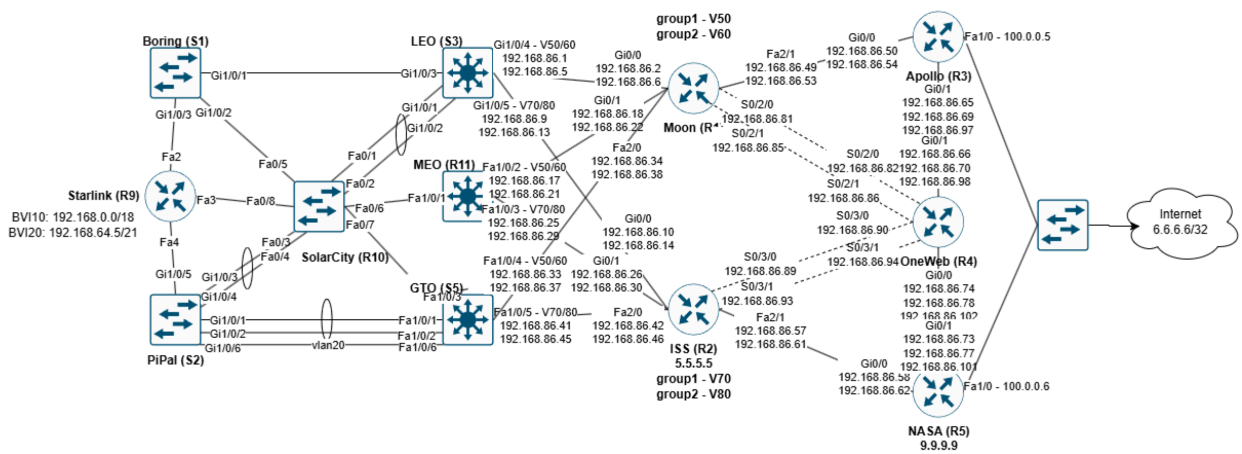
Midterm Practical Exam

Network Diagram:

The network diagram contains all interface/sub-interface/SVI IPs for the Earth area space. Also included are the BVIs for wireless. In total there are 4 main VLANs for SpaceX, Tesla, X.com, and Neuralink. Then coming off of LEO, MEO, GTO we have VLANS 50,60,70,80. Vlans 50 and 60 for SpaceX and Tesla outbound traffic and Vlans 70,80 for X.com and Neurolink outbound VRF traffic.

The ZIP file called “config_routes_stp.zip” contains configurations, route tables (L3 devices) and spanning tree info (MLS and L2 devices) for all respective devices.

Diagram:



Subnetting:

SpaceX

- VLAN10 - /18
- IP Range: 192.168.0.0 - 192.168.63.255
- Subnet Mask: 255.255.192.0
- Wildcard Mask: 0.0.63.255
- Reserved: 192.168.0.1-20

Tesla

- VLAN20 - /20
- IP Range: 192.168.64.0 - 192.168.79.255
- Subnet Mask: 255.255.240.0
- Wildcard Mask: 0.0.15.255
- Reserved: 192.168.64.1-20

X.com VRF

- VLAN30 - /22
- IP Range: 192.168.80.0 - 192.168.83.255
- Subnet Mask: 255.255.252.0
- Wildcard Mask: 0.0.3.255
- Reserved: 192.168.80.1-20

Neuralink VRF

- VLAN40 - /23
- IP Range: 192.168.84.0 - 192.168.85.255
- Subnet Mask: 255.255.254.0
- Wildcard Mask: 0.0.1.255
- Reserved: 192.168.74.1-100

DHCP

DHCP was simple. The configuration was similar between devices except for X.com from NASA DHCP. VRF ip dhcp was not fully working and the host could not grab an IP from that, so instead given that we have full connectivity via OneWeb, DHCP Discover goes to OneWeb and then to NASA and back to OneWeb to traverse VRFs and get DHCP assignment.

SpaceX from Starlink DHCP Config:

```
ip dhcp excluded-address 192.168.0.1 192.168.0.20
!
ip dhcp pool SpaceX
network 192.168.0.0 255.255.192.0
default-router 192.168.0.1
dns-server 5.5.5.5
!
```

X.com from NASA DHCP Config:

```
ip dhcp excluded-address 192.168.80.0 192.168.80.20
!
ip dhcp pool X.com
network 192.168.80.0 255.255.252.0
```

```
default-router 192.168.80.1
dns-server 5.5.5.5
!
```

Tesla and Neuralink from GTO:

```
ip dhcp excluded-address 192.168.64.1 192.168.64.20
ip dhcp pool Tesla
network 192.168.64.0 255.255.240.0
default-router 192.168.64.1
dns-server 5.5.5.5
!
ip dhcp excluded-address 192.168.84.1 192.168.84.20
ip dhcp pool Neuralink
network 192.168.84.0 255.255.254.0
default-router 192.168.84.1
dns-server 5.5.5.5
!
```

HSRP

Tesla GTO -> MEO -> LEO Config:

GTO:

```
interface Vlan10
ip address 192.168.0.4 255.255.192.0
standby 10 ip 192.168.0.1
standby 10 preempt
!
```

MEO:

```
interface Vlan10
ip address 192.168.0.3 255.255.192.0
standby 10 ip 192.168.0.1
standby 10 priority 90
standby 10 preempt
!
```

LEO:

```
interface Vlan10
ip address 192.168.0.2 255.255.192.0
standby 10 ip 192.168.0.1
standby 10 priority 110
standby 10 preempt
!
```

SpaceX LEO -> GTO -> MEO Config:

LEO:

```
interface Vlan20
ip address 192.168.64.2 255.255.240.0
standby 20 ip 192.168.64.1
```

```
standby 20 priority 90
standby 20 preempt
!
```

GTO:

```
interface Vlan20
ip address 192.168.64.4 255.255.240.0
standby 20 ip 192.168.64.1
standby 20 priority 110
standby 20 preempt
!
```

MEO:

```
interface Vlan20
ip address 192.168.64.3 255.255.240.0
standby 20 ip 192.168.64.1
standby 20 preempt
!
```

X.com MEO -> LEO -> GTO Config:

LEO:

```
interface Vlan30
ip address 192.168.80.2 255.255.252.0
standby 20 ip 192.168.80.1
standby 20 preempt
ip vrf forwarding group2
!
```

GTO:

```
interface Vlan30
ip address 192.168.80.4 255.255.252.0
standby 20 ip 192.168.80.1
standby 20 priority 90
standby 20 preempt
ip vrf forwarding group2
!
```

MEO:

```
interface Vlan30
ip address 192.168.80.3 255.255.252.0
standby 20 ip 192.168.80.1
standby 20 priority 110
standby 20 preempt
ip vrf forwarding group2
!
```

Nuerolink MEO -> GTO -> LEO Config:

MEO:

```
interface Vlan40
ip address 192.168.84.3 255.255.254.0
standby 20 ip 192.168.84.1
```

```
standby 20 priority 110
standby 20 preempt
ip vrf forwarding group2
!
```

GTO:

```
interface Vlan40
ip address 192.168.84.4 255.255.254.0
standby 20 ip 192.168.84.1
standby 20 preempt
ip vrf forwarding group2
!
```

LEO:

```
interface Vlan40
ip address 192.168.84.2 255.255.254.0
standby 20 ip 192.168.84.1
standby 20 priority 90
standby 20 preempt
ip vrf forwarding group2
!
```

Wireless

For wireless, I configured both Logan_Tesla and Logan_SpaceX networks. Logan_SpaceX is encrypted with a password: 'password'. As per the instructions, these wireless connections cannot directly talk with each other due to both networks being on different radios and going through separate BVIs to achieve inter VLAN connectivity.

Logan_Tesla config:

```
dot11 ssid Logan_Tesla
vlan 20
authentication open
guest-mode
!
interface Dot11Radio1
ssid Logan_Tesla
!
!
interface Dot11Radio1.20
encapsulation dot1Q 20
bridge-group 20
!
interface Vlan20
no ip address
bridge-group 20
!
ip address 192.168.64.5 255.255.240.0
!
```

Logan_SpaceX config:

```
dot11 ssid Logan_SpaceX
vlan 10
authentication open
authentication key-management wpa
guest-mode
wpa-psk ascii 0 password
!
interface Dot11Radio0
encryption vlan 10 mode ciphers aes-ccm
!
ssid Logan_SpaceX
!
interface Dot11Radio0.10
encapsulation dot1Q 10
bridge-group 10
!
interface Vlan10
no ip address
bridge-group 10
!
interface BVI10
ip address 192.168.0.5 255.255.192.0
!
```

Name Resolution from ISS

EtherChannel (PagP, LACP)

We were asked to put special connections between various places in the Mars Area Space. These special connections are EtherChannel, being both PagP and LACP.

LEO <-> SolarCity (LACP)

LEO:

```
interface Port-channel1
switchport trunk encapsulation dot1q
switchport mode trunk
!
interface GigabitEthernet1/0/1
switchport trunk encapsulation dot1q
switchport mode trunk
channel-group 1 mode active
!
interface GigabitEthernet1/0/2
switchport trunk encapsulation dot1q
switchport mode trunk
channel-group 1 mode active
!
```

SolarCity:

```
interface FastEthernet0/1
switchport mode trunk
channel-group 1 mode active
!
interface FastEthernet0/2
switchport mode trunk
channel-group 1 mode active
!
interface Port-channel1
switchport mode trunk
!
```

PiPal <-> SolarCity (PagP)

SolarCity:

```
interface FastEthernet0/3
switchport mode trunk
channel-group 2 mode auto
!
interface FastEthernet0/4
switchport mode trunk
channel-group 2 mode auto
!
interface Port-channel2
switchport mode trunk
bandwidth 500000
!
```

PiPal:

```
interface GigabitEthernet1/0/3
switchport trunk encapsulation dot1q
switchport mode trunk
channel-group 2 mode desirable
!
interface GigabitEthernet1/0/4
switchport trunk encapsulation dot1q
switchport mode trunk
channel-group 2 mode desirable
!
interface Port-channel2
switchport trunk encapsulation dot1q
switchport mode trunk
bandwidth 500000
!
```

PiPal <-> GTO (LACP)

GTO:

```
interface FastEthernet0/1
switchport mode trunk
channel-group 1 mode active
!
```

```
interface FastEthernet0/2
switchport mode trunk
channel-group 1 mode active
!
interface Port-channel1
switchport mode trunk
bandwidth 500000
!
```

PiPal

```
interface GigabitEthernet1/0/1
switchport trunk encapsulation dot1q
switchport mode trunk
channel-group 1 mode active
!
interface GigabitEthernet1/0/2
switchport trunk encapsulation dot1q
switchport mode trunk
channel-group 1 mode active
!
interface Port-channel1
switchport trunk encapsulation dot1q
switchport mode trunk
bandwidth 500000
!
```

As a note, all other ports on the Mars Area Space that arent set for LACP or PagP are set to trunk ports.

Port-Security

Port security was configured on Boring for all ports that were not in use. I used mac-address sticky to track for only one user and made a timeout interval of 60 sec.

```
int rangeGi1/0/11-52
switchport port-security maximum 1
switchport port-security mac-address sticky
exit
errdisable recovery cause psecure-violation
errdisable recovery interval 60
```

STP Paths

For STP paths, I influenced paths with the priority and increased the priority by a factor of 4096:

SpaceX: Boring - Starlink - Pipal – GTO

```
Boring:
spanning-tree vlan 10 root primary
Starlink:
```

```
spanning-tree vlan 10 root secondary
PiPal:
spanning-tree vlan 10 priority 32768
GTO:
spanning-tree vlan 10 priority 36864
```

Tesla: Pipal - Starlink - Solarcity - Boring – LEO

```
PiPal
spanning-tree vlan 20 root primary
Starlink:
spanning-tree vlan 20 root secondary
SolarCity:
spanning-tree vlan 20 priority 32768
Boring:
spanning-tree vlan 20 priority 36864
LEO:
spanning-tree vlan 20 priority 40960
```

There is a slight problem for this, because SolarCity is more desirable than Boring is from LEO, we must set the priority lower and increase the port cost for the Port-Channel, so it takes the Gi1/0/3 interface instead. This config will remain similar on other connections that share two bridges that have a lower priority but is not the next bridge in the path, instead is the other bridge:

```
interface Port-channel1
switchport trunk encapsulation dot1q
switchport mode trunk
spanning-tree port-priority 112
spanning-tree cost 100
!
```

I also had to change priorities on SolarCity to prefer Starlink over PiPal because that is the path you wanted, but traditionally due to port speed, connections, and priority would take the path straight to PiPal instead.

X.com GTO - Pipal - Starlink - Boring – LEO

```
PiPal:
spanning-tree vlan 30 root primary
Starlink:
spanning-tree vlan 30 root secondary
Boring
spanning-tree vlan 30 priority 32768
LEO:
spanning-tree vlan 30 priority 36864
```

Neurolink MEO - Solarcity - Pipal - Starlink - Boring – LEO

```
MEO:
spanning-tree vlan 40 root primary
SolarCity:
spanning-tree vlan 40 root secondary
PiPal
```

```
spanning-tree vlan 40 priority 32768
Starlink:
spanning-tree vlan 40 priority 36864
Boring:
spanning-tree vlan 40 priority 40960
LEO:
spanning-tree vlan 40 priority 45056
```

Costs were changed at LEO, Boring and Starlink to follow wanted path as opposed to best physical path.

DNS:

DNS was simple, give a domain-name and make sure there is connectivity to the given address. This was hosted on the ISS, and when elon.com was pinged, the Earth switch would receive ping requests from said user.

```
ip domain name iss.com
ip host ELON 77.77.77.77
ip host elon.com 6.6.6.6
```

Routing:

VRFs:

There was a global routing table, and a VRF that was named ‘group2’ that contained both X.com and Neurolink. The VLANS that were encompassed on the Mars Area Space for VRF group2 were VLANS 30,40,70,80 while global table had: 10,20,50,60.

All layer 3 enabled devices had to have the command: ‘ip vrf group2’ to enable the VRF.

Example configuration for MLS VRF:

```
interface Vlan30
ip vrf forwarding group2
ip address 192.168.80.3 255.255.252.0
ip helper-address 9.9.9.9
standby 30 ip 192.168.80.1
standby 30 priority 110
standby 30 preempt
standby 30 track 1 decrement 20
standby 30 track 2 decrement 20
!
```

Example configuration for Router VRF:

```
interface GigabitEthernet0/0.60
encapsulation dot1Q 60
ip vrf forwarding group2
ip address 192.168.86.6 255.255.255.252
!
```

Each router and MLS had two personalities with two different routing tables and could only communicate via One-Web.

EIGRP

EIGRP was a simple configuration, here is a sample configuration for both MLS and Router:

MLS:

```
router eigrp 1
!
address-family ipv4 vrf group2 autonomous-system 1
  network 192.168.80.0 0.0.3.255
  network 192.168.84.0 0.0.1.255
  network 192.168.86.0
  passive-interface Vlan30
  passive-interface Vlan40
exit-address-family
!
```

Router:

```
router eigrp 1
!
address-family ipv4 vrf group2 autonomous-system 1
  network 192.168.86.0
exit-address-family
!
```

To explain the config, EIGRP was chosen to be on the VRF and required a special command to allow for routing to happen via the VRF. Also, shown on the MLS we can see that passive-interface was configured on internal VLAN customers. This was done so that HSRP could take care of redundancy on the L2 side and there would not be any L3 leakage into that area. From these configurations, they remain the same throughout the respective device (MLS or Router). One-Web configuration is a bit different but will be explained in the redistribution section.

For traffic engineering with EIGRP, a delay was set on the *outbound* interface/SVI to control the flow of traffic. Here on MEO we can see that VLAN50 is desirable as next hop over VLAN 60 due to the 'delay' command:

```
interface Vlan50
ip address 192.168.86.17 255.255.255.252
ip ospf cost 1
!
interface Vlan60
ip vrf forwarding group2
ip address 192.168.86.21 255.255.255.252
delay 10000
!
```

OSPF

OSPF configuration was also fairly simple. OSPF existed on the global routing table and served VLANs 10 and 20 with MLS facing outbound traffic passing over VLANs 50,60. Here is another example configuration for both MLS/Router:

MLS:

```
router ospf 1
router-id 2.2.2.2
log-adjacency-changes
passive-interface Vlan10
passive-interface Vlan20
network 2.2.2.2 0.0.0.0 area 0
network 192.168.0.0 0.0.63.255 area 0
network 192.168.64.0 0.0.15.255 area 0
network 192.168.86.0 0.0.0.255 area 0
!
```

Router:

```
router ospf 1
router-id 4.4.4.4
network 4.4.4.4 0.0.0.0 area 0
network 192.168.86.0 0.0.0.255 area 0
!
```

We can see that the config is similar to EIGRP. First, we have network commands to advertise wanted networks, and also the passive-interface command on the MLS to prevent L3 leakage into the customer networks.

Traffic engineering was done by manipulating OSPF by configuring a high cost for un-desirable outbound traffic. Here is an example configuration showing Moon Router preferring the Serial link over FastEthernet:

```
interface FastEthernet2/1.10
encapsulation dot1Q 10
ip address 192.168.86.49 255.255.255.252
ip ospf cost 100
!
interface Serial0/2/0
ip address 192.168.86.81 255.255.255.252
ip ospf cost 1
no fair-queue
clock rate 2000000
!
```

Redistribution

Redistribution took place on One-Web. The initial thought/implementation was supposed have One-Web with a VRF and for a Serial cable connected to itself carry over the redistributed information as these old Cisco routers do not support redistribution of routing protocols from a VRF to a global network and vice versa within the device itself. Instead, I opted for one global network just on One-Web to make cabling simpler and configuration less complicated. While there are security concerns having both customers on the same table, I ultimately thought for the time being this would be the best/most simple implementation.

Redist. Configuration One-Web:

```
router eigrp 1
network 192.168.86.68 0.0.0.3
network 192.168.86.76 0.0.0.3
network 192.168.86.84 0.0.0.3
network 192.168.86.92 0.0.0.3
redistribute ospf 1 metric 1 1 1 1 1
!
router ospf 1
router-id 8.8.8.8
redistribute eigrp 1 subnets
network 8.8.8.8 0.0.0.0 area 0
network 192.168.86.64 0.0.0.3 area 0
network 192.168.86.72 0.0.0.3 area 0
network 192.168.86.80 0.0.0.3 area 0
network 192.168.86.88 0.0.0.3 area 0
!
```

Default Route

The default route was set in OSPF via default-information originate. This meant that all OSPF neighbors were told where to go for any information with a gateway of last resort, that being either NASA or Apollo.

```
default-information originate metric 1000 metric-type 1
ip route 0.0.0.0 0.0.0.0 100.0.0.100
```

Stateful NAT

Stateful NAT was configured with setting a primary, backup and a peer. This configuration changed the inside global IP to 30.0.0.0/29. This was so Elon Musk could reach the space he knew which was 30.0.0.0/29 without using the internal space for translation. PAT was used for actual IP translation with any IP being allowed to translate. Primary was set for Apollo and NASA was the backup.

Apollo Config:

```
ip nat Stateful id 10
 primary 192.168.86.97
  peer 192.168.86.101
  mapping-id 90
ip nat pool ELON_POOL 30.0.0.1 30.0.0.6 prefix-length 29
ip nat inside source list 1 pool ELON_POOL mapping-id 90 overload
```

NASA Config:

```
ip nat Stateful id 10
 backup 192.168.86.101
  peer 192.168.86.97
  mapping-id 90
ip nat pool ELON_POOL 30.0.0.1 30.0.0.6 netmask 255.255.255.240
ip nat inside source list 1 pool ELON_POOL mapping-id 90 overload
```

Port Forwarding/SSH

For port forwading, on each SNAT router, ip nat was configured to include the device IP, the inside global IP, the protocol port number, and finally the port forwarded number in which an external user could access that device via SSH and port.

SNAT Routers config:

```
ip nat inside source static tcp 1.1.1.1 22 interface FastEthernet1/0 1001
ip nat inside source static tcp 2.2.2.2 22 30.0.0.1 1002 extendable
ip nat inside source static tcp 3.3.3.3 22 30.0.0.1 1003 extendable
ip nat inside source static tcp 4.4.4.4 22 30.0.0.1 1004 extendable
ip nat inside source static tcp 5.5.5.5 22 30.0.0.1 1005 extendable
ip nat inside source static tcp 7.7.7.7 22 30.0.0.1 1007 extendable
ip nat inside source static tcp 8.8.8.8 22 30.0.0.1 1008 extendable
ip nat inside source static tcp 9.9.9.9 22 30.0.0.1 1009 extendable
```

ISS Example SSH config:

```
ip domain name iss.com
ip ssh version 2
username admin privilege 15 secret 5 admin
crypto key generate rsa ----> 1024
line vty 0 4
login local
transport input ssh
line vty 5 15
login local
transport input ssh
!
```