse,
  - La metrica utilizzata e' il numero di hop;
  - i distance-vector inviati dai router RA/B/C/D su ciascuna delle loro interfacce sono



	Router A		Router B			Router C		Router D		
	Interf-1	Interf-2	Interf-1	Interf-2	Interf-3	Interf-1	Interf-2	Interf-1	Interf-2	Interf-3
→ LAN 1	inf	1	2	inf	2	4	inf	3	3	inf
→ LAN 2	2	inf	inf	1	1	3	inf	2	2	inf
→ LAN 3	4	inf	3	3	inf	inf	1	2	inf	2
→Internet	3	inf	2	2	inf	2	inf	inf	1	1

Esercizio 3 (cnt'd)

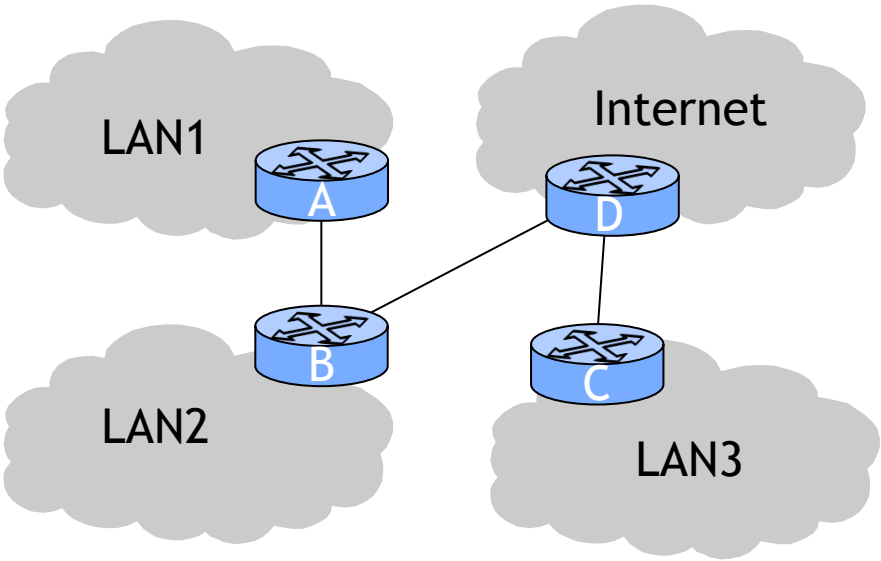
- Domande:
  - Si disegni la topologia del backbone.
  - Si scrivano le tabelle di routing dei router RA/B/C/D.
  - Si dica se in caso di guasti ad uno qualsiasi dei canali punto-punto si possano verificare dei routing loop. Se si, se ne specifichi la natura (permanenti, transitori, ...). Si motivi la risposta.

Esercizio 3: soluzione

- Router A
  - Interf.1: da A a LAN1
  - Interf.2: da A a B
- Router B
  - Interf.1: da B a LAN2
  - Interf.2: da B a A
  - Interf.3: da B a D
- Router C
  - Interf.1: da C a LAN3
  - Interf.2: da C a D
- Router D
  - Interf.1: da D a Internet
  - Interf.2: da D a B
  - Interf.3: da D a C
- In caso di guasti ai link, la rete viene partizionata, per cui ci saranno sicuramente dei routing loop permanenti (vedi esercizio precedente)

da A	dist	next
LAN1	1	dir
LAN2	2	B
LAN3	4	B
Internet	3	B

da D	dist	next
LAN1	3	B
LAN2	2	B
LAN3	2	C
Internet	1	dir



da B	dist	next
LAN1	2	A
LAN2	1	dir
LAN3	3	D
Internet	2	D

da C	dist	next
LAN1	4	D
LAN2	3	D
LAN3	1	dir
Internet	2	D