Overview

Routing Algorithms
- Link-state – From last week
- Distance-vector – TODAY

Recap of Routing So Far

Questions:
- What is the objective of routing?
- Does routing occur between hosts or routers?
- What are differences between centralized (global) and decentralized algorithms?
- What are examples of each?
- Amount of information initially
- How information is shared/spread
- Synchronous or asynchronous?
  - (see pathologies as well)

Algorithm 2: Distance Vector

Rather than using global information, a distance vector algorithm is:

- distributed:
  - each node communicates only with directly-attached neighbors
- iterative:
  - continues until no nodes exchange info.
  - self-terminating: no “signal” to stop
- asynchronous:
  - nodes need not exchange information or iterate in lock step!
Distance Vector Algorithm

**Bellman-Ford Equation**, an important relationship among costs of least-cost paths

Define

$$d_x(y) := \text{cost of least-cost path from } x \text{ to } y$$

Then

$$d_x(y) = \min \{ c(x,v) + d_v(y) \}$$

where min is taken over all neighbors v of x

Distance Vector Routing Algorithm

**Distance Table data structure**

- each node has its own
  - row for each possible destination
  - column for each directly-attached neighbor
- example: in node X, for destination Y via neighbor Z:

$$D^X(Y,Z) = \text{distance from } X \text{ to } Y, \text{ via Z as next hop}$$

$$= c(X,Z) + \min_w \{D^Z(Y,w)\}$$

Bellman-Ford Equation

Clearly, $$d_v(z) = \infty$$, $$d_x(z) = \infty$$, $$d_u(z) = \infty$$

B-F equation says:

$$d_v(z) = \min \{ c(u,v) + d_v(z), c(u,x) + d_x(z), c(u,w) + d_w(z) \}$$

$$= \min \{ , , \} \infty$$

The node that achieves the minimum, is the next hop in the shortest path $\rightarrow$ forwarding table

Distance Table: example with complete information

<table>
<thead>
<tr>
<th>Destination</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

$$D^E(C,D) = c(E,D) + \min_w \{D^D(C,w)\}$$

$$= c(E,D) + \min_w \{D^D(C,w)\}$$

$$D^E(A,D) = c(E,D) + \min_w \{D^D(A,w)\}$$

$$= c(E,D) + \min_w \{D^D(A,w)\}$$

$$D^E(A,B) = c(E,B) + \min_w \{D^B(A,w)\}$$

$$= c(E,B) + \min_w \{D^B(A,w)\}$$
### Distance table to forwarding table

<table>
<thead>
<tr>
<th>Destination</th>
<th>Outgoing Link to Use</th>
<th>Cost to Destination Via</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Distance vector algorithm

**Basic idea:**
- Each node begins with $D_x(y)$
  - An estimate of the cost of the least-cost path from itself to node $y$, for all nodes in $N$
- Each node periodically sends its own distance vector estimate to neighbors
- When a node $x$ receives new DV estimate from neighbor, it updates its own DV using B-F equation, and sends any update to its neighbors
  \[
  D_x(y) \leftarrow \min \{c(x,y) + D_y(y)\} \quad \text{for each node } y \in N
  \]
- Under normal conditions, the estimate $D_x(y)$ converge to the actual least cost $d_x(y)$

### Distance Vector Algorithm: example for obtaining complete information

<table>
<thead>
<tr>
<th>X</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

\[
D^X(Y,Z) = c(X,Z) + \min_w(D^Z(Y,w))
\]

\[
D^X(Z,Y) = c(X,Y) + \min_w(D^Y(Z,w))
\]

### Distance Vector Algorithm: obtaining info

<table>
<thead>
<tr>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

\[
D^Y(X,Z) = c(Y,Z) + \min_w(D^Z(X,w))
\]

\[
D^Y(X,Y) = c(X,Y) + \min_w(D^X(Y,w))
\]

\[
D^Y(X,Y) = c(X,Y) + \min_w(D^X(Y,w))
\]
Comparison of LS and DV algorithms

- Information requirements
- Message complexity
- Convergence time varies
- Robustness: what happens if router malfunctions?
- Oscillations possible?
- Loops possible?

Summary

Forwarding:
- Leads to questions of addressing
  - Assignment of IP addresses
  - NAT, IPv6 ...

Routing:
- Routing objectives
- Routing notation
- Routing classification
- Link state v. Distance Vector
- Hierarchical structure