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 updat

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)

Il nodo "i" aggiorna la propria tabella

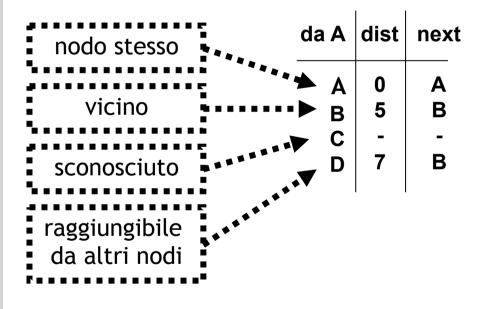
Se arriva un update negativo, lo dobbiamo registrare

- ☐ 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)



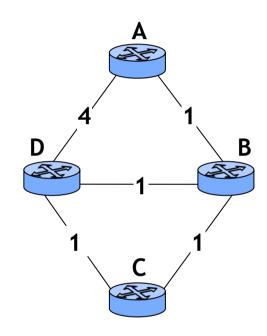
Distance Vector (ad es. inviato da A)

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 splithorizon 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 B C	0 1 2 2	A B B
ט		

dopo il guasto del link A-D

→ nessun cambiamento

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

Tabelle subito dopo il guasto del link A-B

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



Tabelle dopo il guasto

da A dist next

da C

Distance Vector ricevuti dai vicini

Tabelle dopo l'iterazione

А	A B C D	0 inf inf inf	A - -
	da B	dist	next
В	A B C D	inf 0 1	B C D
	da C	dist	next
С	A B C D	2 1 0 1	B B C D
	da D	dist	next
D	A B C D	2 1 1 0	B B C D
	<i>/</i> 1		

	da	D	dist							
		A B C D	2 1 1 0							
da C	dist			da D	dist		_			
A B C D	2 1 0 1			A B C D	2 1 1 0					
	da	В	dist		da	a D	dist		_	
	I	A B C D	inf 0 1	ı		A B C D	2 1 1 0		I	
da A	dist		_ <u>d</u>	la B	dist		da	С	dist	
A B C D	0 inf inf inf			A B C D	inf 0 1			A B C D	2 1 0 1	

	1	1	
da A	dist	next	
A B C D	0 5 5 4	A D D	
da B	dist	next	
A B C D	3 0 1 1	C ← B C D	poteva scegliere anche D
da C	dist	next	
A B C D	3 1 0 1	D B C D	
A	3 1 0	D B C	

Tabelle dopo l'iterazione

C 5 D D 4 D	Α	A B C D	0 5 5 4	
-------------	---	------------------	------------------	--

da A | dist | next

	Α	3	С
D	В	0	В
В	С	1	С
	D	1	D

da C | dist | next

Α	3	D
В	1	В
С	0	С
D	1	D

	da D	dist	next
	Α	3	С
D	В	1	В
	С	1	С
	D	0	D

da D	dist
A	3
B	1
C	1
D	0

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

da B	dist	da D	dist
Α	3	Α	3
В	0	В	1
B C	1	С	1
D	1	D	0

da A	dist	da B	dist	da C	dist	
A B C D	0 5 5 4	A B C D	3 0 1	A B C D	3 1 0 1	

da A	dist	next	
A B C D	0 5 5 4	A D D	
da B	dist	next	
A B C D	4 0 1 1	C ← B C D	poteva scegliere anche D
da C	dist	next	
A B	4 1 0	D ← B C	poteva scegliere anche B
A	4	 D ← _	scegliere
A B	4 1 0	D ← B C	scegliere

Tabelle dopo l'iterazione

	Α	0	Α
Α	В	5	D
<i>,</i> ,	С	5	D
	D	4	D

da A | dist | next

da B | dist | next

da C | dist | next

	Α	4	С
	В	0	В
٦ <u> </u>	С	1	С
	<u> </u>	1	ח

	A	4	D
	В	1	В
C	С	0	С
	D	1	D

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

da D	dist
A	4
B	1
C	1
D	0

da C	dist	da D	dist
A B C	4	A B C	4 1
D	1	D	Ö

da B	dist	da D	dist
Α	4	Α	4
В	0	В	1
B C	1	С	1
D	1	D	0

da A	dist	da B	dist	da C	dist
A B C D	0 5 5 4	A B C D	4 0 1	A B C D	4 1 0 1

da A	dist	next	
A B C D	0 5 5 4	A D D D	
da B	dist	next	
A B C D	5 0 1 1	C ← B C D	poteva scegliere anche D
da C	dist	next	
A B C D	5 1 0 1	D ← B C	poteva scegliere anche B
da D	dist	next	
A B C	4 1 1	A B C	¥611-510

Tabelle dopo l'iterazione

	Α	0	Α
ΙΑΙ	В	5	D
•	С	5	D
	D	4	D

da A | dist | next

Α	5	С
В	0	В
С	1	С
D	1	ח

da B dist next

da C dist next

da D | dist | next

С	A	5	D
	B	1	B
	C	0	C
	D	1	D

	Α	4	Α
D	В	1	В
	С	1	B
	D	0	D

da D	dist
A	4
B	1
C	1
D	0

da C	dist	da D	dist
A B C	5 1 0	A B C	4 1 1 0

da B	dist	da D	dist
A	5	Α	4
В	0	В	1
B C	1	С	1
D	1	, D	0

da A	dist	da B	dist	da C	dist	
A B C D	0 5 5 4	A B C D	5 0 1	A B C D	5 1 0 1	

da A	dist	next	
A B C D	0 5 5 4	A D D	
da B	dist	next	
A B C D	5 0 1 1	D ← B C D	deve scegliere D
da C	dist	next	
da C A B C D	5 1 0	next D ← B C D	poteva scegliere solo D
A B C	5 1 0	D ← B C	scegliere

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

	da A	aist	next
	A B	0 inf	A
$I \cap I$	C	inf	_
	D	inf	-
	da B	dist	next
	Α	inf	-
B	В	0	В
	С	1	С
	D	1	D
	da C	dist	next
	Α	2	В
$I \cap I$		1	В
C	В	2 1 0	B C
С		1 0 1	
С	B C		С
<u>с</u>	B C D	dist	C D
С	B C D	dist	C D next
C	B C D	dist	C D next

	da	D	dist						
		A B C D	2 1 1 0						
da C	dist			da D	dist		_		
A B C D	inf inf 0			A B C D	inf inf 1				
	da	В	dist		da	a D	dist		_
	1	A B C D	inf 0 inf 1			A B C D	2 1 inf 0		ı
da A	dist		_ <u>d</u>	аВ	dist		da	С	dist
A B C D	0 inf inf inf			A B C D	inf 0 1 inf			A B C D	2 1 0 inf

da A	dist	next
A B C D	0 5 5 4	A D D
da B	dist	next
A B C D	inf 0 1	B C D
	I	
da C	dist	next
A B C D	3 1 0 1	D B C D
A B C	3 1 0	D B C



da C

da A

Tabelle dopo l'iterazione

	da A	dist	next
	Α	0	Α
			-
	B C	5 5	-
	D	4	-
	da B	dist	next
	Α	inf	_
\Box	В	0	В
B	C	1	B C
	Ď	1	D
	da C	dist	next
	A		D B
С	A B		D B
С	A	3 1 0 1	D
С	A B C	3 1 0	D B C
С	A B C D	3 1 0 1	D B C D
С	A B C D	3 1 0 1 dist	D B C D
C	A B C D	3 1 0 1	D B C D

		İ					
	da D	dist					
	Α	2					
	В	1					
	B C D	1 1 0					
	D	0					
ı			ı				
C	dist	da D	dist		_		
	3	Α	3				
	inf	B C D	inf				
	0	С	1				
	1	D	0				
	da B	dist	da	a D	dist		_
	Α	inf		Α	inf		
	В	0		В	1		
	C D	inf		С	inf		
	D	1	ı	D	0		I
	dist	da B	dist		da	С	dist
	0	Α	inf			A	inf
	inf	В	0		Į	В	1
	inf	С	1			C	0
	inf	D	inf			D	inf

da A	dist	next
A B C D	0 5 5 4	A D D
da B	dist	next
A B C D	4 0 1 1	C B C D
da C	dist	next
A B C D	inf 1 0	next - B C D
A B C	inf 1 0	- В С



da C

Tabelle dopo l'iterazione

	da A	dist	next
	Α	0	Α
	В		D
	C	5 5	D
	D	4	D
	da B	dist	next
	Α	4	С
В	В		В
B	C	0 1	С
	D	1	D
	da C	dist	next
	Α	inf	-
	В	1	В
	С	0	С
	D	1	D
	da D	dist	next
	Α	4	A
ן ח ו		1	В
	В	•	_
	B C	1	C
	C B	1 1 0	

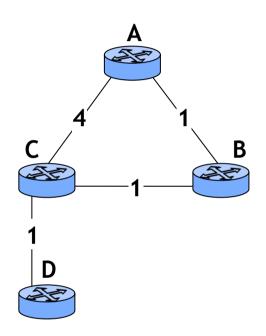
	da	D	dist							
		A B C D	2 1 1 0							
da C	dist			da D	dist		_			
A B C D	inf inf 0 1			A B C D	4 inf 1 0					
	da	В	dist		da	D	dist		_	
	ı	A B C D	inf 0 inf 1		ı	A B C D	4 1 inf 0		ı	
da A	dist			la B	dist		da	С	dist	_
A B C D	0 inf inf inf			A B C D	4 0 1 inf			A B C D	inf 1 0 inf	

da A	dist	next
A B C D	0 5 5 4	A D D
da B	dist	next
A B C D	5 0 1 1	D B C D
da C	dist	next
A B C D	5 1 0 1	D B C D
A B C	5 1 0	D B C



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 splithorizon 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	nex
A B C D	0 1 2 3	A B B
da B	dist	nex
A B C D	1 0 1 2	A B C C
da C	dist	nex
A B C D	2 1 0 1	B B C D
B C	0	B C

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	nex
Α	1	Α
В	1	В
С	0	С
D	1	D



Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l'iterazione

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

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

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

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

Α	1	A
A B C D	1 0	B C
С	0	С
D	inf	-
da D	dist	next
	inf	
A	inf	_
D		
A B C D	inf	_

da C | dist | next



Tabelle dopo il guasto

Distance Vector ricevuti dai vicini

Tabelle dopo l'iterazione

anche B

Distance Vector ricevuti dai vicini

Tabelle dopo l'iterazione

4	<u>d</u>	A B C D	0 1 1 2	next A B C C	da B A B C D	1 0 1 2	da C A B C D	dist 1 1 0 inf	da A A B C D	0 1 1 3	next A B C B
	da	а В	dist	next	da A	dist	da C	dist	da B	dist	next
[3	A B C D	1 0 1 2	A B C C	A B C D	0 1 1 2	A B C D	1 1 0 inf	A B C D	1 0 1 3	A B C A
	da	а С	dist	next	da A	dist	da B	dist	da C	dist	next
		A B C D	1 1 0 inf	A B C	A B C D	0 1 1 2	A B C D	1 0 1 2	A B C D	1 1 0 3	A B C A
									9	pote cegli	

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

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

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



Distance Vector ricevuti dai vicini			Tabelle dopo l'iterazione				Distance Vector ricevuti dai vicini				Tabelle dopo l'iterazione						
da A	A A B C D	0 1 1 4	next A B C B	da B A B C D	1 0 1 4	da C A B C D	dist 1 1 0 4	da A A B C D	0 1 1 5	next A B C B	da B A B C D	dist 1 0 1 5	da C A B C D	dist 1 1 0 4	- da A A B C D	0 1 1 6	next A B C B
da B	A B C D	1 0 1 4	next A B C A	A B C D	0 1 1 4	da C A B C D	1 1 0 4	A B C D	1 0 1 5	next A B C A	da A A B C D	0 1 1 5	da C A B C D	1 1 0 4	A B C D	1 0 1 6	next A B C A
da C	A B C D	1 1 0 4	A B C A	A B C D	0 1 1 4	da B A B C D	1 0 1 4	da C A B C D	1 1 0 5	next A B C A	da A A B C D	0 1 1 5	da B A B C D	1 0 1 5	da C A B C D	1 1 0 6	next A B C A

103

...all' infinito

Soluzione con Split Horizon (+ poison reverse)

ricevuti dai vicini l' iterazione ricevuti dai A					
ricevuti dai vicini l' iterazione ricevuti dai A	ector	Distance Vect	Tabelle dopo		
A dist next A inf A inf A 0 A B 0 B 1 B C 1 C D 2 D inf D 3 B C 1 C D 3 B C C C C C C C C C		icevuti dai vi		zione	
A	dist	dist da C	dist	next	
	3 1	0 B C	0 1 1 inf	A B C	
da B dist next da A dist da C dist da B dist next da A dist da C	dist d	dist da C	dist	next	
B B 0 B inf B inf B inf C 1 C 1 C 1 C 1	3 1 0	inf B C	1 0 1 inf	A B C	
		-	dist	next	
		7	1 1	A B	
	C 1	1 c	0 4	C CII STUD	

Soluzione con Split Horizon (+ poison reverse)

			Dis	tanc	e Vecto	or	Tabe	lle d	dopo
		,	rice	vuti	dai vic	ini	l' it€	eraz	ione
da A	dist	next	da B	dist	da C	dist	─ da A	dist	next
A B C D	0 1 1 inf	A B C	A B C D	inf 0 1 inf	A B C D	inf 1 0 4	A B C	0 1 1 5	A B C
da B	dist	next	da A	dist	da C	dist	da B	dist	next
A B C D	1 0 1 inf	A B C	A B C D	0 inf 1 inf	A B C D	1 inf 0 4	A B C D	1 0 1 5	A B C C
da C	dist	next	da A	dist	da B	dist	da C	dist	next
A B C D	1 1 0 4	A B C A	A B C D	0 1 inf inf	A B C D	1 0 inf inf	A B C D	1 1 0 inf	A B C
	A B C D da C A B C	A 0 B 1 C 1 D inf da B dist A 1 B 0 C 1 D inf da C dist A 1 B 1 C 0	A 0 A B 1 B C 1 C D inf - da B dist next A 1 A B 0 B C 1 C D inf - da C dist next A 1 A B 1 C C 0 C	rice da A dist next A D A B C D da B C D da B dist next da A A B C D da B dist next da A A B B C C D da C dist next A D B B C C D da C dist next A A B B C C D da C dist next A A B B C C C C C C C C C C C C C C C C	A	ricevuti dai vice A	da A dist next A 0 A B 0 B 1 B 1 B C 1 C 0 C 1 C D inf D 4 D inf - D da A dist da C dist A 1 A A 0 A 1 B inf C 1 C C 1 C C 0 D A 1 D inf - D inf D A 1 A A 1 A A 1 A A 1 A A 1 B 0 A 1 B 0 A 1 B 0 A 1 B 0 A 1 B 0 A 1 B 0 A 1 B 0 <	ricevuti dai vicini l' ite da A dist next	ricevuti dai vicini l' iteraz da A dist next

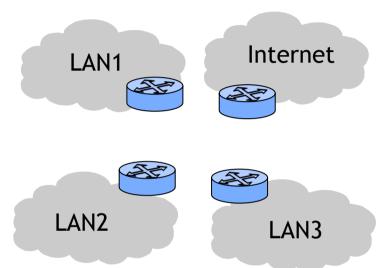
Siamo tornati al punto di partenza, con la distanza verso D uguale a 5 invece che uguale a 2!

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



	Rout	ter A	Router B			Rout	er 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.

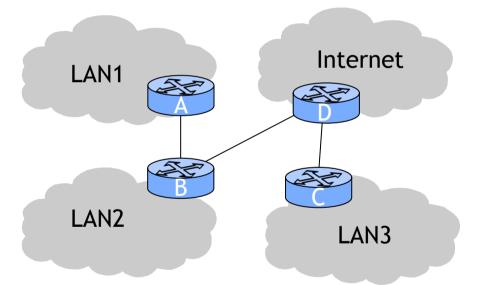


Esercizio 3: soluzione

□ 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	da D	dist	next
LAN1	1	dir	LAN1	3	В
LAN2	2	В	LAN2	2	В
LAN3	4	В	LAN3	2	С
Internet	3	В	Internet	1	dir



da B	dist	next
LAN1	2	Α
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