CS 348: Computer Networks

- Routing (RIP); 4th Sept 2011

Instructor: Sridhar Iyer
IIT Bombay
Routing - Clicker Question

1. How does a router know which path to forward a packet on?
   A) It broadcasts an ARP request to find the destination.
   B) The route is given in the IP header of the packet.
   C) It constructs a routing table and does a lookup to find route.
   D) It forwards the packet to the default gateway.

   • Choose one of the options below.

   1. All of the above
   2. A), C) and D)
   3. C) and D)
   4. C) only
Routing Questions

● How does a router know which path to forward a packet on?
  ▪ Need to construct and lookup routing tables

● How does a router know whether its link with its neighbour is up or down?
  ▪ Need to exchange periodic “keep-alive” messages.

● How does a router know whether a distant link is up or down?
  ▪ Need to propagate “link-state” information.
Routing

• Routing is the mechanism of forwarding messages towards the destination node based on its address.

• Routers decide routes for packets, based on destination address and topology.

• Routers need to learn global information and compute routes to various networks.

• Exchange information with other routers to learn network topology.

• Maintain a table of available routes and their conditions.

• Use table along with path algorithms to determine the best route for a packet.
Routing methods

• Static routing: Default routes are specified at boot time

• Dynamic methods:
  • Source-based: Specify route at source (DSR)
  • Distance-vector routing: Set up next-hops to destinations looking at neighbors’ routing tables (RIP)
  • Link-state routing: Get map of network in terms of link states and calculate best route (but specify only the next-hop) (OSPF)
Distance vector routing

• “Vector” of distances to each possible destination at each router

• How to find distances?
  • Distance to local network is 0
  • Look in neighbors’ distance vectors, and add link cost to reach the neighbor
  • Find minimum distance to destination
DV: Bellman-Ford algorithm

Iteration at each node:

1. *wait* for (change in local link cost or message from neighbor)

2. *recompute* distance table

3. if least cost path to any destination has changed, *notify* neighbors
DV Example: RIP

Final Table

via

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

via

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DV: Another example
DV problem: Count to infinity

A
1
B
1
C

INITIAL
A  2  B
B  1  C

① BC GOES DOWN

② EXCHANGE

③ EXCHANGE

④ EXCHANGE

⑤ STABLE
Count to Infinity problem

- Configuration A->B->C.
- If C fails, B needs to update and thinks there is a route through A
- A needs to update and thinks there is a route thru B

- No clear solution, except to set “infinity” to be small (eg 16 in RIP)

- Split-horizon: If A’s route to C is thru B, then A advertises C’s route (only to B) as infinity
Dealing with the problem

• Path vector
  • DV carries path to reach each destination

• Split horizon
  • never tell neighbor cost to X if neighbor is next hop to X

• Triggered updates
  • exchange routes on change, instead of on timer; faster count up to infinity
RIP information

• Distance vector
  • Cost metric is hop count; Infinity = 16
• RIPv1 defined in RFC 1058
  • uses UDP at port 520
• trigger for sending of distance vectors
  • 30-second intervals; 180s Refresh Timeout
  • routing table update; split horizon
  • format can carry upto 25 routes (512 bytes)
• RIPv2 defined in RFC 1388
  • uses IP multicasting (224.0.0.9); VLSM etc
Hierarchical routing

- Technique used to build large networks

- Minimizes use of network resources
  - router memory
  - router computing resources
  - link bandwidth

- Flat routing: linear increase in routing table size
- Hierarchical: logarithmic increase in routing table size
Hierarchical routing: example
Hierarchical routing example

• Consider a router in 10.1.1
  • assume 16 entries in each level (16 routes within 10.1.1)
  • with flat routing, 9*16 = 144 entries/router

• With 3 level hierarchy, a router in 10.1.1
  • has 16 entries for the routes in 10.1.1.*
  • One entry each for 10.1.2.*, 10.1.3.* and 10.1.*.* (default)
  • for a total of 19 entries.

• Marked reduction in routing table size
Routing issues

• Simplicity and Performance:
  • Size of the routing table should be kept small
  • Minimize number of control messages exchanged

• Correctness and Robustness:
  • Packet should be eventually delivered to the destination
  • Must be flexible to cope with changes in the topology and failures
  • No formation of routing loops or frequent toggling of routes

• Fairness and Optimality:
  • Every node should have a fair chance while transmitting packets
  • Balancing between minimizing mean packet delay v/s maximizing total network throughput