**CONFIGURING SINGLE AREA OPEN SHORTEST PATH FIRST LAB**

******

***BY: ALEKSIE SANCHEZ MEDEIROS***

******

***Purpose***

To learn how to configure single area OSPFv2 and OSPFv3 connectivity among 5 point-to-point routers on a network.

***Background***

OSPF (Open Shortest Path First) is an interior routing protocol that uses a link-state routing algorithm to determine the best course for a packet to go through to reach a destination inside of a network. Before OSPF and link-state routing algorithms became one of the main routing protocols of the modern era there existed a distance-vector routing protocol known as RIP (Routing Information Protocol).

RIP is a distance-vector routing protocol that determines the best path for a packet based on a routing metric known as ‘distance’, distance being the number of ‘hops’ or other routers a packet would have to go through in order to reach the destination. RIP would only use the hop count between each router to create the best path to a network, however due to using hop count as the main parameter for routing, this would normally quickly create loops in the network that would cause packets to never reach their destination. One of the fixes to this problem was to have RIP have a hop count limit of fifteen routers. This would severely limit the amount of routers in a network using RIP and would affect further scalability. In an attempt to fix looping and incorrect routing information from being spread, RIP would take longer time to complete a routing table. Not only that but in the original RIP (RIPv1) would broadcast a packet asking for routing table information to every router using RIP and would repeat the message every thirty seconds. Although this was effective in consistently maintaining an updated routing table, the broadcasts would also create a massive flow of packets every thirty seconds from all the routers in a network, including the ones that were not actively using RIP. Though the issue of broadcasts was fixed later in RIPv2 which would instead only ask for routing tables from nearby routers instead of the whole network, RIPv2 still maintained a limit on scalability by keeping the hop count limit to 15.

As a solution to the shortcomings of the limitations and network convergence time delay of RIP, a different routing protocol known as link-state routing would be used to create paths to destinations in other networks. This system would divide networks into separate areas, where the routers would not need to have information on all the areas of a network but instead of just the areas that the routers are currently in, effectively reducing network convergence time. Furthermore, by also requiring a backbone area to connect all the areas together, routing loops were mitigated. In addition, the routers would not send their entire routing table to the routers’ neighbors but would instead send information that would allow routers to advertise or request information on the state of links between routers. This means that every router would create their own map of the entire network topology, preventing loops if a failure occurs as every device would internally calculate the best route once told of a link failure, although it does create a bit of traffic in the network. In comparison to RIP’s hop count, OSPF would use path cost as its routing metric to determine the best route to a network, path cost was determined by dividing the bandwidth over the speed of the interface where any values that are less than one would round up to 1. This value can also be modified manually in case the network administrator would want a different route to take precedence, due to faster interface speed or other reasons. All of this combined together, in addition to being scalable past a hop count of 15 and a lower difficulty of implementation compared to newer distance vector routing protocols meant that OSPF became more widely used in networks.

Single Area OSPF, is the simplest structure of OSPF where the entire network has an internal map of the links, since there is no division to separate parts of the network into a hierarchy. Once Single Area OSPF is configured correctly, the neighbors find out what other routes it is connected to, and distribute the information from their own links as advertisements. When information from a router in the network arrives, it fact checks information with that of other routers connected to the same link and adds in information and begins creating the routing table for the entire network, including finding the shortest path and remembering the first step of the route required to reach the network in which that first step sends it to the routers next first step repeating the process until the information is delivered. OSPFv3 is the IPv6 form of OSPF with a few differences, OSPFv3 only uses link-local addresses for neighbor discovery, instead of running per subnet.

***Lab summary***

At the end of the lab we had configured IPv4 and IPv6 connectivity between two computers on opposite ends of a 5 router point-to-point network using OSPFv2 and OSPFv3 to establish routes between all routers without needing to manually reference each possible route inside the configuration of the routers.

***Lab Commands***

***router ospf 1***is used for turning on the OSPF routing process and to give it a process-id. The id used to precisely identify what process is being configured as a router could be running multiple instances of OSPF.

***network [network address] [wildcard mask] area 0*** tells the OSPF instance which network and what the subnet that can be advertised to the other participating OSPF routers. The area 0 portion denotes what area that network is participating in.

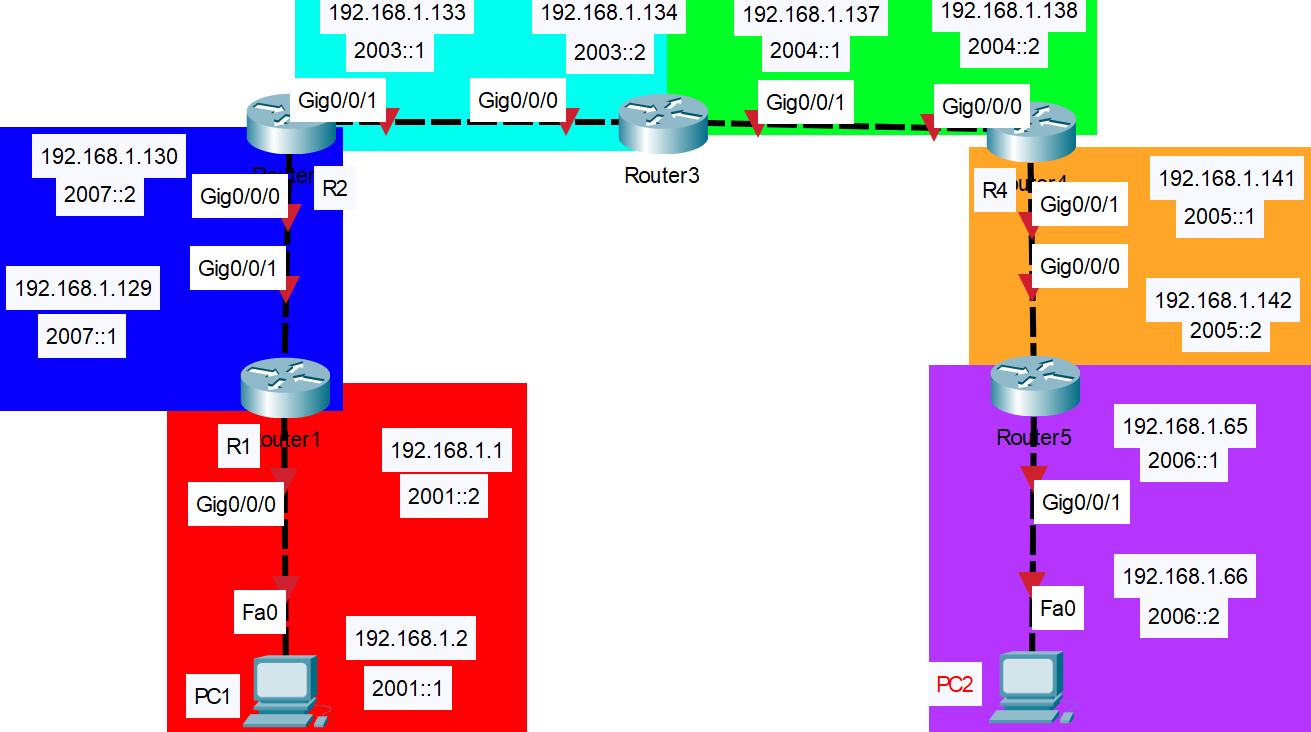
***router-id* *[32 bit address]*** is used as a unique identificator of the router specifically for the OSPFv2 and v3 process, it can be manually configured or automatically chosen as the loopback or ip address interface.

***ip ospf 1 area 0*** is utilized to tell the interface that it should be using the first OSPF process and should be communicating with other routes using area 0.

***ipv6 router ospf 1***is similar to ***router ospf 1*** but instead used for IPv6 and it does not require writing the networks that should be advertised and needs a unique router-id.

***ipv6 ospf 1 area 0***is akin to ***ip ospf 1 area 0*** but for IPv6

***Network Diagram***



***Router Configurations***

***R1:***

!

! Last configuration change at 22:30:08 UTC Thu Sep 8 2022

!

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

!

hostname R1

!

boot-start-marker

boot-end-marker

!

!

vrf definition Mgmt-intf

!

address-family ipv4

exit-address-family

!

address-family ipv6

exit-address-family

!

!

no aaa new-model

!

!

!

!

!

!

!

!

!

!

!

!

!

!

ipv6 unicast-routing

!

!

!

!

!

!

!

subscriber templating

multilink bundle-name authenticated

!

!

!

!

license udi pid ISR4321/K9 sn FDO21482HZX

!

spanning-tree extend system-id

!

!

redundancy

mode none

!

!

vlan internal allocation policy ascending

!

!

!

!

!

!

interface GigabitEthernet0/0/0

ip address 192.168.1.129 255.255.255.252

ip ospf 1 area 0

negotiation auto

ipv6 address 2002::1/64

ipv6 address 2007::1/64

ipv6 ospf 1 area 0

!

interface GigabitEthernet0/0/1

ip address 192.168.1.1 255.255.255.192

ip ospf 1 area 0

negotiation auto

ipv6 address 2001::2/64

ipv6 ospf 1 area 0

!

interface Serial0/1/0

no ip address

shutdown

!

interface Serial0/1/1

no ip address

shutdown

!

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

!

interface Vlan1

no ip address

shutdown

!

router ospf 1

network 192.168.1.0 0.0.0.63 area 0

network 192.168.1.128 0.0.0.3 area 0

!

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

!

!

ipv6 router ospf 1

router-id 1.1.1.1

!

!

!

!

control-plane

!

!

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

!

!

end

***R2:***

!

hostname R2

!

boot-start-marker

boot-end-marker

!

!

vrf definition Mgmt-intf

!

address-family ipv4

exit-address-family

!

address-family ipv6

exit-address-family

!

!

no aaa new-model

!

!

!

!

!

!

!

!

!

!

!

!

!

!

ipv6 unicast-routing

!

!

!

!

!

!

!

subscriber templating

multilink bundle-name authenticated

!

!

!

!

license udi pid ISR4321/K9 sn FDO21482DWJ

!

spanning-tree extend system-id

!

!

redundancy

mode none

!

!

vlan internal allocation policy ascending

!

!

!

!

!

!

interface GigabitEthernet0/0/0

description To R1 G0/0/1

ip address 192.168.1.133 255.255.255.252

ip ospf 1 area 0

negotiation auto

ipv6 address 2003::1/64

ipv6 ospf 1 area 0

!

interface GigabitEthernet0/0/1

description Connects to R3 G0/0/0

ip address 192.168.1.130 255.255.255.252

ip ospf 1 area 0

negotiation auto

ipv6 address 2002::2/64

ipv6 address 2007::2/64

ipv6 ospf 1 area 0

!

interface Serial0/1/0

!

interface Serial0/1/1

!

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

!

interface Vlan1

no ip address

shutdown

!

router ospf 1

network 192.168.1.128 0.0.0.3 area 0

network 192.168.1.132 0.0.0.3 area 0

!

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

!

!

ipv6 router ospf 1

router-id 2.2.2.2

!

!

!

!

control-plane

!

!

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

!

!

end

***R3:***

!

hostname R3

!

boot-start-marker

boot-end-marker

!

!

vrf definition Mgmt-intf

!

address-family ipv4

exit-address-family

!

address-family ipv6

exit-address-family

!

!

no aaa new-model

!

!

!

!

!

!

!

!

!

!

!

!

!

!

ipv6 unicast-routing

!

!

!

!

!

!

!

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

!

!

!

!

license udi pid ISR4321/K9 sn FDO214420HW

!

spanning-tree extend system-id

!

!

redundancy

mode none

!

!

vlan internal allocation policy ascending

!

!

!

!

!

!

interface GigabitEthernet0/0/0

description Connects to R2 G0/0/1

ip address 192.168.1.137 255.255.255.252

ip ospf 1 area 0

negotiation auto

ipv6 address 2004::1/64

ipv6 ospf 1 area 0

!

interface GigabitEthernet0/0/1

description Connects to R4 G0/0/0

ip address 192.168.1.134 255.255.255.252

ip ospf 1 area 0

negotiation auto

ipv6 address 2003::2/64

ipv6 ospf 1 area 0

!

interface Serial0/1/0

no ip address

shutdown

!

interface Serial0/1/1

no ip address

shutdown

!

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

!

interface Vlan1

no ip address

shutdown

!

router ospf 1

network 192.168.1.132 0.0.0.3 area 0

network 192.168.1.136 0.0.0.3 area 0

!

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

!

!

ipv6 router ospf 1

router-id 3.3.3.3

!

!

!

!

control-plane

!

!

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

!

!

end

***R4:***

!

! Last configuration change at 21:45:40 UTC Thu Sep 8 2022

!

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

!

hostname R4

!

boot-start-marker

boot-end-marker

!

!

vrf definition Mgmt-intf

!

address-family ipv4

exit-address-family

!

address-family ipv6

exit-address-family

!

!

no aaa new-model

!

!

!

!

!

!

!

!

!

!

!

!

!

!

ipv6 unicast-routing

!

!

!

!

!

!

!

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

!

!

!

!

license udi pid ISR4321/K9 sn FDO214421D1

!

spanning-tree extend system-id

!

!

redundancy

mode none

!

!

vlan internal allocation policy ascending

!

!

!

!

!

!

interface GigabitEthernet0/0/0

description Connects R3 G0/0/1

ip address 192.168.1.141 255.255.255.252

ip ospf 1 area 0

negotiation auto

ipv6 address 2005::1/64

ipv6 ospf 1 area 0

!

interface GigabitEthernet0/0/1

description Connects to R5 g0/0/0

ip address 192.168.1.138 255.255.255.252

ip ospf 1 area 0

negotiation auto

ipv6 address 2004::2/64

ipv6 ospf 1 area 0

!

interface Serial0/1/0

no ip address

shutdown

!

interface Serial0/1/1

no ip address

shutdown

!

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

!

interface Vlan1

no ip address

shutdown

!

router ospf 1

network 192.168.1.136 0.0.0.3 area 0

network 192.168.1.140 0.0.0.3 area 0

!

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

!

!

ipv6 router ospf 1

router-id 4.4.4.4

!

!

!

!

control-plane

!

!

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

!

!

end

***R5:***

!

hostname R5

!

boot-start-marker

boot-end-marker

!

!

vrf definition Mgmt-intf

!

address-family ipv4

exit-address-family

!

address-family ipv6

exit-address-family

!

!

no aaa new-model

!

!

!

!

login on-success log

!

!

!

!

!

!

!

subscriber templating

ipv6 unicast-routing

multilink bundle-name authenticated

!

!

!

crypto pki trustpoint TP-self-signed-859896477

enrollment selfsigned

subject-name cn=IOS-Self-Signed-Certificate-859896477

revocation-check none

rsakeypair TP-self-signed-859896477

!

!

crypto pki certificate chain TP-self-signed-859896477

certificate self-signed 01

3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 05050030

30312E30 2C060355 04031325 494F532D 53656C66 2D536967 6E65642D 43657274

69666963 6174652D 38353938 39363437 37301E17 0D323230 39303832 31313332

335A170D 33303031 30313030 30303030 5A303031 2E302C06 03550403 1325494F

532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3835 39383936

34373730 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02

82010100 C9F225AF 4B6ADF33 559D4FF7 F0AFF96B 4C37D734 5BE541CE 17408F3E

A2AD8FD2 BD3F7809 40BD3B4D 1592F042 7EAC48F1 DC1DF153 A25CBC8D 6A6D17C4

F503BC80 BAF3311D ACC92CCE 40458DF5 80AB8937 C1AB4E9E 03CF01FA 3B870FB8

01E5D012 2EA30412 742A21CC EF220F53 57F08197 E1009ACE BEA1DD9A 2EC2B839

48ED23EF B5653A64 5EF7EB16 F333AEFE 2609A60D 60B8B007 4CBB12F2 35D35150

19095CE6 39A4721F 84ECE62C EE05AD90 6941F25A 0589CCA6 15349D8E FDDFE824

A68EDDE2 513CA7CD C876388D D9410FFF 4D8F6B21 DE53029F A3EE9460 35D75FF2

52E235A3 30BC5E52 EA2F890A C983B03A 2BB394BE D13B4F7B EE921365 AE764EE6

DC18EC51 02030100 01A35330 51300F06 03551D13 0101FF04 05300301 01FF301F

0603551D 23041830 16801430 374799BC 5D153388 C26DCA51 5D6B20F4 DD48CC30

1D060355 1D0E0416 04143037 4799BC5D 153388C2 6DCA515D 6B20F4DD 48CC300D

06092A86 4886F70D 01010505 00038201 010014B6 F876F359 B8CF468B A70B3DDE

245D928C FE50B500 56DB2279 C62ECBC2 0D7AAAC1 8BE1A819 5B837A7B 96C877C0

9BACA2A1 611C0729 08A326B5 309FA8B0 BEE7D936 A9E09834 8D92AFBE CE34E692

A2554B17 71DC3B24 FC68B952 26003841 452D57DE B72AE1E8 8FE3FE7B B6B94922

501898A0 550CFA71 C24B5201 EFADE2E0 40353E04 08840B7E F02DECE0 0C0DD2B5

C653752B 97F9919A 226AF7A4 9AB274A4 61297BFC 918FAAC4 12070EE7 29BFCF6D

0EEF9B88 D35AEDF7 7AD6BA0D CE3A734C 4D438388 98CD3D30 48D57E86 06C1B001

586C81E0 8C71678D 905CB5E9 116FFF8F A44F0E3A F9E2E2D3 A4B325CB 215E1FCF

B52A97F2 1A2DCC2B 132A7CD3 EFDE12E1 92BF

quit

!

license udi pid ISR4321/K9 sn FLM240608PJ

no license smart enable

diagnostic bootup level minimal

!

spanning-tree extend system-id

!

!

!

!

redundancy

mode none

!

!

!

!

!

!

!

!

interface GigabitEthernet0/0/0

ip address 192.168.1.65 255.255.255.192

negotiation auto

ipv6 address 2006::1/64

ipv6 ospf 1 area 0

!

interface GigabitEthernet0/0/1

ip address 192.168.1.142 255.255.255.252

negotiation auto

ipv6 address 2005::2/64

ipv6 ospf 1 area 0

!

interface GigabitEthernet0/1/0

no ip address

shutdown

negotiation auto

!

interface GigabitEthernet0/1/1

no ip address

shutdown

negotiation auto

!

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

!

router ospf 1

router-id 5.5.5.5

network 192.168.1.64 0.0.0.63 area 0

network 192.168.1.140 0.0.0.3 area 0

!

ip forward-protocol nd

ip http server

ip http authentication local

ip http secure-server

ip tftp source-interface GigabitEthernet0

!

!

ipv6 router ospf 1

router-id 5.5.5.5

!

!

!

!

!

control-plane

!

!

line con 0

transport input none

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

!

!

!

!

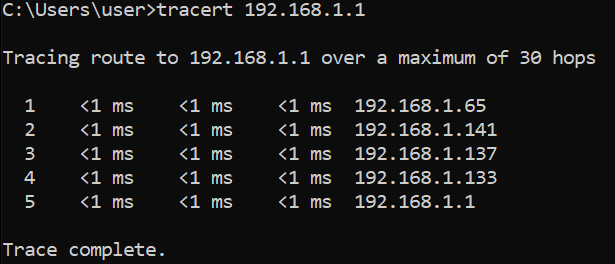
!

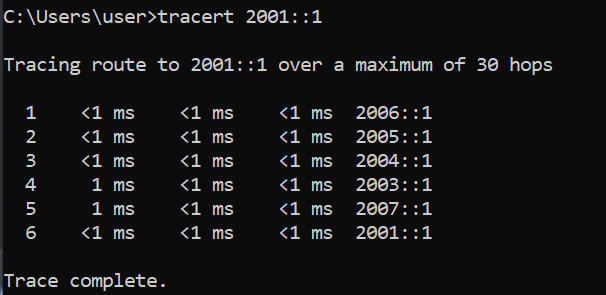
!

end

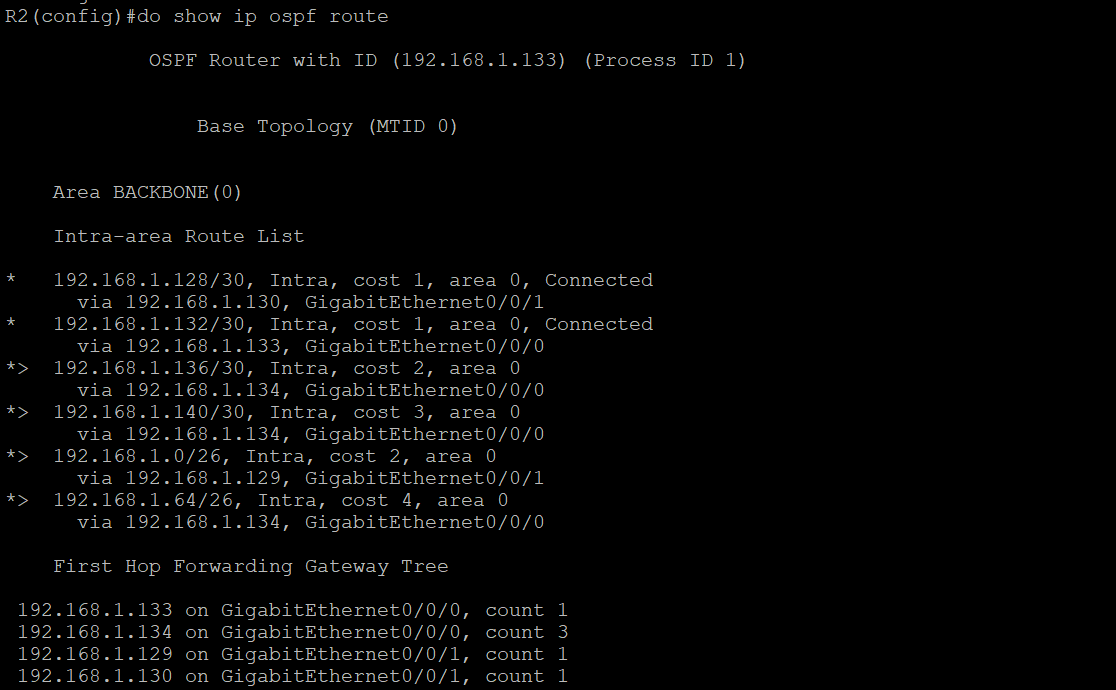
***Screenshots***

*Shown below is a picture of a traceroute from PC 2 to Router 1 made to test connectivity across the network*

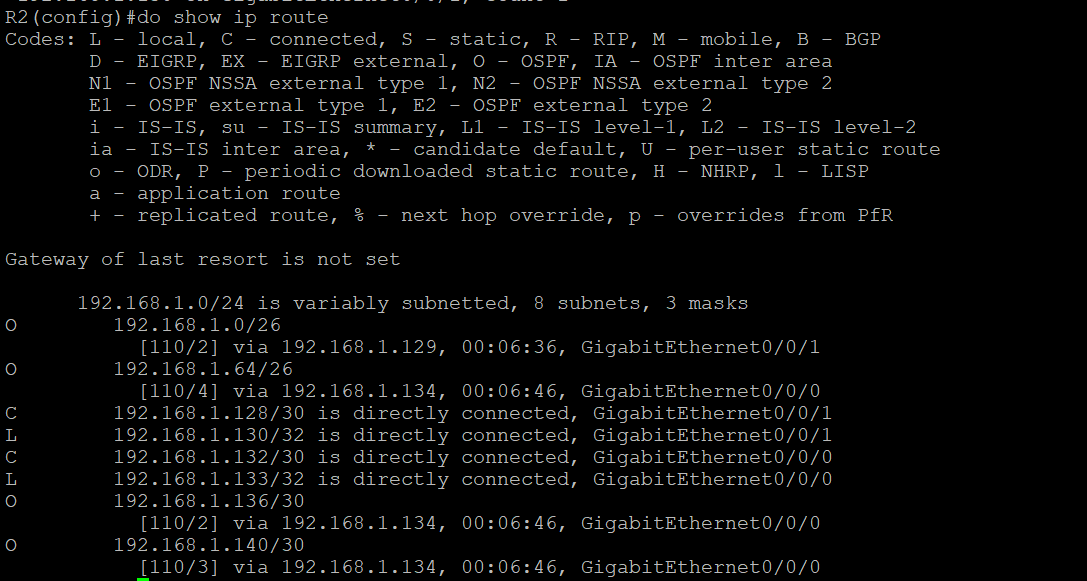


*Shown below is a trace route with a destination of PC1 made to test connectivity across the network*

*Shown below is the OSPF routing table of R2. This tells R2 where the best routes to send each packet entering R2.*



*Shown below is the entire routing table of R2, this tells R2 where to send each packet entering R2; note the different letters next to each entry that state where each route was created*



***Problems***

One of the first issues that I ran into when beginning our OSPF lab was subnetting ip addresses properly as I had opted for a more ‘realistic’ approach to our subnetting with the routers having point to point subnet masks and our pc connections having more available ip addresses. This would cause many issues with overlapping ip addresses that would cause perplexing difficulties when determining what went wrong with the OSPF neighboring process between interfaces. To fix this, I allocated the first 128 ip addresses towards the 2 larger 64 address subnet blocks and the rest were left towards the point-to-point networks between the routers.

On the physical aspect, many of the cables that I was using were longer than needed to be and would tangle with each other, causing a chaotic mess to determine if the diagram was being properly followed, I combatted this issue by finding or terminating new shorter cables that could be used in our lab without leaving a dangling mess, and also recreated the diagram to make more sense to my eyes.

With OSPFv3, the pc and router pings and traceroutes could go through the 2002::/64 network but could not reach the two routing devices on that specific network, this issue turner out to be because the 2002::/64 network is a privately reserved ipv6 address block that was used for

***Conclusion***

In the end, I had learnt about how OSPFv3 doesn’t require configuration of network advertisements and only requires an ipv6 address in the interfaces, ipv6 routing, and a router id. I have also learnt that IPv6 also has reserved address blocks towards certain functions. I now understand that cable management is key to understanding what is going on in a network and how to create a simple point-to-point network that uses single area OSPFv2 and OSPFv3 for connectivity.