Today:

1. Distance-Vector Protocol
2. Link-State Protocol (using Dijkstra’s Algorithm)

Routing

1. Determining 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

### Distance-Vector (DV) vs. Link-State (LS) Protocol

<table>
<thead>
<tr>
<th>What info is sent</th>
<th>DV routing table</th>
<th>LS Link and its cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route Computation</td>
<td>Distributed</td>
<td>Centralized</td>
</tr>
<tr>
<td>Bandwidth Consumption</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Convergence time</td>
<td>Slow</td>
<td>Fast</td>
</tr>
<tr>
<td>Robustness to misconfiguration</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Autonomy</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Memory and CPU time for route computation</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>
Dijkstra's shortest path algorithm

Given: the map of whole network, (link cost of each link)
question: How to calculate routing table for a node.

Algorithm:

```
20 * Initially

21 nodeset = {all nodes}  //nodeset: set of nodes we haven't processed
22 spost = [0] for me
23          [∞] for all other nodes;  //spost: shortest path cost
24 route = { - (doesn't care) for me
25         [?] (unknown) for all other nodes
26 * while nodeset isn't empty
27    find u, the node in nodeset with smallest spost
28    remove u from nodeset
29    for v in [u's neighbors in nodeset] //update u's neighbors if necessary
     update spost
     if spost(v) + linkcost(u,v) < current value of spost(v)
         update spost(v) = spost(v) + linkcost(u,v)
20 route(v) = { link from u to v if u is me
        route(u) otherwise
```
Example 1:

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

Solution:

Initially, assign the nodes with the smallest cost.

Network:

- A, B, C, D
- Shortest path cost
- (0, cost)

Routing Tables:

<table>
<thead>
<tr>
<th>Node</th>
<th>outgoing edge</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>5</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>7</td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>8</td>
</tr>
</tbody>
</table>

Routing Table:

<table>
<thead>
<tr>
<th>Node</th>
<th>outgoing edge</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>5</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>7</td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>8</td>
</tr>
</tbody>
</table>

**Routing Table**

- A (0, 5, 7, 6)
- B (0, 2, 4)
- C (0, 2, 4, 8)
- D (0, 6, 8)

Minimum cost paths:

- From A to B: 5
- From A to C: 7
- From A to D: 6
- From B to C: 2
- From B to D: 4
- From C to D: 8

**Routing Table**

<table>
<thead>
<tr>
<th>Node</th>
<th>outgoing edge</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>5</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>7</td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>8</td>
</tr>
</tbody>
</table>

Routing Table:

- A (0, 5, 7, 6)
- B (0, 2, 4)
- C (0, 2, 4, 8)
- D (0, 6, 8)

**Routing Table**

- A (0, 5, 7, 6)
- B (0, 2, 4)
- C (0, 2, 4, 8)
- D (0, 6, 8)
**Example 2.**

Consider the network below:

![Network Diagram](image)

**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:**

<table>
<thead>
<tr>
<th>Step</th>
<th>Node Set</th>
<th>Shortest Path Cost ($sp_{cost}$)</th>
<th>Route ($s$)</th>
<th>Route ($A$)</th>
<th>Route ($B$)</th>
<th>Route ($C$)</th>
<th>Route ($D$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$S, A, B, C$</td>
<td>$sp_{cost}(S) = 2$</td>
<td>$sp_{cost}(A) = 6$</td>
<td>$sp_{cost}(B) = 0$</td>
<td>$sp_{cost}(C) = 0$</td>
<td>$sp_{cost}(D) = 0$</td>
<td>$sp_{cost}(E) = 0$</td>
</tr>
<tr>
<td>1</td>
<td>$A, B, C$</td>
<td>$sp_{cost}(S) = 2$</td>
<td>$sp_{cost}(A) = 4$</td>
<td>$sp_{cost}(B) = 4$</td>
<td>$sp_{cost}(D) = 0$</td>
<td>$sp_{cost}(E) = 0$</td>
<td>$sp_{cost}(F) = 0$</td>
</tr>
<tr>
<td>2</td>
<td>$B, C$</td>
<td>$sp_{cost}(S) = 2$</td>
<td>$sp_{cost}(A) = 4$</td>
<td>$sp_{cost}(B) = 4$</td>
<td>$sp_{cost}(D) = 0$</td>
<td>$sp_{cost}(E) = 0$</td>
<td>$sp_{cost}(F) = 0$</td>
</tr>
<tr>
<td>3</td>
<td>$B, C$</td>
<td>$sp_{cost}(S) = 2$</td>
<td>$sp_{cost}(A) = 4$</td>
<td>$sp_{cost}(B) = 4$</td>
<td>$sp_{cost}(D) = 0$</td>
<td>$sp_{cost}(E) = 0$</td>
<td>$sp_{cost}(F) = 0$</td>
</tr>
<tr>
<td>4</td>
<td>$C$</td>
<td>$sp_{cost}(S) = 2$</td>
<td>$sp_{cost}(A) = 4$</td>
<td>$sp_{cost}(B) = 4$</td>
<td>$sp_{cost}(D) = 0$</td>
<td>$sp_{cost}(E) = 0$</td>
<td>$sp_{cost}(F) = 0$</td>
</tr>
</tbody>
</table>

So finally, the routing table of node $D$ is:

<table>
<thead>
<tr>
<th>Dest</th>
<th>Route</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>$L1$</td>
<td>2</td>
</tr>
<tr>
<td>$A$</td>
<td>$L1$</td>
<td>4</td>
</tr>
<tr>
<td>$B$</td>
<td>$L1$</td>
<td>4</td>
</tr>
<tr>
<td>$C$</td>
<td>$L1$</td>
<td>5</td>
</tr>
<tr>
<td>$D$</td>
<td>$L1$</td>
<td>0</td>
</tr>
</tbody>
</table>
Example 3

Consider the network as that in example 2 (shown below again).

Question: Calculate routing table for node 0 using distance-vector protocol.

Solution:

Step 0:

- $(A, -, 0)$
- $(S, -, 0)$
- $(B, -, 0)$
- $(C, -, 0)$
- $(D, -, 0)$

$(dest, route, cost)$

Step 1:

- $(A, -, 0)$
- $(S, AS, 2)$
- $(D, AD, 6)$
- $(B, SB, 2)$
- $(C, SB, 2)$
- $(B, -, 0)$
- $(C, DBS, 1)$
- $(S, SB, 2)$
- $(D, CD, 7)$
- $(C, -, 0)$
- $(B, BC, 1)$
- $(D, CD, 7)$
Step 2:

So. The final routing table of node D is:

<table>
<thead>
<tr>
<th>Dest</th>
<th>Route</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>SD</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>SD</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>SD</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>SD</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

Which is same with what we got in Example 2.

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

Consider same network as 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 seconds. The time to send a message on a link and to integrate advertisement is negligible. No advertisement 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 node sends packets to destination D, and no other destination, which link would carry the most traffic?

Solution:

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's a distance-vector protocol.

At time $t=10 \, s$:

- D advertises to $S$, $A$, $C$.
- After hearing this advertisement:
  - $S$ updates its routing table: $\text{cost}(S, D) = 2$ (After hearing from D)
  - $A$ updates its routing table: $\text{cost}(A, D) = 6$ (After hearing from D)
  - $C$ updates its routing table: $\text{cost}(C, D) = 7$ (After hearing from D)

Note: Only $S$'s routing table is correct.

At time $t=110 \, s$:

- $S$, $A$, $C$ all advertise about $D$.
- After hearing these advertisements:
  - $A$ updates its routing table: $\text{cost}(A, D) = 4$ (After hearing from $S$)
  - $B$ updates its routing table: $\text{cost}(B, D) = 4$ (After hearing from $S$)
  - $C$'s routing table doesn't change, so: $\text{cost}(C, D) = 7$.

Note: $A$'s 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 advertisements:
  - $C$ updates its routing table: $\text{cost}(C, D) = 5$ (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.