Building Internetworks (Manually)

Gates, Ozturk, & Shah
Cisc 7332X Section T6 – Local Area Network
Brooklyn College 2019

***Abstract*—This is a detailed review of our completion of the project "Building Internetworks (Manually)". The objective of this project was as follows,**

- **Practice planning multiple networks**
- **To connect networks to form an internetwork**
- **Implement the internetwork using a few Linux virtual machines**
- **To understand CIDR notation**
- **To plan networks by determining IPv4 address spaces including network number and network mask**
- **Assign Ipv4 addresses to Linux VMs on the networks using Linux commands**
- **To configure packet forwarding for Ipv4 on Linux VMs serving network gateways**
- **Diagnose and test the networks using ICMP (ping) and packet capturing (tcpdump)**

I. Project Description Overview

In this project we were tasked in establishing multiple virtual machines that we could then use for building internetworks. Our objectives include assigning IPv4 addresses to Linux VMs on the networks using Linux commands, to plan and implement routing tables, configure packet forwarding for IPv4 on the Linux VMs serving network gateways, and to diagnose and test the networks using ICMP (ping) and packet capturing (tcpdump).

II. Linux virtual machine settings

A. Virtual Machine 1

VM1 has two ethernet adapters, one in the NAT mode, and the other in the Internal Network mode and the name of the ethernet is ethernet1.

B. Virtual Machine 2

VM2 has three ethernet adapters and all are in the Internal Network mode and the three adapters are named ethernet1, ethernet2 and ethernet3, respectively.

C. Virtual Machine 3

VM3 has one ethernet adapter that is in the Internal Network mode and the name of the ethernet of the adapter is ethernet3.

D. Virtual Machine 4

VM4 has one ethernet adapter that is in the Internal Network mode and the name of the ethernet of the adapter is ethernet3.

III. Software

- Windows 10/MacOS as host operating systems
- Oracle VM VirtualBox version 4.2.16 and above running on the host
- Debian Linux 8.x as guest operating systems

Linux commands that were used in this experiment are **route, ip, ping, sysctl,** and, **tcpdump**.

Linux system configuration files that are of concern are **/etc/network/interfaces** and **/etc/sysctl.conf**

IV. Contents of config files

For each of the virtual machines, we went into the /etc/sysctl.conf files and added the following lines shown in *Figure 1*:

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Figure 1.

This allows the disabling of IPv6. In desiring a cleaner setup, configured the network adapters to be IPv4 only.

```
debian@dVMbase:~$ ping -c 1 172.23.3.1
PING 172.23.3.1 (172.23.3.1) 56(84) bytes of data.
64 bytes from 172.23.3.1: icmp_seq=1 ttl=64 time=0.782 ms

--- 172.23.3.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.782/0.782/0.782/0.000 ms
```

Figure 2.

Shown in *Figure 2.* is an example of a transverse betweenVM1 and VM4. As you can see, 1 packet is transmitted from VM1 and is received by VM4 with 0% packet loss.

Below is a routing table example of both VM1 and VM4

showing the IP addresses of each along with their gateway address, and netmask.

```
debian@dVMbase:~$ sudo route -n
[sudo] password for debian:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG    0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
172.20.136.224  172.22.199.62   255.255.255.248 UG    0      0        0 eth1
172.22.199.0    0.0.0.0         255.255.255.192 U     0      0        0 eth1
172.23.3.0      172.22.199.62   255.255.255.248 UG    0      0        0 eth1
```

Figure 3a. *VM 1 Routing Table*

```
debian@dVMbase:~$ sudo route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.20.136.224  172.23.3.4      255.255.255.248 UG    0      0        0 eth0
172.22.199.0    172.23.3.4      255.255.255.192 UG    0      0        0 eth0
172.23.3.0      0.0.0.0         255.255.255.192 U     0      0        0 eth0
```

Figure 3b. *VM 4 Routing Table*

V. WHAT BENEFITS DOES VLAN OFFER?

VLANs provide a number of advantages, such as ease of administration, confinement of broadcast domains, and reduced broadcast traffic.

VI. HOW WOULD WE BUILD THE NETWORK AUTOMATICALLY?

In this experiment we build internetworks manually, examining and editing routing tables as well as creating private IP addresses, but to automize the creation of networks we can call upon service providers who have pre-built public (or private) networks. (e.g Verizon, Google, etc.)