Lecturer: Richard Frisby

Title: Dynamic Routing Report

Name: Stephen Long

Student Number: 20035687

Introduction to Dynamic Routing

Dynamic routing protocols share information between routers in order to construct topologies of networks. Depending on the particular protocol, this is either done by sending updates at regular intervals (RIP) or when changes occur to the topology (EIGRP &OSPF). Routers using dynamic routing protocols use this shared information to construct their routing tables. They can then use the routing table to determine the best path to a desired destination. There are both advantages and disadvantages to using a dynamic routing protocol. If the topology in question was a large topology, it would be more favourable to use dynamic routing as opposed to static routing as not only would it be time consuming and confusing to configure static routing on a large topology, dynamic routing protocols react when there are changes to topologies, such as an interface going down, or a destination becoming unreachable. A disadvantage is that they are more difficult to configure than static routes. A router running a dynamic routing protocol will inform its neighbours of the topology change, via an update. Its neighbours will then inform its neighbours, and so on, until the topology has successfully converged.

Difference between static routing and dynamic routing

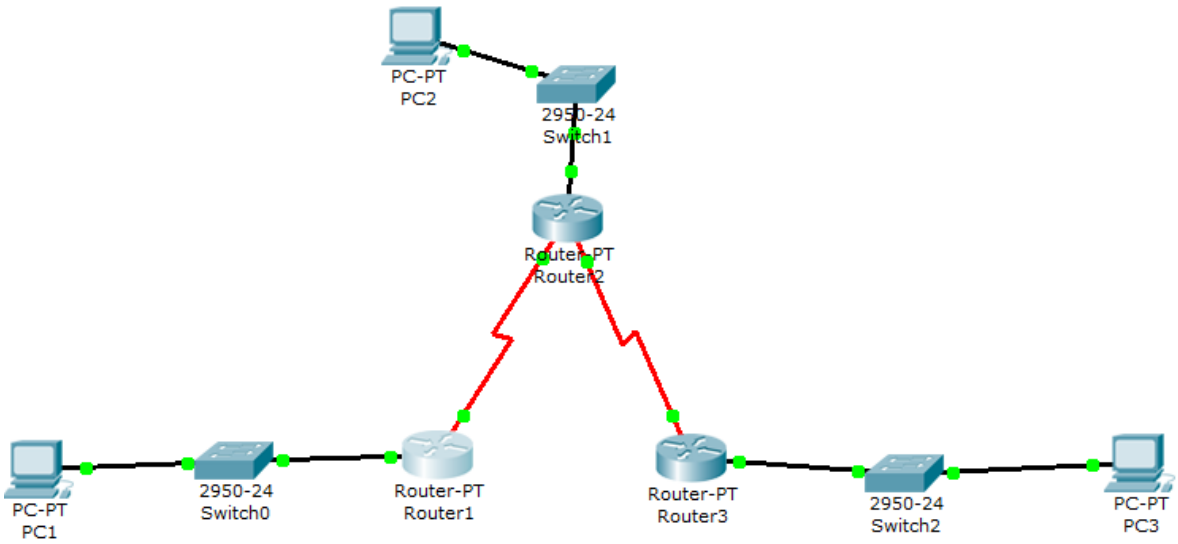
Static routes are easier to configure than dynamic routes however they do not have the ability to adapt and calculate alternate routes to a destination if a particular path goes down, or an interface becomes inactive.

As I will discuss, dynamic routing protocols use algorithms to determine the shortest path (based on metrics) to get a packet to a destination. They have the intelligence to adapt when a route goes down. This involves informing other routers of the status of interfaces and updating routing tables to store information regarding the topology.

RIP VERSION 1

RIP v1 is a distance vector protocol that works by sending broadcast messages to its adjacent neighbours every 30 seconds, informing them of its routing table. It has an Administrative Distance (AD) of 120. This means that it is considered to be less “trustworthy” than other protocols such as EIGRP(90), and OSPF(110). RIP uses a hop count metric, meaning that it determines the best path to its destination by measuring how many hops it would take to send a packet there. The maximum hop count for routers running RIP is 15, meaning that any network over 15 hops away, is considered to be unreachable and is therefore given a TTL value of 16. If a network is unavailable, such as an interface goes down, or it is for any reason unreachable, it is assigned a value of 16 which informs other routers it is unreachable.

Configuring RIPv1 on the topology shown



To enable RIP, enter *router rip* in global config mode and enter the network you wish to advertise.

```
R1(config)#router rip
R1(config-router)#network 192.168.1.0
R1(config-router)#network 192.168.2.0
R1(config-router)#end
```

```
R2(config)#router rip
R2(config-router)#network 192.168.2.0
R2(config-router)#network 192.168.3.0
R2(config-router)#network 192.168.4.0
R2(config-router)#end
R2#
```

```
R3(config)#router rip
R3(config-router)#network 192.168.4.0
R3(config-router)#network 192.168.5.0
R3(config-router)#end
```

To verify that the router s are in fact supporting rip, we can view the routing table using *show ip route*.

```
C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, Serial2/0
R    192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:19, Serial2/0
R    192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:19, Serial2/0
R    192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:19, Serial2/0
```

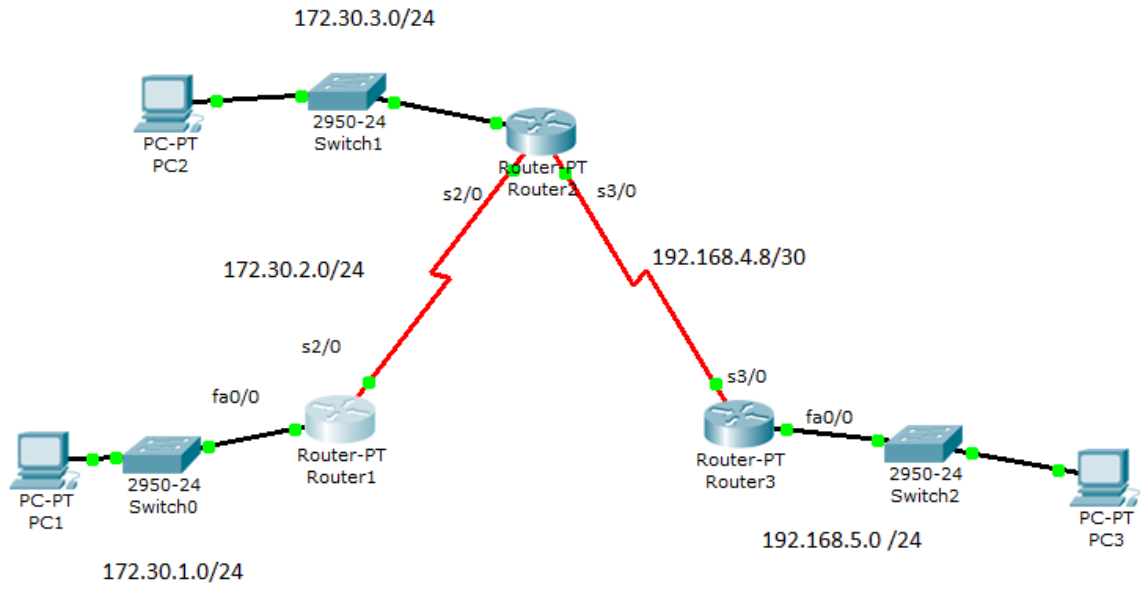
Examining the screenshot above we can see that R1 is aware off all other networks in the topology. The 192.168.3.0 network is accessible via the Serial2/0 port on R2, and is 1 hop away. If there was another path to the same destination, and the hop count was equal, the router would use load balancing, which is a method whereby the router sends packets to a destination via more than one

route. If I were to add another path to the same destination, via a different protocol, the router would check the admin distance value, and whichever is deemed the most "trustworthy"(had the lowest AD), which is the first number in the bracket(120 for RIP), would be the chosen protocol.

```
Routing for Networks:
        192.168.1.0
        192.168.2.0
Passive Interface(s):
```

We can also see which networks the router is advertising each time it sends its updates by looking at the networks under "Routing for Networks" as seen in the diagram above.

CONFIGURING RIPV1: Classless Networks and with subnets



After successfully creating the topology shown and applying the following addressing table, we can then configure RIP on the routers.

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	Fa0/0	172.30.1.1	255.255.255.0	N/A
	S0/0/0	172.30.2.1	255.255.255.0	N/A
R2	Fa0/0	172.30.3.1	255.255.255.0	N/A
	S0/0/0	172.30.2.2	255.255.255.0	N/A
	S0/0/1	192.168.4.9	255.255.255.252	N/A
R3	Fa0/0	192.168.5.1	255.255.255.0	N/A
	S0/0/0	192.168.4.10	255.255.255.252	N/A
PC1	NIC	172.30.1.10	255.255.255.0	172.30.1.1
PC2	NIC	172.30.3.10	255.255.255.0	172.30.3.1
PC3	NIC	192.168.5.10	255.255.255.0	192.168.5.1

```
R1(config)#router rip
R1(config-router)#network 172.30.0.0
R1(config-router)#passive-interface fa0/0
R1(config-router)#end
R1#
```

After configuring RIPv1 on an interface, we can then inform the router not to send updates out an interface by using the *passive-interface* command followed by the specified interface.

Why stop a router sending out updates?

Sending out updates can consume bandwidth unnecessarily, causing delays in the network. If RIP is configured on a stub network, it would make sense to use *passive interface* on an interface, as if the router only has one way in which to reach a particular network, there is no need to constantly send updates informing it of this when a static route could be applied. Also, RIPv1 sends packets out on a broadcast network, meaning that anyone connected to the network will receive the broadcast meaning packets can easily be intercepted.

After configuring RIP on the three routers we can check using *show ip route* to see if the tables consist of the correct information.

```
     172.30.0.0/24 is subnetted, 3 subnets
C       172.30.1.0 is directly connected, FastEthernet0/0
C       172.30.2.0 is directly connected, Serial2/0
R       172.30.3.0 [120/1] via 172.30.2.2, 00:00:23, Serial2/0
R    192.168.4.0/24 [120/1] via 172.30.2.2, 00:00:23, Serial2/0
R    192.168.5.0/24 [120/2] via 172.30.2.2, 00:00:23, Serial2/0
R1#
```

Here we can see that R1 is aware of all other networks in the topology. Taking for example, the last entry in the table, we can see that it is 2 hops away, and accessible via 172.30.2.2.

```
     172.30.0.0/24 is subnetted, 3 subnets
R       172.30.1.0 [120/1] via 172.30.2.1, 00:00:14, Serial2/0
C       172.30.2.0 is directly connected, Serial2/0
C       172.30.3.0 is directly connected, FastEthernet0/0
     192.168.4.0/30 is subnetted, 1 subnets
C       192.168.4.8 is directly connected, Serial3/0
R    192.168.5.0/24 [120/1] via 192.168.4.10, 00:00:20, Serial3/0
R2#
```

```
R    172.30.0.0/16 [120/1] via 192.168.4.9, 00:00:11, Serial3/0
     192.168.4.0/30 is subnetted, 1 subnets
C       192.168.4.8 is directly connected, Serial3/0
C    192.168.5.0/24 is directly connected, FastEthernet0/0
R3#
```

We can see that all the tables hold the correct information.

To verify that the *passive interface* command discontinued the sending of packets from the fa0/0 interface on R1, we can use *show ip protocols* to check the current passive interfaces

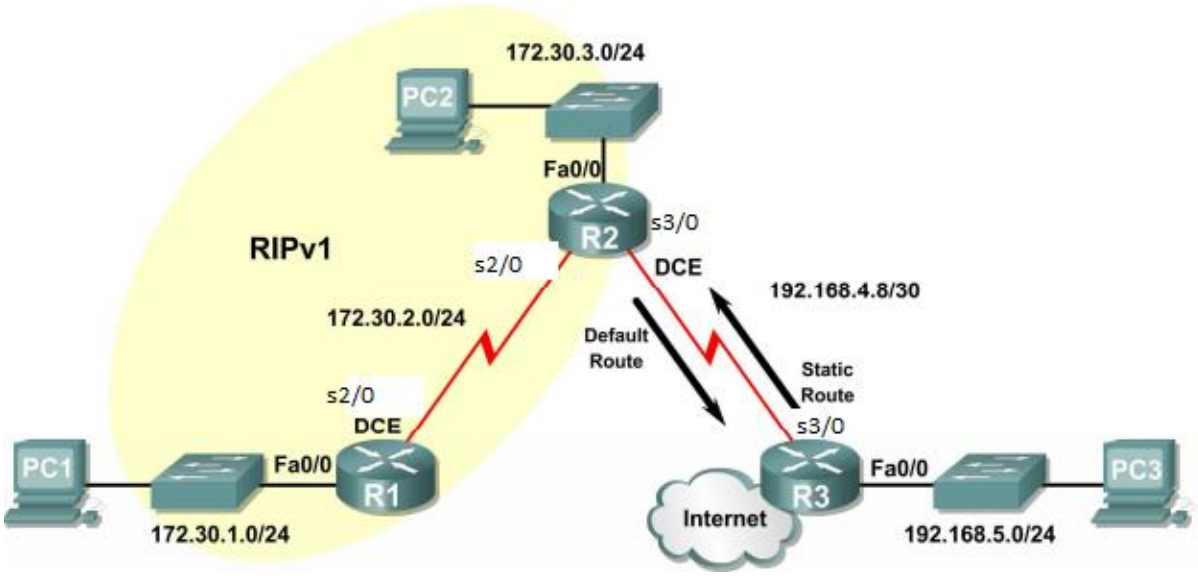
```
Routing for Networks:
        172.30.0.0
Passive Interface(s):
        FastEthernet0/0
```

We can also use *debug ip rip* to view the packets being sent and received by the router.

```
R1#debug ip rip
RIP protocol debugging is on
R1#RIP: sending  v1 update to 255.255.255.255 via Serial2/0 (172.30.2.1)
RIP: build update entries
      network 172.30.1.0 metric 1
RIP: received v1 update from 172.30.2.2 on Serial2/0
      172.30.3.0 in 1 hops
      192.168.4.0 in 1 hops
      192.168.5.0 in 2 hops
```

We can see that R1 is sending packets out its Serial2/0 interface but not out is fa0/0 interface due to the *passive interface* command. The router is also receiving packets via its Serial connection informing it of other networks. As RIP is a distance vector protocol, R1 does not have any knowledge of the structure of the topology outside its immediate neighbours, it merely knows the other networks distance in hops, which it considers when sending packets.

Scenario C: Running RIPv1 on a Stub Network



Her I have configured a static route between R3 and R2

```
R3(config)#ip route 172.30.0.0 255.255.252.0 serial 3/0
R3(config)#
```

As R2 is running RIP, it sends out updates every 30 seconds to its neighbours. We must tell it to include this static route information in its updates by using the *default information originate* command. Now when updates are send to R1, R1 will be informed of the static route.

Why use a static route between R2 and R3?

The reason a static route is configured between R2 and R3 is because R3 is not using RIP, so updates are not exchanged, as the routers need to both be configured using the same protocol to exchange information.

The static route can be seen in the last line of the screenshot below. R2 knows that if it receives a packet and does not know what to do with it, send in out the Serial 3/0 interface, which leads to R3.

```
     172.30.0.0/24 is subnetted, 3 subnets
R       172.30.1.0 [120/1] via 172.30.2.1, 00:00:18, Serial2/0
C       172.30.2.0 is directly connected, Serial2/0
C       172.30.3.0 is directly connected, FastEthernet0/0
     192.168.4.0/30 is subnetted, 1 subnets
C       192.168.4.8 is directly connected, Serial3/0
S*   0.0.0.0/0 is directly connected, Serial3/0
R2#
```

If we examine R1, we see that after we told R2 to include the static route in its updates via the *default-information originate* command, the next update sent to R1 included the static route between R2 and R3.

```
     172.30.0.0/24 is subnetted, 3 subnets
C       172.30.1.0 is directly connected, FastEthernet0/0
C       172.30.2.0 is directly connected, Serial2/0
R       172.30.3.0 [120/1] via 172.30.2.2, 00:00:14, Serial2/0
R*   0.0.0.0/0 [120/1] via 172.30.2.2, 00:00:14, Serial2/0
R1#
```

In the screenshot below, we can see that the static route has been configured on R3 and that if it wants to send packets to the 172.30.0.0 network, it sends them out the Serial 3/0 interface.

```
     172.30.0.0/22 is subnetted, 1 subnets
S       172.30.0.0 is directly connected, Serial3/0
     192.168.4.0/30 is subnetted, 1 subnets
C       192.168.4.8 is directly connected, Serial3/0
C    192.168.5.0/24 is directly connected, FastEthernet0/0
R3#
```

The functionality of this topology is better than the previous topology. R3 is no longer consuming bandwidth by sending updates every 30 seconds (RIP is disabled). The functionality of the topology is not compromised as R2 still informs R1 of the static route between itself and R3.

RIP v1 downfalls

RIP v1 is no longer commonly used for a number of reasons. It is a classfull protocol, meaning it does not send subnet mask information in its updates. RIPv1 does not support networks that have been sub netted with more than one subnet mask. Not supporting VLSM means that if a network chooses

to run RIP, more address spaces may be wasted as VLSM cannot be used to minimize losses. It also greatly limits the application abilities of RIP in large networks.

Routers running RIPv1 send out updates every 30 seconds, informing its neighbours of its routing table. The downfall of this is that RIP sends out these messages, even if no changes have been made to the topology of the network. Sending these packets every 30 seconds can consume bandwidth unnecessarily when no changes have occurred, as opposed to other protocols such as OSPF that only send out updates when there has been a topology change, thus using less resources and being more accurate as the updates are instant, not periodic.

The hop count metric is not reliable as it only takes into consideration the number of hops before a destination is reached, disregarding the speed at which some links are operating at. A router running RIP would consider a route with a hop of 2 routers and extremely poor bandwidths to be a better route than a route with 3 routers of very good bandwidths.

Conclusion regarding RIPv1

Although RIPv1 can be configured with minimal effort, its lack of compatibility with networks supporting VLSM make it instantly inferior to other protocols. I think that RIP would be suitable for a small stub network as it is easy to configure, but certainly not for larger networks that require sub netting. The fact that it sends out periodic updates means that it will always consume a certain percentage of bandwidth unless the network administrator has reduced the amount of interfaces sending updates, meaning that a protocol that only sends updates regarding topology changes may be more favourable.

RIP v2

RIPv2 is a more recent updated version of RIPv1. It retains many of RIPv1's characteristics and adds some improvements. RIPv2 is a classless protocol, meaning it does include subnet mask information in its updates. It also supports VLSM. Similarly to RIPv1 , RIPv2 sends fixed updates at regular intervals regardless of whether or not a topology change has occurred. It also uses a hop count metric and its AD is the same as RIPv1(120).RIPv2 uses multicast messages to update routing tables as opposed to the broadcast messages sent by RIPv1. RIPv2 also supports manual route summarization, so instead of sending a packet containing 8.0.0.0 /8, 9.0.0.0/8, 10.0.0.0/8, 11.0.0.0/8 the overall size of the packet could be reduced by summarizing the route and instead just sending the 8.0.0.0 252.0.0.0

The configuration of RIPv2 on routers is much the same as RIPv1, however in relation to the screenshot below, the command *version 2* would need to be inserted after the first line.

```
R1(config)#router rip
R1(config-router)#network 192.168.1.0
R1(config-router)#network 192.168.2.0
R1(config-router)#end
```

RIPv1 is very wasteful of a network's resources. All updates sent out by RIPv1 are broadcast updates, meaning that any router or host on the network will get updates regardless of whether or not they are relevant. This means that resources are consumed encapsulating, sending, and decapsulating packets. RIPv2 uses multicast messages so only those interested will receive the information.

In comparison to other protocols, RIP is rather inferior, as it is not conducive to larger networks, as its hop count limitations means its range is limited to 15 hops. The periodic updates mean that bandwidth is constantly being used, which is not desirable especially when the protocol does not make forwarding decisions based on bandwidth and link speed. Although the inclusion of VLSM make it an improvement of RIPv1, with regard to the standard of other protocols, RIPv2 is still inferior. Link State routers create topologies and have a faster convergence, making them much more desirable.

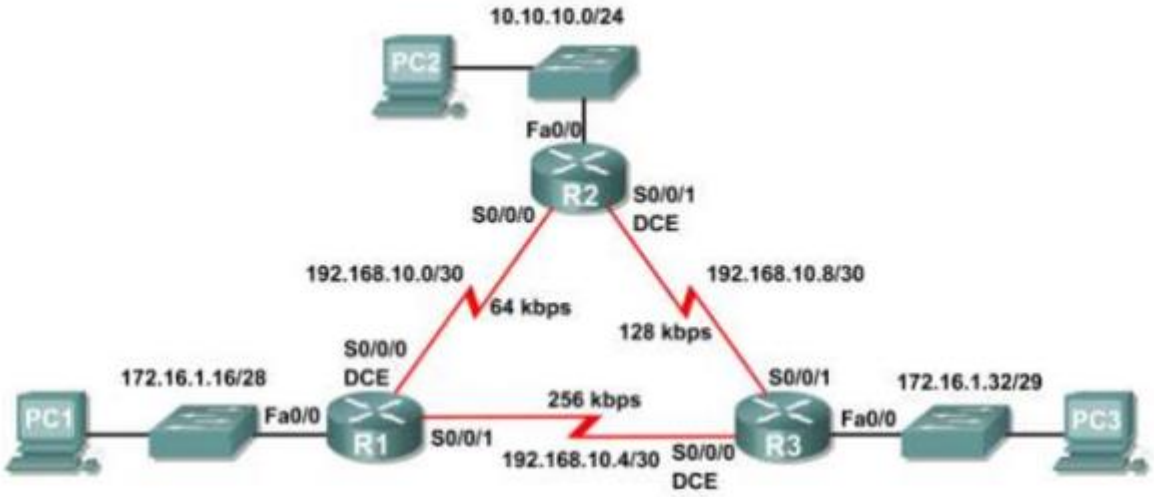
OSPF

OSPF is a link state protocol. Like RIP,OSPF(Open Shortest Path First) uses the SPF algorithm. It has an Administrative Distance of 110, and its metric is based on bandwidth and cost. OSPF has a high speed of convergence as it only sends updates when there is a topology change. OSPF routers elect a DR (Designated Router) and a (Back-up Designated Router). The DR receives information regarding topology changes, and then distributes that information throughout the network until the network has converged. OSPF is very scalable, as it uses a system whereby “areas” are constructed, with each area having its own border router, which can be used to connect networks/areas.

Differences between RIP and OSPF

Unlike RIP, OSPF does not send out periodic updates every 30 seconds. Instead, it only sends out updates containing information regarding a topology change. These updates are known as LSA’s and do not consist of an entire routing table, but instead just information regarding the topology change. OSPF is conducive to larger networks whereas RIP is not due to its limitations regarding hop count. OSPF also supports VLSM. Its compatibility with VLSM and its fast convergence time make OSPF an ideal choice for larger networks.

Basic OSPF configuration



```
R1(config)#router ospf 1
R1(config-router)#network 172.16.1.16 0.0.0.15 area 0
R1(config-router)#network 192.168.10.0 0.0.0.3 area 0
R1(config-router)#network 192.168.10.4 0.0.0.3 area 0
R1(config-router)#end
```

In the screenshot above, ospf has been configured to advertise three networks. Area 0 is usually the base area of a network. R1 now knows to advertise these networks in the updates it sends. After doing this with the other routers, we can check their routing tables to see if they are aware of the various networks in the topology.

Wildcard Masks

When configuring OSPF, we use a wildcard mask. The purpose of a WCM is to tell the router what part of an IP address to use. The wildcard mask is an inverse of a subnet mask. But why do we use wildcard masks? Wildcard masks allow us to identify the part of an IP address that we essentially don't care about. So if for example, we want to disregard any packets send from a 172.16.56.0 network, the wildcard mask essentially looks at the last octet and then disregards the bits in the last octet because as far as its concerned, once the other octets match 172.16.56.0, it has found a match.

So if for example we were to block the network in the example, we would use the following command where the wildcard mask essentially tells the router to disregard what information is in the final octet, just don't accept packets form the 172.16.56.0 network.

```
#deny 172.16.56.0 0.0.0.255
```

If we examine the routing table of R1, we can see that it has received updates from R2 and R3 as it has constructed its database regarding the converged network. R1 now knows that to reach the 172.16.1.32 network, it must send packets out its s0/0/1 interface.

```
     10.0.0.0/24 is subnetted, 1 subnets
O       10.10.10.0 [110/65] via 192.168.10.2, 00:04:44, Serial0/0/0
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.1.16/28 is directly connected, FastEthernet0/0
O       172.16.1.32/29 [110/65] via 192.168.10.6, 00:02:36, Serial0/0/1
     192.168.10.0/30 is subnetted, 3 subnets
C       192.168.10.0 is directly connected, Serial0/0/0
C       192.168.10.4 is directly connected, Serial0/0/1
O       192.168.10.8 [110/128] via 192.168.10.2, 00:02:08, Serial0/0/0
                     [110/128] via 192.168.10.6, 00:02:08, Serial0/0/1
R1#
```

OSPF Master/Slave

OSPF is a protocol where by routers establish a master-slave relationship. The master router will instigate communication between the routers, and the slave will receive the updates and respond accordingly. A slave cannot initialize communication between the two. In cases where there are multiple routers on a network, a similar approach is taken. A DR and a BDR are elected. The DR is essentially the master, and so it is responsible for receiving and distributing update information amongst the routers. In the case that this router becomes inactive, the BDR (which has been actively listening) becomes the DR and thus the network is not compromised. The system of having one router that is elected a master means that there are not numerous instances of update packets flooding across networks consuming bandwidth unnecessarily.

Router ID's

We need to be able to distinguish between routers. There are three ways in which a *router ID* can be assigned. It is prioritized as follows:

1. *router id* command sets ID
2. Highest IP of Loopback address
3. Highest IP address of router

After using *show ip protocols* on R1 we can see that the router ID has been set to the IP address as out of all the IP's on its interfaces this is currently the highest. If we were to assign a Loopback address, this IP would take priority, just as the router id entered using *router id* would override that.

```
Router ID 192.168.10.5
```

By configuring Loopback addresses, the current highest IP's will be overridden and will no longer be the router ID's. In the screenshot below, a loopback address with an IP of 10.1.1.1 255.255.255.255 is set, if we then compare the Router ID from before and after, we can see how the loopback address is prioritized.

```
R1(config)#interface loopback 0

%LINK-5-CHANGED: Interface Loopback0, changed stat

%LINEPROTO-5-UPDOWN: Line protocol on Interface Lo

R1(config-if)#ip address 10.1.1.1 255.255.255.255
R1(config-if)#exit
```

After copying the *running-config* to *startup-config* and reloading, the new router ID for R1 is as follows

```
Router ID 10.1.1.1
```

After using *router-id 10.4.4.4* the loopback address is disregarded as the router ID has the highest priority.

```
Router ID 10.4.4.4
Number of areas in th
Maximum path: 4
```

On examining the routing table for R1 we can see that R1 has successfully built a topology of the network. This consists of a number of OSPF routes, denoted by the O beside each line. We can see OSPF's AD in the bracket, along with the cost of each route. If we were to install alternative paths in this topology that all led to the same areas as the OSPF routes, OSPF would be used over RIP due to the lower AD. Take for example, the first OSPF route in the table. It has a cost of 65.

```
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.1/32 is directly connected, Loopback0
O       10.10.10.0/24 [110/65] via 192.168.10.2, 00:09:52, Serial0/0/0
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.1.16/28 is directly connected, FastEthernet0/0
O       172.16.1.32/29 [110/65] via 192.168.10.6, 00:16:52, Serial0/0/1
     192.168.10.0/30 is subnetted, 3 subnets
C       192.168.10.0 is directly connected, Serial0/0/0
C       192.168.10.4 is directly connected, Serial0/0/1
O       192.168.10.8 [110/128] via 192.168.10.6, 00:09:52, Serial0/0/1
                     [110/128] via 192.168.10.2, 00:09:52, Serial0/0/0
R1#
```

After configuring the bandwidth on the different interfaces, the routing table is updated to reflect the changes.

```
Process ID 1, Router ID 10.4.4.4, Network Type POINT-TO-POINT, Cost: 1562
Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0
```

DR and BDR election process in OSPF

Routers running RIP send periodic updates to their neighbours. The process of swapping information to converge a network in OSPF is more in depth. With the exception of a point to point network, a DR and a BDR have to be elected. The DR is designated to receive updates and to distribute them amongst its area. The BDR listens, and if the DR fails, the BDR becomes the DR. So how is a DR/BDR elected? All routers running OSPF are automatically assigned a priority of 1. We can see this, by using *show IP ospf interfaces* command.

```
 Router ID 10.4.4.4, Network Type
; is 1 sec, State DR, Priority 1
ıter (ID) 10.4.4.4, Interface addr
```

Here we see the router has a default priority of 1. If a router has a priority of 0, it is not eligible for election. If the priority of the routers are the same, the router ID, will be examined. As mentioned earlier, if that is not set, the highest loopback address will become the ID, or if not that, the highest IP on the router. When a router is elected DR, the BDR is elected in a similar manner. Other routers are then refered to as DRothers. If a BDR or DR fails, the DRothers are eligible for election. Any failed DR or BDR that then becomes active again, does not simply regain its place as a DR or BDR, but instead becomes a DRother(regardless of priority).

A network administrator may want to influence which router becomes the DR. this can be done by manually configuring a higher priority for a particular router. Using the *ip ospf priority* command allows us to manually set a priority to a router.

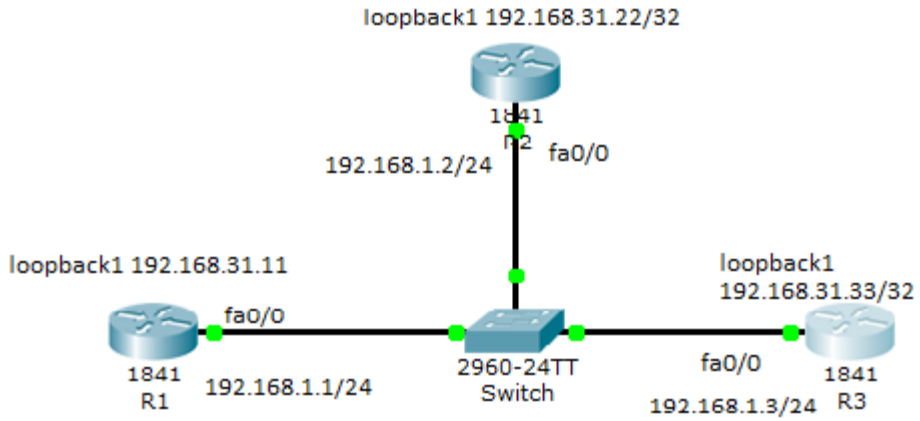
```
R1(config)#interface fa0/0
R1(config-if)#ip ospf priority 100
R1(config-if)#end
```

Now instead of the router having a priority of 1 like all the other routers, it has a priority of 100, meaning it will become DR when an election takes place.

```
 Router ID 10.4.4.4, Network Type BI
y is 1 sec, State DR, Priority 100
```

If it occurs that two or more routers share the highest priority, the router with the highest IP address will become the DR/BDR.

OSPF on a multi-access network



In the topology shown, the routers have been configured with the corresponding IP addresses. A DR needs to be elected, and no priority has been configured. By using show *ip ospf interface* on R3 we see that it has become the DR as it has the highest IP. We can see from the screenshots below that all three routers had the same priority and that their states were determined by the highest IP.

```
R3#show ip ospf interface
FastEthernet0/0 is up, line protocol is up
  Internet address is 192.168.1.3/24, Area 0
  Process ID 1, Router ID 192.168.31.33, Network
  Transmit Delay is 1 sec, State DR, Priority 1
```

```
2#show ip ospf interface
astEthernet0/0 is up, line protocol is up
  Internet address is 192.168.1.2/24, Area 0
  Process ID 1, Router ID 192.168.31.22, Network
  Transmit Delay is 1 sec, State BDR, Priority 1
```

```
R1#show ip ospf interface
FastEthernet0/0 is up, line protocol is up
  Internet address is 192.168.1.1/24, Area 0
  Process ID 1, Router ID 192.168.31.11, Network Type
  Transmit Delay is 1 sec, State DROTHER, Priority 1
```

I then used the *ip ospf priority* to change to manually determine who will become the DR/BDR/DROTHER.

R1 priority = 255

R2 priority = 0

R3 priority = 100

```
R1#show ip ospf interface
FastEthernet0/0 is up, line protocol
  Internet address is 192.168.1.1/24,
  Process ID 1, Router ID 192.168.31.
  Transmit Delay is 1 sec, State DR,
```

```
R2#show ip ospf interface
FastEthernet0/0 is up, line protocol is u
  Internet address is 192.168.1.2/24, Are
  Process ID 1, Router ID 192.168.31.22,
  Transmit Delay is 1 sec, State DROTHER,
```

```
R3#show ip ospf interface
FastEthernet0/0 is up, line protocol
  Internet address is 192.168.1.3/24,
  Process ID 1, Router ID 192.168.31.
  Transmit Delay is 1 sec, State BDR,
```

Note: Must shutdown interfaces and then use *no shutdown* for the changes to take effect.

OSPF allows for extensive control over a network. The ability to assign areas to subnetted networks allows an administrator to have freedom over the configuration of the network.

Extras

Routing Loops

Routing loops are a hazard to any network. Packets traversing a network with no way of getting to their destination use up precious resources and are a drain on bandwidth and CPU. Such loops occur due to incorrectly configured routes, or slow convergence on a network. There are ways to minimize the threat of routing loops, such as running protocols with fast convergence times, or implementing split horizon. OSPF has fast a fast convergence time, and anytime there is a topology change, updates are sent immediately. RIP uses count to infinity, which means packets can be given a TTL field, resulting in them being dropped if they exceed a certain TTL.

Split Horizon

Split horizon is a method where by a router does not advertise a network out through the interface from which the update regarding that network came. This helps prevent against routing loops.

Hello Packets & Dead Interval

Routers running OSPF send each other hello packets which are essentially there to confirm that the route is up and that no interface has gone down. If a router sends a hello packet and gets a response, it knows the route is active. The dead interval in OSPF refers to the number of seconds a router waits to hear a response from another router before it advertises it as being down. It is customary for the dead interval to be four times the hello interval.

Conclusion regarding dynamic routing protocols

When choosing a protocol to install on the routers in a network there are a number of factors to consider, such as scalability and cost. After learning how to configure RIP and OSPF I think that OSPF is the superior protocol. Link state protocols are more advanced than distance vector protocols. If I were to construct a network I would be inclined to choose OSPF or another link state protocol as opposed to either version of RIP. Even though it is more burdening on the resources, its convergence speed, event driven updates, and ability to construct a topology of the given network make it a more superior protocol in my opinion. Obviously the type of network is a factor in determining what protocols to use, however I think that the level of control the administrator has over a network running OSPF makes it the better protocol to choose.