RIP on Cisco Hardware

- DV Routing on Cisco Hardware
  - Kerry Wood (woodk@alum.rpi.edu)
  - Yong Xia (TA)
  - Shiv Kalyanaraman
  - shivkuma@ecse.rpi.edu
  - http://www.ecse.rpi.edu/Homepages/shivkuma
RIPv1 Overview

- Originally designed for Xerox PARC Universal Protocol, when it was called GWINFO.
- BSD began shipping with RIP installed as *routed* in 1982, leading to its wide acceptance.
- Mutated and modified by many vendors for their own use (i.e. Apples RTMP is an extension of RIP).
- Documented in XNS Internet Transport Protocols publication (1981) and in RFC 1058.
RIP: Key Facts

- Distance Vector protocol using Bellman-Ford algorithm.
- Hop count is only metric for path determination.
- Maximum hop count is 15, 16 is unreachable.
- Routing updates are BROADCAST every 30 seconds by default.
- RIP is capable of load balancing over multiple paths.
- RIP supports both TCP/IP routing and IPX.
- RIP is classful; it does not keep track of subnet masks.

Who cares if it supports IPX?

Many of the RIP implementations that are left are because it routes IPX! The only truly viable alternative for an organization is to go to EIGRP, which is Cisco proprietary.
RIP: Tables

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination</td>
<td>32 bit destination address</td>
</tr>
<tr>
<td>Metric</td>
<td>Hop count to dest.</td>
</tr>
<tr>
<td>Next-hop</td>
<td>Next router along path.</td>
</tr>
<tr>
<td>Updated flag</td>
<td>Flag indicating if this route was recently updated.</td>
</tr>
<tr>
<td>Timers</td>
<td>Timers associated with entry.</td>
</tr>
</tbody>
</table>

The routing table contains nothing more than a list of destinations, associated metrics, and maintenance information.
RIP: Timers (defaults)

- **Update interval**: 30 sec. Every half minute, each router broadcasts its routing table in entirety to its neighbors.
- **Invalid timer**: 180 sec. If no update about the route is received in this time, it is marked as invalid and advertised unreachable. However, it continues to forward packets until flush time expires.
- **Holddown**: 180 sec. When a router receives a packet declaring a route unreachable, this timer is started. No routes with worse metrics are accepted until it expires.
- **Flush timer**: 240 sec. How long after the route is declared down, before it is completely removed. Starts incrementing immediately after last update is received.
RIP packets are encapsulated in UDP packets (max 512 bytes). A maximum of 25 routes can be held in any single update packet.
RIP: Traffic Pt2

- So, assuming 100 routes in every router..
- At 408 bytes per 25 routes updated..
- Broadcast twice per minute..

We end up with roughly 1.632Kbytes or 13,506Kbits per minute.

- On a point to point link, there are two routers issuing these updates.. So the traffic doubles.
- On a broadcast medium like standard Ethernet, we have N-times as much traffic!

*Obviously, this doesn’t scale particularly well for large networks!*
RIP: Subnetting

- RIP is capable of only rudimentary subnetting.
- V2 has VLSM capability, but V2 is not widely deployed.
- In V1, routers initially look for the class barriers, A, B, and C.

However, it is possible to have a global subnet mask. That is, all networks use the same mask. Since the routing updates contain no subnet information, this must be configured on each individual router. Think that causes problems?
RIP: Summary of Issues

- RIP v1 constantly sends out updates, necessary or not.
- RIP v1’s convergence time is dependent upon the setup of the timers. Bad routes can take MINUTES to disappear.
- RIP v1 has no VLSM support. You either have to use lots of classful networks, or use a universal subnet mask.
- RIP v1 has no authentication method. Plug a box into a RIP v1 network advertising a destination with metric 0, and everyone will use it! (Or at least try to).
Cisco Implementation

The first step in a setup is addressing the links. Technically, you can do this after you enable RIP, but it will make more sense to do it now..

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int s0
Router(config-if)#ip address 172.16.0.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#int sl
Router(config-if)#ip address 172.16.1.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#
```
Enabling the Router

Now, we can enable the RIP process on the router, this is done by issuing the “router rip” command from (config) mode.

```
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#
```

Router configuration commands:
- auto-summary: Enable automatic network number summarization
- default: Set a command to its defaults
- default-information: Control distribution of default information
- default-metric: Set metric of redistributed routes
- distance: Define an administrative distance
- distribute-list: Filter networks in routing updates
- exit: Exit from routing protocol configuration mode
- flash-update-threshold: Specify flash update threshold in second
- help: Description of the interactive help system
- input-queue: Specify input queue depth
- maximum-paths: Forward packets over multiple paths
- neighbor: Specify a neighbor router
- network: Enable routing on an IP network
- no: Negate a command or set its defaults
- offset-list: Add or subtract offset from IGRP or RIP metrics
- output-delay: Interpacket delay for RIP updates
- passive-interface: Suppress routing updates on an interface
- redistribute: Redistribute information from another routing protocol
- timers: Adjust routing timers
- traffic-share: Algorithm for computing traffic share for alternate
Specifying Network

Once the router is enabled, we specify which network it should be routing for with a “network <ip_address>” command.

If we had debugging enabled, we would instantly see the messages corresponding to the startup of the daemon.
Viewing Setup

To verify that the protocol is up and ready to go, we can issue a “show ip protocols” command. This would generally show details of all running IP routing protocols, since we only have RIP, it is the only one shown.

```
Router#show ip proto
Routing Protocol is "rip"
    Sending updates every 30 seconds, next due in 3 seconds
    Invalid after 180 seconds, hold down 180, flushed after 240
    Outgoing update filter list for all interfaces is
    Incoming update filter list for all interfaces is
    Redistributing: rip
    Default version control: send version 1, receive any version
    Interface        Send  Recv  Key-chain
    Serial0           1     1  2
    Serial1           1     1  2
    Routing for Networks:
    172.16.0.0
    Routing Information Sources:
    Gateway         Distance  Last Update
    Distance: (default is 120)
```
Fun RIP Commands 1

auto-summary (router)
- RIP can “fake” some hierarchy, by summarizing on class boundaries. You can turn this on, the question is “Where would you?”.

default-information originate (router)
- A router can generate a default route and inject it in the network. If no other routes qualify, this one is used. Why would you use this command?

ip rip receive/send version (interface command)
- If you’re exclusively using V2 in a network, you wouldn’t want to accept insecure V1 packets.

ip split-horizon (interface command)
- Split horizon can be enabled or disabled on an interface. Knowing that frame-relay circuits can have multiple PVC’s on one physical interface, why is this command useful?
Fun RIP Commands 2

timers basic *update invalid holddown flush* (router)

- Looking at those timers, you’d think there would be a way to play with them. This is it. IOS allows you to modify all the values associated with RIP, generally with NO checks. You could set the invalid timer to 2 seconds and the update interval to 10 seconds. Every 10 seconds you would have a usable route for 2 ticks. (Or would you? How does holddown and flush affect this?)

```
version {1|2} (router)
```

- Set the overall routing process version. This obviously overrides all the other version setup commands.
Here's a look at the lab setup for the OSPF lab. You may want to pick a new address range to play with, or try to get these to work. But log into a router, work together, and see if you can get the network routing packets.
Telnet and Teams

We don’t have as many routers to use as last time due to connectivity problems. Grab one from the following list...

Available routers are:

7000 Series:
- 7000_1, telnet port: 2129
- 7000_2, telnet port: 2132
- 7000_3, telnet port: 2131

2500 Series:
- 2524_1, telnet port: 2127
- 2500_1, telnet port: 2123
- 2500_2, telnet port: 2124
- 2500_3, telnet port: 2125
- 2500_4, telnet port: 2126

To connect, telnet to:

litec-wti.ecse.rpi.edu <PORT_NUM>