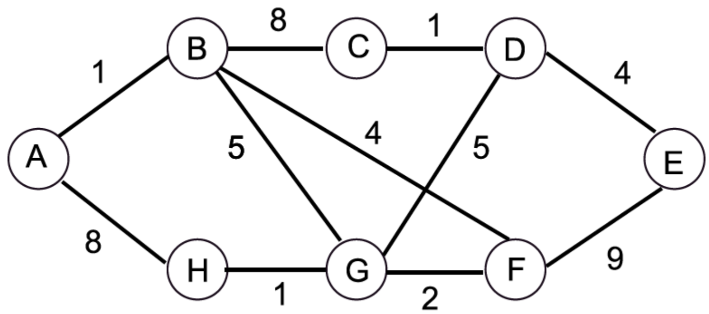
ELEN E6761: Communication Networks

Homework 6: Solutions

---

1. Computation process of Dijkstra's algorithm at node A.

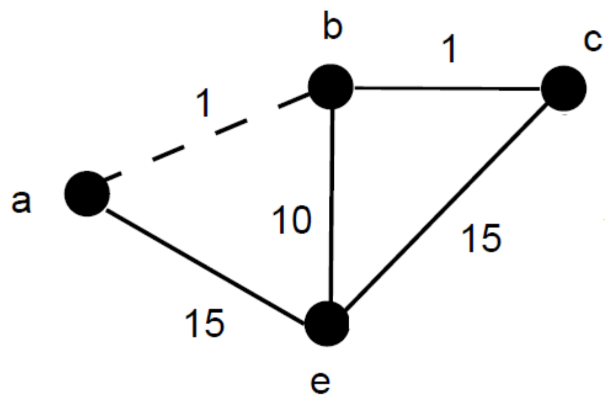


step	SPT	$(D(B), P(B))$	$(D(C), P(C))$	$(D(D), P(D))$	$(D(E), P(E))$	$(D(F), P(F))$	$(D(G), P(G))$	$(D(H), P(H))$
0	A	(1,A)	(∞,)	(∞,)	(∞,)	(∞,)	(∞,)	(8,A)
1	A,B		(9,B)			(5,B)	(6,B)	(8,A)
2	A,B,F		(9,B)		(14,F)		(6,B)	(8,A)
3	A,B,F G		(9,B)	(11,G)	(14,F)			(7,G)
4	A,B,F G,H		(9,B)	(11,G)	(14,F)			
5	A,B,F G,H,C			(10,C)	(14,F)			
6	A,B,F G,H,C D				(14,F)			
7	A,B,F G,H,C D							

2. Node A is the only destination. Note that the algorithm is terminated when there is no change in distance vector.

(D_B, H_B)	(D_C, H_C)	(D_D, H_D)	(D_E, H_E)	(D_F, H_F)	(D_G, H_G)	(D_H, H_H)
(1,A)	(∞,)	(∞,)	(∞,)	(∞,)	(∞,)	(8,A)
(1,A)	(9,B)	(∞,)	(∞,)	(5,B)	(6,B)	(8,A)
(1,A)	(9,B)	(10,C)	(14,F)	(5,B)	(6,B)	(7,G)
(1,A)	(9,B)	(10,C)	(14,F)	(5,B)	(6,B)	(7,G)

3. Transient loop:
   Using Dijkstra's algorithm to construct routing tables:



step	SPT	$(D(a), P(a))$	$(D(c), P(c))$	$(D(e), P(e))$
0	b	(∞,)	(1,c)	(10,e)
1	b,c	(∞,)		(10,e)
2	b,c,e	(25,e)		
3	b,c,e,a			

step	SPT	$(D(a), P(a))$	$(D(b), P(b))$	$(D(e), P(e))$
0	c	(∞,)	(1,b)	(15,e)
1	c,b	(2,b)		(11,b)
2	c,b,a			(11,b)
3	c,b,a,e			

step	SPT	$(D(a), P(a))$	$(D(b), P(b))$	$(D(c), P(c))$
0	e	(15,a)	(10,b)	(15,c)
1	e,b	(11,b)		(11,c)
2	e,b,c	(11,b)		
3	e,b,c,a			

b :

des.	link
a	(b,e)
c	(b,c)
e	(b,e)

c :

des.	link
a	(c,b)
b	(c,b)
e	(c,b)

e :

des.	link
a	(e,b)
b	(e,b)
c	(e,b)

According to above tables, packets from e to a will loop between nodes b and e:
e → b → e → b → e → ...

4. Node A is the only destination.

i)

(D_A, H_A)	(D_B, H_B)	(D_C, H_C)	(D_D, H_D)
(0,A)	(1,A)	(2,B)	(3,C)
(0,A)	(3,C)	(2,B)	(3,C)
(0,A)	(3,C)	(4,B)	(3,C)
(0,A)	(5,C)	(4,B)	(5,C)
(0,A)	(5,C)	(6,B)	(5,C)
⋮	⋮	⋮	⋮

ii)

(D_A, H_A)	(D_B, H_B)	(D_C, H_C)	(D_D, H_D)
(0,A)	(1,A)	(2,B)	(3,C)
(0,A)	(3,C)	(2,B)	(3,C)
(0,A)	(3,C)	(4,B)	(3,C)
⋮	⋮	⋮	⋮
(0,A)	(9,C)	(10,B)	(9,C)
(0,A)	(11,C)	(10,B)	(10,A)
(0,A)	(11,C)	(11,D)	(10,A)
(0,A)	(12,C)	(11,D)	(10,A)
(0,A)	(12,C)	(11,D)	(10,A)

After link AB goes down, it will take 12 iterations for all nodes to find the alternative path to node A. (Number of rows in above table minus 1, since the first row is initial state.)
Note that the algorithm is terminated when there is no change in distance vector.

iii) we will calculate the distance table for each node to show better how Poisoned Reverse works.

D^B	A	C
A	1	∞

D^C	B	D
A	2	∞

D^D	C	A
A	3	10

$\downarrow$

D^B	A	C
A	∞	∞

D^C	B	D
A	2	∞

D^D	C	A
A	3	10

$\downarrow$

D^B	A	C
A	∞	∞

D^C	B	D
A	∞	∞

D^D	C	A
A	3	10

$\downarrow$

D^B	A	C
A	∞	∞

D^C	B	D
A	∞	∞

D^D	C	A
A	∞	10

$\downarrow$

D^B	A	C
A	∞	∞

D^C	B	D
A	∞	11

D^D	C	A
A	∞	10

$\downarrow$

D^B	A	C
A	∞	12

D^C	B	D
A	∞	11

D^D	C	A
A	∞	10

$\downarrow$

D^B	A	C
A	∞	12

D^C	B	D
A	∞	11

D^D	C	A
A	∞	10