Recitation 20 11-29-2011

Today:

① Distance - Vector Protocol

② Link-state Protocol (Using Dijkstra's Algorithm)

$\boxed{\text{Routing}}$

1. Determing Live neighbours
   * common to distance-vector and link-state Protocol
   * Hello protocol.
2. Advertisement Step
   * send some information
3. Integration step
   * compute routing table using info from advertisement.

$\boxed{\text{Distance-Vector (DV) v.s Link-State (LS) protocol}}$

	DV	LS
What info is sent	routing table	Link and its cost
Route computation	Distributed	Centralized
Bandwidth Consumption	Low	High
Convergence time	SLOW	Fast
Robustness to misconfiguration	Low	High
Autonomy	High	Low
Memory and CPU time for route computation	Low	High

Dijkstra's Shortest Path Algorithm

Given: the map of whole network, (Link cost of each link)

Question: How to calulate routing table for a node.

Algorithm:

L0 * Initially

L1 nodeset = {all nodes} //nodeset: set of nodes we haven't processed

L2. $\text{spcost} = \begin{cases} 0 & \text{for me} \\ \infty & \text{for all other nodes;} \end{cases}$ //spcost: shortest path cost

L3. $\text{route} = \begin{cases} - \text{(doesn't care)} & \text{for me} \\ ? \text{ (unknown)} & \text{for all other nodes} \end{cases}$ // route: routing table

L4. * while nodeset isn't empty

L5 find u, the node in nodeset with smallest spcost

L6 remove u from nodeset

L7 for v in [u's neighbors in nodeset] //update u's neighbors if necessary

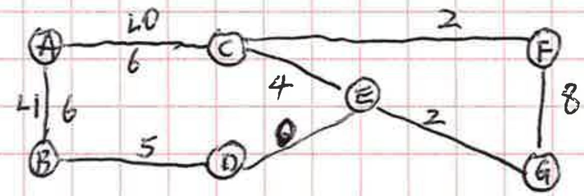
~~update sp~~

L8 if spcost(u) + linkcost(u, v) < current value of spcost(v)

L9 update spcost (v) = spcost(u) + linkcost(u, v)

L10 $\text{route(v)} = \begin{cases} \text{link from } u \text{ to } v & \text{if } u \text{ is me} \\ \text{route}(u) & \text{otherwise.} \end{cases}$

Example 1:



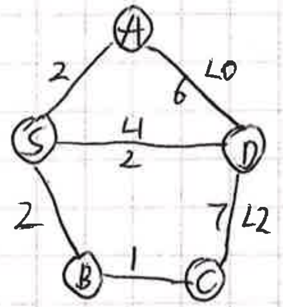
Question: For the network shown above, find the minimum cost path from node A to every other node.

Solution:

	the nodes with smallest cost(u)	nodeset	~~short~~ shortest path cost (spcost)	routing tables (route)
Initially.		A, B, C, D E, F, G	spcost(A)=0 spcost(B)=∞ spcost(C)=∞ spcost(D)=∞ spcost(E)=∞ spcost(F)=∞ spcost(G)=∞	route(A)=– route(B)=? route(C)=? route(D)=? route(E)=? route(F)=? route(G)=?
step 0:	A	B, C, D E, F, G	spcost(A)=0 spcost(B)=6 spcost(C)=6 spcost(D)=∞ spcost(E)=∞ spcost(F)=∞ spcost(G)=∞	route(A)=– route(B)=L1 route(C)=L0 route(D)=? route(E)=? route(F)=? route(G)=?
step 1.	B	C, D, E F, G	spcost(A)=0 spcost(B)=6 spcost(C)=6 spcost(D)=11 spcost(E)=∞ spcost(F)=∞ spcost(G)=∞	route(A)=– route(B)=L1 route(C)=L0 route(D)=L1 route(E)=? route(F)=? route(G)=?
step 2:	C	D, E F, G	spcost(A)=0 spcost(B)=6 spcost(C)=6 spcost(D)=11 spcost(E)=10 spcost(F)=8 spcost(G)=∞	route(A)=– route(B)=L1 route(C)=L0 route(D)=L1 route(E)=L0 route(F)=L0 route(G)=?
step 3:	F	D, E, G	spcost(A)=0 spcost(B)=6 spcost(C)=6 spcost(D)=11 spcost(E)=10 spcost(F)=8 spcost(G)=16	route(A)=– route(B)=L1 route(C)=L0 route(D)=L1 route(E)=L0 route(F)=L0 route(G)=L0
step 4:	E	D, G	spcost(A)=0 spcost(B)=6 spcost(C)=6 spcost(D)=10 spcost(E)=10 spcost(F)=8 spcost(G)=12	route(A)=– route(B)=L1 route(C)=L0 route(D)=L0 route(E)=L0 route(F)=L0 route(G)=L0
step 5.	D	G	spcost(A)=0 spcost(B)=6 spcost(C)=6 spcost(D)=10 spcost(E)=10 spcost(F)=8 spcost(G)=12	route(A)=– route(B)=L1 route(C)=L0 route(D)=L0 route(E)=L0 route(F)=L0 route(G)=L0
step 6.	G		spcost(A)=0 spcost(B)=6 spcost(C)=6 spcost(D)=10 spcost(E)=10 spcost(F)=8 spcost(G)=12	route(A)=– route(B)=L1 route(C)=L0 route(D)=L0 route(E)=L0 route(F)=L0 route(G)=L0

Example 2.

Consider the network below



Question: find the shortest cost paths from node D to every other node using Dijkstra's algorithm.
(e.g calculate node D's routing table)

Solution:

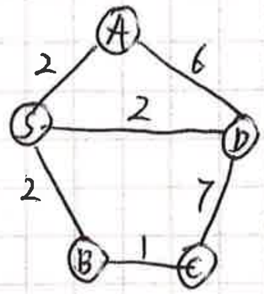
	The node with smallest path (u)	nodeset	shortest path cost (splost	
Initially		S, A, B, C, D	splost(S)=∞ splost(A)=∞ splost(B)=∞ splost(C)=∞ splost(D)=0	route(S)=? route(A)=? route(B)=? route(C)=? route(D)=–
Step 0.	D	S, A, B, C	splost(S)=2 splost(A)=6 splost(B)=∞ splost(C)=7 splost(D)=0	route(S)=L1 route(A)=L0 route(B)=? route(C)=L2 route(D)=–
Step 1	S	A, B, C	splost(S)=2 splost(A)=4 splost(B)=4 splost(C)=7 splost(D)=0	route(S)=L1 route(A)=L1 route(B)=L1 route(C)=L2 route(D)=–
Step 2	A	B, C	splost(S)=2 splost(A)=4 splost(B)=4 splost(C)=7 splost(D)=0	route(S)=L1 route(A)=L1 route(B)=L1 route(C)=L2 route(D)=–
Step 3	B	C	splost(S)=2 splost(A)=4 splost(B)=4 splost(C)=5 splost(D)=0	route(S)=L1 route(A)=L1 route(B)=L1 route(C)=L1 route(D)=–
Step 4	C		splost(S)=2 splost(A)=4 splost(B)=4 splost(C)=5 splost(D)=0	route(S)=L1 route(A)=L1 route(B)=L1 route(C)=L1 route(D)=–

So. Finally, the routing table of node D is:

Dest	route	cost
S	L1	2
A	L1	4
B	L1	4
C	L1	5
D		0

Example 3

conside the network as that in example 2 (shown betown again)

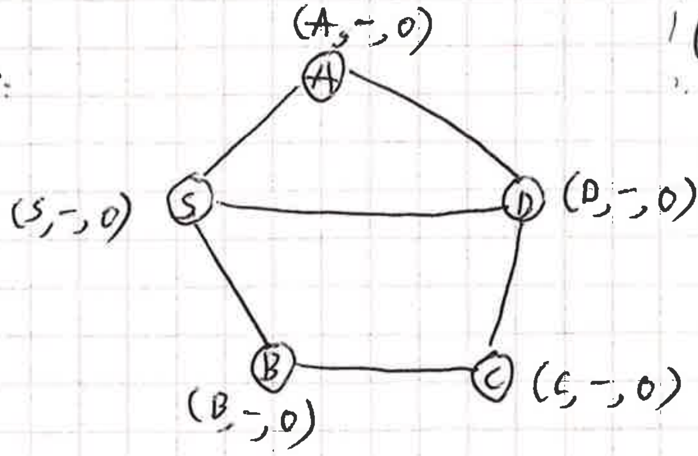


Question: Calculate routing table for node D using distance-vector protocol.

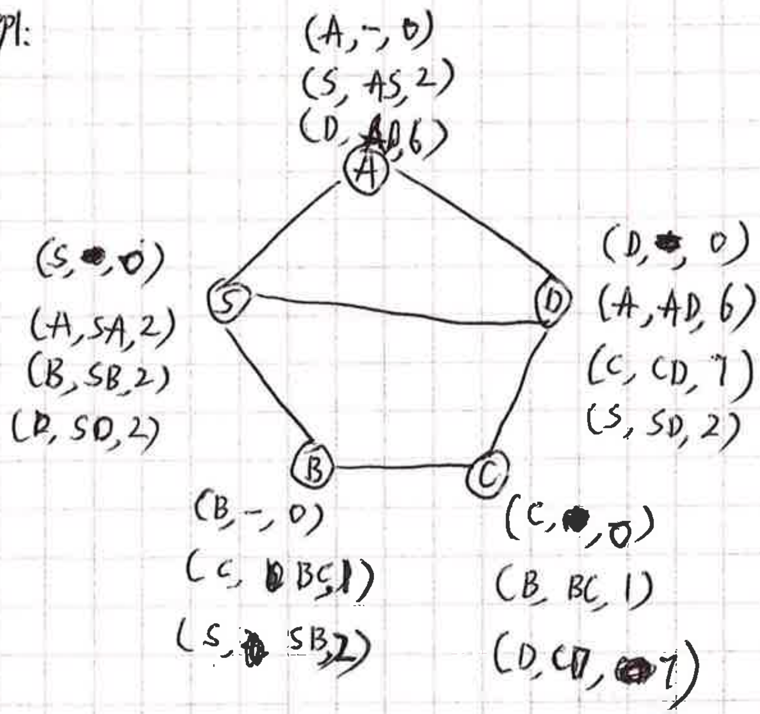
Solution:

(dest, route, cost)

step 0:



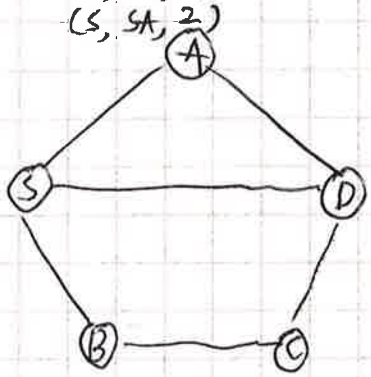
step 1:



step 3:

(A, -, 0)
(B, SA, 4)
(C, AD, 13)
(D, SA, 4)
(S, SA, 2)

(S, -, 0)
(A, SA, 2)
(B, SB, 2)
(D, SD, 2)
(C, SB, 3)

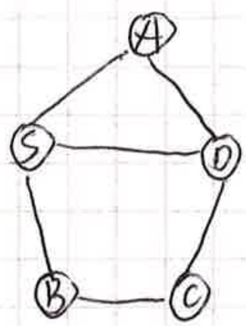


(D, -, 0)
(A, SD, 4)
(B, SD, 4)
(C, CD, 7)
(S, SD, 2)

(B, -, 0)
(A, SB, 4)
(C, BC, 1)
(D, SB, 4)
(S, SB, 2)

(C, -, 0)
(B, BC, 1)
(A, CD, 13)
(D, CD, 7)
(S, BC, 3)

step 4:



← (D, -, 0)
(A, SD, 4)
(B, SD, 4)
(C, SD, 5)
(S, SD, 2)

So. The final routing table of node D is:

Dest	route	cost
S	SD	2
A	SD	4
B	SD	4
C	SD	5
D	-	0

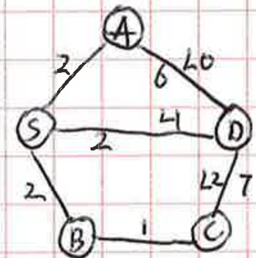
Which is same with what we got in Example 2.

(Here, SD is equal to L1 in Example 2)

Example 4:

Consider same network in example 2.

The network is shown below again for convenience:



Now we implement 6.02 distance-vector protocol

Each node sends its distance-vector advertisement every 100 sends.

The time to send a message on a link and to integrate advertisment is negligible

No advertisment losts.

Q1: At time 0, all nodes except D are up and running

At time 10 seconds, node D turns on and immediately sends a route advertisements for itself to all its neighbours.

what is minimum time at which each of the other nodes is guaranteed to have a correct routing table entry corresponding to a minimum-cost path to reach D.

Q2 If every nodes sends packets to destination D, and no other destination which link would carry the most traffic.

Solution 4

1) From Example 2, we know minimum cost between node D and other nodes is:

$cost(S, D) = 2$

$cost(A, D) = 4$

$cost(B, D) = 4$

$cost(C, D) = 5$

Here, using distance-vector protocol.

At time $t=10$ s,

D advertises to S, A, C.

After hearing this advertisment.

S updates its routing table: cost (S, D) = 2 (After hearing from D) ✓

A updates its routing table: cost(A, D) = 6 (After hearing from D)

C updates its routing table: cost(C, D) = 7 (After hearing from D)

Note, only S's routing table is correct.

At time $t=110$ s.

D, S, A, C all advertises about D.

After hearing these advertisemet,

A updates its routing table. cost (A, D) = 4 (After hearing from S) ✓

B updates its routing table: cost (B, D) = 4 (After hearing from S) ✓

C's routing table doesn't change. so cost (C, D) = 7.

Note: A and B's routing table is correct.

At time $t=210$ s.

All nodes (S, A, B, C, D) advertise about D.

After hearing these advertisement table.

C update its routing table: cost (C, D) = 5 (After hearing from B). ✓

Note. ~~C's routing table is correct~~.

In short: Node S: 10 s.

Node A: 110 s

Node B: 110 s

Node C: 210 s.

2) Every node's best route to D is via S.

So. the link between S and D has the most traffic.

(S → D)