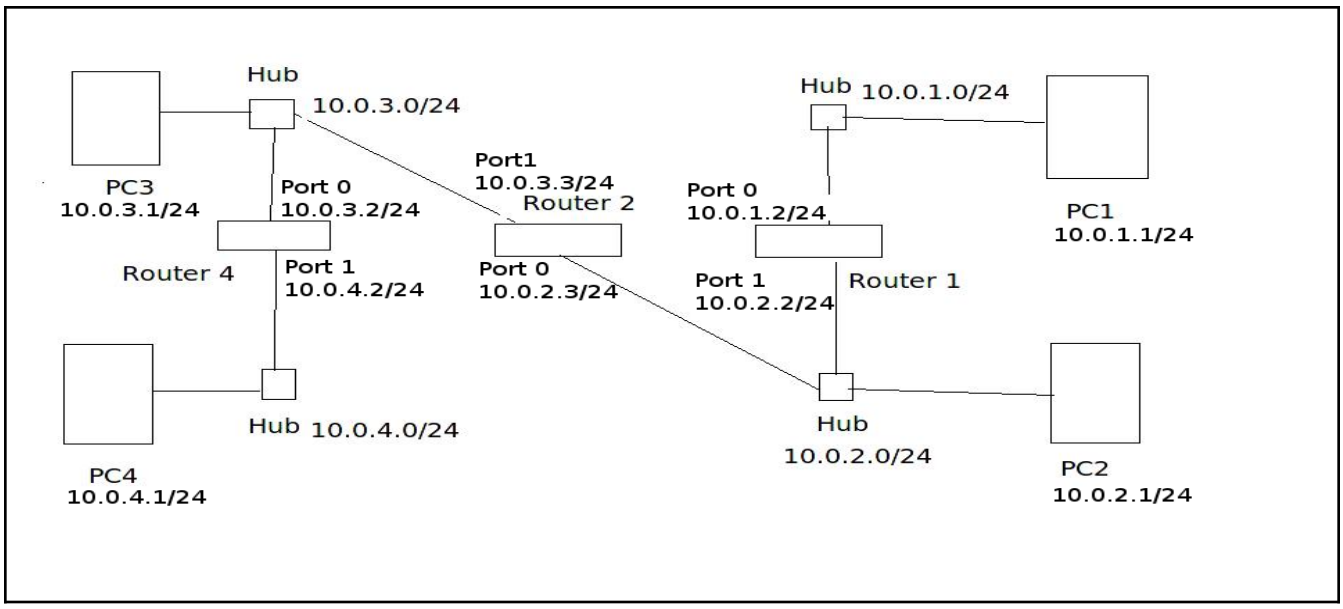
Andrew Zurn
CSCI 312 - Schnepf
Lab 4 Report
5/1/13

Introduction:

The main objective in this lab is to use the RIP dynamic routing protocol that was discussed in class on a small internetwork of machines and routers, and also to experience what would happen in the event of a link failure, and how this protocol would adapt in the case that some redundant links were present. We first set up our network as such displayed in the image below. The IP addresses used for each machine are also present in the image.



The machines were configured using the */sbin/ip addr add <address> dev em1*, and then given a default gateway to the router they are connected to using the ip route add default via <router_address> command. These routes are set up as follows: PC1 to Router1:Port0, PC2 to Router2:Port0, PC3 to Router2:Port1, PC4 to Router4:Port1.

The addresses of the routers and their ports(interfaces) were configured using the *ip address IP_Address Subnet_Mask* command in the global configuration terminal of the Cisco routers that were used. Once this was done, we are able to continue and begin working with the dynamic RIP protocol.

Using RIP and Determining Effectiveness

Now, as stated above, the routers need to be configured to using the RIP protocol to begin broadcasting to each other, and learning of each other's addresses on the network. This is done by entering into enable mode, and going into the global configuration terminal. Then a series of commands are given to the router to begin the routing protocol. Those are as follows:

- Router(config)# no ip routing
- Router(config)# ip routing (This is done to reset our routing tables to make sure nothing out of the ordinary happens)

- Router(config)# router rip (This brings us to the router configuration terminal, and starts the RIP protocol
- Router(config-router)# version 2 (tells the router to use RIPv2)
- Router(config-router)# network 10.0.0.0 (tells RIP to send updates to those on the 10.x.x.x network)

Finally, we return to the enable mode terminal, and clear our IP routes to again start with fresh tables to be sure that nothing abnormal happens (this is done using the *clear ip route* * command).

The routing tables established by the RIP protocol for routers 2 and 4 are displayed below.

router2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route **(Codes Table OMITTED here after)**

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 4 subnets
C 10.0.2.0 is directly connected, FastEthernet0/0
C 10.0.3.0 is directly connected, FastEthernet0/1
R 10.0.1.0 [120/1] via 10.0.2.2, 00:00:07, FastEthernet0/0
R 10.0.4.0 [120/1] via 10.0.3.2, 00:00:00, FastEthernet0/1

Router4#show ip route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 4 subnets
R 10.0.2.0 [120/1] via 10.0.3.3, 00:00:14, FastEthernet0/0
C 10.0.3.0 is directly connected, FastEthernet0/0
R 10.0.1.0 [120/2] via 10.0.3.3, 00:00:14, FastEthernet0/0
C 10.0.4.0 is directly connected, FastEthernet0/1

As can be seen, the RIP protocol used did indeed turn up to find every network within our internetwork, and properly configured our routing tables to hold the paths from each network to every other network.

Looking at a quick breakdown of the router 2‘s table, we see two connected networks and two networks found outside of a direct connection. Router 2‘s directly connected networks are 10.0.2.0 (on Port 0) and 10.0.3.0 (on Port 1). Router 2 also has connections to 10.0.1.0 (through 10.0.2.2, which is Port 1 or Router 1), and also to 10.0.4.0 (through 10.0.3.2, which is Port 0 of Router 4) Router 4 has a similar breakdown, where it has connections to the 10.0.3.0 network

and the 10.0.4.0 network, and also connections to the 10.0.2.0 network through 10.0.3.3 (Port 1 of Router 2), and also to 10.0.1.0 (again through Port 1 of Router 3).

Now, the linux command *traceroute ip_address* is used from PC1 to PC4, and its output is displayed below.

```
[netlab@PC1 ~]$ traceroute 10.0.4.1
traceroute to 10.0.4.1 (10.0.4.1), 30 hops max, 60 byte packets
 1  10.0.1.2 (10.0.1.2)  2.553 ms  3.391 ms  5.278 ms
 2  10.0.2.3 (10.0.2.3)  3.372 ms  4.675 ms  5.249 ms
 3  10.0.3.2 (10.0.3.2)  4.393 ms  4.967 ms  5.549 ms
 4  10.0.4.1 (10.0.4.1)  3.053 ms  3.049 ms  3.532 ms
```

The output of our traceroute to PC4 shows the routers between the source and destination, and also displayed their round trip time (RTT) for the three packets that are sent to and from each router. As can be seen, the packets travel from PC1 to 10.0.1.2 (Port0 of Router1), to 10.0.2.3 (Port0 of Router2), to 10.0.3.2 (Port0 of Router4), and finally to its destination of 10.0.4.1.

Overall, our RIP protocol configured our routers successfully, and do use and display the proper routes based off of our diagram.

Finally, Wireshark is used to dive into some of the more specific inner-workings of the RIP protocol and the packet information that it uses. A set of questions is answered, that highlights these specifics.

1. What is the destination IP address of RIP packets?
    1.1. The destination address in every observed packet from every router is 224.0.0.9. Below is one packet from both PC1 and PC3‘s wireshark capture showing this.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.3.2	224.0.0.9	RIPv2	66	Response

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.1.2	224.0.0.9	RIPv2	66	Response

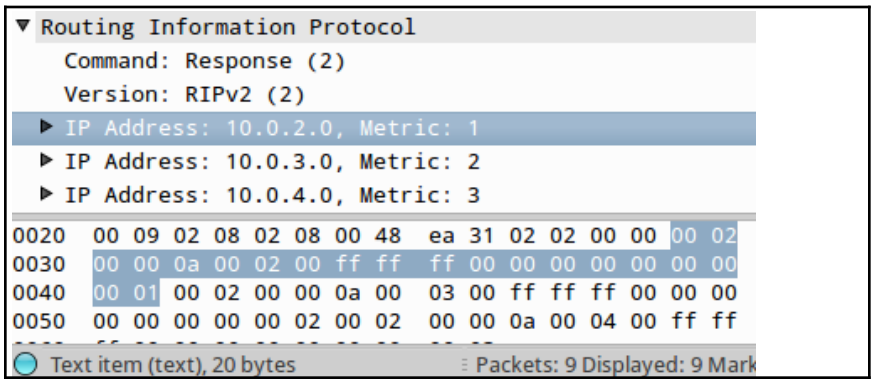
    This is actually the special multicast address that routers use to broadcast these RIP packets to other routers. (RFC 1723)
2. Do routers forward RIP packets?
    2.1. Routers will not forward RIP packets, but they will update their own tables with any new information that they might receive from update packets sent to them, and will send out a triggered update in response to new changes in the connections in the network. This is evident because no RIP response packets from any other machines outside of the one being monitored are seen (if it would forward these packets, the routing tables would become corrupted as they would receive packets from across the network with a weight between nodes as 1, and thus update their tables with the invalid information). A few packets from a capture on PC3 highlights this.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.3.2	224.0.0.9	RIPv2	66	Response

No.	Time	Source	Destination	Protocol	Length	Info
2	15.362265	10.0.3.3	224.0.0.9	RIPv2	86	Response

No.	Time	Source	Destination	Protocol	Length	Info
3	28.529289	10.0.3.2	224.0.0.9	RIPv2	66	Response
No.	Time	Source	Destination	Protocol	Length	Info
4	42.068024	10.0.3.3	224.0.0.9	RIPv2	86	Response
No.	Time	Source	Destination	Protocol	Length	Info
5	56.426784	10.0.3.2	224.0.0.9	RIPv2	66	Response

3. What types of routing RIP messages do you observe? The type is indicated by the value in the field *command.* What role does each message type play?
    3.1. In the captures, only response packets are being picked up by Wireshark (as seen above). A filter on port 520 was used to capture all RIP packets. As per RFC 2453, all RIP packets are sent and received on port 520, and thus, this should have picked up both response and request packets. The request packet asks for a system to send its routing table information. The response packet may be a response to a request, or just sent as an timed update, and contains the routing table information for that sender. (RFC 2453)
4. Describe the information that you find in a RIP message. How many bytes does each routing table entry take?
    4.1. In the message of the RIP packet, we will see the *command* type (in our case, all responses), the *version* of RIP being used (RIPv2), and then the routing table information for that router (IP Address followed by a weighting metric). Below is an image of the information in the table. As shown in the image below, each table entry takes 20 bytes of information.



Convergences and Failures

In the final part of the lab, the internetwork is slightly rearranged and in the final step of the lab, a network outage is produced, and a work-around is established by our routing protocol. To start off, we bring down port0 or Router1, and connect it to the hub that connects Router4 and PC4, and add it to the 10.0.4.0/24 network (port0 becomes 10.0.4.3/24 and also PC4's gateway address is changed to Router1 rather than Router4). The new routing tables after this change in the internetwork is made, and is displayed for both Router2 and Router4 below.

router2#show ip route

Gateway of last resort is not set

```
    10.0.0.0/24 is subnetted, 3 subnets
C       10.0.2.0 is directly connected, FastEthernet0/0
C       10.0.3.0 is directly connected, FastEthernet0/1
R       10.0.4.0 [120/1] via 10.0.3.2, 00:00:28, FastEthernet0/1
                 [120/1] via 10.0.2.2, 00:00:15, FastEthernet0/0
```

Router4#show ip route

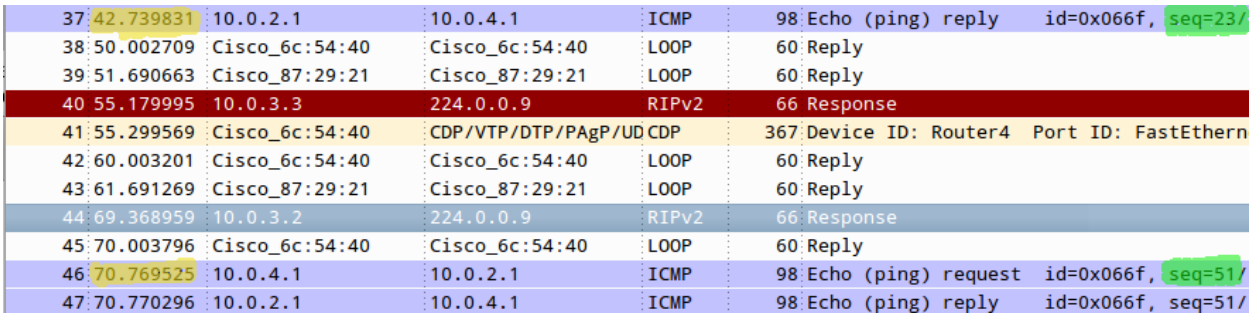
Gateway of last resort is not set

```
    10.0.0.0/24 is subnetted, 3 subnets
R       10.0.2.0 [120/1] via 10.0.3.3, 00:00:06, FastEthernet0/0
                 [120/1] via 10.0.4.3, 00:00:21, FastEthernet0/1
C       10.0.3.0 is directly connected, FastEthernet0/0
C       10.0.4.0 is directly connected, FastEthernet0/1
```

What is seen here is that we have the same directly connected networks as earlier, but now our network found by RIP has two routes to reach its final destination. Router2 can now reach the 10.0.4.0 network either via 10.0.3.2 (port0 of Router2 and sends through the FastEthernet0/1 interface of that router), or through 10.0.2.2 (port1 of Router 1 and sends through the FastEthernet 0/0 interface of that router), to reach it's final destination. Router4 can now reach the 10.0.2.0 in the same fashion using the links found above (which are the just the opposite directions of the routes that Router2 uses to reach Router4).

Finally, a link outage is produced to show how RIP will update the routing tables and a new path to an end destination is found. A ping command from PC4 to PC2 is issued, whereupon the link from port0 of Router1 to the 10.0.4.0 network is disconnected, and our machine's pings no longer reach PC2.

RIP, after sometime when its tables are updated, sees this outage and bridges that connection by telling PC4 to send its pings via 10.0.3.2, that is, through Router2. The time that this takes is about 28 seconds before our networks update, and the new path is used. The evidence of this is displayed below, where it is seen that no pings are picked up by PC3 between time 42.7 and 70.7 (the times are highlighted on the left). Additionally, the seq number of the last seen ICMP packet before the failure is fixed is number 23, and then is picked up again at 51. This data is coherent with the data from the terminal output, which is also displayed below (and also highlighted on the right in the wireshark output).



37	42.739831	10.0.2.1	10.0.4.1	ICMP	98	Echo (ping) reply id=0x066f, seq=23/
38	50.002709	Cisco_6c:54:40	Cisco_6c:54:40	LOOP	60	Reply
39	51.690663	Cisco_87:29:21	Cisco_87:29:21	LOOP	60	Reply
40	55.179995	10.0.3.3	224.0.0.9	RIPv2	66	Response
41	55.299569	Cisco_6c:54:40	CDP/VTP/DTP/PAgP/UD	CDP	367	Device ID: Router4 Port ID: FastEthern
42	60.003201	Cisco_6c:54:40	Cisco_6c:54:40	LOOP	60	Reply
43	61.691269	Cisco_87:29:21	Cisco_87:29:21	LOOP	60	Reply
44	69.368959	10.0.3.2	224.0.0.9	RIPv2	66	Response
45	70.003796	Cisco_6c:54:40	Cisco_6c:54:40	LOOP	60	Reply
46	70.769525	10.0.4.1	10.0.2.1	ICMP	98	Echo (ping) request id=0x066f, seq=51/
47	70.770296	10.0.2.1	10.0.4.1	ICMP	98	Echo (ping) reply id=0x066f, seq=51/

64 bytes from 10.0.2.1: icmp_req=23 ttl=62 time=1.22 ms **(Last ping seen by PC3)**
From 10.0.4.3 icmp_seq=30 Destination Host Unreachable
From 10.0.4.3 icmp_seq=31 Destination Host Unreachable
From 10.0.4.3 icmp_seq=32 Destination Host Unreachable
From 10.0.4.3 icmp_seq=33 Destination Host Unreachable
From 10.0.4.3 icmp_seq=34 Destination Host Unreachable
From 10.0.4.3 icmp_seq=35 Destination Host Unreachable
From 10.0.4.3 icmp_seq=36 Destination Host Unreachable
From 10.0.4.3 icmp_seq=37 Destination Host Unreachable
From 10.0.4.3 icmp_seq=38 Destination Host Unreachable
From 10.0.4.3 icmp_seq=39 Destination Host Unreachable
From 10.0.4.3 icmp_seq=40 Destination Host Unreachable
From 10.0.4.3 icmp_seq=41 Destination Host Unreachable
From 10.0.4.3 icmp_seq=42 Destination Host Unreachable
From 10.0.4.3 icmp_seq=43 Destination Host Unreachable
From 10.0.4.3 icmp_seq=44 Destination Host Unreachable
From 10.0.4.3 icmp_seq=45 Destination Host Unreachable
From 10.0.4.3 icmp_seq=46 Destination Host Unreachable
From 10.0.4.3 icmp_seq=47 Destination Host Unreachable
From 10.0.4.3 icmp_seq=48 Destination Host Unreachable
From 10.0.4.3 icmp_seq=49 Destination Host Unreachable
From 10.0.4.3: icmp_seq=50 Redirect Host(New nexthop: 10.0.4.2)
From 10.0.4.3 icmp_seq=50 Redirect Host64 bytes from 10.0.2.1: icmp_req=51 ttl=62 time=3.46 ms **(First re-seen by PC3)**

Conclusion

The main focus of this lab was to examine the RIP protocol and the packets it uses, and watch the protocol operate in the event of a network failure. It was seen that the RIP protocol was able to establish routes between each network forming our internetwork, and that it was able to, after about 28 seconds, establish a new route when one of our other routes failed. This shows the dynamic nature of routing protocols, and what might actually happen to the routers and routes that establish the Internet when a failure occurs. This lab was very insightful as to how dynamic routing protocols need to be, how well they work, and was an excellent way to see firsthand some of the material that was discussed in class.