

Fondamenti di Internet e Reti

Esercizi

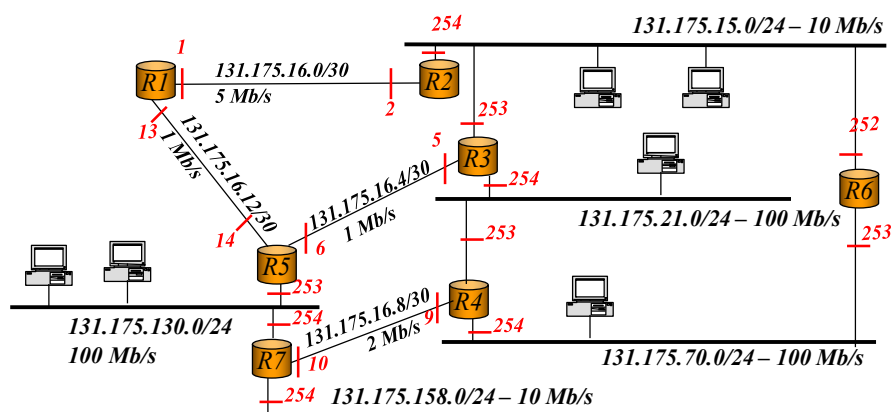
4 - Esercizi sul livello di rete **Parte B: Instradamento**

4b.1 - Esercizio

Si consideri la rete in figura. Si rappresenti, mediante un grafo, la rete per il calcolo dei cammini minimi (solo i nodi e gli archi – no reti).

1) Si calcoli il cammino minimo tra R1 e tutti gli altri nodi mediante l'algoritmo di *Dijkstra* supponendo che ciascun arco abbia peso unitario.

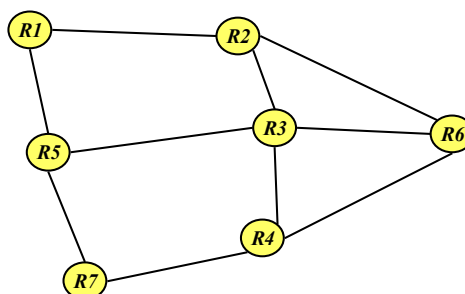
2) Si ripeta il calcolo assegnando a ciascun arco un peso pari a $100/C$ dove C è la velocità del *link* in Mb/s.



Soluzione

1)

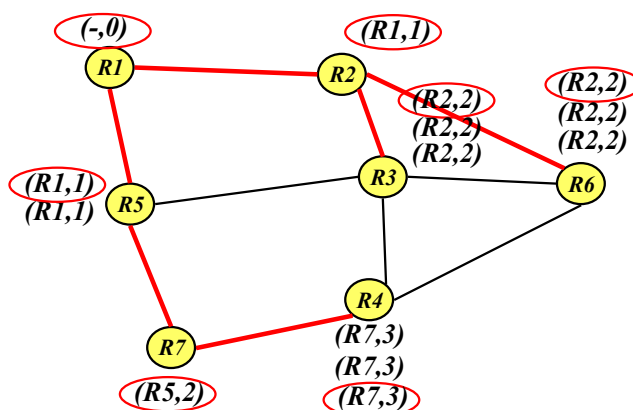
Il grafo che rappresenta la rete sopra è il seguente, rappresentando, come richiesto nel testo, solo *router* e collegamenti.



La tabella seguente mostra l'evoluzione dell'algoritmo di *Dijkstra*. Lo *Step 0* si riferisce all'inizializzazione. Ad ogni passo (*step*) sono indicate le etichette rese permanenti (in rosso) ed i *router* che non vengono considerati al passo specifico (#); si ricorda che ad ogni passo, possono essere modificate le etichette dei soli nodi che sono vicini di nodi già parte dell'albero dei cammini minimi (nodi con etichetta rossa al passo precedente).

	R1	R2	R3	R4	R5	R6	R7
Step 0	(-,0)	(-,inf)	(-,inf)	(-,inf)	(-,inf)	(-,inf)	(-,inf)
Step 1		(1,R1)	#	#	(1,R1)	#	#
Step 2			(2,R2)	#	(1,R1)	(2,R2)	#
Step 3			(2,R2)	#		(2,R2)	(2,R5)
Step 4			(2,R2)	(3,R7)		(2,R2)	
Step 5			(2,R2)	(3,R7)			
Step 6				(3,R7)			

L'albero dei cammini minimi sarà il seguente:

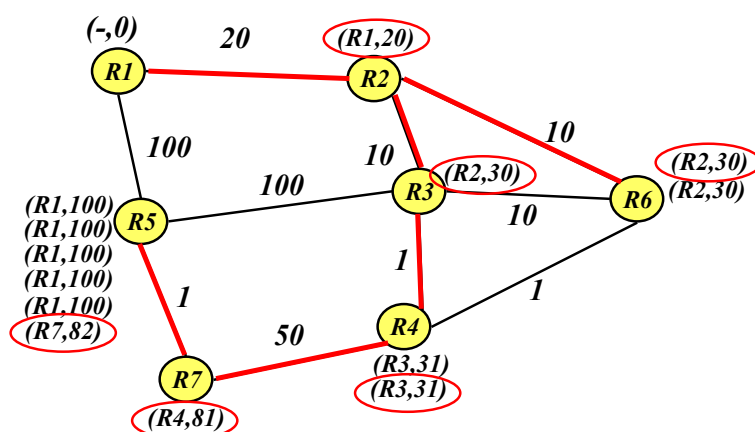


2)

Applicando l'algoritmo di *Dijkstra* con la nuova metrica $100/C$, si ottiene il seguente albero dei cammini minimi:

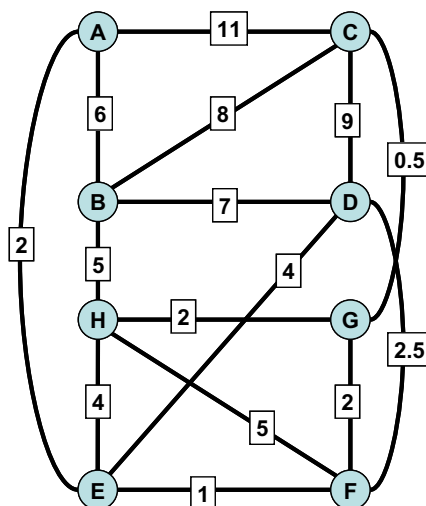
	R1	R2	R3	R4	R5	R6	R7
Step 0	(-,0)	(-,inf)	(-,inf)	(-,inf)	(-,inf)	(-,inf)	(-,inf)
Step 1		(20,R1)	#	#	(100,R1)	#	#
Step 2			(30,R2)	#	(100,R1)	(30,R2)	#
Step 3				(31,R3)	(100,R1)	(30,R2)	#
Step 4				(31,R3)	(100,R1)		#
Step 5					(100,R1)		(81,R4)
Step 6					(82,R7)		

La figura seguente mostra l'albero dei cammini minimi.



4b.2 - Esercizio

Data la rete rappresentata in figura (in cui su ogni *link* è riportato il costo) si trovi l'albero dei cammini minimi del nodo B. Si riporti nella tabella seguente ad ogni passo e per ogni nodo x l'etichetta: (D_x, p_x) , dove p_x è il nodo precedente di x nel percorso e D_x è la distanza al passo corrente del nodo x dal nodo radice.



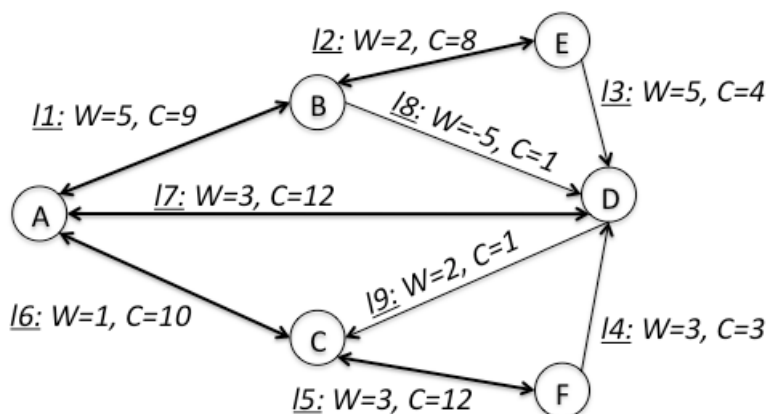
Soluzione

La tabella con l'evoluzione dell'algoritmo di *Dijkstra* è riportata qui di seguito. La notazione è la stessa dell'esercizio precedente; lo *Step 0* si riferisce all'inizializzazione. Ad ogni passo (*step*) sono indicate le etichette rese permanenti (in rosso) ed i *router* che non vengono considerati al passo specifico (#); si ricorda che ad ogni passo, possono essere modificate le etichette dei soli nodi che sono vicini di nodi già parte dell'albero dei cammini minimi (nodi con etichetta rossa al passo precedente).

	A	B	C	D	E	F	G	H
Step 0	(inf,-)	(0,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)
Step 1	(6,B)		(8,B)	(7,B)	#	#	#	(5,B)
Step 2	(6,B)		(8,B)	(7,B)	(9,H)	(10,H)	(7,H)	
Step 3			(8,B)	(7,B)	(8,A)	(10,H)	(7,H)	
Step 4			(8,B)		(8,A)	(9.5, D)	(7,H)	
Step 5			(7.5, G)		(8,A)	(9,G)		
Step 6					(8,A)	(9,G)		
Step 7						(9,G)		

4b.3 - Esercizio

Sia data la seguente rete, dove per ogni *link* è indicata la coppia (W,C), con W peso della metrica di *routing* e C capacità del *link*. Si presti attenzione alla presenza di *link* monodirezionali per cui è indicato il verso di percorrenza



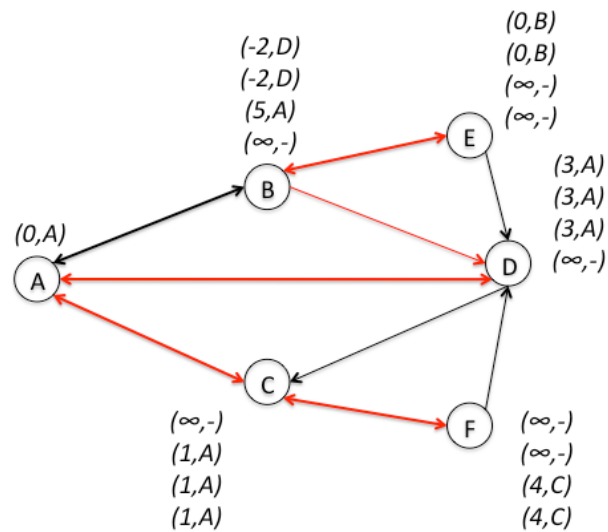
Si chiede di:

- Calcolare l'albero dei cammini minimi da tutti i nodi al nodo A (attenzione al verso di percorrenza dei *link*) con l'algoritmo più efficiente. E' necessario mostrare il processo di aggiornamento delle etichette
- Calcolare l'albero dei cammini minimi da tutti i nodi al nodo A con il vincolo che i percorsi non possano passare attraverso *link* con capacità inferiore a 5

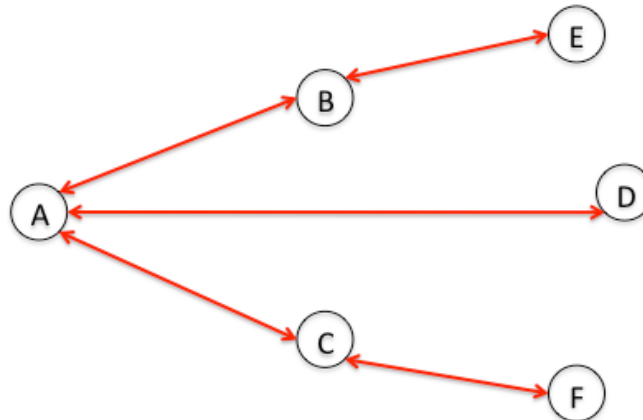
Soluzione

- Il *link* B-D ha peso negativo, quindi dobbiamo usare l'algoritmo di *Bellman-Ford*. La tabella seguente riporta l'evoluzione dell'algoritmo.

	A	B	C	D	E	F
Step 0	(0,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)
Step 1		(5,A)	(1,A)	(3,A)	(inf,-)	(inf,-)
Step 2		(-2,D)	(1,A)	(3,A)	(7,B)	(4,C)
Step 3		(-2,D)	(1,A)	(3,A)	(0,B)	(4,C)
Step 4		(-2,D)	(1,A)	(3,A)	(0,B)	(4,C)
Step 5		(-2,D)	(1,A)	(3,A)	(0,B)	(4,C)



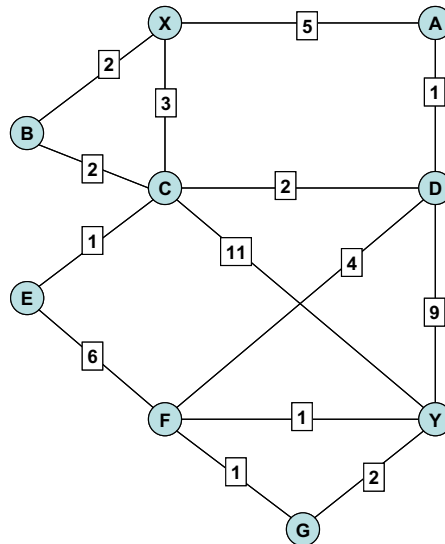
b) Eliminando gli archi con capacità $C < 5$, il grafo risulta già essere un albero. Non è necessario applicare alcun algoritmo per cammini minimi.



4b.4 - Esercizio

E' data la rete geografica a commutazione di pacchetto in figura, in cui su ogni *link* è riportato il costo. Applicare l'algoritmo di Dijkstra per trovare il percorso a minimo costo tra i nodi X e Y.

Trovare il percorso alternativo a minimo costo tra X e Y dopo aver posto a ∞ il costo dei collegamenti attraversati dal primo percorso.



Soluzione

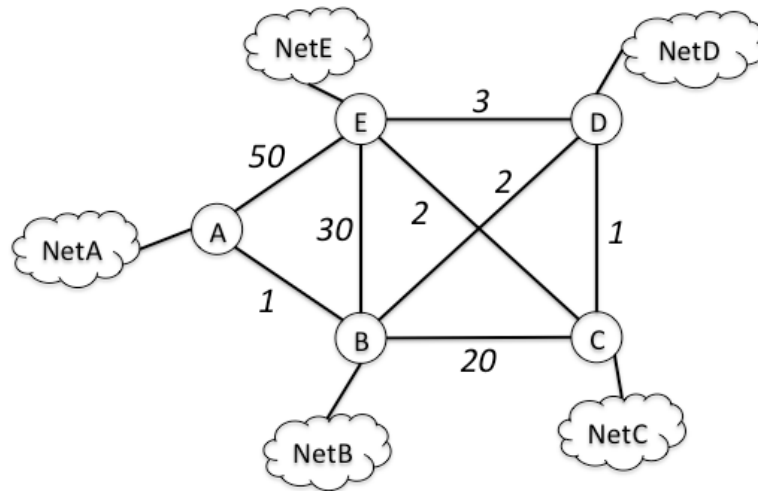
Applicando l'algoritmo di *Dijkstra* si ottiene la seguente evoluzione (la notazione in tabella è la stessa dell'Esercizio 4c-1). Lo *Step 0* si riferisce all'inizializzazione. Ad ogni passo (*step*) sono indicate le etichette rese permanenti (in rosso) ed i router che non vengono considerati al passo specifico (#); si ricorda che ad ogni passo, possono essere modificate le etichette dei soli nodi che sono vicini di nodi già parte dell'albero dei cammini minimi (nodi con etichetta rossa al passo precedente).

	X	A	B	C	D	E	F	G	Y
Step 0	(0,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)
Step 1		(5, X)	(2, X)	(3, X)	#	#	#	#	#
Step 2		(5, X)		(3, X)	#	#	#	#	#
Step 3		(5, X)			(5, C)	(4, C)	#	#	(14, C)
Step 4		(5, X)			(5, C)		(10, E)	#	(14, C)
Step 5		(5, X)					(9, D)	#	(14, C)
Step 6							(9, D)	#	(14, C)
Step 7								(10, F)	(10, F)
Step 8									(10, F)

Il percorso a costo minimo tra X e Y è: X-C-D-F-Y (costo: 10)

Eliminando il suddetto percorso e ripetendo il procedimento sul nuovo grafo di rete, il secondo percorso a costo minimo tra X e Y risulta essere X-B-C-E-F-G-Y (costo: 14)

4b.5 - Esercizio



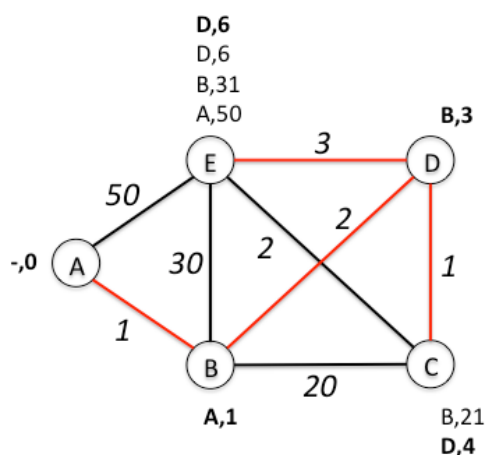
Sia data la rete in figura in cui sono indicati 5 *router* (X) e 5 reti afferenti $\text{Net}(X)$. I costi dei collegamenti (bidirezionali) sono indicati accanto ai rispettivi *link*, si assumano pari ad 1 i costi dei collegamenti tra *router* e reti. Si chiede di:

- Calcolare l'albero dei cammini minimi con destinazione in A (considerando solo i *router*). Si utilizzi l'algoritmo più efficiente, rendendo comprensibili i passaggi svolti.
- Considerando l'instradamento calcolato al punto a), indicare i *Distance vector* inviati da A. A chi vengono inviati?
- Indicare i *Distance vector* con *Split-Horizon* (non *Poisonous-Reverse*) inviati da A. A chi vengono inviati?

Soluzione

- Il grafo di rete presenta archi con pesi solo positivi. Possiamo usare l'algoritmo di *Dijkstra*.

	A	B	C	D	E
Step 0	(0,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)
Step 1		(1,A)	#	#	(50,A)
Step 2			(21,B)	(3,B)	(31,B)
Step 3			(4,D)		(6,D)
Step 4					(6,D)



- b) Il *router* A invia i propri *distance vector* ai *router* vicini B ed E. Il contenuto dei DV è il seguente:

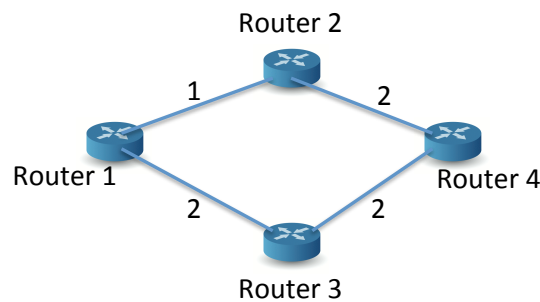
NetA,1; NetB,2; NetC,5; NetD,4; NetE,7.

- c) Il *router* A non include nei DV le destinazioni che il *router* stesso raggiunge usando come next hop il destinatario del DV. Il *router* A usa il *router* B come next hop per tutte le destinazioni (tranne la rete NetA), quindi i contenuti dei due DV inviati a B ed E sono:

NetA,1; NetB,2; NetC,5; NetD,4; NetE,7. Inviato ad E
NetA,1. Inviato a B

4b.6 - Esercizio

Scrivere il contenuto di tutti i pacchetti di *distance vector* inviati dal *router 1* agli altri *router* nei due casi: (i) *distance vector* base, (ii) *distance vector* con *Split Horizon* con *Poisonous Reverse* e *Hop-limit=16*. La figura riporta la topologia di rete e la tabella di *routing* del *router 1*.



Destinazione	Next Hop	Costo
Net 1	Router 2	4
Net 2	Router 3	3
Net 3	Router 2	2
Net 4	Router 3	3
Net 5	Local	Local

Soluzione

Distance vector base:

Il router 1 invia lo stesso messaggio a tutti i router collegati:

Net 1:4, Net2:3, Net3:2, Net4:3, Net5:local

Distance vector con *Split Horizon*

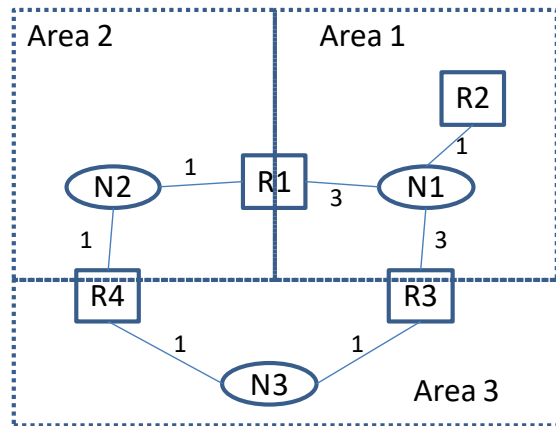
Il router 1 invia messaggi diversi a Router 2 e router 3. Le destinazioni che il router 1 raggiunge usando come next hop il destinatario del DV sono indicate con costo uguale a *hop limit* nei *distance vector* inviati.

A router 2: Net 1: 16, Net 2: 3, Net 3: 16, Net 4: 3, Net 5: local

A router 3: Net 1: 4, Net 2: 16, Net 3: 2, Net 4: 16, Net 5: local

4b.7 - Esercizio

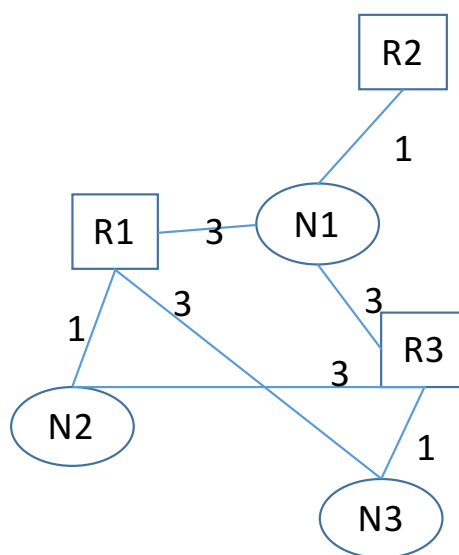
All'interno del dominio di *routing* in figura si usa il protocollo OSPF. Il dominio di *routing* è suddiviso in 3 aree come mostrato in figura (linee tratteggiate). Disegnare la topologia del dominio di *routing* “vista” dal router R2 (i numeri accanto ad ogni *link* rappresentano i costi di utilizzo delle interfacce corrispondenti).



Soluzione

Il protocollo OSPF opera secondo questi due principi: (i) ogni *router* mantiene la rappresentazione di dettaglio di tutti i *link* presenti nella sua area, (ii) ogni *router* mantiene una rappresentazione sintetica di tutte le destinazioni (reti) raggiungibili al di fuori della sua area.

Nel caso specifico, il Router R2 avrà visione di dettaglio di tutti i *link* dell'Area 1, e “vedrà” tutte le reti esterne all'Area 1 (N2 e N3) come raggiungibili sia attraverso R1 che attraverso R3.



4b.8 - Esercizio

La tabella seguente rappresenta il grafo di connettività di una rete IP con 5 nodi su cui è attivo un protocollo di *routing* di tipo *link state*.

Da	Verso	Link	Costo	Sequence Number
A	B	1	2	2
A	D	3	2	2
B	A	1	2	2
B	C	2	2	2
B	E	4	2	2
C	B	2	2	2
C	E	5	2	2
D	A	3	2	2
D	E	6	2	2
E	B	4	2	2
E	C	5	2	2

Per ciascuno dei messaggi di *Link State Update* seguenti inviati da A dire come cambia il data base di rete:

Da A, Verso B, *Link* 1, Costo=1, Sequence Number=1

Da A, Verso B, *Link* 1, Costo=3, Sequence Number =3

Da A Verso F, *Link* 7, Costo=1, Sequence Number=1

Soluzione

Da A, Verso B, *Link* 1, Costo=1, Sequence Number=1

Il messaggio di *Link State Update* ha un numero di sequenza minore rispetto all'informazione sul *link* A-B memorizzata all'interno del data base del *router*. Il messaggio viene scartato e non viene apportata alcuna modifica al data base. Eventualmente il *router* invia alla sorgente del LSU la versione aggiornata dello stato del collegamento.

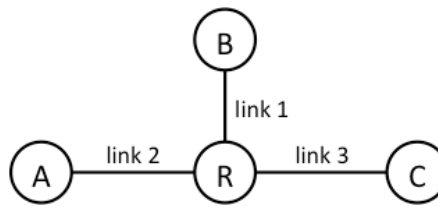
Da A, Verso B, *Link* 1, Costo=3, Sequence Number =3

Il messaggio di *Link State Update* ha un numero di sequenza maggiore rispetto all'informazione sul *link* A-B memorizzata all'interno del data base del *router*. La prima riga del data base diventa: da: A, verso: B, *link*: 1, costo:3, SN:3

Da A Verso F, *Link* 7, Costo=1, Sequence Number=1

Il *link* 7 (A-F) non è noto al *router*. Il *router* aggiunge una riga al proprio data base del tipo: da: A, verso: F, *link*:7, costo: 1, SN: 1

4b.9 - Esercizio



In figura sono rappresentati i *router* A, B, C e R, i costi di attraversamento di ogni *link* sono tutti uguali e pari a 2. Le informazioni di raggiungibilità del *router* R sono le seguenti:

Destination	Cost	Next Hop
Net A	8	B
Net B	6	B
Net C	5	A
Net D	2	C

Il *router* R riceve dal *link* 1 il seguente DV: (NetA,4), (NetB,7), (NetC,4) e successivamente dal *link* 2 il seguente DV: (NetB,2), (NetC,3), (NetE,7).

Si indichino:

- Le informazioni di raggiungibilità di R dopo il primo DV
- Le informazioni di raggiungibilità di R dopo il secondo DV
- A valle del secondo DV, il DV inviato da R nella modalità *Split Horizon* con *Poisonous Reverse*.

Soluzione

Dopo primo DV

Net	NH	Cost
NetA	B	6
NetB	B	9
NetC	A	5
NetD	C	2

Dopo secondo DV

Net	NH	Cost
NetA	B	6
NetB	A	4
NetC	A	5
NetD	C	2
NetE	A	9

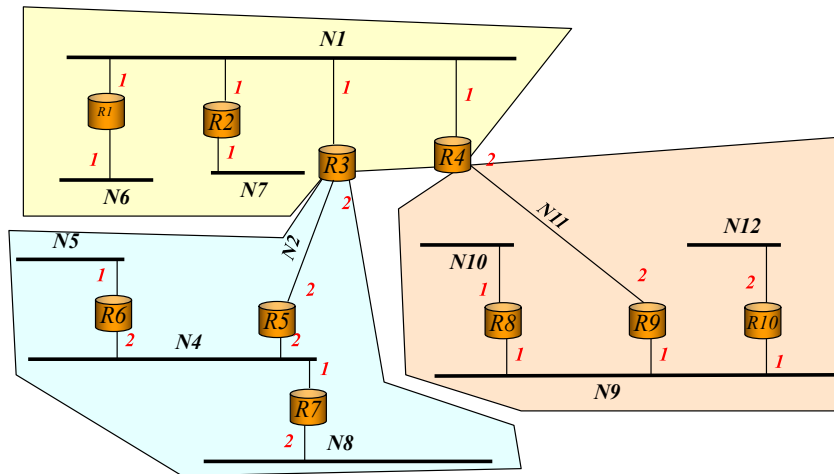
DV inviato ad A: (NetA,6),(NetB,infty),(NetC,infty),(NetD,2),(NetE,infty)

DV inviato ad B: (NetA,infty),(NetB,4),(NetC,5),(NetD,2),(NetE,9)

DV inviato ad C: (NetA,6),(NetB,4),(NetC,5),(NetD,infty),(NetE,9)

4b.10 - Esercizio

Si consideri la rete in figura dove sono indicati *router*, reti e costo associato alle interfacce dei *router*.

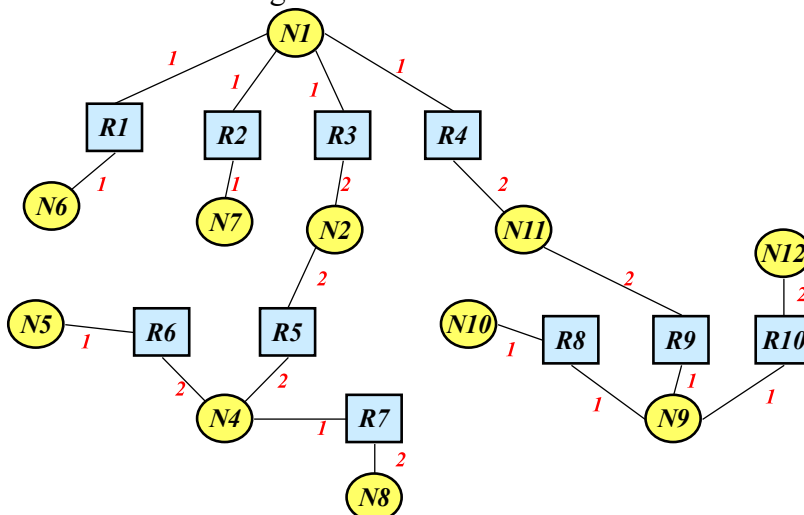


Si supponga di utilizzare il protocollo di *routing* OSPF. Si disegni il grafo della rete nell'ipotesi che si utilizzi una sola area per l'intera rete (si indichi un nodo per ogni *router* – quadrato - e per ogni rete –tondo). Si divida come mostrato in figura la rete in tre aree (area 0, area 1 e area 2) e si disegnino i grafi che rappresentano la rete vista dal *router* R1, R7 ed R10.

Soluzione

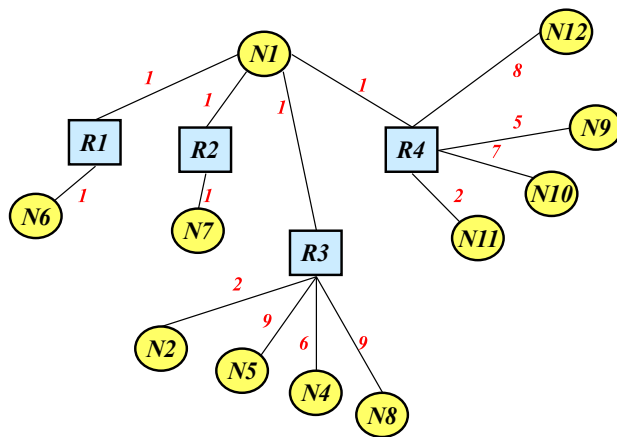
Unica area

Nel caso il dominio di *routing* non sia partizionato in aree (unica area), tutti i *router* del dominio hanno informazione di dettagli su tutti i *link* del dominio.



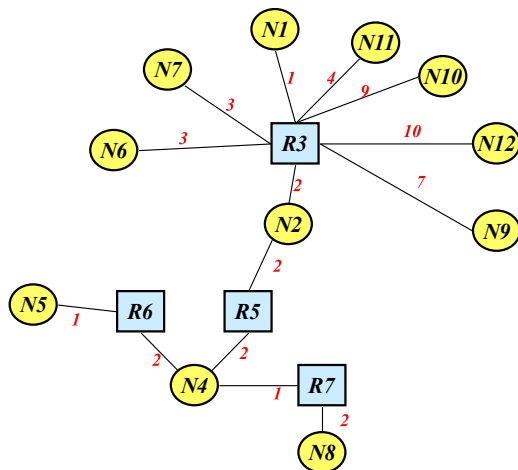
Visto da R1

Nel caso specifico, il *Router* R1 avrà visione di dettaglio di tutti i *link* della sua area, e “vedrà” tutte le reti esterne alla sua area come raggiungibili attraverso R3 e R4.



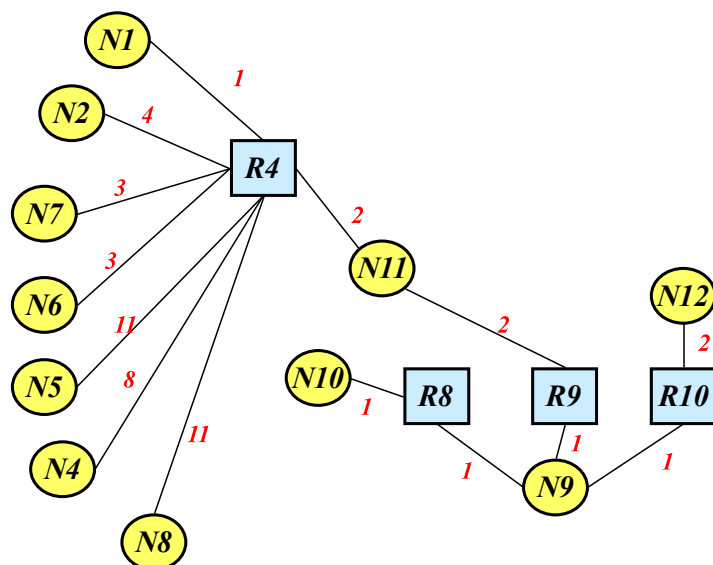
Visto da R7

Nel caso specifico, il Router R7 avrà visione di dettaglio di tutti i *link* della sua area, e “vedrà” tutte le reti esterne alla sua area come raggiungibili attraverso R3.



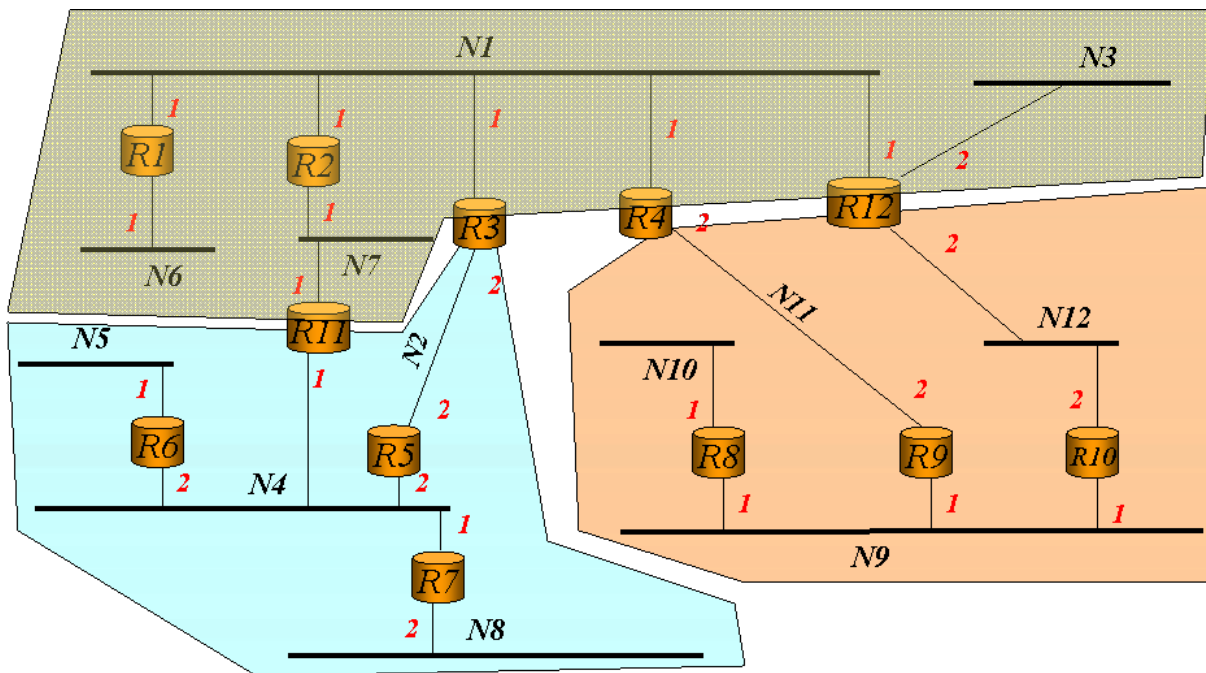
Visto da R10

Nel caso specifico, il Router R10 avrà visione di dettaglio di tutti i *link* della sua area, e “vedrà” tutte le reti esterne alla sua area come raggiungibili attraverso R4.



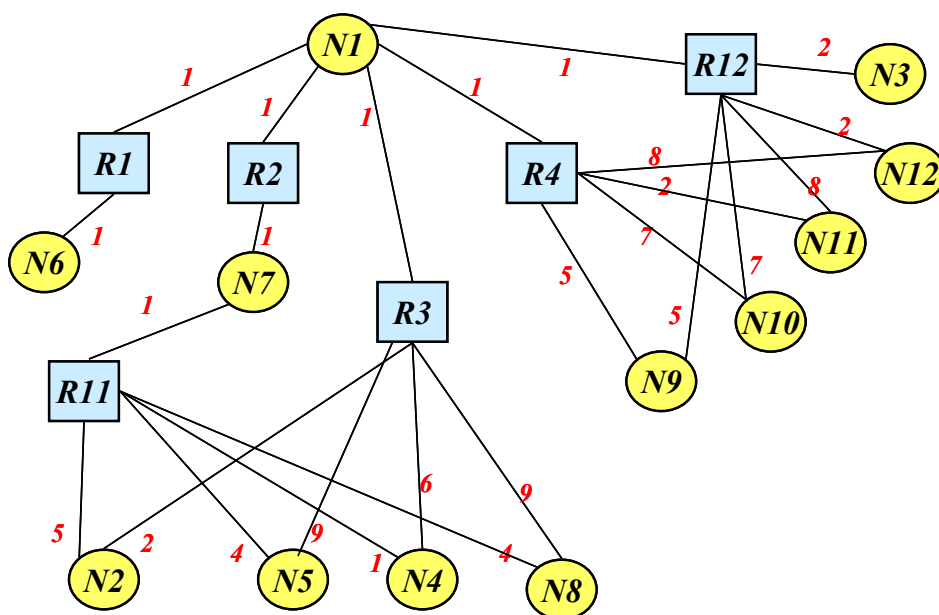
4b.11 - Esercizio

Si consideri la rete in figura dove sono indicati *router*, reti e costo associato alle interfacce dei *router*. Si supponga di utilizzare il protocollo di *routing* OSPF. Si divida come mostrato in figura la rete in 5 aree e si disegnino i grafi che rappresentano la rete vista dal *router* R1, R7, ed R8.



Soluzione

Visto da R1

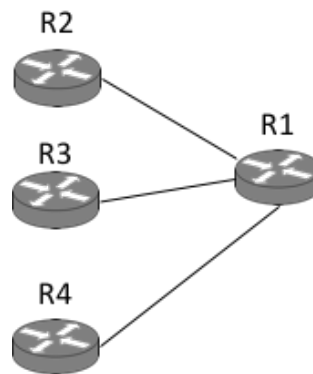


Visto da R7 e R8 è simile. I *router* vedono tutta la propria area e solo le reti esterne raggiungibili dai *router* di bordo area.

4b.12 - Esercizio

La figura riporta la topologia di rete e la tabella di *routing* di R1. Scrivere il contenuto di tutti i pacchetti di *distance vector* inviati dal router R1 agli altri router nei due casi: (a) *distance vector* base, (b) *distance vector* con *Split Horizon* con *Poisonous Reverse* e Hop-limit=16.

Network	Cost	Next-hop
1.1.1.0/24	1	R4
1.1.2.0/24	3	R2
1.1.3.0/24	2	R2
1.1.4.0/24	4	R3
1.1.5.0/24	3	R3
0.0.0.0	3	R4



Soluzione

(a)

verso tutti

1.1.1.0/24	1
1.1.2.0/24	3
1.1.3.0/24	2
1.1.4.0/24	4
1.1.5.0/24	3

(b)

verso R2

1.1.1.0/24	1
1.1.2.0/24	16
1.1.3.0/24	16
1.1.4.0/24	4
1.1.5.0/24	3

verso R3

1.1.1.0/24	1
1.1.2.0/24	3
1.1.3.0/24	2
1.1.4.0/24	16
1.1.5.0/24	16

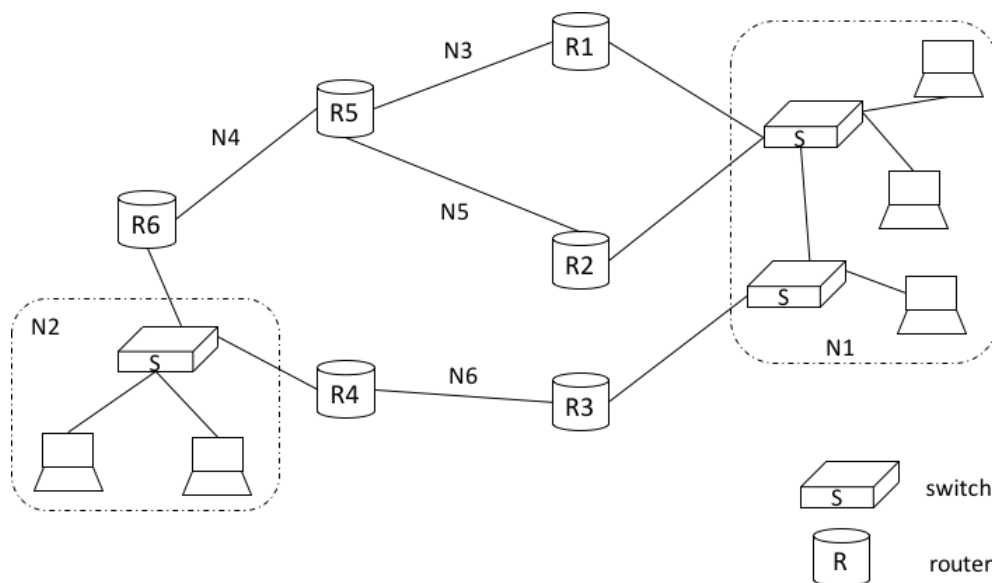
verso R4

1.1.1.0/24	16
1.1.2.0/24	3
1.1.3.0/24	2
1.1.4.0/24	4
1.1.5.0/24	3

4b.13 - Esercizio

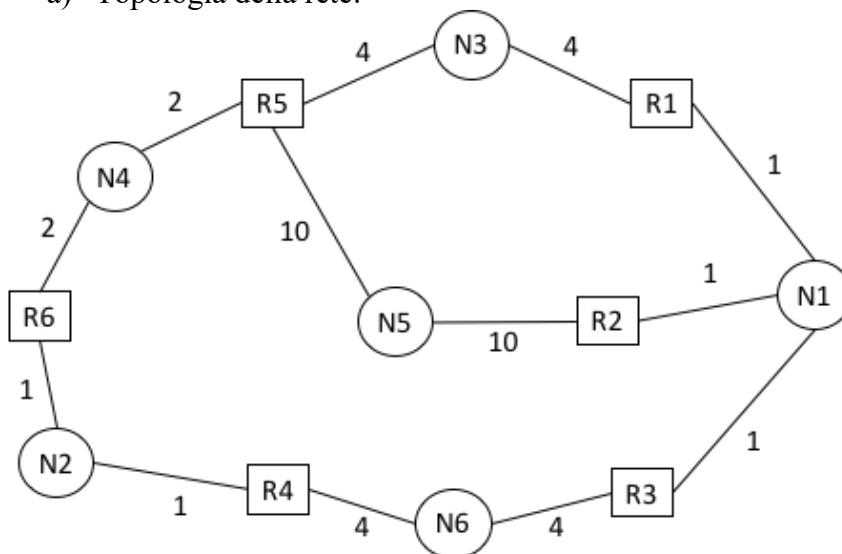
Si consideri la rete in figura composta da 6 *router* e 6 reti (inclusi i collegamenti punto-punto).

- Si rappresenti la rete con un grafo che ha come nodi i *router* (indicati con quadrati) e le reti (indicati cerchi), e come archi le interfacce dei *router* con le reti. Il peso di ciascun arco del grafo è pari a $100/C$, dove C è la capacità della rete in Mb/s. Le velocità delle reti sono: N1 – 100 Mb/s, N2 – 100 Mb/s, N3 – 25 Mb/s, N4 – 50 Mb/s, N5 – 10 Mb/s, N6 – 25 Mb/s.
- Si calcolino i cammini minimi da R1 a tutte le reti usando l'algoritmo di *Dijkstra*. Si scriva la tabella di *routing* di R1.
- Si assuma la reti usi il protocollo OSPF e sia divisa in tre aree. Area 0 (R4, R5, R6, N2, N4), Area 1 (R1, R2, R3, R5, N1, N3, N5), Area 2 (R3, R4, N6). Si disegni la topologia della rete vista da R1. La tabella di *routing* di R1 si modifica rispetto al punto b)?



Soluzione

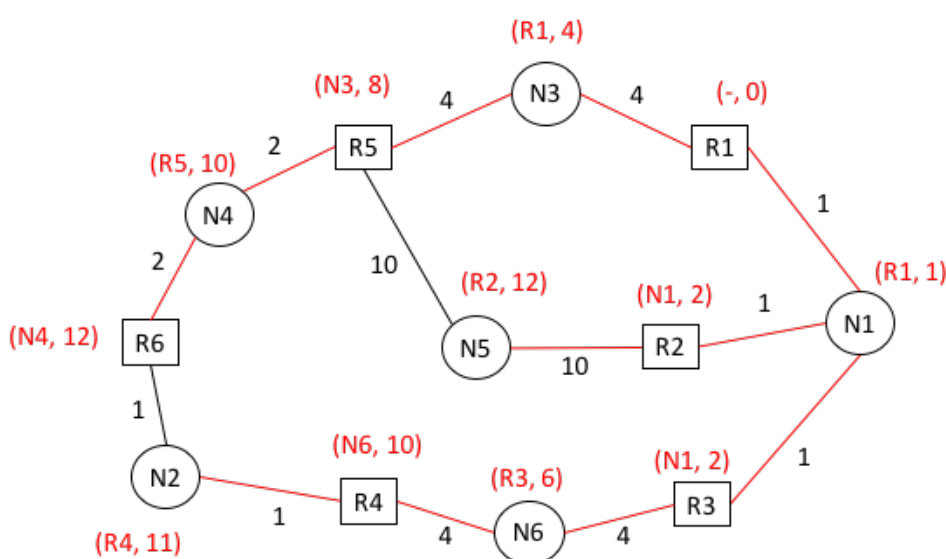
a) Topologia della rete:



b) cammini minimi da R1: la tabella seguente mostra l'evoluzione dell'algoritmo di *Dijkstra*. Lo *Step 0* si riferisce all'inizializzazione. Ad ogni passo (*step*) sono indicate le etichette rese permanenti (in rosso) ed i router che non vengono considerati al passo specifico (#); si ricorda

che ad ogni passo, possono essere modificate le etichette dei soli nodi che sono vicini di nodi già parte dell'albero dei cammini minimi (nodi con etichetta rossa al passo precedente)

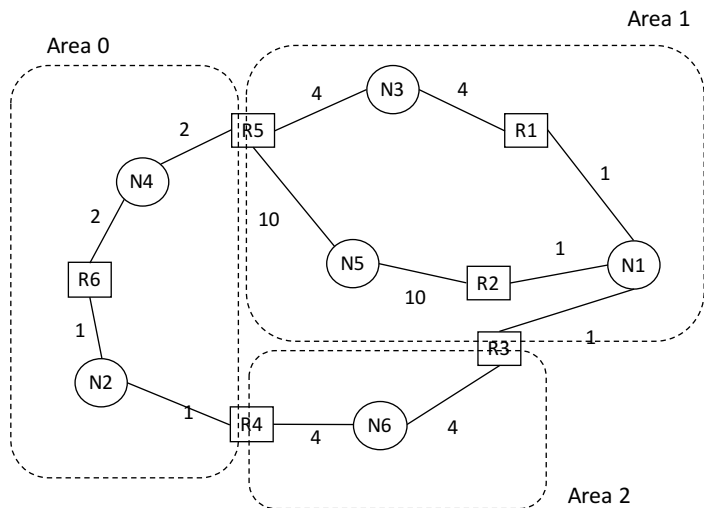
	R1	R2	R3	R4	R5	R6	N1	N2	N3	N4	N5	N6
Step 0	(0,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)
Step 1		#	#	#	#	#	(1,R1)	#	(4,R1)	#	#	#
Step 2		(1,N1)	(2,N1)	#	#	#		#	(4,R1)	#	#	#
Step 3			(2,N1)	#	#	#		#	(4,R1)	#	(12,R2)	#
Step 4				#	#	#		#	(4,R1)	#	(12,R2)	(6,R3)
Step 5				#	(8,N3)	#		#		#	(12,R2)	(6,R3)
Step 6				(10,N6)	(8,N3)	#		#		#	(12,R2)	
Step 7				(10,N6)		#		#		(10,R5)	(12,R2)	
Step 8				(10,N6)		(12,N4)		#			(12,R2)	
Step 9						(12,N4)		(11,R4)			(12,R2)	
Step 10						(12,N4)					(12,R2)	
Step 11											(12,R2)	



b) tabella di *routing* di R1:

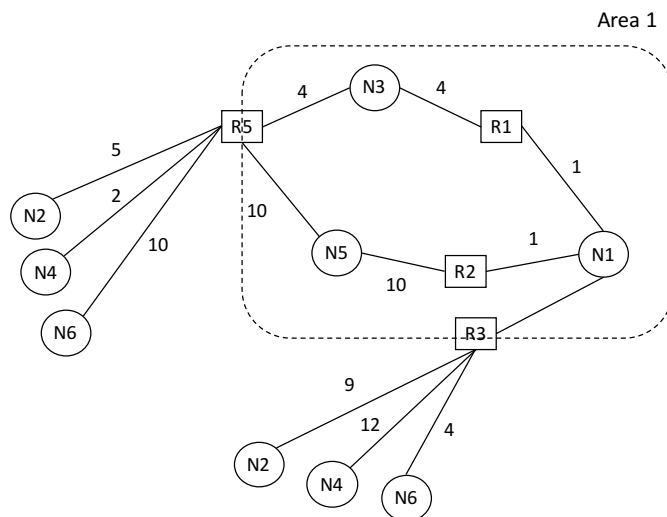
Network	Next-hop
N1	local
N3	local
N2	R3
N4	R5
N5	R2
N6	R3

c) divisione in aree:



c) topologia vista da R1:

(La topologia è quella dettagliata dell'area 1 a cui R1 appartiene e un riassunto della raggiungibilità attraverso gli area border router R5 (il valore riportato è la distanza da R5 alle reti esterne) ed R3 (il valore riportato è la distanza da R3 alle reti esterne)).



La tabella di R1 non cambia.

4b.14 - Esercizio

Si supponga che i quattro *router* in figura usino il protocollo RIP versione 2. In figura è anche riportata la tabella di *routing* per il *router* R1. Si assuma che il costo dei collegamenti tra R1 e tutti gli altri *router* sia uguale a 1.

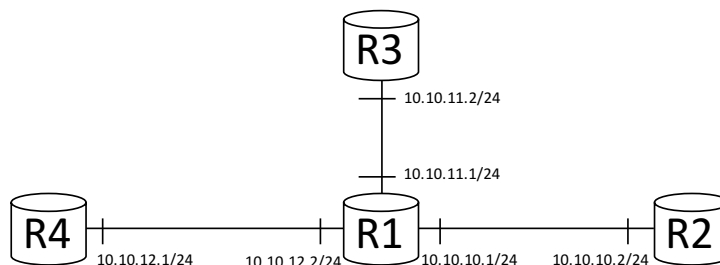


Tabella di *routing* di R1

Destinazione	Next Hop	Costo
131.175.124.0/24	10.10.10.2	3
131.180.0.0/16	10.10.11.2	6
131.175.16.0/24	10.10.12.2	9
131.175.9.0/24	10.10.10.2	11

- Indicare il contenuto dei messaggi di risposta RIPv2 inviati da R1 a tutti gli altri *router* nel caso in cui si usi la versione con *Poisonous Reverse* del protocollo di *routing*.
- Dire come cambia la seguente tabella di *routing* del *router* R1 quando riceve un messaggio di risposta RIPv2 da R3 con il seguente contenuto:

```
131.175.124.0/24    costo: 1
131.180.0.0/16     costo: 16
131.180.12.0/23    costo: 11
131.175.9.0/24     costo: 13
```

Soluzione

a)

A R2

```
131.175.124.0/24    costo: 16
131.180.0.0/16     costo: 6
131.175.16.0/24    costo: 9
131.175.9.0/24     costo: 16
```

A R3

```
131.175.124.0/24    costo: 3
131.180.0.0/16     costo: 16
131.175.16.0/24    costo: 9
131.175.9.0/24     costo: 11
```

A R4

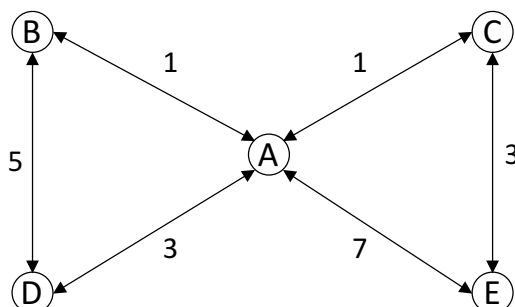
131.175.124.0/24	costo: 3
131.180.0.0/16	costo: 6
131.175.16.0/24	costo: 16
131.175.9.0/24	costo: 11

b)

131.175.124.0/24	10.10.11.2	2
131.180.0.0/16	10.10.11.2	16
131.180.12.0/23	10.10.11.2	12
131.175.16.0/24	10.10.12.2	9
131.175.9.0/24	10.10.10.2	11

4b.15 - Esercizio

Sia data la rete in figura.



Sono indicati i nodi (A,B,C,D,E) ed il costo di attraversamento di ogni collegamento. Nella rete è attivo un algoritmo di *routing* secondo cammini minimi arrivato a convergenza. Ipotizzando che gli stessi nodi siano le destinazioni da raggiungere, si chiede di:

- Indicare i *Distance vector* (no *Split Horizon*) inviati dal nodo A (attenzione: contenuto e destinatario del DV)
- Indicare i *Distance vector* inviati dal nodo A in caso di *Split Horizon*, senza *Poisonous Reverse* (attenzione: contenuto e destinatario del DV)
- Nel caso in cui il nodo A riceva dal nodo C il seguente DV: (B,2), (D,1), (E,5), (F,3), riempire le tabelle di instradamento del nodo A sottostanti, una è prima della ricezione del DV, l'altra subito dopo.

PRIMA		
Dest.	Costo	Next-Hop

DOPO		
Dest.	Costo	Next-Hop

Soluzione

Punto a)

DV verso B, C, D, E: A0; B1; C1; D3; E4

Punto b)

DV verso B: A0; C1; D3; E4

DV verso C: A0; B1; D3

DV verso D: A0; B1; C1; E4

DV verso E: A0; B1; C1; D3; E4

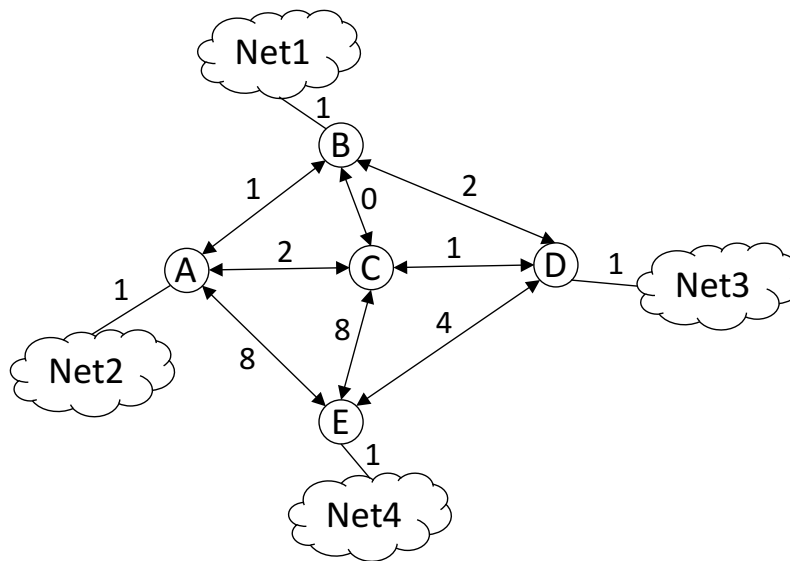
Punto c)

A riceve dal nodo C il seguente DV: (B,2), (D,1), (E,5), (F,3).

PRIMA		
Dest.	Costo	Next-Hop
A	0	Dir
B	1	B
C	1	C
D	3	D
E	4	C

DOPO		
Dest.	Costo	Next-Hop
A	0	dir
B	1	B
C	1	C
D	2	C
E	6	C
F	4	C

4b.16 - Esercizio



Nella rete in figura è rappresentato il grafo di una rete in cui sono presenti dei *router* (A, B, C, D, E) e 4 reti (Net1, Net2, Net3, Net4). I costi di attraversamento sono indicati accanto ad ogni *link*, i *link* sono bidirezionali e simmetrici.

Si chiede di:

- Calcolare mediante l'algoritmo di *Bellman-Ford* l'albero dei cammini minimi con sorgente A e destinazioni tutti gli altri *router* (si omettano le reti nel grafo). Indicare:
 - nella Tabella A, il valore dell'etichetta ad ogni step in cui il nodo viene analizzato: nel caso lo step successivo non modifichi l'etichetta dello step precedente occorre riscrivere l'etichetta dello step precedente.
 - nella figura sopra, l'albero trovato
- Sulla base dell'albero dei cammini calcolato al punto precedente, indicare i *Distance vector* (DV) relativi alle reti Net1, Net2, Net3 e Net4, inviati dal *router* A ai propri vicini nella modalità *Split Horizon* con *Poisonous Reverse*. Per ogni DV inviato indicare chiaramente il destinatario del DV e le reti raggiungibili comunicate con i rispettivi costi.

Soluzione

Punto a)

Nodo A	Nodo B	Nodo C	Nodo D	Nodo E
(0,-)	(inf,-)	(inf,-)	(inf,-)	(inf,-)
(0,-)	(1,A)	(2,A)	(inf,-)	(8,A)
(0,-)	(1,A)	(1,B)	(3,B)	(8,A)
(0,-)	(1,A)	(1,B)	(2,C)	(7,D)
(0,-)	(1,A)	(1,B)	(2,C)	(6,D)

Punto b)

	$A \Rightarrow B$	$A \Rightarrow C$	$A \Rightarrow E$
Net 1	inf	2	2
Net 2	1	1	1
Net 3	inf	3	3
Net 4	inf	7	7