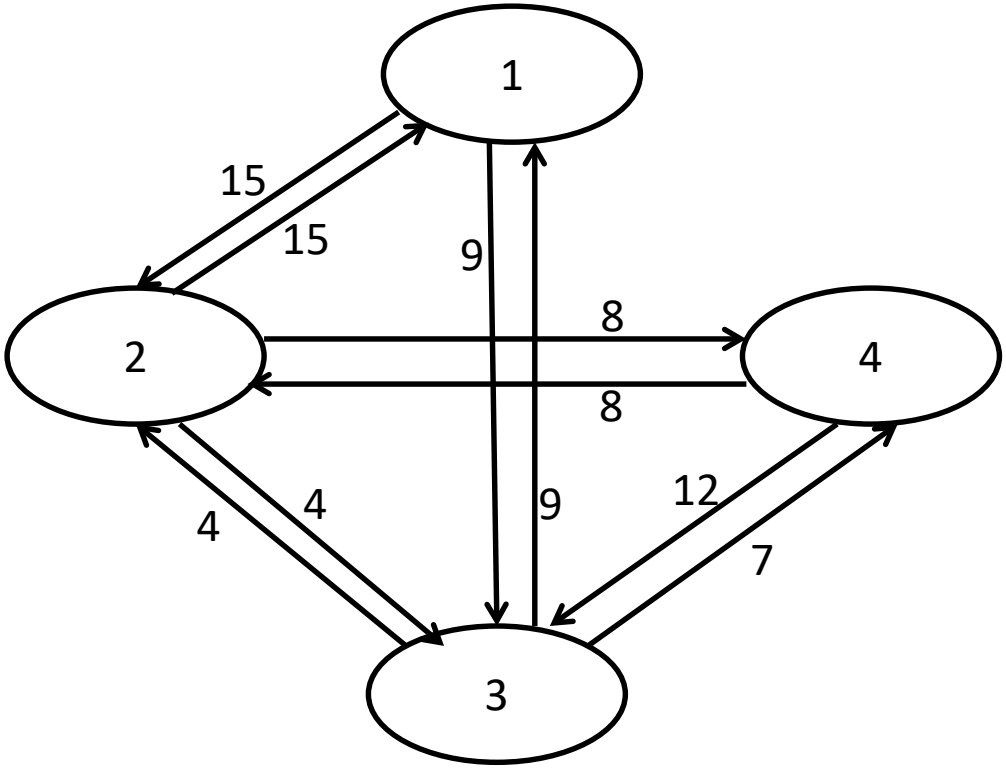
Assignment 3 – Part II

Huang Qun
SHB, 120
qhuang@cse.cuhk.edu.hk

Distance Vector Algorithm

- Each router x maintains:
  - Cost Table: distance to each neighbor
  - Neighbors' Distance Vectors
  - Routing Table
- Calculation:
  - $d_x(v)$:= cost of least-cost path from x to v
  - $d_x(y) = \min_v \{c(x,v) + d_v(y)\}$
  - $d_x(y)$ is stored in Neighbor's Distance Vector
  - c(x,v) is stored in Cost Table

Distance Vector Algorithm



Distance Vector Algorithm

- For Router 1 at the beginning:

Cost Table	
Router	Cost
2	15
3	9

Neighbors' Distance Vectors				
Router	To Router 1	To Router 2	To Router 3	To Router 4
2	-1	0	-1	-1
3	-1	-1	0	-1

Distance Vector Algorithm

- Use the former two tables to generate the Routing Table

Routing Table		
Router	Next Hop	Distance
1	1	0
2	2	15
3	3	9
4	-1	-1

Distance Vector Algorithm

- When Router 1 receives DV from other Router
  - add the DV received counter
  - change the Neighbors' Distance Vectors
  - recalculate the Routing Table
  - the routing table changed, propagate to its neighbors
- When Router 1 receives update command from Agent
  - decide whether to change(see tutorial slides Page 8-13)
  - if yes, change the Cost Table
  - recalculate the Routing Table

Example 1

- Router 1 receives a DV from Router 3(you must define your own format)

3	9	4	0	7

- Add the counter first!
- Change the Neighbors' Distance Vectors

Neighbors' Distance Vectors				
Router	To Router 1	To Router 2	To Router 3	To Router 4
2	-1	0	-1	-1
3	9	4	0	7

Example 1

- Recalculate the routing table

Routing Table		
Router	Next Hop	Distance
1	1	0
2	3	13
3	3	9
4	3	16

- Since the Routing Table changed, propagate!

Example 2

- Router 1 receives an Update command from agent: update:4,1,1
- (since there is no edges between Router 1 and 4 before, and edges are bi-directional, so Router 1 must change)
- Change the Cost Table

Cost Table	
Router	Cost
2	15
3	9
4	1

Example 2

- Recalculate the routing table

Routing Table		
Router	Next Hop	Distance
1	1	0
2	3	13
3	3	9
4	4	1

- Not propagate!

Traps, tips, and tricks...

- Trap #1: multiple threads is not required
  - Tips: a while loop works well in both Agent and Router program.
  - See the work flow of the two programs.

Traps, tips, and tricks...

- Trap #2: router IDs may not be consecutive.
  - Tips: map your own index to the router ID. (recommended)
  - Tricks: it won't hurt you much if you just use a sparse array.
  - Further, you can also use a sparse array to represent the neighbors and/or other things you need.

Traps, tips, and tricks...

- Trap #3: if you use -1 to represent infinity or no edges between routers, remember that (-1)+(-1)≠-1.
  - Tips: always use if ... then ... clause.

Traps, tips, and tricks...

- Trap #4: the network may not always be strongly connected.
  - Tips: use extensive test cases to test your program.

Traps, tips, and tricks...

- Trap #5: the network byte order may not be as you wish.
  - Tips: always use htonl() and ntohl().

Traps, tips, and tricks...

- Trap #6: TCP does not preserve message boundaries.
  - Tips: use a loop to send/receive message;
  - see the Socket Programming Tutorial and what you have done in previous assignments.

Traps, tips, and tricks...

- Trap #7: any two neighboring routers must be bi-directional connected, but the distance is unidirectional. Don’t mix the these two things.
  - Tips: each router only need to store the outer edges.

Traps, tips, and tricks...

- Trap #8: the agent doesn't know the topology of the network.
  - Tips: when update, agent sends commands to both the two related routers, then let the router decide whether to perform the update operations.

Suggestion

- Note: the following schedule is purely from my own experience. You may or may not employ it.
  - Step 1: consider the protocol formats, define data structures needed.
  - Step 2: write the configuration file parser, print them out and check whether the parsing is correct.
  - Step 3: write the agent program and a simple framework of router program, test if the agent can communicate with the routers.
  - Step 4: fill in the details of router program.

Some Reminders

- Do NOT cheat in any form.
  - Don’t dream that you could simulate everything without real network communications.
- Compile and test your programs in the VMs.

End

- Q & A

- Thank you!