Announcements

- Anonymous feedback form on website
- Won’t get to mobility
  - Read 4.2.5 if you are interested
  - Come to office hours if you have questions
- Homework 1 due next time
  - Questions?

Most important ideas from last time

- Clarification of spanning trees
- Soft state
- Why use routing instead of spanning trees
- Routing - “Routing is cool”
  - Using shortest paths
  - Routing vs. forwarding
- Distance vector routing
  - Counting-to-infinity problem

Questions from last time

- Today we’ll address:
  - Split horizon & poison reverse unclear
  - Do people use distance vector?
  - If path vector isn’t used to avoid counting to infinity, what is?
  - How to determine link costs?
- Later we’ll address:
  - Routing at a global scale
Count To Infinity Problem

A 2  B 1

The Internet

Approaches to routing

- Distance vector:
  - Distribute information about all nodes to my neighbors

- Link state:
  - Distribute information about my neighbors to all nodes

Link State Routing

- Idea:
  Each router learns the complete network topology and computes its own best paths

- Two phases:
  1.
  2.

Why link state routing?

- In DV, routers hide their computation, making it difficult to decide what to do when there are changes
  - e.g.,

- With LS, faster convergence and hopefully better stability
- But it is more complex...
**Link State Assumptions**

- Each router knows
  - Who neighbors are
  - Cost to each neighbor
  - Whether each link is up or down

**LSP Flooding**

- Each router periodically floods a link state packet (LSP)
- LSP contains
  - Cost
  - Whether each link is up or down
- Each router maintains a database of most recent LSPs

---

**Shortest Paths: Dijkstra’s Algorithm**

- Graph algorithm for single-source shortest path

\[
S \leftarrow \emptyset \\
Q \leftarrow \text{<all nodes keyed by distance>} \\
\text{While } Q \neq \emptyset \\
\quad u \leftarrow \text{extract-min}(Q) \\
\quad S \leftarrow S \cup \{u\} \\
\quad \text{for each node } v \text{ adjacent to } u \\
\quad \text{“relax” the cost of } v
\]

**Relaxation step**

If \( \text{cost}(u) + \text{cost}(u,v) < \text{cost}(v) \)

update cost(v)
Dijkstra Example – Your answer

Dijkstra Example – Class answer

What if LSPs get lost?

• What could happen?

• What can we do?

What if a link or router fails?

• Need to remove old data. How?

• What if it fails repeatedly?
What happens when a router fails and restarts?

- What sequence number should it use? Don’t want data ignored.

Cost Metrics

- How should we choose cost?
  - To get high bandwidth, low delay, or low loss?
  - Does cost depend on the load?

Static vs. Dynamic Metrics

- Static Metrics
  - Hopcount is easy but treats OC3 (155 Mbps) same as T1 (1.5 Mbps)
  - Can tweak result with manually assigned costs

- Dynamic Metrics
  - Depend on load; try to avoid hotspots (congestion)
  - But can lead to oscillations

Revised ARPANET Cost Metric
### Alternatives to routing

- Goal: Avoid packet reordering

- Source routing:

- Virtual circuits:

### Establishing a virtual circuit

- Send circuit setup request

- Each node records state for next hop in path for this VC

- What if request is dropped along path?

---

### Hard state vs. Soft state

- Hard state:

- Soft state:

### What if a link fails?

- With hard state:

- With soft state:
### Network Service Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Datagram</th>
<th>Virtual circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metaphor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other terms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packet forwarding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Router state</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Datagrams or Virtual Circuits?

- **Pros and Cons?**
  - Simplicity/robustness versus stronger resource allocation
- **We return to these tradeoffs later**
  - Quality of Service (QOS)
  - At the heart of current Internet evolution
  - Intserv (connection oriented)
    vs Diffserv ("connectionless")

### Key Concepts

- How can we get packets across the network efficiently in the face of node and link failures?
  - Learning bridges & spanning trees
  - Distance vector routing
  - Link state routing
  - Virtual circuits
  - Soft state and learning

### Next time...

- Where in a system should we place functionality?
- Saltzer et al., “End-to-End Arguments in System Design”