**Interior Gateway Protocols: RIP & OSPF**

Shivkumar Kalyanaraman  
Rensselaer Polytechnic Institute  
shivkuma@ecse.rpi.edu

Based in part upon slides of Prof. Raj Jain (OSU), S. Keshav (Cornell), J. Kurose (U Mass)

---

**Routing Tables & static routing**

**Dynamic routing (inter- and intra-domain)**

**Distance vector vs Link state routing**

**RIP, RIPv2**

**OSPF**

**Refs: Chap 9, 10.**

**Books: “Routing in Internet” by Huitema,**

"Interconnections" by Perlman

---

**Routing vs. Forwarding**

- **Forwarding:** select an output port based on destination address and routing table
- **Routing:** process by which routing table is built
- **Function of finding paths in a network.**
- **Can display routing table using “netstat -rn”**

---

**Routing Table Structure**

- **Fields:** destination, gateway, flags, ...
- **Destination:** can be a host address or a network address. If the ‘H’ flag is set, it is the host address.
- **Gateway:** router/next hop IP address. The ‘G’ flag says whether the destination is directly or indirectly connected.
- **U flag:** Is route up?
- **G flag:** router (indirect vs direct)
- **H flag:** host (dest field: host or n/w address?)

---

**Route Table Setup**

- Route Table setup by:  
  a) ‘route’ command [Static]  
  b) ICMP redirect message [Static]  
  c) routing daemon. Eg: ‘routed’ [Dynamic]

---

**Static Routing**

- Upon booting, default routes initialized from files.  
  Eg: /etc/rc.net in AIX, /etc/netstart in BSD,  
  /etc/rc.local in SUN/Solaris  
- Use ‘route’ command to add new routes eg:  
  route add default sun 1
Static Routing (Continued)

- ICMP redirect: sent to host by router when a “better” router exists on the same subnet.
- Alt: router discovery ICMP messages
  - Router solicitation request from host
  - Router advertisement messages from routers

Both ICMP methods are a form of “configuration” of static routes.

Dynamic Routing Model

- A node makes a local choice depending on global topology: this is the fundamental problem

Key problem

- How to make correct local decisions?
- Each router must know something about global state
- Global state
  - Inherently large
  - Dynamic
  - Hard to collect
- A routing protocol must intelligently summarize relevant information
- Hard issues: consistency, completeness, scalability

Requirements

- Minimize routing table space
  - Fast to look up
  - Less to exchange
- Minimize number & frequency of control pkts
- Robustness: avoid
  - Black holes
  - Loops
  - Oscillations
- Use optimal path

Choices ...

- Centralized vs. distributed routing
  - Centralized is simpler, but prone to failure and congestion
- Source-based vs. hop-by-hop
  - How much is in packet header?
  - Intermediate: loose source route
- Single vs. multiple path
  - Primary and alternative paths
- State-dependent vs. state-independent
  - Do routes depend on current network state (e.g. delay)

Detour: Telephony routing

- 3-level hierarchy, with a fully-connected core
- AT&T: 135 core switches with nearly 5 million circuits
- LECs may connect to multiple cores
Telephony Routing algorithm

- If endpoints are within same CO, directly connect
- If call is between COs in same LEC, use one-hop path between COs
- Otherwise send call to one of the cores
- Only major decision is at toll switch
  - one-hop or two-hop path to the destination toll switch. (why don’t we need longer paths?)
- Essence of problem:
  - which two-hop path to use if one-hop path is full

Features of telephone routing

- Stable load
  - can predict pairwise load throughout the day
  - can choose optimal routes in advance
- Extremely reliable switches
  - downtime is less than a few minutes per year
  - can assume that a chosen route is available
  - can’t do this in the Internet
- Single organization controls entire core
  - can collect global statistics and implement global changes
- Very highly connected network
  - Connections require resources (but all need the same)

Internet Dynamic Routing

- Internet organized as “autonomous systems” (AS).
- Interior Gateway Protocols (IGPs) within AS.
  - Eg: RIP, OSPF, HELLO
- Exterior Gateway Protocols (EGPs) for AS to AS routing.
  - Eg: EGP, BGP-4

Dynamic Routing Methods

- Source-based: chart route at src, given a map.
- Link state routing: Get map of network (in terms of link states) and calculate best route locally
- Distance vector: At every node, set up distance signposts to destination nodes (a vector)
  - Setup this by peeking at neighbors’ signposts.
**Distance Vector routing**

- “Vector” of distances (signposts) 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 which direction yields minimum distance to particular destination. Turn signpost that way.
  - Keep checking if neighbors change their signposts and modify local vector if necessary.
  - Called the “Bellman-Ford algorithm”
- Idea: At node i, for destination j:
  - Find neighbor/next-hop x that minimizes the sum
    \[ D(i,j) = C(i,x) + D(x,j) \]

**Distance Vector Routing Algorithm**

- Iterative:
  - continues until no nodes exchange info.
- Asynchronous:
  - each node has its own row for each possible destination
  - column for each directly-attached neighbor to node
  - example: in node X, for dest. Y via neighbor Z.
  - idea: At node i, for destination j:
    - Find neighbor/next-hop x that minimizes the sum
      \[ D(i,j) = C(i,x) + D(x,j) \]

**Distance Table: example**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>D(C,D)</td>
<td>2+2 = 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D(A,D)</td>
<td>2+3 = 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D(A,B)</td>
<td>8+6 = 14</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Distance Vector Algorithm**

1. Initialization:
2. for all adjacent nodes x:
3. \[ D^0(x,v) = \infty \] /* the “ operator means “for all rows” */
4. \[ D^0(v,v) = c(x,v) \]
5. for all destinations, y
6. send min \[ D^0(y,w) \] to each neighbor */ w over all X's neighbors */
Distance Vector Algorithm (cont.):

8 loop
9 wait (until I see a link cost change to neighbor V
10 or until I receive update from neighbor V)
11
12 if (c(X,V) changes by d)
13 /* change cost to all dest's via neighbor v by d */
14 for all destinations y: D(y,V) = D(y,V) + d
15
16 else if (update received from V wrt destination Y)
17 /* shortest path from V to some Y has changed */
18 /* V has sent a new value for its min. DW(Y,W) */
19 /* call this received new value is "newval" */
20 for the single destination y: D(Y,V) = c(X,V) + newval
21 if we have a new min. D(Y,w) for any destination Y
22 send new value of min. D(Y,w) to all neighbors
23 forever

Distance Vector Algorithm: example

Link cost changes:
- If Z routes through Y to get to X:
  - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)

Distance Vector: link cost changes

Link cost changes:
- good news travels fast
- bad news travels slow - "count to infinity" problem!
RIP: Routing Information Protocol
- Uses hop count as metric (max: 16 is infinity)
- Tables (vectors) “advertised” to neighbors every 30 s.
  - Each advertisement: upto 25 entries
- No advertisement for 180 sec: neighbor/link declared dead
- Routes via neighbor invalidated
- Neighbors in turn send out new advertisements (if tables changed)
- Link failure info quickly propagates to entire net
- Poison reverse used to prevent ping-pong loops (distance = 16 hops)

RIPv1 Problems (Continued)
- Split horizon/poison reverse does not guarantee to solve count-to-infinity problem
- 16 = infinity => RIP for small networks only!
- Slow convergence
- Broadcasts consume non-router resources
- RIPv1 does not support subnet masks (VLSMs)
- No authentication

RIPv2
- Why? Installed base of RIP routers
- Provides:
  - VLSM support
  - Authentication
  - Multicasting
  - “Wire-sharing” by multiple routing domains,
  - Tags to support EGP/BGP routes.
- Uses reserved fields in RIPv1 header.
- First route entry replaced by authentication info.

Link State Protocols
- Key: Create a network “map” at each node.
- 1. Node collects the state of its connected links and forms a “Link State Packet” (LSP)
- 2. Flood LSP => reaches every other node in the network and everyone now has a network map.
- 3. Given map, run Dijkstra’s shortest path algorithm (SPF) => get paths to all destinations
- 4. Routing table = next-hops of these paths.

Dijkstra’s algorithm
- Net topology, link costs known to all nodes
  - accomplished via “link state broadcast”
- All nodes have same info
- Computes least-cost paths from one node (“source”) to all other nodes
- Gives routing table for that node
- Iterative: after k iterations, know least cost path to k dest.’s

Dijkstra’s Algorithm
1. Initialization:
   2. $N = \{A\}$
   3. for all nodes $v$
   4. if $v$ adjacent to $A$
   5. then $D(v) = c(A,v)$
   6. else $D(v) = \infty$
7. 
8. Loop
9. find $w$ not in $N$ such that $D(w)$ is a minimum
10. add $w$ to $N$
11. Update $D(v)$ for all $v$ adjacent to $w$ and not in $N$:
12. $D(v) = \min(D(v), D(w) + c(w,v))$
13. new cost to $v$ is either old cost to $v$ or known
14. shortest path cost to $w$ plus cost from $w$ to $v^*$
15. until all nodes in $N$
Dijkstra's algorithm: example

<table>
<thead>
<tr>
<th>Step</th>
<th>start N</th>
<th>D(B,p(B))</th>
<th>D(C,p(C))</th>
<th>D(D,p(D))</th>
<th>D(E,p(E))</th>
<th>D(F,p(F))</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ADEB</td>
<td>3.E</td>
<td>4.E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ADEBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dijkstra's algorithm, discussion

Algorithm complexity: n nodes
- each iteration: need to check all nodes, w, not in N
- n(n+1)/2 comparisons: O(n^2)
- more efficient implementations possible: O(nlogn)

Oscillations possible:
- e.g., link cost = amount of carried traffic

Link State: Topology Dissemination
- A.k.a. LSP distribution
- 1. Flood LSPs on links except incoming link
  - Require at most 2E transfers for n/w with E edges
- 2. Sequence numbers to detect duplicates
  - Why? Routers/links may go down/up
  - Problem: wrap-around => have large seq # space, lollipop sequence

Lollipop Sequence

- Need a unique start sequence number
- Comparison: a is older than b if:
  - a < 0 and a < b
  - a > 0, a < b, and b - a < N/4
  - a > 0, b > 0, a > b, and a - b > N/4

Topology Dissemination (Continued)
- 3. Age field (similar to TTL)
  - Periodically decremented after acceptance
  - Zero => discard LSP & request everyone to do so
  - Router awakens => knows that all its old LSPs would have been purged and can choose a new initial sequence number

Recovering from a partition
- On partition, LSP databases can get out of synch
- Databases described by database descriptor records
- Routers on each side of a newly restored link talk to each other to update databases (determine missing and out-of-date LSPs) => selective synchronization
OSPF “advanced” features (not in RIP)

- Security: all OSPF messages authenticated (to prevent malicious intrusion); TCP connections used
- Multiple same-cost paths allowed (only one path in RIP)
- For each link, multiple cost metrics for different TOS (e.g., satellite link cost set “low” for best effort; high for real time)
- Integrated uni- and multicast support:
  - Multicast OSPF (MOSPF) uses same topology database as OSPF
  - Hierarchical OSPF in large domains.

Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
- Link-state advertisements only in area
- Each node has detailed area topology; only know direction (shortest path) to nets in other areas.
- Area border routers: “summarize” distances to nets in own area, advertise to other Area Border routers.
- Backbone routers: run OSPF routing limited to backbone.
- Boundary routers: connect to other ASs (generate “external” records)

Example

External and summary records

- If a domain has multiple gateways
  - External records tell hosts in a domain which one to pick to reach a host in an external domain
    - A.E.G allows 6.4.0.0 to discover shortest path to 5.* is through 6.0.0.0
  - Summary records tell backbone (and areas) which gateway to use to reach an internal node
    - A.E.G allows 5.0.0.0 to discover shortest path to 6.4.0.0 is through 6.0.0.0

E-IGRP (Interior Gateway Routing Protocol)

- CISCO proprietary; successor of RIP (late 80s)
- Several metrics (delay, bandwidth, reliability, load etc)
- Uses TCP to exchange routing updates
- Loop-free routing via Distributed Updating Alg. (DUAL) based on diffused computation
  - Freeze entry to particular destination
  - Diffuse a request for updates
  - Other nodes may freeze/propagate the diffusing computation (tree formation)
  - Unfreeze when updates received.
  - Tradeoff: temporary un-reachability for some destinations
Link State vs. Distance Vector

- Link State (LS) advantages:
  - More stable (aka fewer routing loops)
  - Faster convergence than distance vector
  - Easier to discover network topology, troubleshoot network.
  - Can do better source-routing with link-state
  - Type & Quality-of-service routing (multiple route tables) possible

- Caveat: With path-vector-type (paths instead of distances) DV routing, these differences blur…

Summary

- Basic routing concepts, Route Tables
- Distance vector, Bellman-Ford, RIP, RIPv2
- Link state, Dijkstra, OSPF
- DUAL/EIGRP etc