

ESERCIZIO 4: La Figura 4.1 illustra il collegamento, mediante link seriali, tra quattro router di una rete internet. Tale figura riporta inoltre il costo relativo a ciascun link seriale. Il candidato:

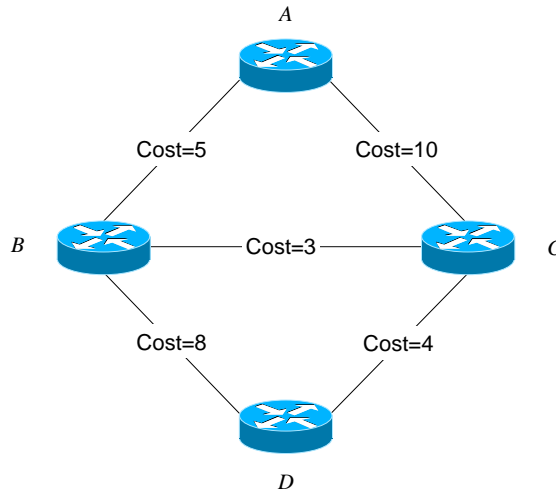


Figura 4.1: Struttura della rete internet con i costi associati a ciascun link seriale

1. specifichi il link state database;
2. descriva l'algoritmo di Dijkstra utilizzato dal router A per costruire la routing table.

Supponiamo adesso che ciascun link di collegamento sia caratterizzato da una banda trasmissiva pari a quella specificata in Figura 4.2 e supponiamo che il router A, utilizzando l'algoritmo di Dijkstra, voglia calcolare la routing table con il vincolo che la banda minima garantita verso un qualunque router di destinazione sia di 60 Mbps. Il candidato:

3. specifichi in che modo l'algoritmo di Dijkstra deve essere modificato per tener conto di tale vincolo;
4. specifichi la routing table del router A.

RISOLUZIONE

1. La Figura 4.3 illustra il Link State Database contenuto in ciascun router dell'internet di Figura 4.1. In corrispondenza di ciascun router (riportato nella prima riga) vi sono tanti elementi della forma (router adiacente, costo)

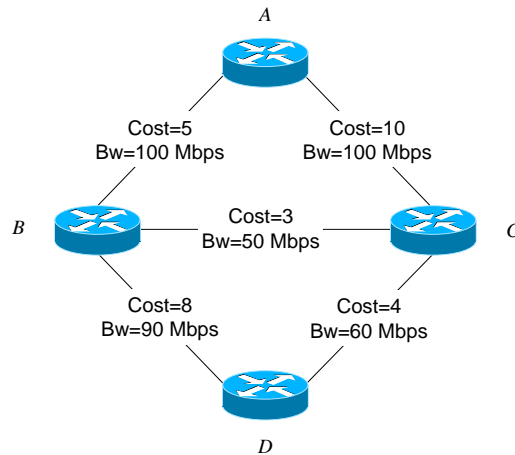


Figura 4.2: Costi e banda trasmissiva associati a ciascun link seriale

A	B	C	D
(B,5)	(A,5)	(A,10)	(B,8)
(C,10)	(C,3)	(B,3)	(C,4)
	(D,8)	(D,4)	

Figura 4.3: Link State Database

quanti sono i router ad esso adiacenti. Per esempio, il router *A* è adiacente ai router *B* e *C* che sono raggiungibili da *A* con costi di 5 e 10 rispettivamente.

2. La costruzione della tabella di routing tramite l'algoritmo di Dijkstra avviene attraverso i passi illustrati e commentati in Figura 4.4. La routing table di *A* è riportata in Figura 4.5. Dal punto di vista del Router *A* la topologia della rete è illustrata in Figura 4.6 dalla quale si evince che il traffico proveniente dal router *A* non attraverserà mai i links da *A* a *C* e da *B* a *D*. Questi ultimi due links possono essere utilizzati per rendere il sistema tollerante ai guasti. Se ad esempio cade il link tra i routers *C* e *D* (Figura 4.6) il link tra *B* e *D* può offrire un cammino alternativo. In condizioni normali tali link costituiscono però un peso aggiuntivo dal punto di vista dei costi di gestione dell'intera rete.

<i>Step</i>	<i>CONF</i>	<i>TENT</i>	<i>Commenti</i>
1	(A,0,-)	Empty	
2	(A,0,-)	(B,5,B) (C,10,C)	Inseriamo in <i>TENT</i> i router adiacenti al router A. Questi sono i router B e C che possono essere raggiunti da A con costi di 5 e 10 rispettivamente.
3	(A,0,-) (B,5,B)	(C,10,C)	Spostiamo in <i>CONF</i> il router che può essere raggiunto da A con il costo minore. Nel caso in esame esso è ovviamente il router B.
4	(A,0,-) (B,5,B)	(C,10,C) (C,8,B) (D,13,B)	Inseriamo in <i>TENT</i> i router adiacenti al router B. Il primo item (C,10,C) viene sostituito con l'item (C,8,B) in quanto il router C è raggiungibile da B a minor costo.
5	(A,0,-) (B,5,B) (C,8,B)	(D,13,B)	(C,8,B) viene spostato in <i>CONF</i> in quanto item a minor costo rispetto a (D,13,B).
6	(A,0,-) (B,5,B) (C,8,B)	(D,13,B) (D,12,B)	Inseriamo in <i>TENT</i> i router adiacenti al router C. (D,12,B) sostituisce (D,13,B) in quanto a minor costo.
7	(A,0,-) (B,5,B) (C,8,B) (D,12,B)		Spostiamo in <i>CONF</i> l'item (D,12,B). La procedura termina in quanto non vi sono più item in <i>TENT</i> .

Figura 4.4: Steps per costruire la routing table del router A

<i>Node</i>	<i>Cost</i>	<i>Next Hop</i>
A	0	<i>Self</i>
B	5	<i>B (directly connected)</i>
C	8	B
D	12	B

Figura 4.5: Routing table di A

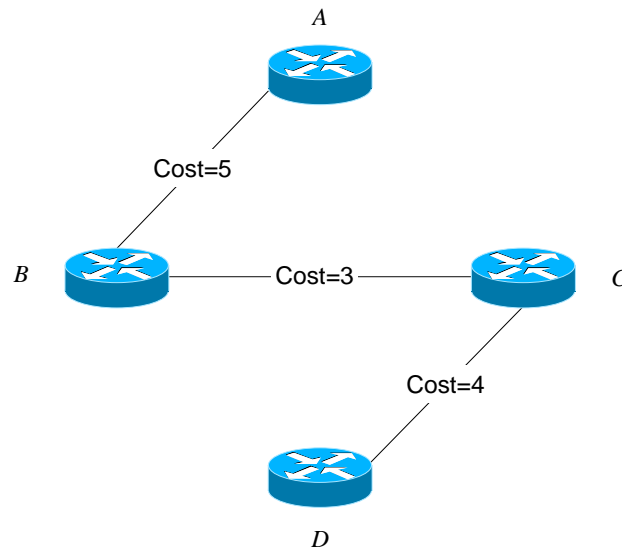


Figura 4.6: Visibilità della rete dal punto di vista del router A

3. L'algoritmo di Dijkstra deve adesso controllare, ad ogni step, la banda minima disponibile. Di conseguenza, i vari items delle liste *TENT* e *CONF* dovranno avere la seguente struttura: $(node, cost, next\ hop, minimum\ bandwidth)$. Con questa estensione l'evoluzione dell'algoritmo di Dijkstra è illustrato in Figura 4.7.

4. La routing table di A è riportata in Figura 4.8.

<i>Step</i>	<i>CONF</i>	<i>TENT</i>	<i>Comments</i>
1	(A,0,self,N/A)	Empty	
2	(A,0,self,N/A)	(B,5,B,100) (C,10,C,100)	Inseriamo in <i>TENT</i> i router adiacenti al router A. Questi sono i router B e C che possono essere raggiunti da A con costi di 5 e 10 rispettivamente e con una banda di 100 Mbps.
3	(A,0,self,N/A) (B,5,B,100)	(C,10,C,100)	Spostiamo in <i>CONF</i> il router che può essere raggiunto da A con il costo minore. Nel caso in esame esso è ovviamente il router B.
4	(A,0,self,N/A) (B,5,B,100)	(C,10,C,100) (D,13,B,90)	Inseriamo in <i>TENT</i> i router adiacenti al router B. L'item (C,8,B,50) non viene inserito in <i>TENT</i> in quanto non rispetta il vincolo sulla banda minima sebbene il costo sia inferiore a (C,10,C,100).
5	(A,0,self,N/A) (B,5,B,100) (C,10,C,100)	(D,13,B,90)	(C,10,C,100) viene spostato in <i>CONF</i> in quanto il costo è inferiore rispetto a (D,13,B,90).
6	(A,0,self,N/A) (B,5,B,100) (C,10,C,100)	(D,13,B,90) (D,14,C,60)	Inseriamo in <i>TENT</i> l'item (D,14,C,60) che sopprimiamo in quanto il costo è più elevato rispetto a (D,13,B,90).
7	(A,0,self,N/A) (B,5,B,100) (C,10,C,100) (D,13,B,90)		Spostiamo in <i>CONF</i> l'item (D,13,B,90). La procedura termina in quanto non vi sono più item in <i>TENT</i> .

Figura 4.7: Algoritmo di Dijkstra modificato

<i>Node</i>	<i>Cost</i>	<i>Next Hop</i>	<i>Minimum Bandwidth</i>
<i>A</i>	0	<i>Self</i>	<i>N/A</i>
<i>B</i>	5	<i>B (directly connected)</i>	100
<i>C</i>	10	<i>C</i>	100
<i>D</i>	13	<i>B</i>	90

Figura 4.8: Routing table di *A*