CSE123A discussion session

2007/03/08    Ryo Sugihara

Topics

• Review
  – Network Layer (2,3): Route computation
    • Distance vector
    • Link state
Where are we now?

Route computation

- Router needs to know “which port to forward”
  - Done by “routing table”

- But how to make routing table?
  - By exchanging routing information
  - What kind of information?

Routing table

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>001*</td>
<td>A</td>
</tr>
<tr>
<td>0*</td>
<td>B</td>
</tr>
<tr>
<td>01*</td>
<td>C</td>
</tr>
<tr>
<td>11*</td>
<td>C</td>
</tr>
<tr>
<td>110*</td>
<td>D</td>
</tr>
<tr>
<td>default</td>
<td>B</td>
</tr>
</tbody>
</table>
Distance vector & Link state

- Two ways to exchange routing info
  - Distance vector
    - Exchange "distance to a destination"
    - Each router knows the next hop for each dest
      - One hop of the shortest path to the dest
  - Link state
    - Exchange "my neighbors (and costs)"
    - Each router knows whole network topology
      - compute the shortest paths by Dijkstra’s algorithm

Example

CSE (132.239.10.0/24)

\[
\begin{array}{c}
R1 & R2 & R3 & R4 & R5 \\
1 & 5 & 2 & 3 & 4 \\
& & 8 & & 10 \\
& & & & \\
\end{array}
\]
**General strategy**

- R1 (somehow) knows CSE is distance one from itself

- R1 tells the neighbors (=R2, R3) about it
  - DV: “CSE is distance one from me”
  - LS: “I have link with CSE (d=1), R2 (d=5), and R3 (d=1)”

**Distance vector**

- R1 tells “CSE is distance one from me!”
  - to R2 and R3
What happens at R2?

- Routing table
  - Net | dist | iface
  - ...
  - CSE 6 | p1 |
  - ...

- Table for each interface
- One routing table

Distance vector

- Router forwards distance info to others when its routing table is updated
What happens at R2?

Routing table

<table>
<thead>
<tr>
<th>Net</th>
<th>dist</th>
<th>iface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSE</td>
<td>6</td>
<td>p2</td>
</tr>
</tbody>
</table>

In case of link failure,

- Switch to backup route
- Problem: “Count-to-infinity problem” (next)
Count-to-infinity problem

R1 broadcasts “I have link with CSE (d=1), R2 (d=5), and R3 (d=1)”
  – It will be flooded to the entire network

Link state
**Link state**

- Each router has the **entire graph**
  - In DV, it only knew next hop for each dest.
- Each router computes next hop for each dest.
  - By Dijkstra’s algorithm

![Link state diagram](image)

**Intelligent flooding**

- Without “intelligence”, LSP may loop
  - If a router received a “newer” LSP, it forwards
  - Otherwise, it discards

- Solution: add **sequence number** to LSP
  - If a router received a “newer” LSP, it forwards
  - Otherwise, it discards

- Other issues:
  - Subtle problems: “source jumping”, “aging”  → lecture notes
  - How Dijkstra’s algorithm works  →