FINAL YEAR REPORT

Title: Secure Redundant Network

Student: Aaron Morrissey

ID:C00239014

Supervisor: James Egan

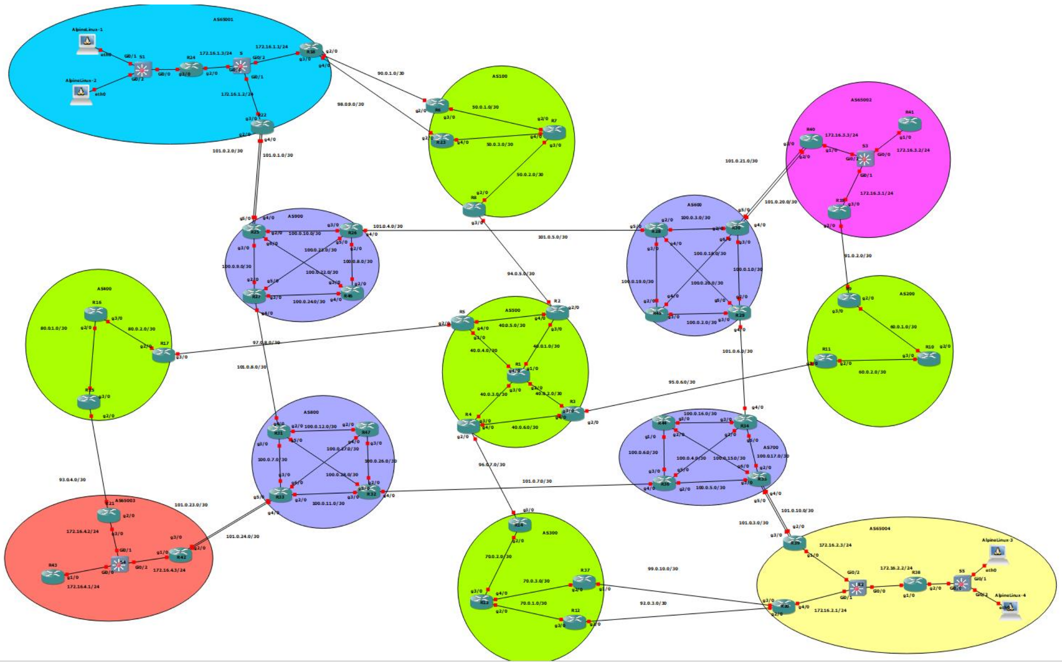
Contents

Abstract

The contents of this paper detail and examine the aims, processes, corrections and conclusions that I came to when designing a WAN network for a hospital group that would attempt to benefit users on the network, both guest and staff members, by improving areas such as Scalability, Security, Performance, etc. However, I would be remiss if I also didn’t mention some of the various practices and methods I originally planned to implement and why I chose to pass on them for a different option.

Introduction

A network is essentially the nervous system of a company or environment that it's been placed in and provides us the ability to achieve numerous goals and tasks, from giving us the ability to communicate with each other over long distances to storing an exorbitant amount of data and information that we could not hold on our own while simultaneous granting us access to it within seconds, as well as protecting us from external and unknown malefactors that are hidden or unknown to us. Given that it is such a monumental requirement for any business or organization these days, the need for an optimal and secure network is crucial for any company to survive and thrive. As such, I have been given the task of designing and implementing a mock network based on a hospital group, whose sites consist of Dublin Carlow, Belfast, and Cork in correlation with its staff's complaints about its current quality of service. This process will require equal parts trial and error as well as simultaneous research and study in order to find the best combination of technologies and practices in order to achieve a functioning and stable network.



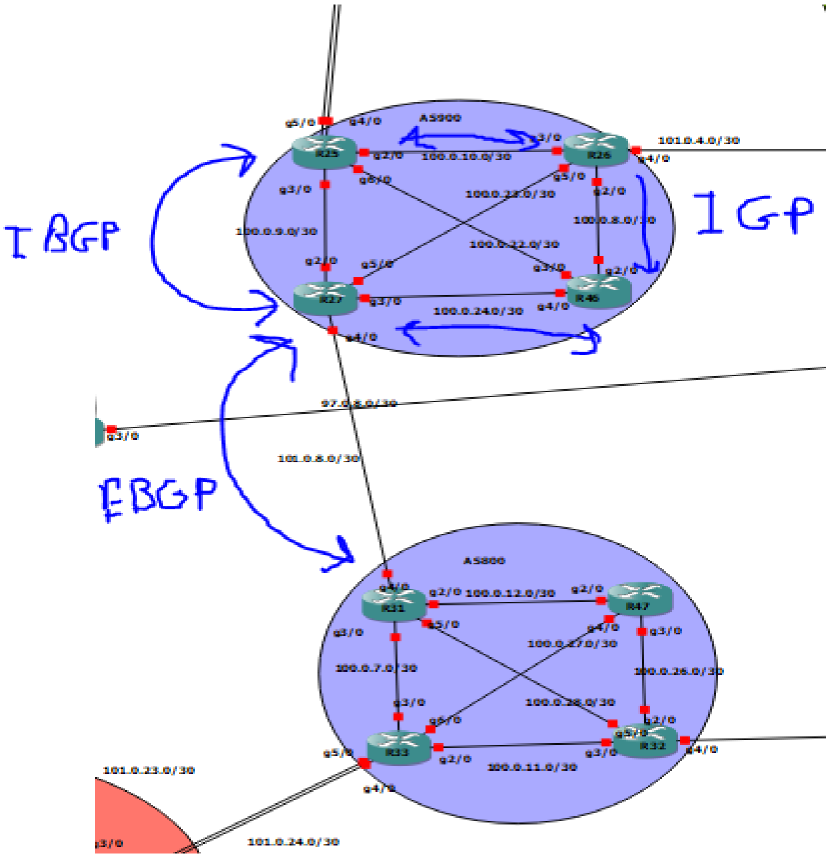
Topology Explanation

From the above image, my topology currently consists of 4 hospital sites at each corner that are connected to 2 ISPs, the green being used to route non-administrative traffic across the various autonomous systems to other sites, and purple, meant for devices using a VPN to communicate with admin devices. Unfortunately, due to delays and poor time management the topology was significantly downsized in regards to hospital site configurations as to allow me to focus more on the core network consisting of the ISP's that would have a greater influence on the network as a whole.

Main technologies implemented

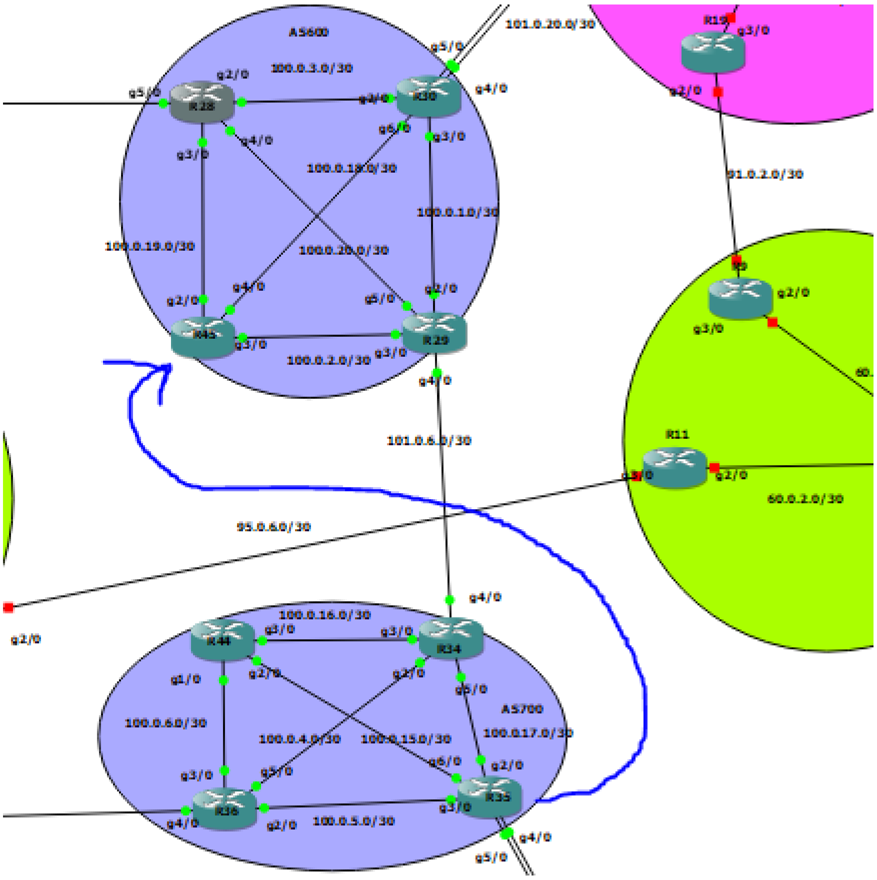
BGP (Border Gateway Protocol)

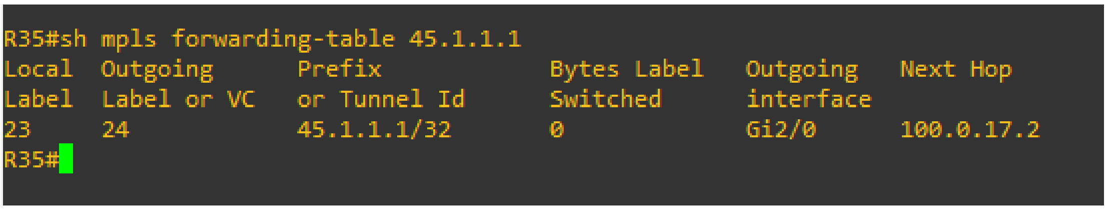
Out of all the technologies and methods that were incorporated into this project, Border Gateway Protocol was the most influential and important as its implementation affected the whole of the network. In basic terms, this protocol is essentially what is used to connect the entire internet, it functions by determining how data is routed over numerous Autonomous Systems (basically a network and the prefixes that belong to it) by the process of exchanging information about routers and their reachability that is passed between system edge peer routers to their internal peers. To clarify, BGP and be classified into two processes, eBGP (external) and iBGP (internal), as you can guess from the names eBGP refers to when BGP neighbours are peering with one another on separate autonomous systems whereas iBGP refers to neighbours that exist within the same autonomous system. IBGP is used to gather routing information from other internal peers whereas eBGP is used to send that route information to an external peer which will then share it internally allowing for routing to occur, in my case, via an IGP.



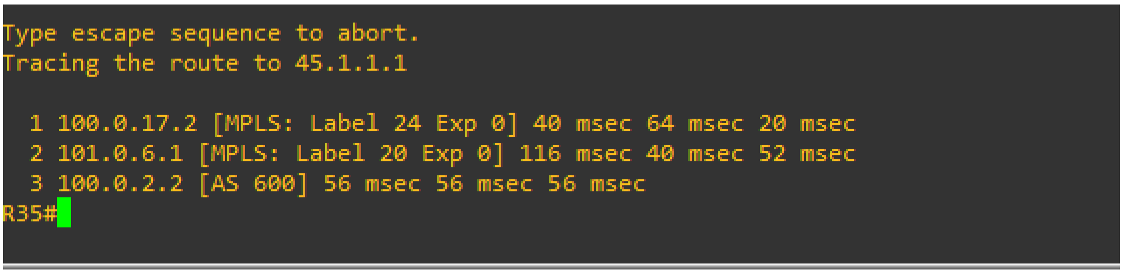
MPLS (Multiprotocol Label Switching)

The other major technology that I wished incorporate into this project was Multiprotocol Label Switching, this is used in place of traditional path routing, so while other protocols would decide based on source and destination addresses, MPLS uses predefined labels added to the routers of autonomous systems that get swapped with one another as the traffic gets routed across the network until it reaches the last hop before the MPLS network ends. For example, take the following scenario where I want to route traffic from router35 to router45. By using the MPLS forwarding-table command followed by our target address, we can see that the outgoing label being used to get to our destination is set as 24. So, if we use a traceroute to it shows that, yes, the first label it will use to get to the destination will be label 24 which will then be swapped out with the label of the next best hop until it reaches its destination.





```
R35#sh mpls forwarding-table 45.1.1.1
Local  Outgoing    Prefix           Bytes Label   Outgoing   Next Hop
Label  Label or VC or Tunnel Id     Switched      interface
23     24          45.1.1.1/32      0             Gi2/0      100.0.17.2
R35#
```



```
Type escape sequence to abort.
Tracing the route to 45.1.1.1

  1 100.0.17.2 [MPLS: Label 24 Exp 0] 40 msec 64 msec 20 msec
  2 101.0.6.1 [MPLS: Label 20 Exp 0] 116 msec 40 msec 52 msec
  3 100.0.2.2 [AS 600] 56 msec 56 msec 56 msec
R35#
```

Main areas of improvement

Performance

MPLS provides many benefits for a network, one of main ones being increased speed of network traffic as well as improved control over it. Due to the process of data not needing to be routed using routing tables, which becomes more complex as well as time-consuming the larger the network gets, it forwards data much faster using predefined labels and as improves performance immensely. Originally this was the technology that would be used to test the performance of VoIP between two admin machines over the internet service providers, however, as I mention I unfortunately had to downscale.

Scalability

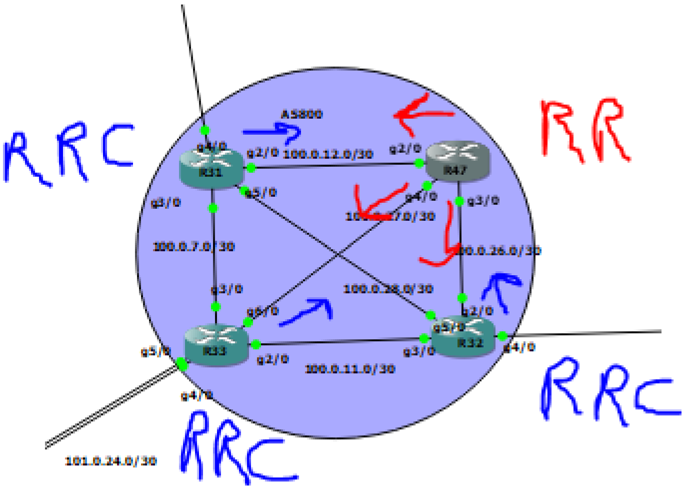
One of the features that BGP provides is its loop prevent mechanism, one for both external and internal purposes. In the case of internal BGP, it follows a rule whereby routes that have been learnt from other internal neighbours are presumed to already be known be all other members within that autonomous. As such, this requires iBGP to be implemented with a full mesh topology in order for all BGP sessions to be established. This method of network can prove to have a few disadvantages, the main two being that, one, mass configuration would be needed on all individual routers, not taking into account the additional commands configured on the router, and two, high unnecessary volumes of BGP sessions. For instance, take a scenario where an autonomous system consists of ten routers, in order for a full routing table to be established all routers must be peering with each other resulting in a total of 45 separate BGP sessions. Additionally, it takes a minimum of two commands in order to establish neighbour peering, not accounting for configs outside of BGP, which need to must be entered 8 more times to the router with all the different IP addresses of its peers, once this process is complete it then must be replicated again on the other 9 routers.

BGP SESSION FORMUAL = N(N-1)/2

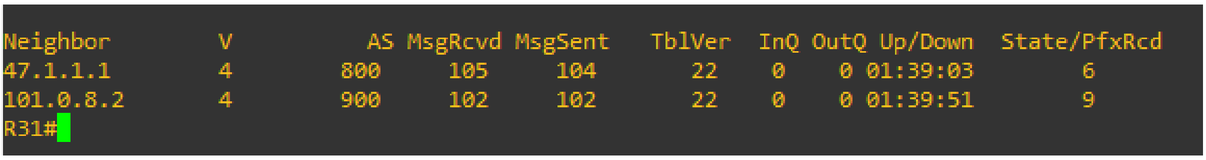
COMMANDS:

- **Neighbor xxxx.xxxx.xxxx.xxxx remote-as xx**
- **Neighbor xxxx.xxxx.xxxx.xxxx update-source xxxx**

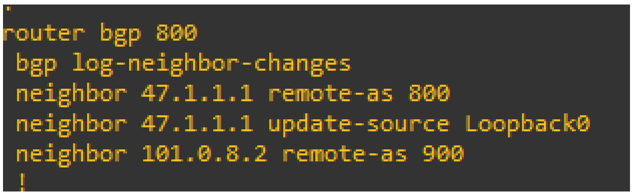
For these reasons I have implemented Route Reflectors (RR) into my topology that's purpose is to essentially accomplish what a full mesh is trying to achieve but with the added benefits of increased scalability and fewer amounts of configurations needed. It achieves this by first establishing a router within the autonomous system as a route reflector, this router will then establish peering sessions with all other routers in the system and classify them as router reflector clients (RRC), we then configure the rest of the device to peer solely with the route reflector instead of all the other routers. The reason for this is because, as the name might suggest, the route reflector is learning information from its clients and reflecting it back out to be seen by the rest, resulting in a full iBGP peering being accomplished without the additional complexities.



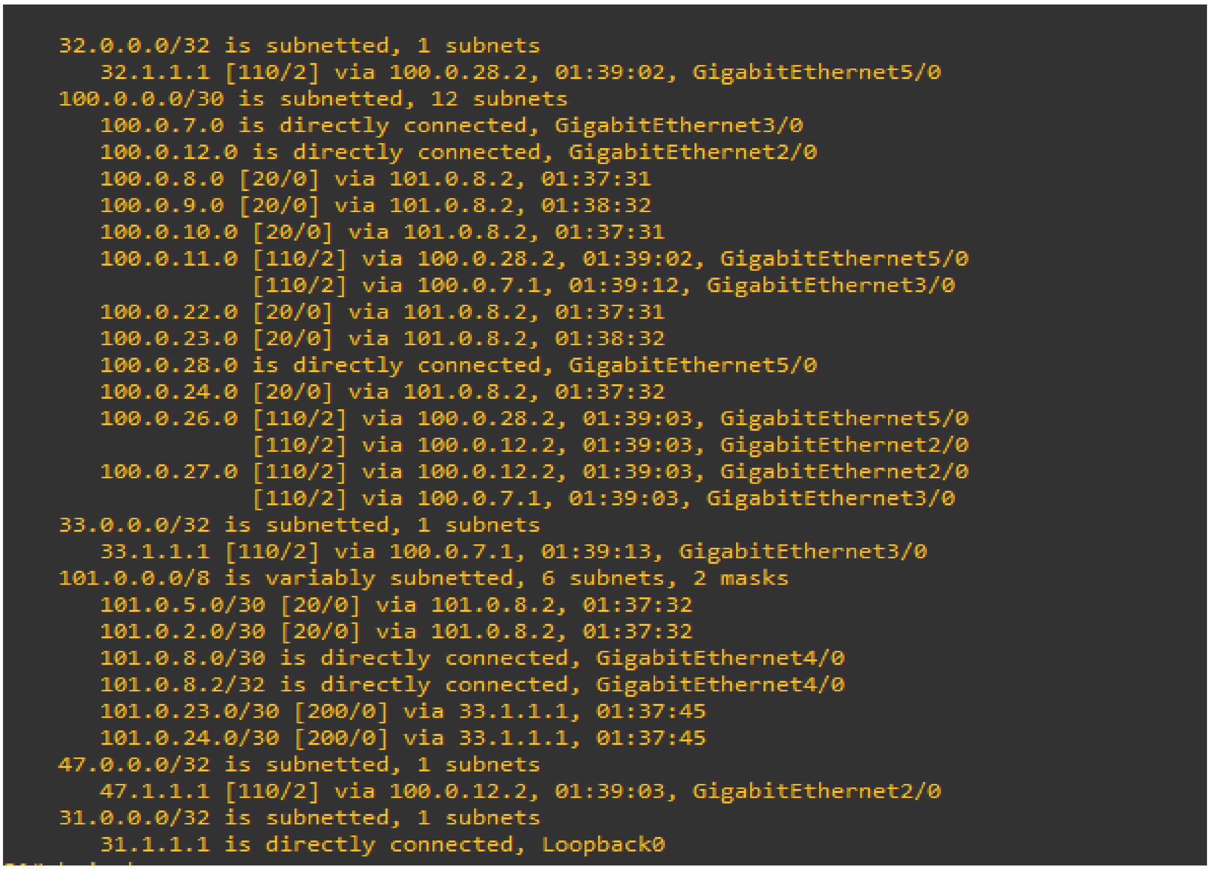
ROUTER REFLECTOR CLIENT CONFIG



```
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
47.1.1.1        4          800     105     104       22    0    0 01:39:03        6
101.0.8.2       4          900     102     102       22    0    0 01:39:51        9
R31#
```

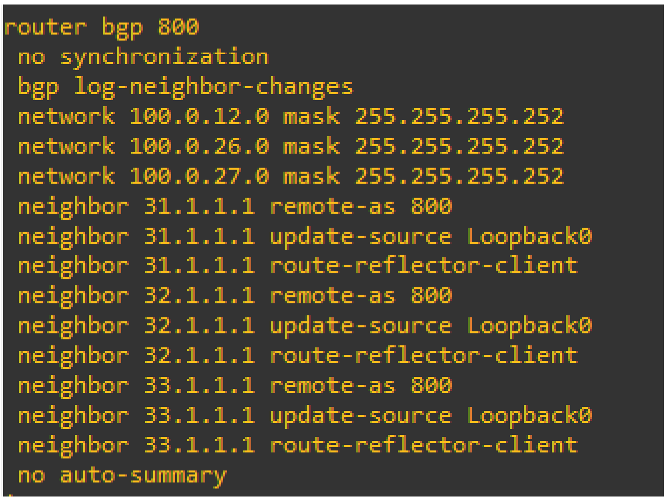


```
router bgp 800
 bgp log-neighbor-changes
 neighbor 47.1.1.1 remote-as 800
 neighbor 47.1.1.1 update-source Loopback0
 neighbor 101.0.8.2 remote-as 900
 !
```

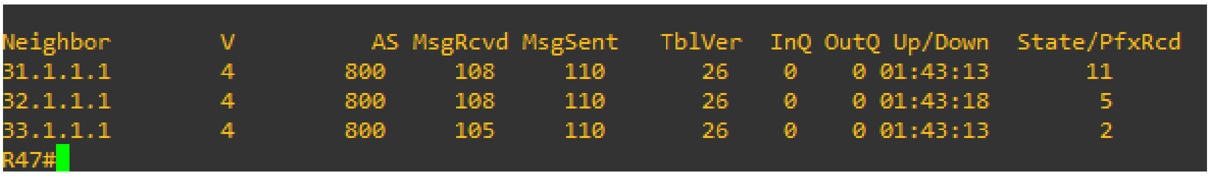


```
    32.0.0.0/32 is subnetted, 1 subnets
       32.1.1.1 [110/2] via 100.0.28.2, 01:39:02, GigabitEthernet5/0
    100.0.0.0/30 is subnetted, 12 subnets
       100.0.7.0 is directly connected, GigabitEthernet3/0
       100.0.12.0 is directly connected, GigabitEthernet2/0
       100.0.8.0 [20/0] via 101.0.8.2, 01:37:31
       100.0.9.0 [20/0] via 101.0.8.2, 01:38:32
       100.0.10.0 [20/0] via 101.0.8.2, 01:37:31
       100.0.11.0 [110/2] via 100.0.28.2, 01:39:02, GigabitEthernet5/0
                  [110/2] via 100.0.7.1, 01:39:12, GigabitEthernet3/0
       100.0.22.0 [20/0] via 101.0.8.2, 01:37:31
       100.0.23.0 [20/0] via 101.0.8.2, 01:38:32
       100.0.28.0 is directly connected, GigabitEthernet5/0
       100.0.24.0 [20/0] via 101.0.8.2, 01:37:32
       100.0.26.0 [110/2] via 100.0.28.2, 01:39:03, GigabitEthernet5/0
                  [110/2] via 100.0.12.2, 01:39:03, GigabitEthernet2/0
       100.0.27.0 [110/2] via 100.0.12.2, 01:39:03, GigabitEthernet2/0
                  [110/2] via 100.0.7.1, 01:39:03, GigabitEthernet3/0
    33.0.0.0/32 is subnetted, 1 subnets
       33.1.1.1 [110/2] via 100.0.7.1, 01:39:13, GigabitEthernet3/0
    101.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
       101.0.5.0/30 [20/0] via 101.0.8.2, 01:37:32
       101.0.2.0/30 [20/0] via 101.0.8.2, 01:37:32
       101.0.8.0/30 is directly connected, GigabitEthernet4/0
       101.0.8.2/32 is directly connected, GigabitEthernet4/0
       101.0.23.0/30 [200/0] via 33.1.1.1, 01:37:45
       101.0.24.0/30 [200/0] via 33.1.1.1, 01:37:45
    47.0.0.0/32 is subnetted, 1 subnets
       47.1.1.1 [110/2] via 100.0.12.2, 01:39:03, GigabitEthernet2/0
    31.0.0.0/32 is subnetted, 1 subnets
       31.1.1.1 is directly connected, Loopback0
```

ROUTER REFLECTOR CONFIG



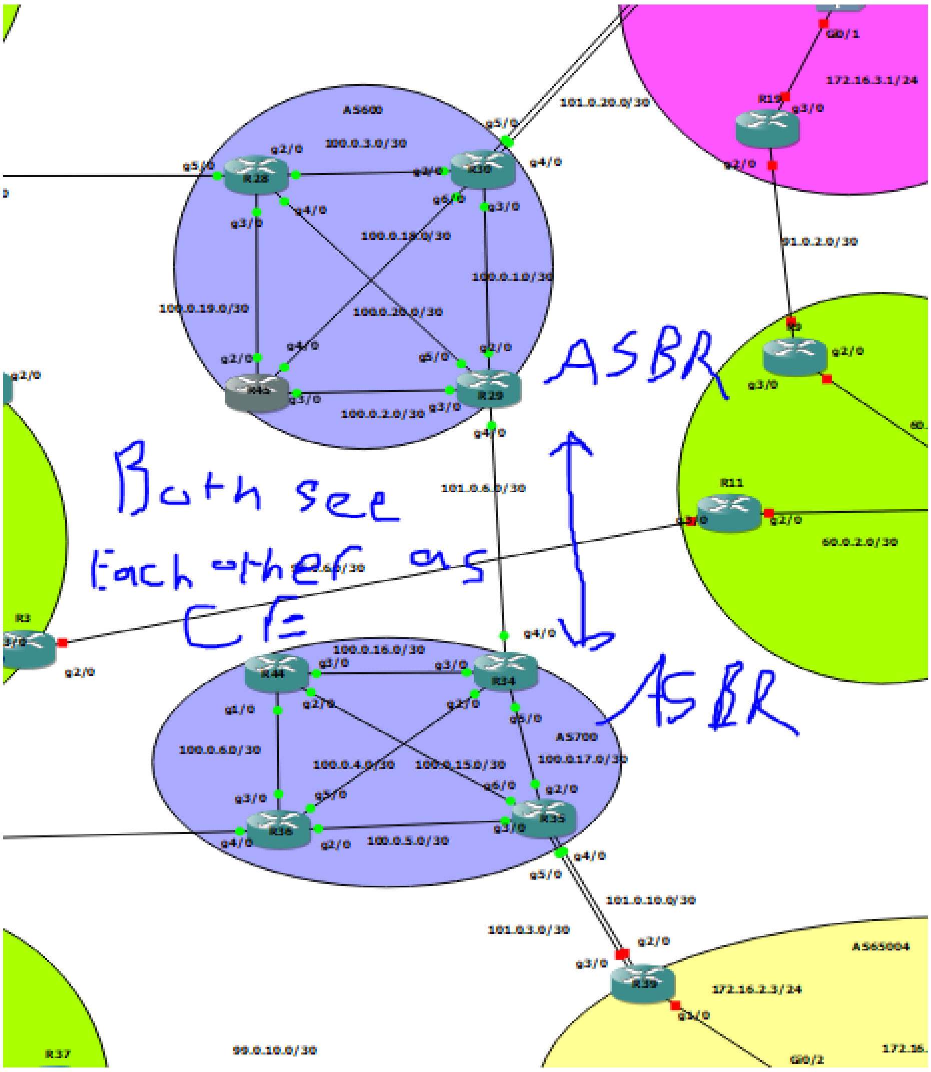
```
router bgp 800
 no synchronization
 bgp log-neighbor-changes
 network 100.0.12.0 mask 255.255.255.252
 network 100.0.26.0 mask 255.255.255.252
 network 100.0.27.0 mask 255.255.255.252
 neighbor 31.1.1.1 remote-as 800
 neighbor 31.1.1.1 update-source Loopback0
 neighbor 31.1.1.1 route-reflector-client
 neighbor 32.1.1.1 remote-as 800
 neighbor 32.1.1.1 update-source Loopback0
 neighbor 32.1.1.1 route-reflector-client
 neighbor 33.1.1.1 remote-as 800
 neighbor 33.1.1.1 update-source Loopback0
 neighbor 33.1.1.1 route-reflector-client
 no auto-summary
```



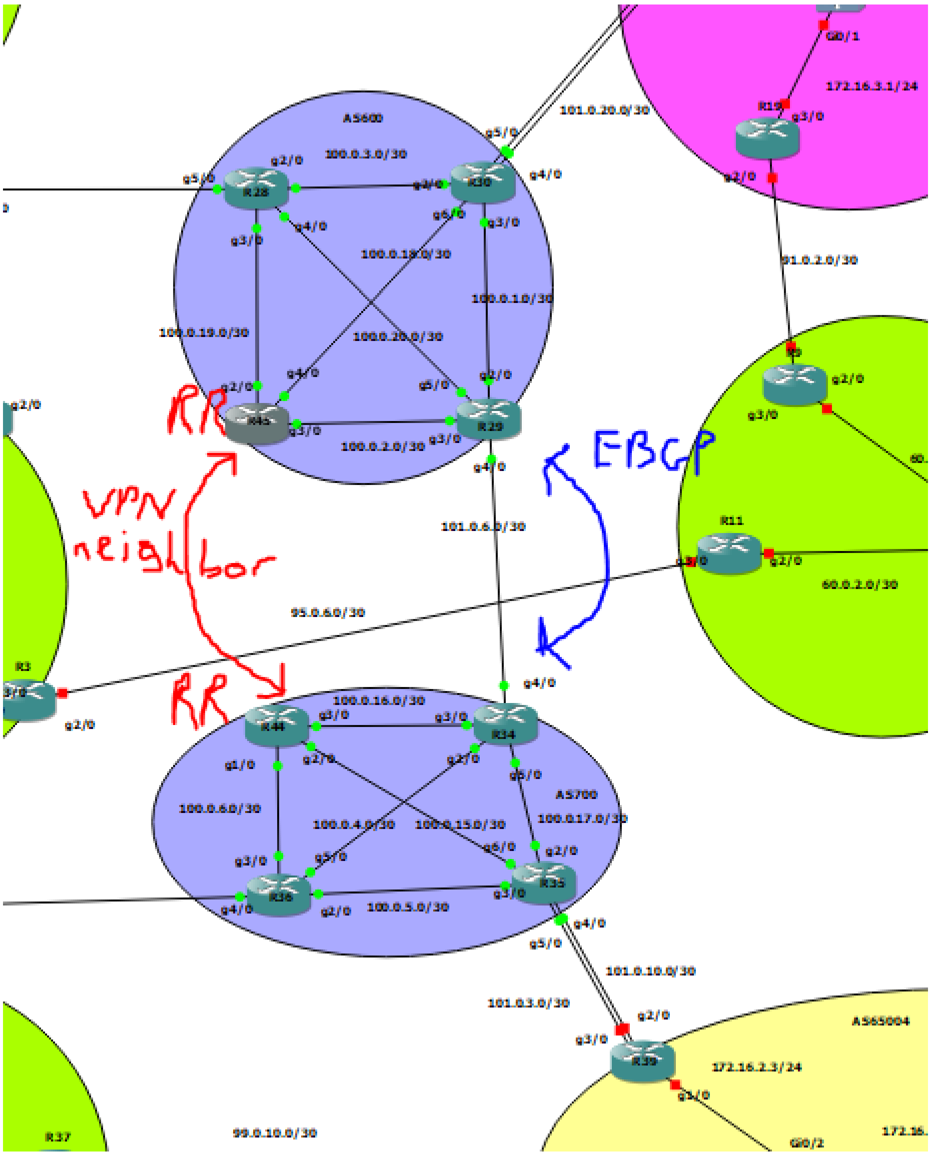
```
Neighbor        V          AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
31.1.1.1        4        800     108     110       26    0    0 01:43:13       11
32.1.1.1        4        800     108     110       26    0    0 01:43:18        5
33.1.1.1        4        800     105     110       26    0    0 01:43:13        2
R47#
```

Security

Although it did not work out as planned, one of the security features I wanted to add into the topology was an Inter-AS MPLS VPN configured for separate ISP’s that would be dedicated to routing traffic for administrative staff and machines at each hospital site. Originally, I intended to implement a method where the virtual routing and forwarding table that resides in an autonomous systems ASBR is directly connected to another autonomous systems ASBR’s virtual routing table in order to utilize the VPN attached to them. However, it became apparent that as the network grow and more autonomous systems were added to the topology, the number of virtual routing protocols attached to the ASBRs would begin to be overwhelming and would result in poor scalability.

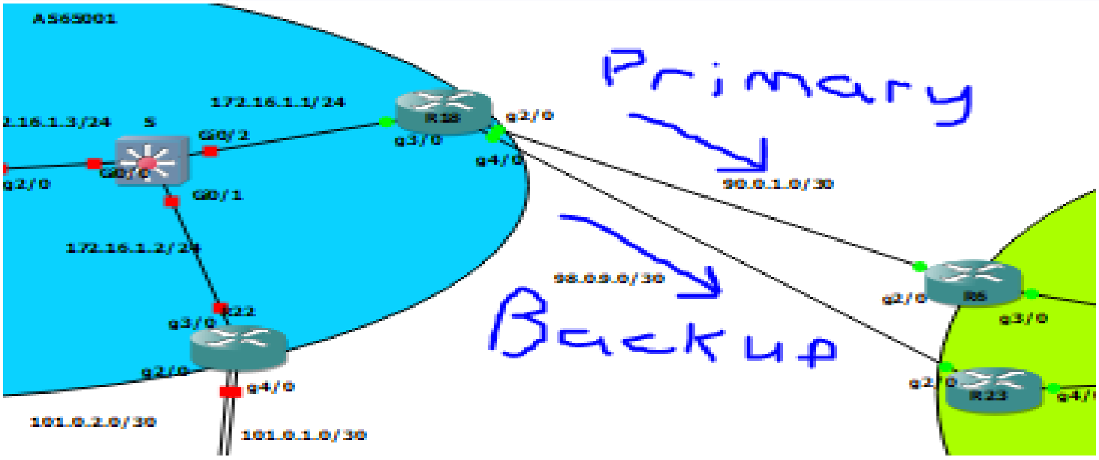


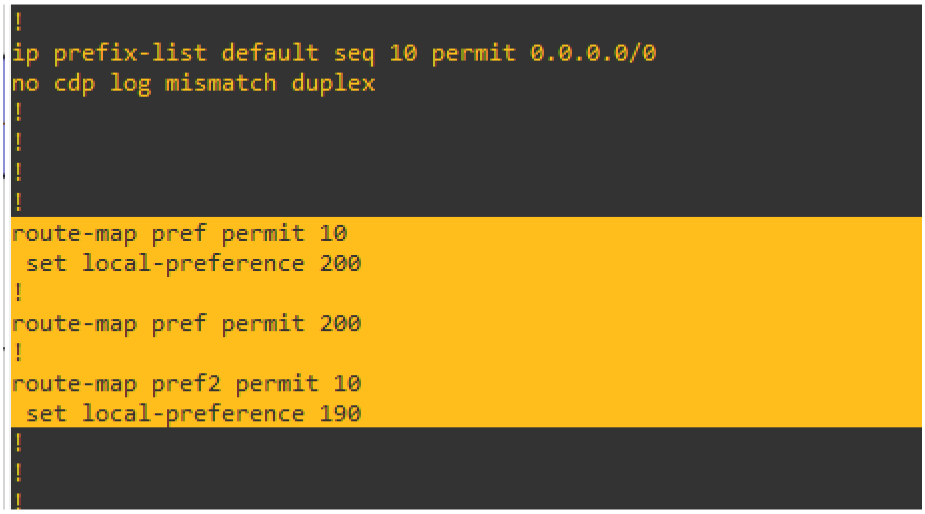
The next method that I tried to implement was using MP-BGP to distribute labelled VPN routes from a provider edge router to the route reflector that would reflect them to the ASBR. The ASBR would then use MP EBGP to further distribute VPN routes to its external peer which would then follow the first two steps but in reverse order, so the ASBR of that autonomous system would distribute the VPN routes to the provider edger via the route reflector. Unfortunately, it was only able to operate up to a point where, yes, the hospital routers know of the routes within the other hospital network, however, the issue lies with those routers not being able to make it past the second hop of the MPLS network.



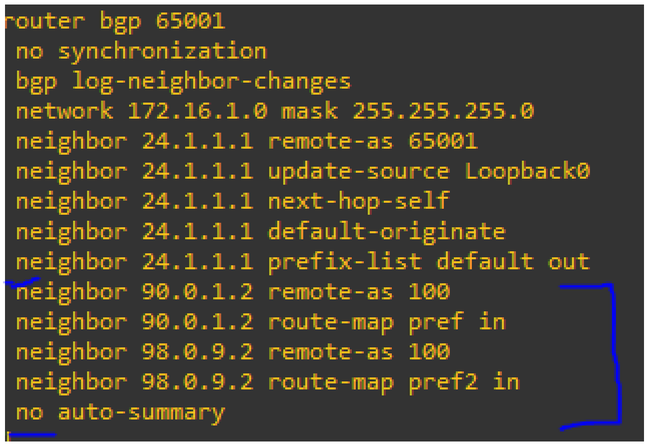
Redundancy

BGP provides network stability whereby ensuring that if a route path were to suddenly become unavailable, for whatever reason, it would quickly try to adapt to the situation by redirecting traffic through a new path or by the next best path set up by the system administrators based on attributes such a Weight or MED. As an example, within my topology, each hospital site has two border routers connecting to a different internet service provider via two links, looking at the top left site I will configure the router connecting to the green ISP with a local preference attribute (this being the second highest BGP path value with Weight being the highest priority). As shown, the router links are peering with the two edge router neighbours in AS100 providing two potential paths to that AS, however, I have configured the BGP protocol on Router18 to establish a primary and secondary route for leaving that network. Local preference routes are ranked based on the highest value set, so by applying a value of 200 to neighbour Router6 from interface g2/0 it becomes the primary route for traffic to be sent through from AS65001 with the next best route being that of Router23 with a value of 190 in the event of a router failure.





```
!
ip prefix-list default seq 10 permit 0.0.0.0/0
no cdp log mismatch duplex
!
!
!
!
route-map pref permit 10
 set local-preference 200
!
route-map pref permit 200
!
route-map pref2 permit 10
 set local-preference 190
!
!
!
```



```
router bgp 65001
 no synchronization
 bgp log-neighbor-changes
 network 172.16.1.0 mask 255.255.255.0
 neighbor 24.1.1.1 remote-as 65001
 neighbor 24.1.1.1 update-source Loopback0
 neighbor 24.1.1.1 next-hop-self
 neighbor 24.1.1.1 default-originate
 neighbor 24.1.1.1 prefix-list default out
 neighbor 90.0.1.2 remote-as 100
 neighbor 90.0.1.2 route-map pref in
 neighbor 98.0.9.2 remote-as 100
 neighbor 98.0.9.2 route-map pref2 in
 no auto-summary
```

Learning Outcomes

From working on this project, it has dawned on me just how much more large and complex the creation and functionality of an organisations network truly is, from being able to establish connectivity between sites miles away from one another while simlutaniously being able to reroute traffic over networks in the event of route failure that could consist of hundreds of devices in seconds is nothing short of mind boggling. While doing simultaneous research it felt as though the material would go on forever as there so much to be done in so many ways. While I might not yet be anywhere near ready to develop a fully function WAN with fully configured fields, I do feel that the time I invested into this project will at the very least prepare me for what is to be expected within a organisational structure as well as the need and importance of core technologies such as BGP and MPLS.

Acknowledgements

I would like to personally thank my supervisor, James Egan, for his patience and guidance when meeting with me throughout the entire year.