Announcement

• Submit mock HW through Moodle by Friday
Intradomain Routing

• Beyond extending LANs to a few segments

• Equipment and link heterogeneity
Basic Algorithms

• Distance Vector Routing

• (Path Vector)

• Link State Routing
Distance Vector Routing Overview

• All the nodes advertise the smallest known cost to a destination
• Next hop is the node on the minimum metric path for each destination
• Loops

http://en.wikipedia.org/wiki/Distance-vector_routing_protocol
The Algorithm

• Distributed Bellman-Ford
• Initialization at the destination: distance = 0
• All nodes periodically broadcast their best known distance to the destination
• On each RX
  – New distance= RX.distance + link length
  – If new distance < distance
    • Update distance and next hop link
• Extend to multiple destinations

DV Routing / One Destination

The diagram shows a network with nodes labeled DST, B, C, D, E, and F. Connections are indicated by edges between the nodes, with numerical values representing costs or distances, such as 0, 1, and 2.

The nodes are connected as follows:
- DST to B: 0
- DST to D: 1
- B to D: 1
- C to D: 1
- C to E: ?
- D to F: 2
- E to F: ?
A Network Deployment
Single Destination DV
DV Routing / Multiple Destinations

Graph:
- DA
- DB
- DD
- DC
- DE

Routing:
- DA: 0
- DB: 0, self
- DD: 1, DD
- DA: 1, DA
- DC
- DE: 0
- DD: 0, self
- DB: 1, DB
- DA: 1, DA
- DF

Directed connections: 7
Routing State

• Amount of State
• Inconsistency
  – States on all the routers don’t agree
• Stale
  – Takes time for changes to be reflected in routing state
• Potential problems
  – Loops and Suboptimal Choices
Convergence

• Periodic messaging
• Packet size?
• Information from one end needs to get all the way to the other end of the network
  – Diameter of the network
  – Frequency of updates
• How many messages/node needed for convergence?
Count to Infinity

• The distance increases to infinity
• Loops

http://en.wikipedia.org/wiki/Count_to_infinity
Split-Horizon

• Do not advertise back to the source

http://en.wikipedia.org/wiki/Split_Horizon
Route Poisoning and Poison Reverse

• Advertise large cost / invalid route to the neighbor from which the route was learned
Path Vector

- Include path in the advertisement

http://en.wikipedia.org/wiki/Path_vector_protocol
Link State Routing Overview

• Neighborhood information sent to all the nodes: \{\text{node } i, \text{neighbors} - [n1,n2,...]\}
  
  – Reliable Flooding of Link State information

• Each node constructs a full network graph and runs Dijkstra

http://en.wikipedia.org/wiki/Link-state_routing_protocol
Dijkstra’s Algorithm

• Each node knows $G = \{V,E\}$
• Initialize:
  – distance = 0 for dst, inf. for the rest of the nodes
• Maintain a queue sorted by distance
• Remove the node $n$ with shortest distance and iterate through all its neighbors $i$
  – New distance($i$) = node cost($n$) + link cost ($i$-$n$)
  If new distance < best known distance
    • Update distance
    • New next hop = $n$
• Sort by distance, iterate through rest of the nodes

http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
Dijkstra’s Algorithm
Routing State

- Amount of memory
- Inconsistency
- Stale
- Potential problems
  - Sometimes Loops and Suboptimal Choices

- Source route with the path
Convergence

• Send updates after initial flooding
• Packet size?
• Usually Faster than DV

http://docwiki.cisco.com/wiki/Routing_Basics#Link-State_Versus_Distance_Vector
Link Metric

• Reflects the cost of the link
  – Hops
  – Latency
  – Capacity

• Additive
  – Can convert some multiplicative metrics to additive
OSPF Packet Types

- Hello
- Database Description
- Link State Request
- Link State Update
- Link State Ack
OSPF

• A link state protocol
• We abstractly say OSPF packets contain
  – Node ID, Seq, TTL, List of neighbors
• Protocol specifications
  – What do these packets look like?
  – Where to find this information?
# OSPF Packet Header

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

[+-----------------+-----------------+-----------------+-----------------+]
| Version # | Type | Packet length |               |
[+-----------------+-----------------+-----------------+-----------------+]
|               | Router ID |               |               |
[+-----------------+-----------------+-----------------+-----------------+]
|               | Area ID |               |               |
[+-----------------+-----------------+-----------------+-----------------+]
|               | Checksum |               | AuType |
[+-----------------+-----------------+-----------------+-----------------+]
|               | Authentication |               |               |
[+-----------------+-----------------+-----------------+-----------------+]
|               | Authentication |               |               |

[RFC2328]
OSPF Packet Header

```
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
RIP Packets

The packet format is shown in Figure 1. Format of datagrams containing network information. Field sizes are given in octets. Unless otherwise specified, fields contain binary integers, in normal Internet order with the most-significant octet first. Each tick mark represents one bit.

<table>
<thead>
<tr>
<th>command (1)</th>
<th>version (1)</th>
<th>must be zero (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>address family identifier (2)</td>
<td>must be zero (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP address (4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>must be zero (4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>must be zero (4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>metric (4)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[RFC1988]

Can RIPv1 and RIPv2 inter-operate?

The RIP packet format is:

```
<table>
<thead>
<tr>
<th>command (1)</th>
<th>version (1)</th>
<th>must be zero (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIP Entry (20)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The same header format is used for RIP-1 and RIP-2 messages (see section 3.4). The format for the 20-octet route entry (RTE) for RIP-2 is:

```
| Address Family Identifier (2) | Route Tag (2) |
| IP Address (4)               |               |
| Subnet Mask (4)              |               |
| Next Hop (4)                 |               |
| Metric (4)                   |               |
```

[RFC2453, 1998]
Naming

• Name must scale to a large number of nodes
• IPv4 Address
  – Classes (A,B,C)
  – Subnets
  – Classless
    • More flexible bit boundaries

http://en.wikipedia.org/wiki/IP_address
Classless Inter-Domain Routing

• Contiguous addresses can be aggregated
  – Why aggregate?

• Two prefixes
  • 239.109.104.0/22: 11101111:01101101:011010xx:xxxxxxxxxx
  • 239.109.108.0/22: 11101111:01101101:011011xx:xxxxxxxxxx

• Aggregate
  • 239.109.104.0/21: 11101111:01101101:01101xxx:xxxxxxxxxx

• Advertise reachable prefix with the routing algorithms

http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing
Routing with CIDR

• Aggregate and announce whenever possible
Forwarding with CIDR

• Longest Prefix Match

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Nexthop</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.b.0.0/23</td>
<td>A</td>
</tr>
<tr>
<td>a.b.1.0/24</td>
<td>C</td>
</tr>
</tbody>
</table>

Where to forward these packets?
- dst: a.b.0.5
- dst: a.b.1.6