CSE 461 - Module 9: IP Routing

Comparison with Link Layer (Bridges)

<table>
<thead>
<tr>
<th></th>
<th>Learning Bridges</th>
<th>Learning Bridges + Spanning Tree</th>
<th>IP Routers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Mechanism</td>
<td>Forwarding table</td>
<td>Forwarding table</td>
<td>Forwarding table</td>
</tr>
<tr>
<td>Destination Addresses</td>
<td>MAC</td>
<td>MAC</td>
<td>IP</td>
</tr>
<tr>
<td>Scale</td>
<td>10's?</td>
<td>10's?</td>
<td>100's</td>
</tr>
<tr>
<td>Assumed network</td>
<td>Loop free (tree)</td>
<td>Arbitrary</td>
<td>Arbitrary</td>
</tr>
<tr>
<td>topology</td>
<td>None</td>
<td>Min-cost spanning tree</td>
<td>Min-cost spanning tree</td>
</tr>
<tr>
<td>Topology computation</td>
<td>None</td>
<td>Min-cost spanning tree</td>
<td>Min-cost spanning tree</td>
</tr>
<tr>
<td>Destinations</td>
<td>Individual network interface cards</td>
<td>Individual network interface cards</td>
<td>Aggregates of IP addresses</td>
</tr>
<tr>
<td>Action on lookup miss</td>
<td>Flood</td>
<td>Flood</td>
<td>Drop packet</td>
</tr>
<tr>
<td>Gathering forwarding table entries</td>
<td>Passive – learn from source fields of frames that come by</td>
<td>Passive – learn from source fields of frames that come by</td>
<td>Active – routers engage in a forwarding table maintenance protocol</td>
</tr>
</tbody>
</table>

DV – Distance Vector

- Distributed Bellman-Ford
  - Periodically tell each neighbor how far you are from every destination
  - When you hear from a neighbor, for each destination D, compare distance to neighbor + neighbor’s distance to D with your currently recorded distance to D.
  - If going through that neighbor is shorter than the path you already knew about
    - Enter the neighbor in your forwarding table as the next hop to reach D
    - Update your distance to D to be distance to neighbor + neighbor’s distance to D
- Robustness? Convergence?
- Traffic pattern
  - Send message w/ all destinations to your neighbors
**DV Issue – Count to infinity**

- What are forwarding tables at A, B, C, and D?
- What happens if the link C-D goes down?
- Ad hoc approaches to the issue
  - Split horizon; poison reverse

**Link State Routing**

- Idea:
  - every router individually acquires information about the current state of all links in the network
  - each one then computes a minimum cost spanning tree rooted at itself
    - if all routers have the same link state information, they compute the same (or at least compatible) trees
  - it then uses the min-cost spanning tree to set its forwarding table
- Each router must tell every other router about the links it's connected to
  - Link state advertisements (LSAs)
  - Flood them (why?)
    - Want reliability (why?)
      - Need sequence numbers
        - How do you use the sequence numbers?
- Traffic pattern
  - Send message about your directly connected links to everyone