Reti: Esercizi

Aymane Chabbaki

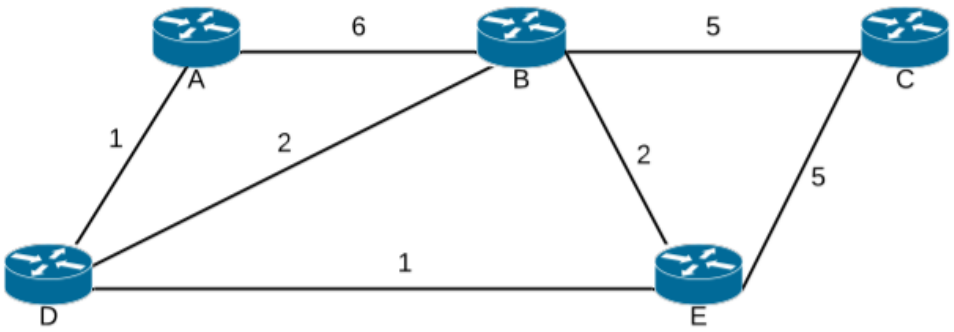
III semestre 2019/2020

Indice

1 Routing: Link State (OSPF)

1.1 Esercizio 1

Usando gli algoritmi propri di OSPF, ed assumendo che i costi siano già stati distribuiti, si calcoli la tabella di instradamento del nodo A.



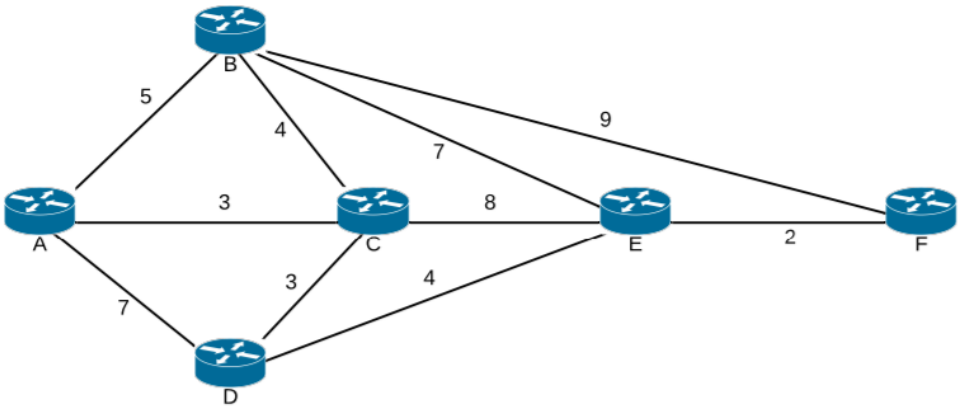
A

Step	N'	$D(B)$	$D(D)$	$D(E)$	$D(C)$
0	A	$6, A$	$1, A$	∞	∞
1	A, D	$3, D$		$2, D$	∞
2	A, D, E	$3, D$			$7, E$
3	A, D, E, B				$7, E$

Destination	Cost	Next Hop
A	0	–
B	3	D
C	7	D
D	1	D
E	2	D

1.2 Esercizio 2

Usando gli algoritmi propri di OSPF, ed assumendo che i costi siano già stati distribuiti, si calcoli la tabella di instradamento del nodo A.



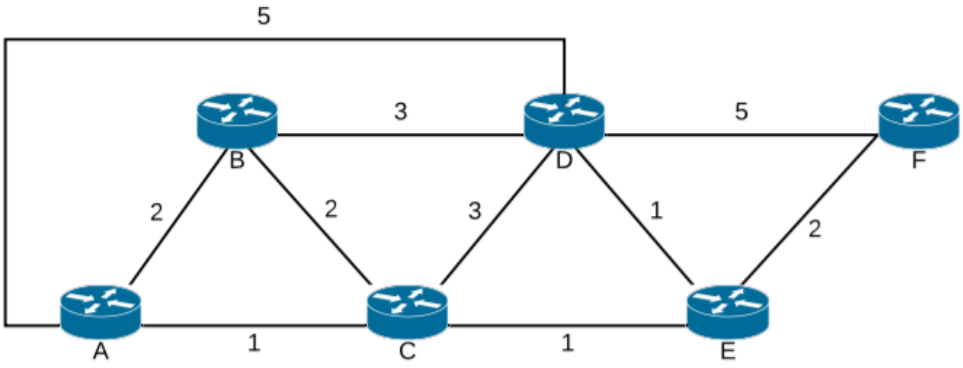
A

Step	N'	$D(B)$	$D(C)$	$D(D)$	$D(E)$	$D(F)$
0	A	$5, A$	$3, A$	$7, A$	∞	∞
1	A, C	$5, A$		$6, C$	$11, C$	∞
2	A, C, B			$6, C$	$11, C$	$14, B$
3	A, C, B, D				$10, C$	$14, B$
4	A, C, B, D, E					$12, E$

Destination	Cost	Next Hop
A	0	–
B	5	B
C	3	C
D	6	C
E	10	C
F	12	C

1.3 Esercizio 3

Usando gli algoritmi propri di OSPF, ed assumendo che i costi siano già stati distribuiti, si calcoli la tabella di instradamento del nodo A.



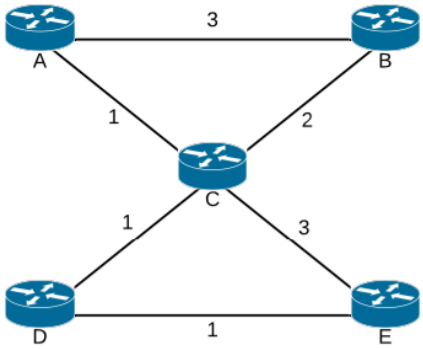
A

Step	N'	$D(B)$	$D(C)$	$D(D)$	$D(E)$	$D(F)$
0	A	$2, A$	$1, A$	$5, A$	∞	∞
1	A, C	$2, A$		$4, C$	$2, C$	∞
2	A, C, E	$2, A$		$3, E$		$4, E$
3	A, C, E, B			$3, E$		$4, E$
4	A, C, E, B, D					$4, E$

Destination	Cost	Next Hop
A	0	–
B	2	B
C	1	C
D	3	C
E	2	C
F	4	C

1.4 Esercizio 4

Usando gli algoritmi propri di OSPF, ed assumendo che i costi siano già stati distribuiti, si calcoli la tabella di instradamento del nodo E.



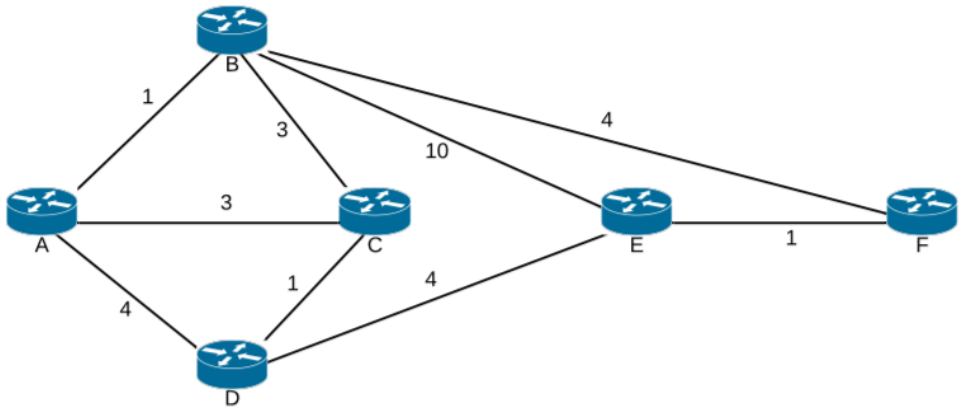
E

Step	N'	$D(A)$	$D(B)$	$D(C)$	$D(D)$
0	E	∞	∞	$3, E$	$1, E$
1	E, D	∞	∞	$2, E$	
2	E, D, C	$3, C$	$4, C$		
3	E, D, C, A		$4, C$		

Destination	Cost	Next Hop
A	3	D
B	4	D
C	2	D
D	1	D
E	0	–

1.5 Esercizio 5

Usando gli algoritmi propri di OSPF, ed assumendo che i costi siano già stati distribuiti, si calcoli la tabella di instradamento dei nodi B, D, F.



B

Step	N'	$D(A)$	$D(C)$	$D(D)$	$D(E)$	$D(F)$
0	B	$1, B$	$3, B$	∞	$10, B$	$4, B$
1	B, A		$3, B$	$5, A$	$10, B$	$4, B$
2	B, A, C			$4, C$	$10, B$	$4, B$
3	B, A, C, D				$8, D$	$4, B$
4	B, A, C, D, F				$5, F$	

Destination	Cost	Next Hop
A	1	A
B	0	–
C	3	C
D	4	C
E	5	F
F	4	F

D

Step	N'	$D(A)$	$D(B)$	$D(C)$	$D(E)$	$D(F)$
0	D	$4, D$	∞	$1, D$	$4, D$	∞
1	D, C	$4, D$	$4, C$		$4, D$	∞
2	D, C, A		$4, C$		$4, D$	∞
3	D, C, A, B				$4, D$	$8, B$
4	D, C, A, B, E					$5, E$

Destination	Cost	Next Hop
A	4	A
B	4	C
C	1	C
D	0	–
E	4	E
F	5	E

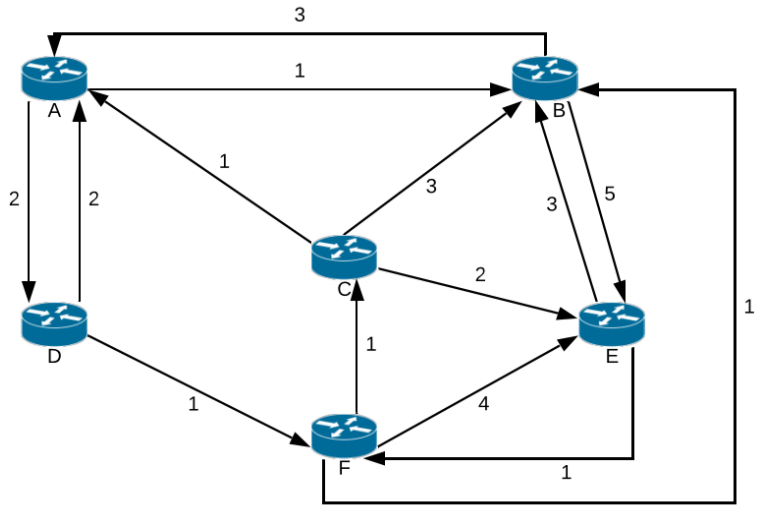
F

Step	N'	$D(A)$	$D(B)$	$D(C)$	$D(E)$	$D(F)$
0	F	∞	$4, F$	∞	∞	$1, F$
1	F, E	∞	$4, F$	∞	$5, E$	
2	F, E, B	$5, B$		$7, B$	$5, E$	
3	F, E, B, A			$7, B$	$5, E$	
4	F, E, B, A, D			$6, D$		

Destination	Cost	Next Hop
A	5	B
B	4	B
C	6	E
D	5	E
E	1	E
F	0	–

1.6 Esercizio 6

Usando gli algoritmi propri di OSPF, ed assumendo che i costi siano già stati distribuiti, si calcoli la tabella di instradamento dei nodi A, C, E.



B

Step	N'	$D(A)$	$D(C)$	$D(D)$	$D(E)$	$D(F)$
0	A	$3, B$	∞	∞	$5, B$	∞
1	B, A		∞	$5, A$	$5, B$	∞
2	B, A, D		∞		$5, B$	$6, D$
3	B, A, D, E		∞			$6, D$
4	B, A, D, E, F		$7, F$			

Destination	Cost	Next Hop
A	3	B
B	0	–
C	7	A
D	5	A
E	5	B
F	6	A

C

Step	N'	$D(A)$	$D(B)$	$D(D)$	$D(E)$	$D(F)$
0	C	$1, C$	$3, C$	∞	$2, C$	∞
1	C, A		$2, A$	$3, A$	$2, C$	∞
2	C, A, B			$3, A$	$2, C$	∞
3	C, A, B, E			$3, A$		$3, E$
4	C, A, B, E					$3, E$

Destination	Cost	Next Hop
A	1	A
B	2	A
C	0	–
D	3	A
E	2	E
F	3	E

E

Step	N'	$D(A)$	$D(B)$	$D(C)$	$D(D)$	$D(F)$
0	E	∞	$3, E$	∞	∞	$1, E$
1	E, F	∞	$2, F$	$2, F$	∞	
2	E, F, B	$5, B$		$2, F$	∞	
3	E, F, B, C	$3, C$			∞	
4	E, F, B, C, A				$5, A$	

Destination	Cost	Next Hop
A	3	F
B	2	F
C	2	F
D	5	F
E	0	–
F	1	F

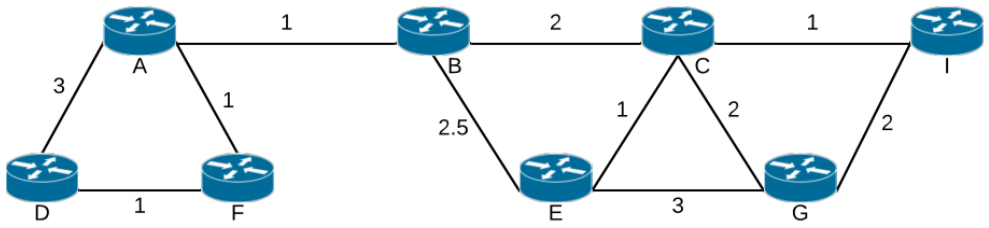
F

Step	N'	$D(A)$	$D(B)$	$D(C)$	$D(D)$	$D(E)$
0	F	∞	$1, F$	$1, F$	∞	$4, F$
1	F, B	$4, B$		$1, F$	∞	$4, F$
2	F, B, C	$2, C$			∞	$3, C$
3	F, B, C, A				$4, A$	$3, C$
4	F, B, C, A, E				$4, A$	

Destination	Cost	Next Hop
A	2	C
B	1	B
C	1	C
D	4	C
E	3	C
F	0	–

1.7 Esercizio 7

Usando gli algoritmi propri di OSPF, ed assumendo che i costi siano già stati distribuiti, si calcoli la tabella di instradamento dei nodi A, G.



A

Step	N'	$D(B)$	$D(C)$	$D(D)$	$D(E)$	$D(F)$	$D(G)$	$D(I)$
0	A	$1, A$	∞	$3, A$	∞	$1, A$	∞	∞
1	A, B		$3, B$	$3, A$	$3.5, B$	$1, A$	∞	∞
2	A, B, F		$3, B$	$2, F$	$3.5, B$		∞	∞
3	A, B, F, D		$3, B$		$3.5, B$		∞	∞
4	A, B, F, D, C				$3.5, B$		$5, C$	$4, C$
5	A, B, F, D, C, E						$5, C$	$4, C$
6	A, B, F, D, C, E, I						$5, C$	

A:

Destination	Cost	Next Hop
A	0	–
B	1	B
C	3	B
D	2	F
E	3.5	B
F	1	F
G	5	B
I	4	B

G

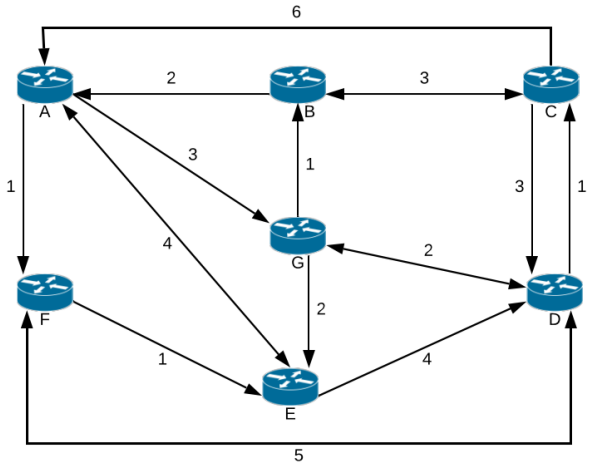
Step	N'	$D(A)$	$D(B)$	$D(C)$	$D(D)$	$D(E)$	$D(F)$	$D(I)$
0	G	∞	∞	$2, G$	∞	$3, G$	∞	$2, G$
1	G, C	∞	$4, C$		∞	$3, G$	∞	$2, G$
2	G, C, I	∞	$4, C$		∞	$3, G$	∞	
3	G, C, I, E	∞	$4, C$		∞		∞	
4	G, C, I, E, B	$5, B$			∞		∞	
5	G, C, I, E, B, A				$8, A$		$6, A$	
6	G, C, I, E, B, A, F				$7, F$			

G:

Destination	Cost	Next Hop
A	5	C
B	4	C
C	2	C
D	7	C
E	3	E
F	6	C
G	0	–
I	2	I

1.8 Esercizio 8

Usando gli algoritmi propri di OSPF, ed assumendo che i costi siano già stati distribuiti, si calcoli la tabella di instradamento dei nodi A, B.



A

Step	N'	$D(B)$	$D(C)$	$D(D)$	$D(E)$	$D(F)$	$D(G)$
0	A	∞	∞	∞	$4, A$	$1, A$	$3, A$
1	A, F	∞	∞	$6, F$	$2, F$		$3, A$
2	A, F, E	∞	∞	$6, F$			$3, A$
3	A, F, E, G	$4, G$	∞	$5, G$			
4	A, F, E, G, B		$7, B$	$5, G$			
5	A, F, E, G, B, D		$6, D$				

A:

Destination	Cost	Next Hop
A	0	–
B	4	G
C	6	G
D	5	G
E	2	F
F	1	F
G	3	G

B

Step	N'	$D(B)$	$D(C)$	$D(D)$	$D(E)$	$D(F)$	$D(G)$
0	B	$2, B$	$3, B$	∞	∞	∞	∞
1	B, A		$3, B$	∞	$6, A$	$3, A$	$5, A$
2	B, A, C			$6, C$	$6, A$	$3, A$	$5, A$
3	B, A, C, F			$6, C$	$4, F$		$5, A$
4	B, A, C, F, E			$6, C$			$5, A$
5	B, A, C, F, E, G			$6, C$			

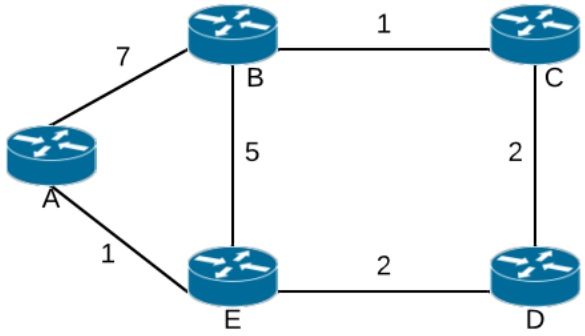
B:

Destination	Cost	Next Hop
A	2	A
B	0	–
C	3	C
D	6	C
E	4	A
F	3	A
G	5	A

2 Routing: Distance Vector

2.1 Esercizio 1

Usando l'algoritmo di routing di tipo Distance-vector, mostrare l'evoluzione delle tabelle di routing per ogni nodo, considerando che i DV vengono inoltrati dai router seguendo l'ordine: D, E, C, A, B.



Inizializzo le tabelle di routing di ogni nodo inserendo solo i nodi direttamente connessi (per semplicità includo tutte le destinazioni anche se sono ignote, queste avranno costo e NH vuoto).

A		
Dst	Cost	NH
A	0	-
B	7	B
C		
D		
E	1	E

B		
Dst	Cost	NH
A	7	A
B	0	-
C	1	C
D		
E	5	E

C		
Dst	Cost	NH
A		
B	1	B
C	0	-
D	2	D
E		

D		
Dst	Cost	NH
A		
B		
C	2	C
D	0	-
E	2	E

E		
Dst	Cost	NH
A	1	A
B	5	B
C		
D	2	D
E	0	-

D manda il suo DV: $\{(C, 2), (E, 2)\}$ a $\{E, C\}$.
E e C appena ricevono il DV, controllano se possono raggiungere una destinazione (o scoprirne una nuova) con un costo minore rispetto a quello che già conoscono.
E per prima cosa controlla il percorso verso se stesso usando la regola dell'algoritmo:

$$DV_D[E].cost + c(E, D) < R_E[E].cost?$$

E sta controllando se $2 < 0$, che è falso, dunque il percorso verso D non viene modificato.
E esegue la stessa procedura per C e visto che $2+2 < \infty$, E scopre un percorso attraverso D per raggiungere C (prima non conosceva C).
C esegue lo stesso procedimento e viene a conoscenza del nodo E.

A		
Dst	Cost	NH
A	0	-
B	7	B
C		
D		
E	1	E

B		
Dst	Cost	NH
A	7	A
B	0	-
C	1	C
D		
E	5	E

C		
Dst	Cost	NH
A		
B	1	B
C	0	-
D	2	D
E	4	*D*

D		
Dst	Cost	NH
A		
B		
C	2	C
D	0	-
E	2	E

E		
Dst	Cost	NH
A	1	A
B	5	B
C	4	*D*
D	2	D
E	0	-

E manda il suo DV: $\{(A,1),(B,5),(4,C),(2,D)\}$ a $\{A,B,D\}$.
A scopre dell'esistenza dei nodi C e D (raggiungibili attraverso il nodo E) e aggiorna il suo percorso verso B attraverso il nodo E.
B viene a conoscenza del nodo D (raggiungibile attraverso il nodo E) e aggiorna il suo percorso verso A attraverso il nodo E.
D viene a conoscenza dei nodi A e B, raggiungibili attraverso il nodo E.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	6	*E*	A			A	3	*E*	A	1	A
B	6	*E*	B	0	-	B	1	B	B	7	*E*	B	5	B
C	5	*E*	C	1	C	C	0	-	C	2	C	C	4	D
D	3	*E*	D	7	*E*	D	2	D	D	0	-	D	2	D
E	1	E	E	5	E	E	4	D	E	2	E	E	0	-

C manda il suo DV a $\{B,D\}$.
B aggiorna il suo percorso verso il nodo D attraverso il nodo C.
D aggiorna il suo percorso verso il nodo B attraverso il nodo C.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	6	E	A			A	3	E	A	1	A
B	6	E	B	0	-	B	1	B	B	3	*C*	B	5	B
C	5	E	C	1	C	C	0	-	C	2	C	C	4	D
D	3	E	D	3	*C*	D	2	D	D	0	-	D	2	D
E	1	E	E	5	E	E	4	D	E	2	E	E	0	-

A manda il suo DV a $\{B,E\}$ (non si sono rotte migliori ne scoperta di nuovi nodi, nulla cambia).

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	6	E	A			A	3	E	A	1	A
B	6	E	B	0	-	B	1	B	B	3	C	B	5	B
C	5	E	C	1	C	C	0	-	C	2	C	C	4	D
D	3	E	D	3	C	D	2	D	D	0	-	D	2	D
E	1	E	E	5	E	E	4	D	E	2	E	E	0	-

B manda il suo DV a $\{A,E,C\}$.
C viene a conoscenza del nodo A, raggiungibile attraverso il nodo B.
Sono stati inviati il set completo di messaggi tra i router, ma non sono ancora arrivato alla convergenza, infatti esiste percorso da C ad A con costo minore rispetto a quello che ha nella sua tabella di routing.

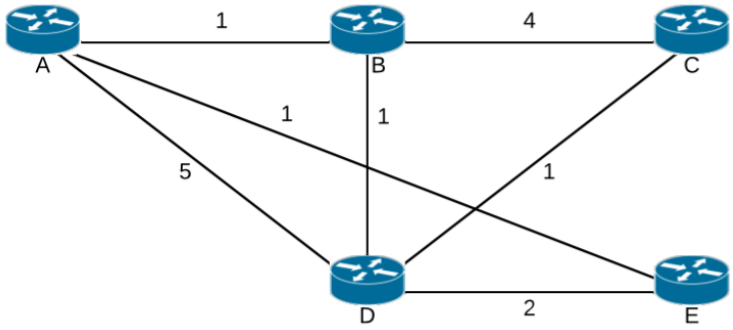
A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	6	E	A	7	*B*	A	3	E	A	1	A
B	6	E	B	0	-	B	1	B	B	3	C	B	5	B
C	5	E	C	1	C	C	0	-	C	2	C	C	4	D
D	3	E	D	3	C	D	2	D	D	0	-	D	2	D
E	1	E	E	5	E	E	4	D	E	2	E	E	0	-

D manda il suo DV a $\{E, C\}$.
C aggiorna il suo percorso verso il nodo A attraverso il nodo E e così facendo ho raggiunto la convergenza.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	6	E	A	5	*E*	A	3	E	A	1	A
B	6	E	B	0	-	B	1	B	B	3	C	B	5	B
C	5	E	C	1	C	C	0	-	C	2	C	C	4	D
D	3	E	D	3	C	D	2	D	D	0	-	D	2	D
E	1	E	E	5	E	E	4	D	E	2	E	E	0	-

2.2 Esercizio 2

Usando l'algoritmo di routing di tipo Distance-vector, mostrare l'evoluzione delle tabelle di routing per ogni nodo, considerando che i DV vengono inoltrati dai router seguendo l'ordine: E, B, A, C, D.



Inizializzo le tabelle di routing di ogni nodo inserendo solo i nodi direttamente connessi (per semplicità includo tutte le destinazioni anche se sono ignote, queste avranno costo e NH vuoto).

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	1	A	A			A	5	A	A	1	A
B	1	B	B	0	-	B	4	B	B	1	B	B		
C			C	4	C	C	0	-	C	1	C	C		
D	5	D	D	1	D	D	1	D	D	0	-	D	2	D
E	1	E	E			E			E	2	E	E	0	-

E manda il suo DV: $\{(A, 1), (D, 2)\}$ a $\{D, A\}$.
D e A scoprono una nuova rotta per raggiungersi con costo inferiore, attraverso E.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	1	A	A			A	3	*E*	A	1	A
B	1	B	B	0	-	B	4	B	B	1	B	B		
C			C	4	C	C	0	-	C	1	C	C		
D	3	*E*	D	1	D	D	1	D	D	0	-	D	2	D
E	1	E	E			E			E	2	E	E	0	-

B manda il suo DV: $\{(A,1),(C,4),(D,1)\}$ a $\{A,C,D\}$.
A e D scoprono un nuovo percorso con costo inferiore attraverso D.
A viene a conoscenza del nodo C (e viceversa) attraverso il nodo B.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	1	A	A	5	*B*	A	2	*B*	A	1	A
B	1	B	B	0	-	B	4	B	B	1	B	B		
C	5	*B*	C	4	C	C	0	-	C	1	C	C		
D	2	*B*	D	1	D	D	1	D	D	0	-	D	2	D
E	1	E	E			E			E	2	E	E	0	-

A manda il suo DV: $\{(B,1),(C,5),(D,2),(E,1)\}$ a $\{B,D,E\}$.
B viene a conoscenza del nodo E (e viceversa) e del nodo C (ma C ancora non sa che esiste B).

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	1	A	A	5	B	A	2	B	A	1	A
B	1	B	B	0	-	B	4	B	B	1	B	B	2	*A*
C	5	E	C	4	C	C	0	-	C	1	C	C	6	*A*
D	2	E	D	1	D	D	1	D	D	0	-	D	2	D
E	1	E	E	2	*A*	E			E	2	E	E	0	-

C manda il suo DV: $\{(A,5),(B,4),(D,1)\}$ a $\{B,D\}$.
Non ci sono modifiche, non ci sono percorsi migliori ne scoperte di nuovi nodi.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	1	A	A	5	B	A	2	B	A	1	A
B	1	B	B	0	-	B	4	B	B	1	B	B	2	A
C	5	E	C	4	C	C	0	-	C	1	C	C	6	A
D	2	E	D	1	D	D	1	D	D	0	-	D	2	D
E	1	E	E	2	A	E			E	2	E	E	0	-

D manda il suo DV: $\{(A,2),(B,1),(C,1),(E,2)\}$ a $\{A,B,C,E\}$.
B scopre un percorso migliore verso C (e viceversa).
C scopre un percorso migliore verso A (ma non viceversa) e scopre anche il nodo E.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	1	A	A	3	*D*	A	2	B	A	1	A
B	1	B	B	0	-	B	2	*D*	B	1	B	B	2	A
C	5	E	C	2	*D*	C	0	-	C	1	C	C	3	*D*
D	2	E	D	1	D	D	1	D	D	0	-	D	2	D
E	1	E	E	2	A	E	3	*D*	E	2	E	E	0	-

E' stato inviato il set completo di messaggi tra i router, ma non abbiamo raggiunto la convergenza; il nodo A deve ancora scoprire il percorso migliore verso C.

E manda il suo DV: $\{(A,1),(B,2),(C,3),(D,2)\}$ a $\{A,D\}$.
A scopre un nuovo percorso verso C con costo inferiore attraverso E.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	1	A	A	3	D	A	2	B	A	1	A
B	1	B	B	0	-	B	3	D	B	1	B	B	2	A
C	4	*E*	C	2	D	C	0	-	C	1	C	C	3	D
D	2	E	D	1	D	D	1	D	D	0	-	D	2	D
E	1	E	E	2	A	E	3	D	E	2	E	E	0	-

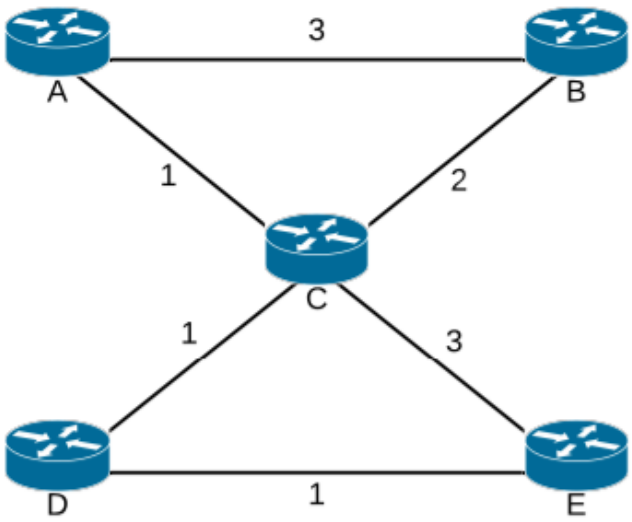
E' stato inviato il set completo di messaggi tra i router, ma non abbiamo raggiunto la convergenza; il nodo A deve ancora scoprire il percorso migliore verso C.

B manda il suo DV: $\{(A,1),(C,2),(D,1),(E,2)\}$ a $\{A,D\}$.
A finalmente scopre il percorso migliore verso C attraverso B, e così facendo raggiungiamo la convergenza.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	1	A	A	3	D	A	2	B	A	1	A
B	1	B	B	0	-	B	3	D	B	1	B	B	2	A
C	3	*B*	C	2	D	C	0	-	C	1	C	C	3	D
D	2	E	D	1	D	D	1	D	D	0	-	D	2	D
E	1	E	E	2	A	E	3	D	E	2	E	E	0	-

2.3 Esercizio 3

Usando l'algoritmo di routing di tipo Distance-vector, mostrare l'evoluzione delle tabelle di routing per ogni nodo, considerando che i DV vengono inoltrati dai router seguendo l'ordine: E, D, C, B, A.



Inizializzo le tabelle di routing di ogni nodo inserendo solo i nodi direttamente connessi (per semplicità includo tutte le destinazioni anche se sono ignote, queste avranno costo e NH vuoto).

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	3	A	A	1	A	A			A		
B	3	B	B	0	-	B	2	B	B			B		
C	1	C	C	2	C	C	0	-	C	1	C	C	3	C
D			D			D	1	D	D	0	-	D	1	D
E			E			E	3	C	E	1	E	E	0	-

E manda il suo DV: $\{(C,3),(D,1)\}$ a $\{D,C\}$.
Nulla cambia, non ci sono tratte migliori ne scoperta di nuovi nodi.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	3	A	A	1	A	A			A		
B	3	B	B	0	-	B	2	B	B			B		
C	1	C	C	2	C	C	0	-	C	1	C	C	3	C
D			D			D	1	D	D	0	-	D	1	D
E			E			E	3	C	E	1	E	E	0	-

D manda il suo DV: $\{(C,1),(E,1)\}$ a $\{C,E\}$.
E scopre un nuovo percorso verso C (e viceversa) con costo inferiore attraverso D.

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	3	A	A	1	A	A			A		
B	3	B	B	0	-	B	2	B	B			B		
C	1	C	C	2	C	C	0	-	C	1	C	C	2	*D*
D			D			D	1	D	D	0	-	D	1	D
E			E			E	2	*D*	E	1	E	E	0	-

C manda il suo DV: $\{(A,1),(B,2),(D,1),(E,2)\}$ a $\{C,E\}$.
A e B vengono a conoscenza dei nodi D e D (e viceversa).

A			B			C			D			E		
Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH	Dst	Cost	NH
A	0	-	A	3	A	A	1	A	A	2	*C*	A	3	*C*
B	3	B	B	0	-	B	2	B	B	3	*C*	B	4	*C*
C	1	C	C	2	C	C	0	-	C	1	C	C	2	D
D	2	*C*	D	3	*C*	D	1	D	D	0	-	D	1	D
E	3	*C*	E	4	*C*	E	2	D	E	1	E	E	0	-

In questo modo si ha raggiunto la convergenza (se si continua con l'inoltro dei DV, non si avranno più aggiornamenti dei percorsi, in quanto si ha già trovato il percorso migliore per ogni nodo).

3 TCP

3.1 Esercizio 1

- Dimensione file $D = 71540\,Bytes$
- SSTHR = 4 segmenti
- RWND = 8 segmenti
- RTO = 1 s
- $T_p = 50\ ms$
- $T_t = 0\ s$

1. Calcola il numero totale di segmenti da inviare (MTU = 1500 Byte):
   - MTU = 1500 Bytes
   - MSS = 1460 Bytes
   - $N = \dfrac{D}{MSS} = \dfrac{71540\,Bytes}{1460\,Bytes} = 49$ segmenti.
2. Calcolare il tempo necessario per trasmettere il file (ignorando i tempi per instaurare il collegamento tra i due host):
   - ciao
3. Calcolare il throughput T percepito dal livello applicativo:
   - $T = \dfrac{D}{T_r} = \dfrac{71540\,Bytes}{0.9\,s} = \dfrac{80\,kB}{s}$