**Last Update:** 4/1/2019

**By:**

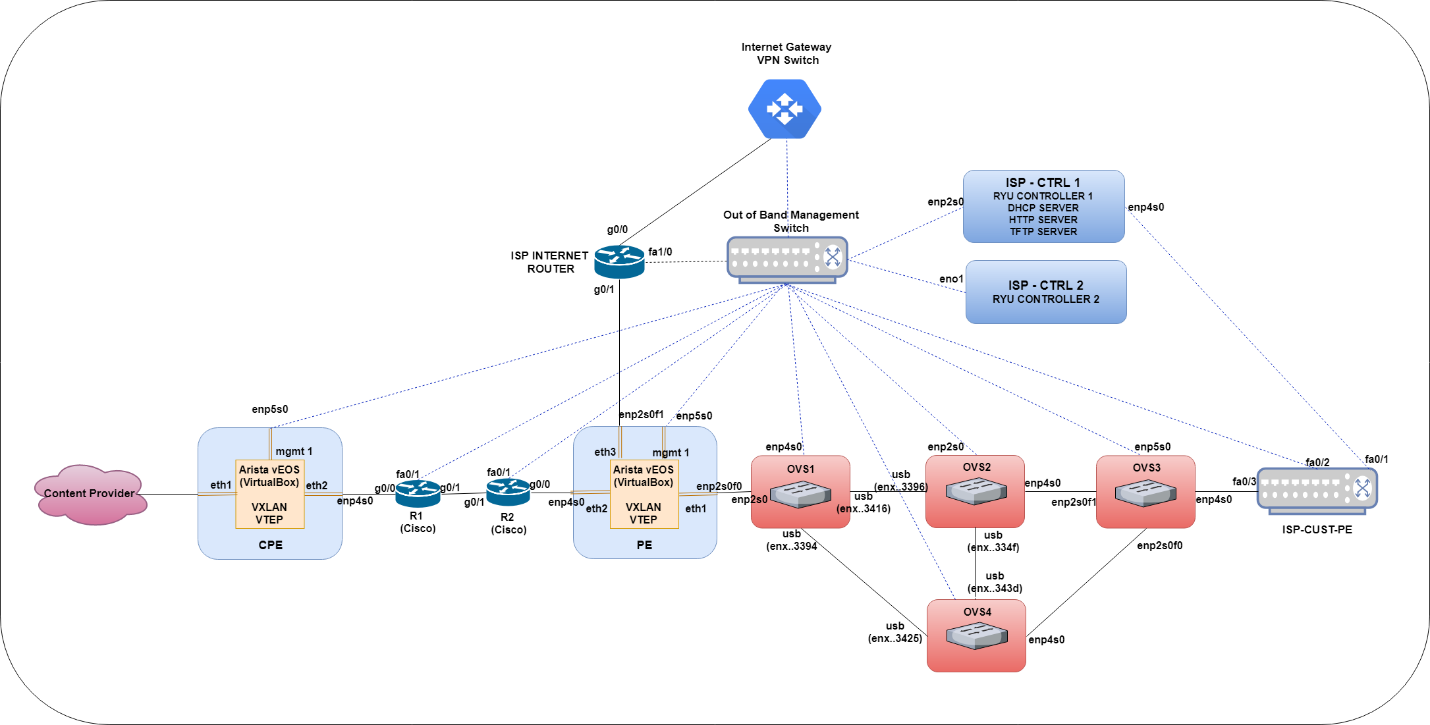
**Team:** ISP-Backbone

**Team Members:** Gaurav, Jom, Karthick, Srinidhi

**Manager:** Srinidhi

**Date:** 3/19/2019 (Week 1)

**Topology:**

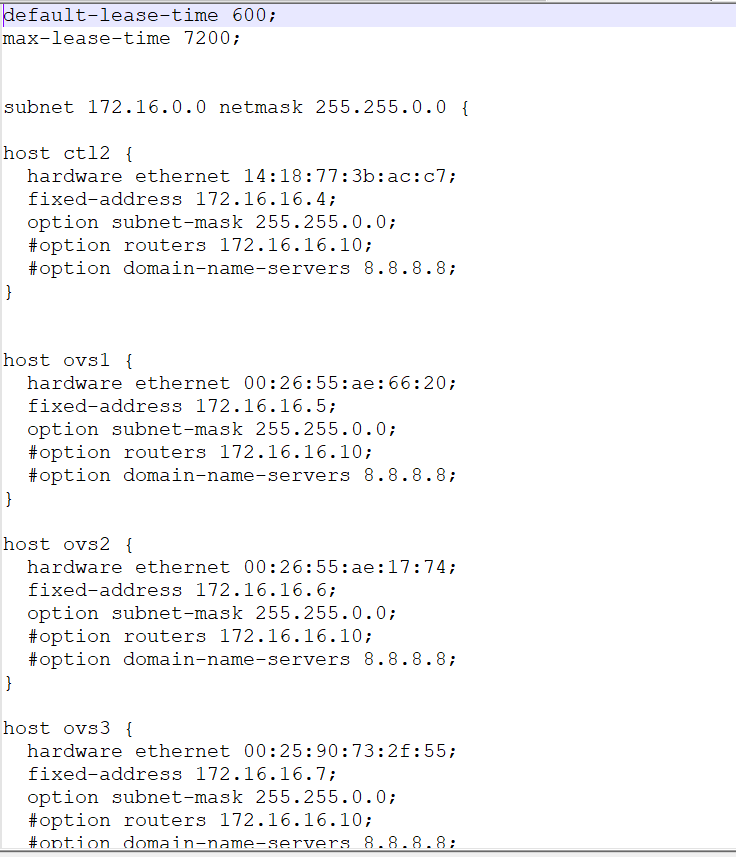


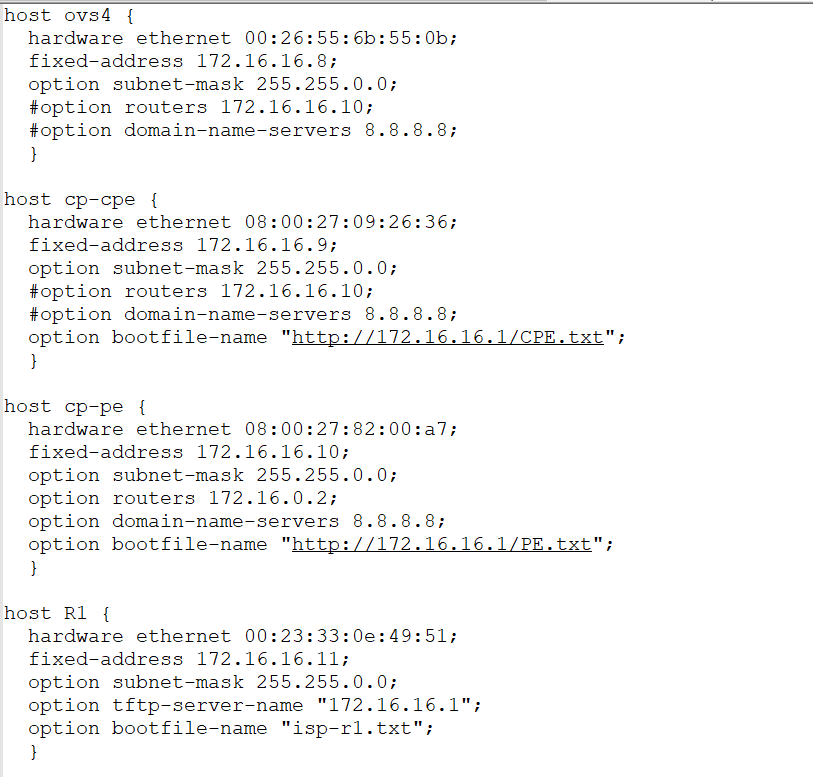
**Detailed Report:**

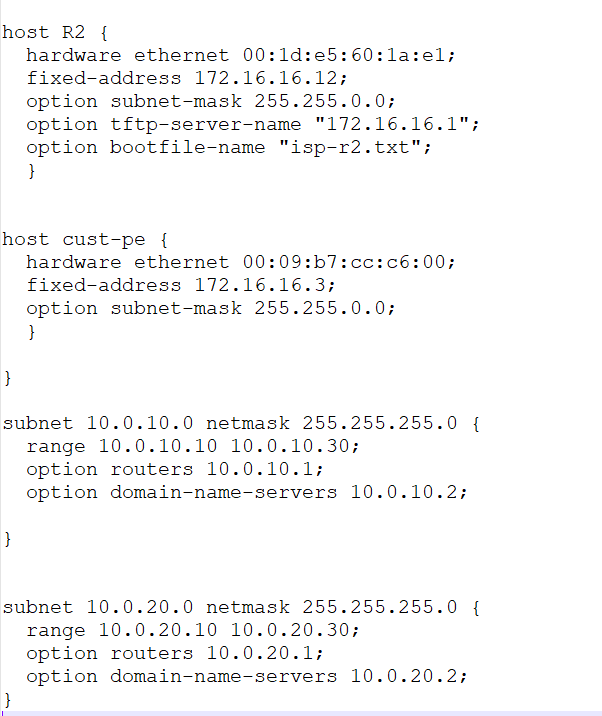
**Basic Goals that were completed:**

* Configured OOB management switch and ISP-CTRL-1 server manually. This is our network services VM which runs DHCP, TFTP, HTTP and the network applications for managing the OVSes. For backup we have ISP-CTRL-2.

Below is the DHCP configuration, where we map specific host mac addresses to a management IP that we provide using DHCP. This way the management IPs remain consistent, easing network management efforts:

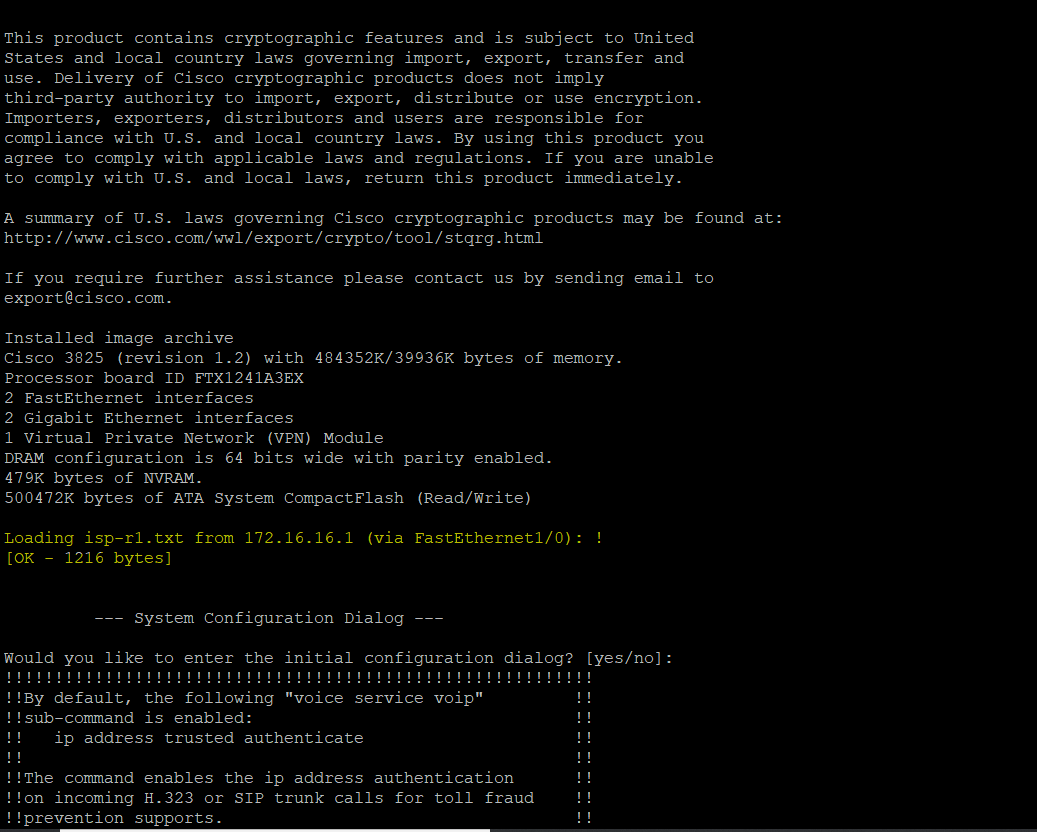




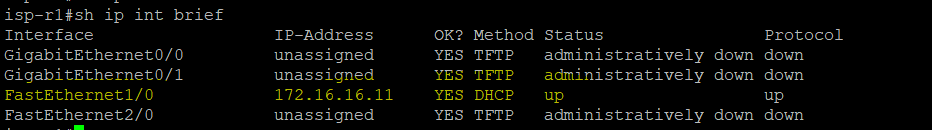


* Installed Ubuntu OS on all other Linux boxes. Pointed primary interface so they got a DHCP IP and installed SSH package along with the OS. Therefore, after the Ubuntu OS installation was complete, the Linux boxes had SSH capability.
* Carried out ZTP on CISCO routers. Upon bootup, the routers requested for DHCP, during which a TFTP server option was passed. The TFTP server option was realized on the ISP-CTRL-1. The routers then pulled their respective configurations.

CISCO routers obtaining DHCP IP and pulling configs from TFTP server:



Interfaces configured from the startup config pulled through TFTP:



**Stretch Goals that were completed:**

**VTEPs (CPE and PE):**

* For the VTEPs, we first installed a base Ubuntu OS on the physical server. We then loaded a vEOS Arista image onto a VM running on Virtualbox on these servers. The VM essentially works like an Arista box. The adapters to the VM is bridged so that any traffic arriving on the server is forwarded to the VM where it is processed on an “Arista 4.21.0F” MLS box image, which supports VXLAN encapsulation. We legally downloaded this image from the following link under software downloads tab:

<https://www.arista.com/en/support>

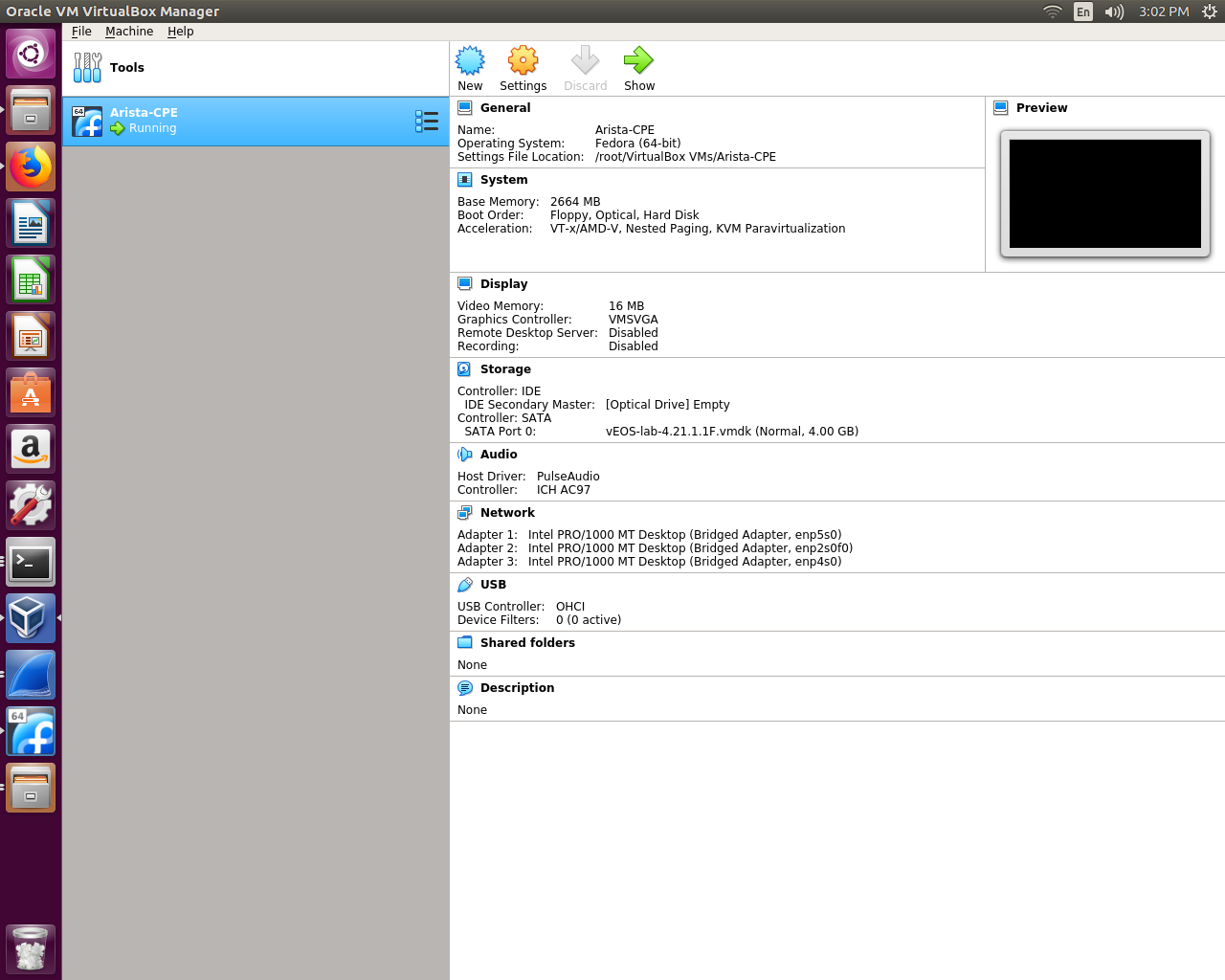
In addition, we also implemented ZTP on this Arista box by providing a HTTP bootfile option. The HTTP server also was running on ISP-CTRL-1 box. Below are screenshots of network settings on the Arista VM and ZTP running successfully:

Below are the extensive steps carried out on the VTEPs (CPE and PE)

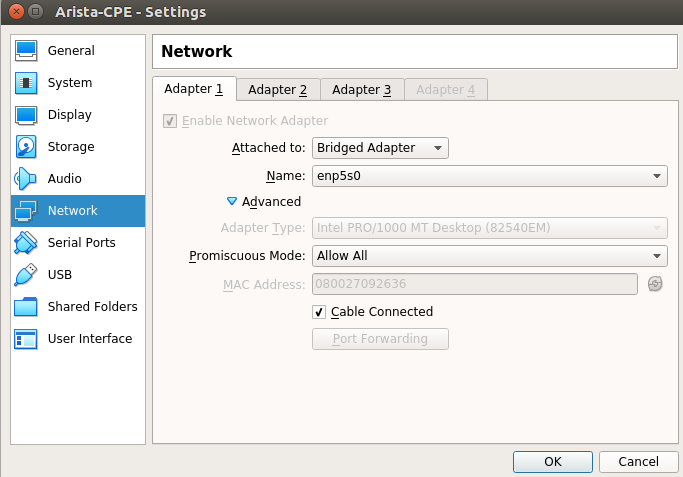
* Installing base Ubuntu.
* Setting up Linux and Virtual box networking,
* ZTP for Arista VMs.
* OSPF underlay connectivity.
* VXLAN overlay connectivity for test VLANs.

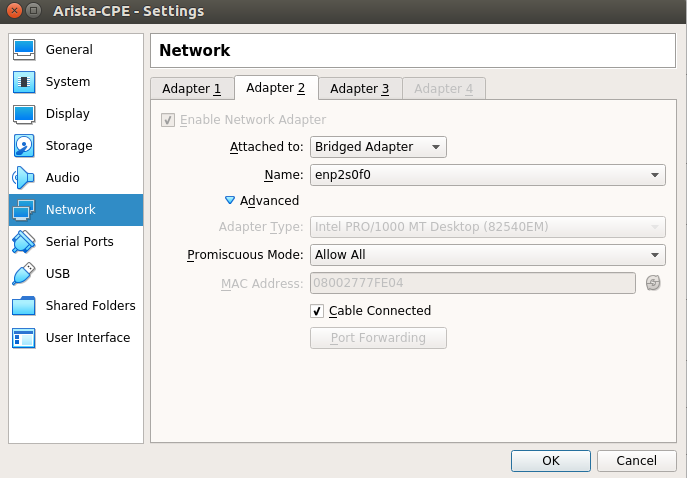
**Virtualbox and Linux Network Configuration:**

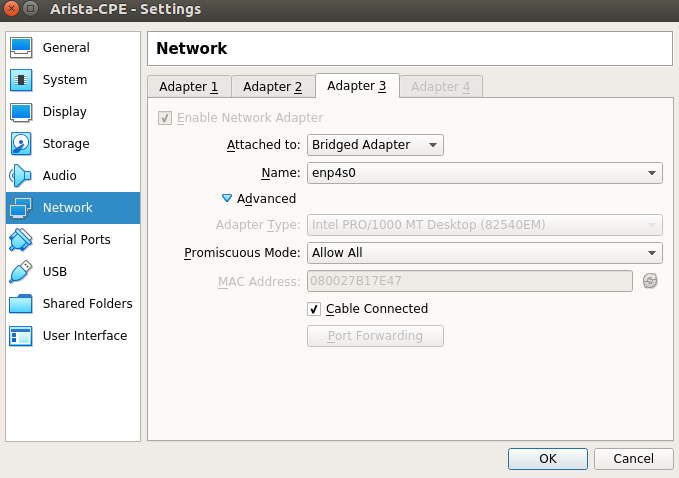
CPE on Virtualbox:



Network Settings for CPE on the CP side:



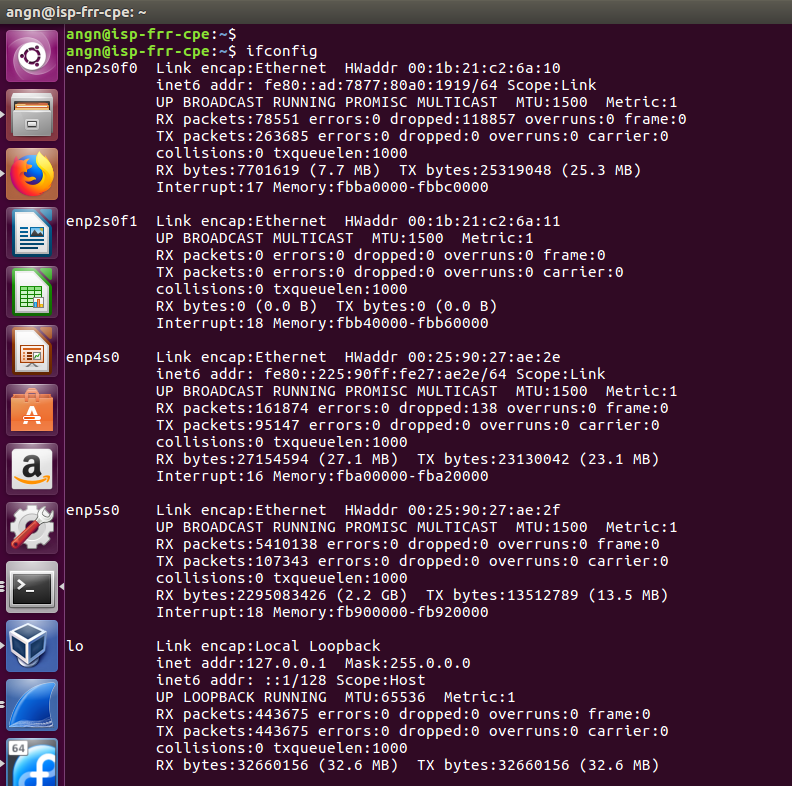




Note that the interfaces need to be bridged and set in promiscuous mode. The first network adapter is the management interface and the following network adapters are named Et1 and Et2, which can be mapped with respect to topology for connections and with respect to config files for the configurations.

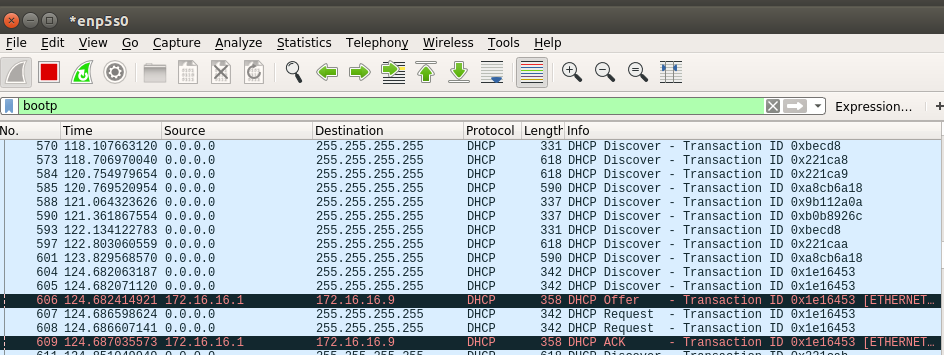
On the CPE Linux server, this is the interface configuration:

(Note PROMISC configuration indicating promiscuous mode is on. No IP on any of the interfaces because they are bridged to the VM which has the IP addresses configured)

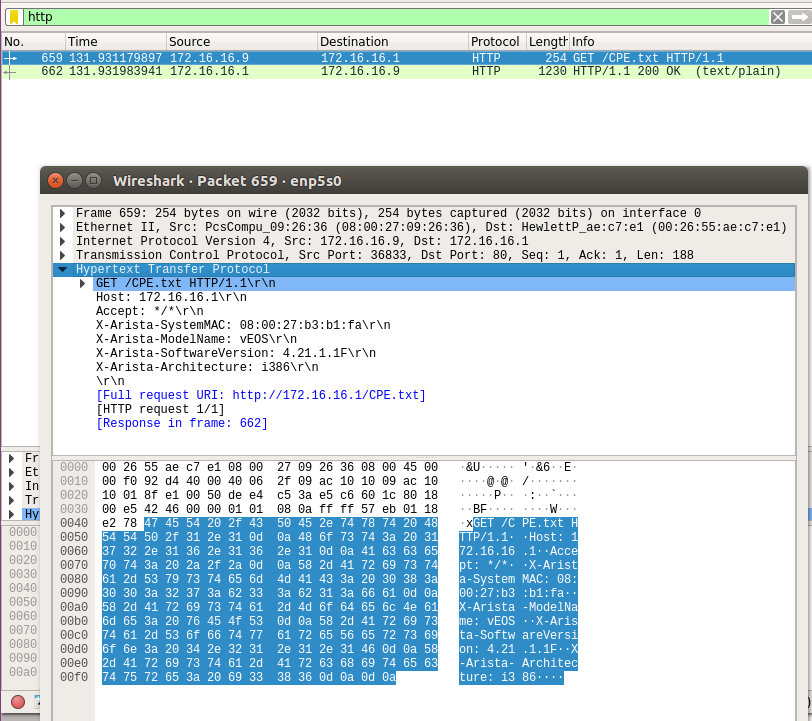


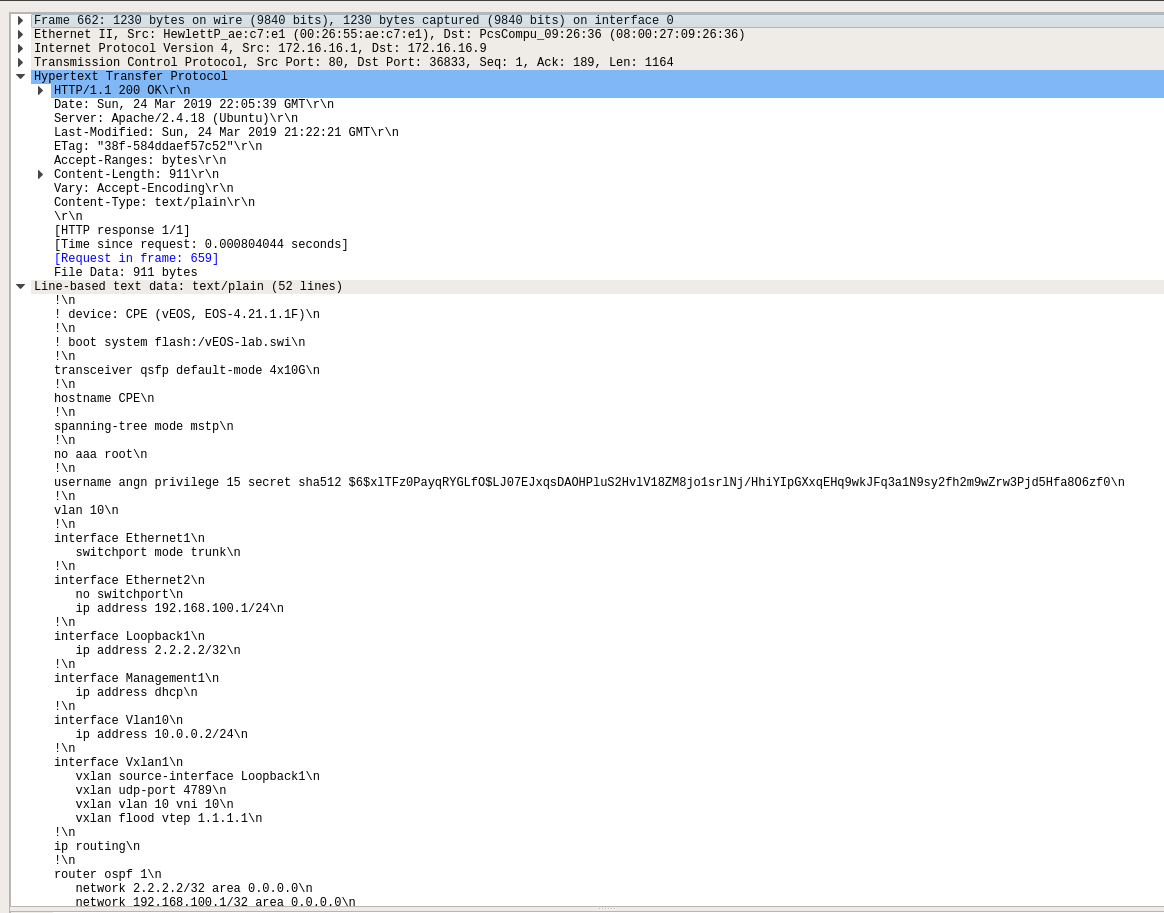
**ZTP**

ZTP through which the Arista CPE VM obtained its DHCP IP and startup configuration from our HTTP server:

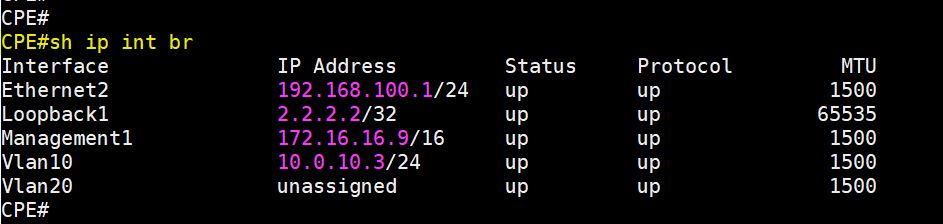
DHCP:

Pulling configuration from HTTP server



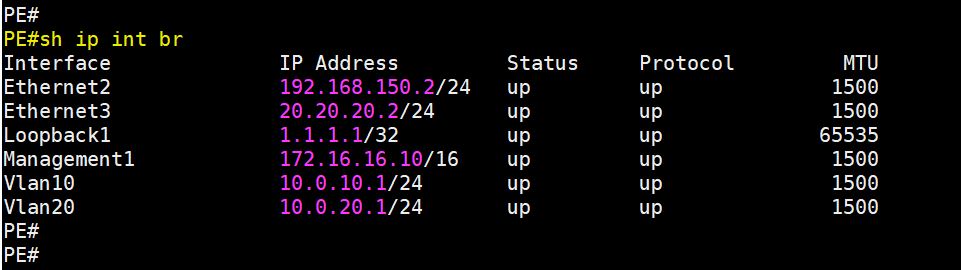


Configuration pulled from ZTP. Below are the interfaces configured upon startup on CPE:



Ethernet1 is a trunk interface.

Similar network settings on Virtualbox and ZTP implementation were carried out on PE. Here are the interfaces configured on PE (towards CP side):

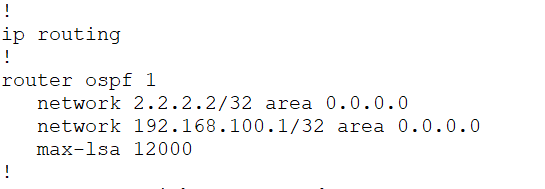


Notice interface vlan 10 and 20. These will be the default gateways for customers plugging into vlan 10 and vlan 20 respectively.

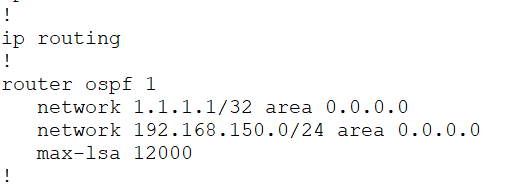
**Implementation of OSPF underlay between CPE and PE for underlay reachability:**

We then configure OSPF of the loopbacks (Loopback1) over which we will have our VXLAN overlay. Below are the OSPF configs on CPE and PE:

CPE:

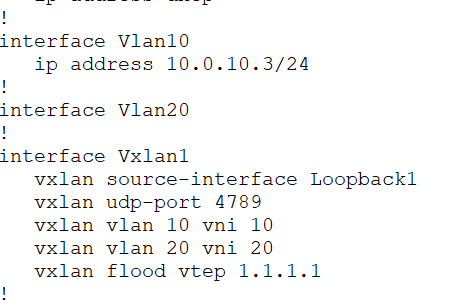


PE:

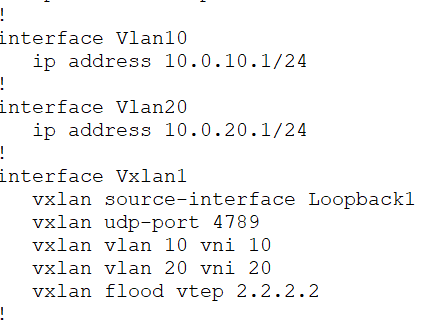


**Implementation of VXLAN overlay**:

CPE:



PE:

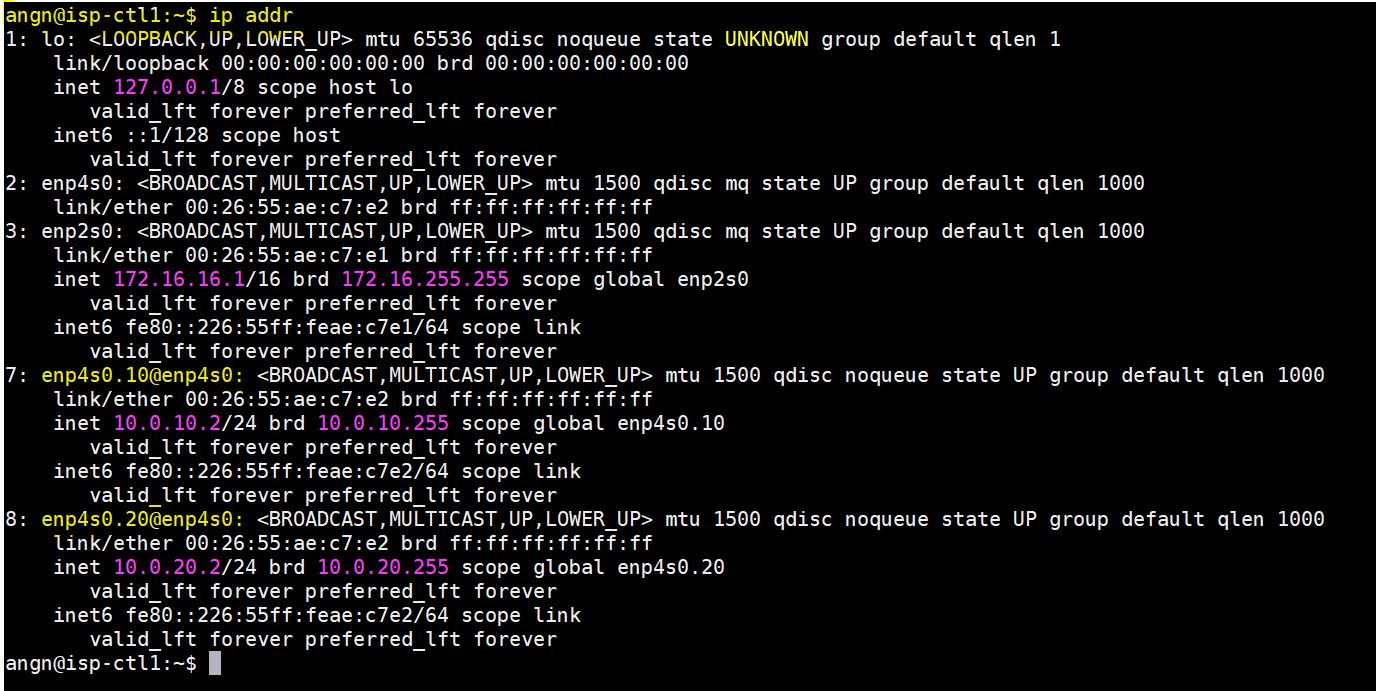


**Implementation of DHCP for a customer plugging into CPE (customer side):**

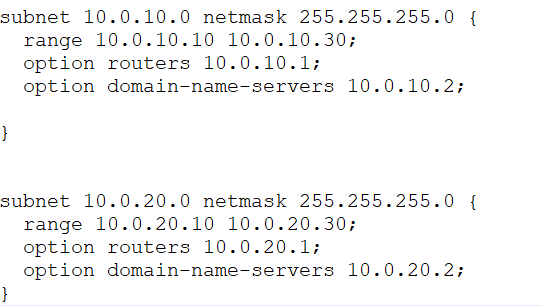
The customer side CPE is a CISCO catalyst switch with access ports in customer VLAN. For testing purposes, port fa0/10 was configured in VLAN 10 for our customer1. The customer plugs into this port. The switch also has a trunk port going to ISP-CTRL-1 which is a DHCP server. This Linux box has sub-interfaces in VLAN 10 and VLAN 20 (and for scaling, it must have a sub-interface in each customer VLAN)

Below is a screenshot of interfaces and sub-interfaces on ISP-CTRL-1, the DHCP server for customers plugging into CPE:

(Notice sub-interfaces enp4s0.10 and enp4s0.20. These interfaces have IP addresses in the customer 1 and customer 2 pools i.e. VLAN 10 and VLAN 20)



Whenever a DHCP request is obtained on these interfaces, its mapped to the appropriate DHCP pool as shown below. Therefore customer 1 in VLAN 10 gets an IP in subnet 10.0.10.0/24 and customer 2 in VLAN 20 gets an IP in subnet 10.0.20.0/24.

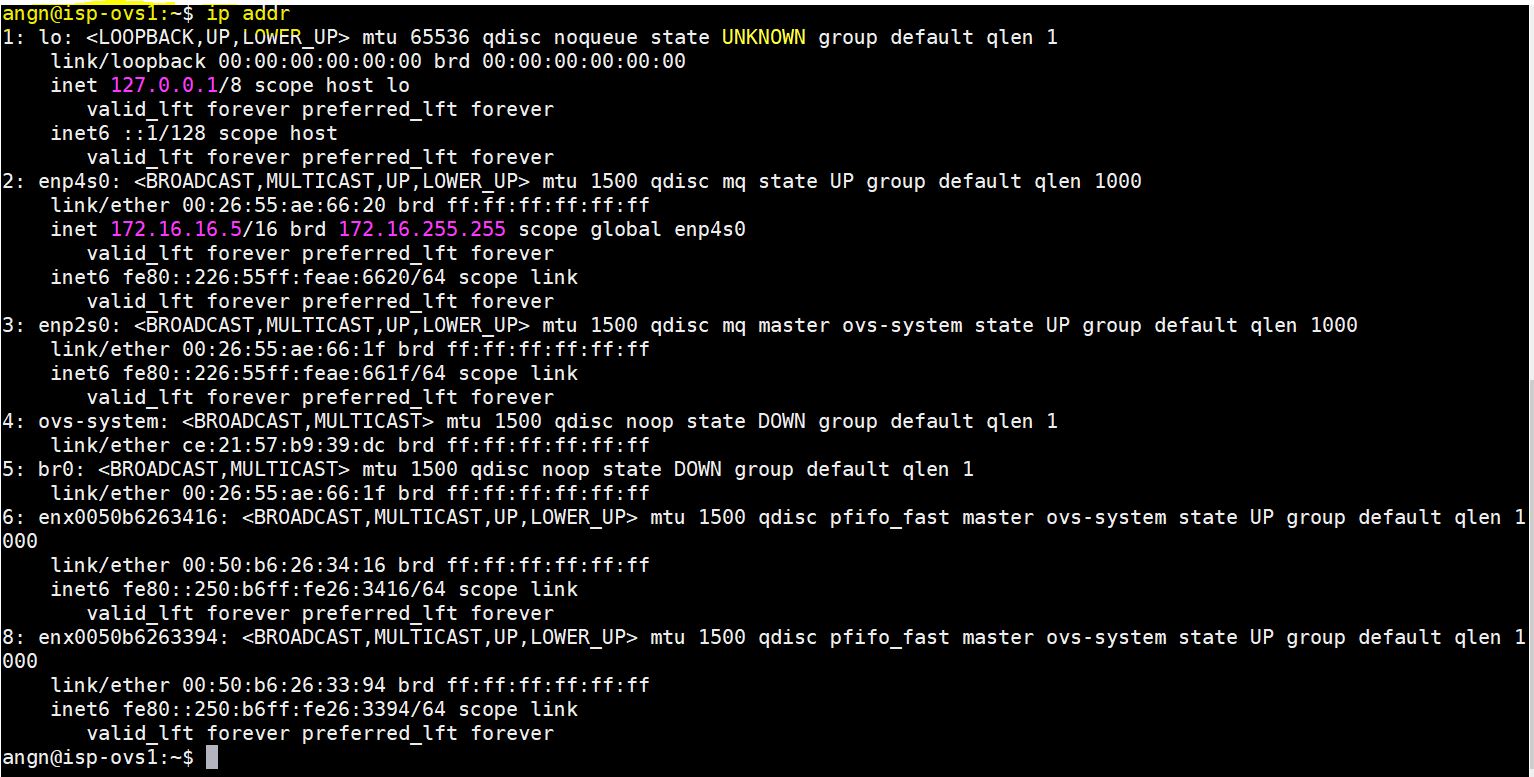


**Installation of Ryu and running simple\_switch\_13.py**

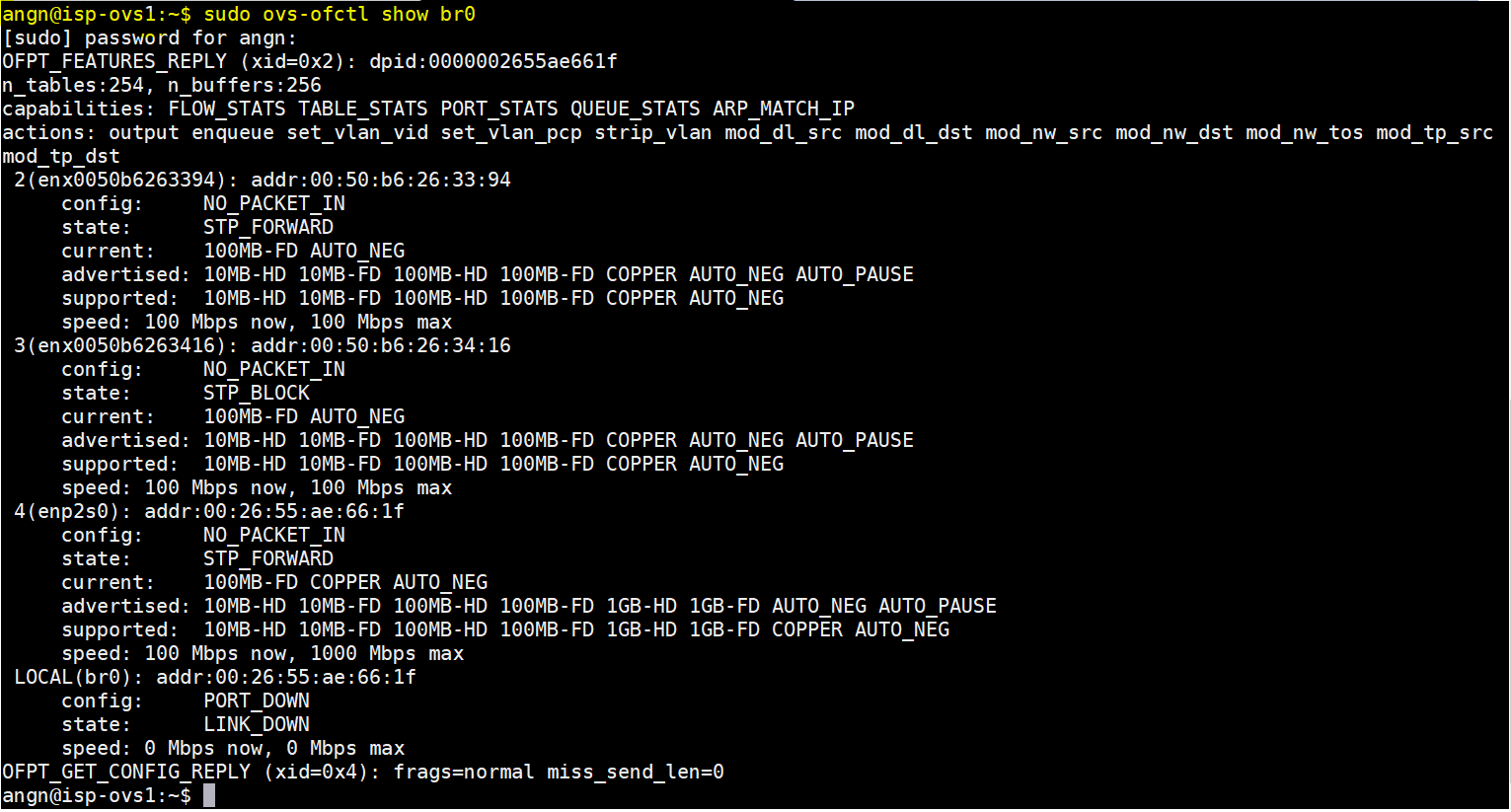
We installed RYU on the controller and configured all OVSes in the topology to connect to both controllers (primary and backup). We also enabled STP on the OVSes to prevent broadcast storms. Below are some screenshots pertaining to this:

OVS-1:

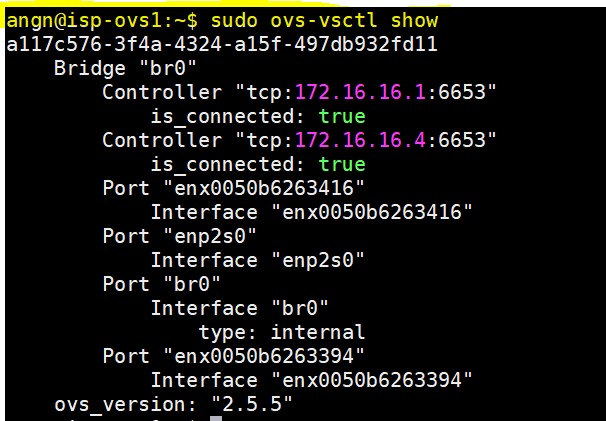
Interface config on Linux:



STP state:

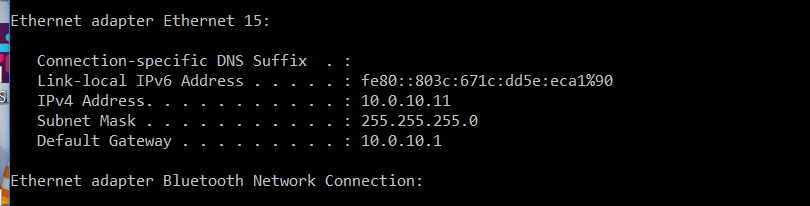


Connectivity to controller(s):



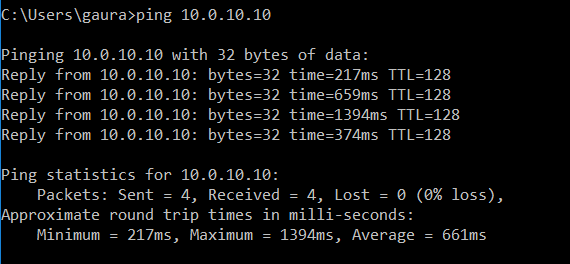
**Final Testing and Verification:**

When customer plugs in laptop to CPE port 10 (which is configured in VLAN 10), the laptop gets an IP in 10.0.10.0/24 network:

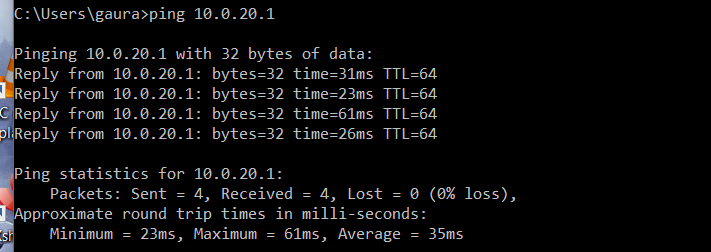


Beyond this point, the customer can:

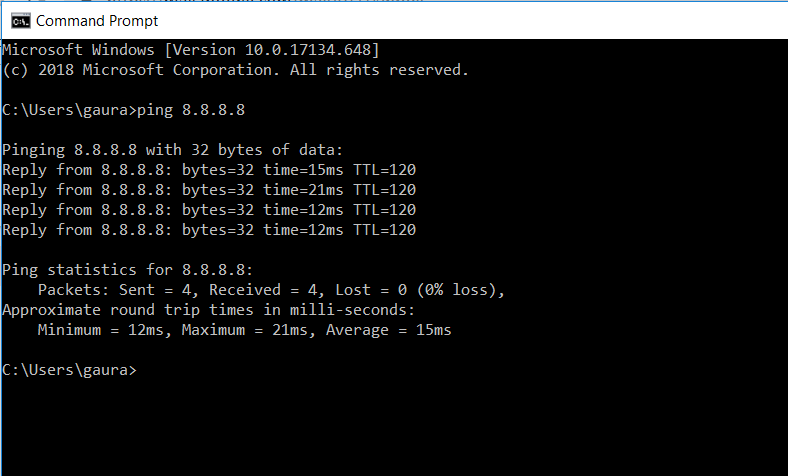
* Reach a host in the same subnet on the CP side in VLAN 10 (over the VXLAN tunnel configured):



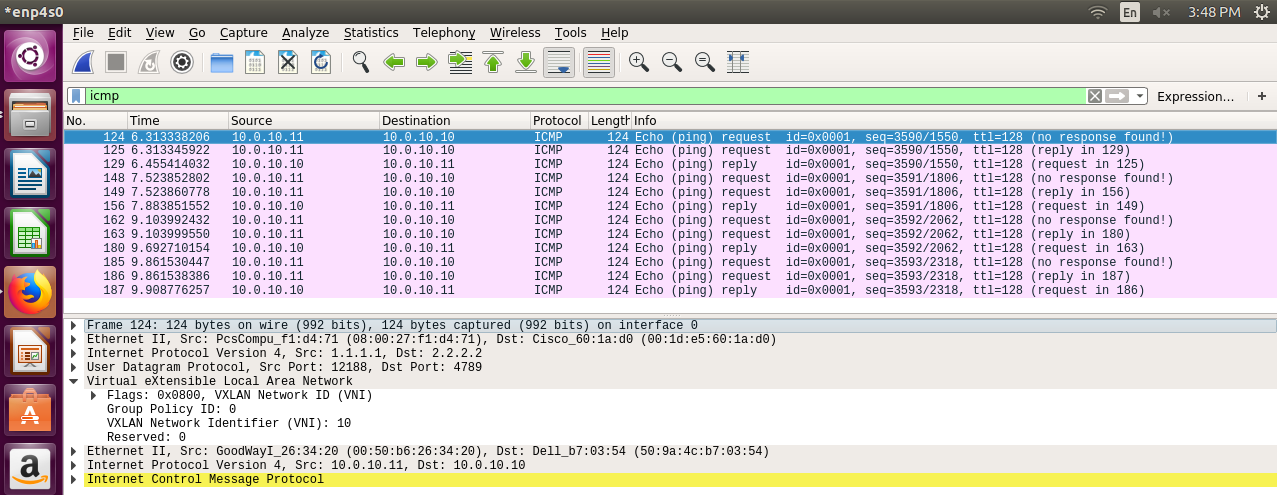
* Reach a host in another customer’s VLAN 20 (in 10.0.20.0/24) subnet:

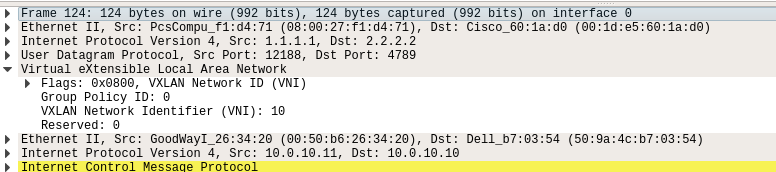


* Reach the Internet:



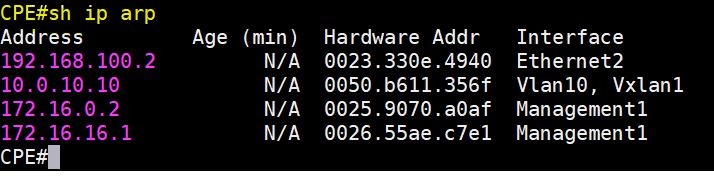
VXLAN encapsulation when traffic is sent over the tunnel:







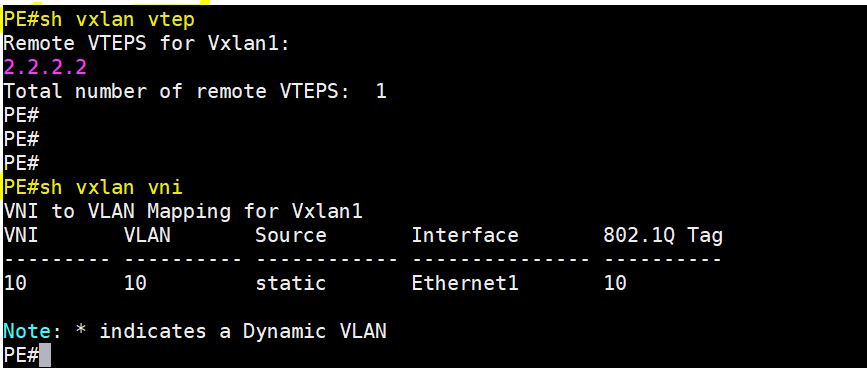
Note that on the CPE when it ARPs for a host in the same VLAN (on the CP side), the ARP gets completed over the VXLAN tunnel just like a normal L2 network even though its over an L3 underlay network. This reinforces the concept of VXLAN:

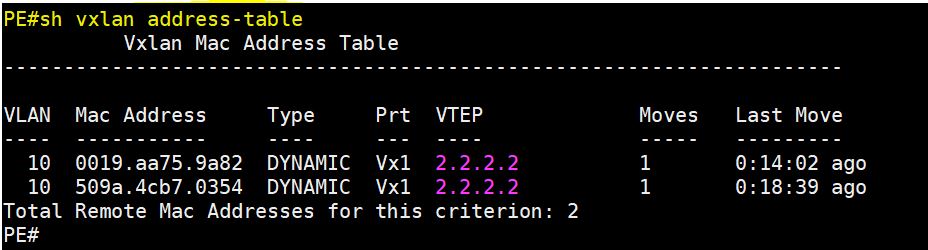




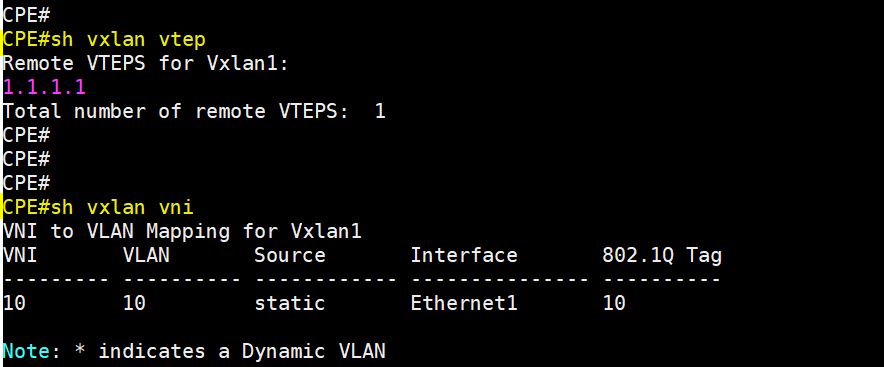
Miscellaneous VXLAN related information:

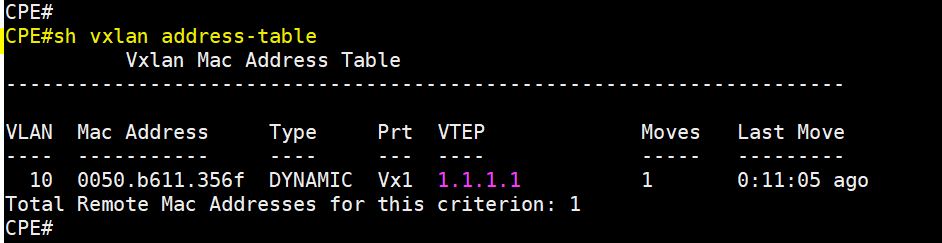
PE:





CPE:





\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Miscellaneous options considered:**

* Considered using FRR as VTEPs but rejected because it does not have MLS support.
* Considered containerizing of VTEPs using Arista cEOS image. Rejected the plan after team meeting where we agreed that for this case implementation complexity of automating container creation, bridge creation and inter-networking outweighed benefits.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Conclusion:**

* All basic goals including all stretch goals have been successfully accomplished.
* All connectivity scenarios over the ISP network have been achieved.

- Intra-VLAN: a customer accessing their own service on the CP side over the

VXLAN tunnel.

- Inter-VLAN: a customer accessing another customer’s service on the CP side.

- Internet: a customer just trying to access the Internet.

* Pending goals for ISP backbone:

- Automate the above scenarios dynamically for each new customer by adding

VLAN-VXLAN mapping and by adding new interface vlans and creating new

VLANs on the switches.

- Implement DNS.

- Implement shortest latency-based routing.

- Develop the REST interface for NSoT read/write.

- Achieve redundancies (controller and network services).

- TBD by future teams (if any other changes)