Internet Routing:
This is a graph
Networks can be modeled as a graph.
Trees on graphs

- You can superimpose a directed tree on a graph
  - No loops (or cycles)
- This tree is rooted at a destination
- Every directed edge (link) in this tree represents a “next hop”
- Forming these trees (per destination) is the essence of the routing problem
- By the way, what do you get if you reverse the directions of the tree edges?
Trees on graphs

- Lots of trees are possible
- Two of interest:
  - Shortest path spanning tree
    - Sum of weights on links from any node to destination is the smallest
  - Minimum weight spanning tree
    - Sum of weights of all links is the smallest
- In the internet, we calculate shortest paths
Setting costs on links

- Each “link” is actually two unidirectional links
  - But we’ll pretend they are bidirectional in the lecture for convenience
- One way: set link cost as the inverse of the BW
  - i.e., take highest BW link, give it a cost of one, weight all other links inverse proportionally
- Does this work?
Setting costs on links

- Ultimate goal is to balance traffic over links
  - No overloaded or underloaded links
- Load on links depends on traffic matrix
  - That is, amount of traffic between every src-dest pair
- In practice, costs are “hand tuned” to get good balance
  - Of course, BW is added or removed as needed
- Link underloaded? Lower its cost a bit . . .
Can we dynamically set costs?

- Measure volume of traffic, increase cost as volume increases, decrease cost as volume decreases
- It turns out that this is hard to get right
  - Route oscillations---must be damped
Can we dynamically set costs?

- If network is lightly loaded, traffic-sensitive routing doesn’t help
  - All routes are good
- If network is overloaded, traffic-sensitive routing also doesn’t help
  - Alternate routes also bad
- What you really want is to throttle sources at times of overload. **This is the big win.**
Multiple link metrics?

- What if we care about BW and latency and MTU and QoS???
- Many people have studied multi-metric routing
  - It gets complex, and it is hard to figure out what to do with it
- In practice: Over-provision, throttle sources!
Three classes of IP routing algorithm in the Internet today

- Distance-vector
  - RIP
- Distance-path
  - A variant of distance-vector
  - BGP
- Link-state
  - OSPF, IS-IS
Distance Vector

- Also known as Bellman-Ford
- Each node’s routing table:
  - The distance to each destination via each neighbor
- Building the forwarding table:
  - The next hop to each destination is the neighbor with the shortest distance
- The algorithm:
  - Periodically tell each neighbor the shortest distance to all destinations
  - This is the so-called “distance vector”
Example RIB, FIB, and routing update message

**RIB**

<table>
<thead>
<tr>
<th>Dest</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>5</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>D2</td>
<td>2</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>D3</td>
<td>11</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>D4</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

**FIB**

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>N2</td>
</tr>
<tr>
<td>D2</td>
<td>N1</td>
</tr>
<tr>
<td>D3</td>
<td>N2</td>
</tr>
<tr>
<td>D4</td>
<td>N1</td>
</tr>
</tbody>
</table>

Update Message: \((D_1:1)(D_2:2)(D_3:2)(D_4:3)\)
Examples

- Show establishment of tree
- Show link addition and changes in tree
- Show link deletion and changes in tree
- Show node removal and count-to-infinity
Count-to-infinity fix

- Split horizon
  - Don’t advertise reachability to a neighbor if the destination is reached via that neighbor
- Also triggered updates
  - Instantly report changes (not just periodically)
  - Count-to-infinity fast!
- This fixes “ping-pong” CTI, but doesn’t solve the general problem…
Example with split horizon
RIP-2 header (RFC 2453)
Distance-path

- The problem with distance-vector is that a node never knows whether a path loops back through itself.

- With distance-path algorithm, the entire path to the destination is reported.
  - This is not so much overhead, because network diameters are generally small.
Distance-path example
Distance-path pros and cons

- New paths have to be advertised even when the distance doesn’t change
  - More traffic overhead…this has caused havoc in BGP

- More policy control (BGP)
  - Path is known, so can make more intelligent selection among alternatives
  - But still “hostage” to policy decisions made before you in the path