CS4150 Computer Networks Laboratory – Assignment 1

Subhash S – 111801042
Computer Science and Engineering, IIT Palakkad

1 Man Pages

1.1 arp

ARP stands for *Address Resolution Protocol.* This protocol is used to resolve IP address of a system to its MAC address and It works between the Link and Network Layer. This command can be used to manipulate the System’s ARP cache which is a collection of Address Resolution Protocol entries.

1.2 ifconfig

It stands for interface configuration and can be used to configure the kernel-resident network interfaces. During the OS boot, it is used to setup the network interfaces and after that it can be used for debugging network issues or tuning network configurations.

1.3 route

A *Routing Table* at a node is a map which decides where to forward the incoming traffic. This command can be used to work with the IP Routing tables. It is mainly used to set up static routes to specific hosts or networks via an interface

1.4 host

This command is used to perform DNS lookup operations. It can be used to find the domain name from the IP address and vice-versa. It also provides options to obtain more details about the IP addresses or domain names using appropriate flags.

1.5 ping

Packet Internet Groper (PING) is used test the reachability of a host and round-trip time for connecting to a host on an IP network. It uses ICMP to send an echo message to the specified host which returns an ICMP reply message if it is available.

1.6 tcpdump

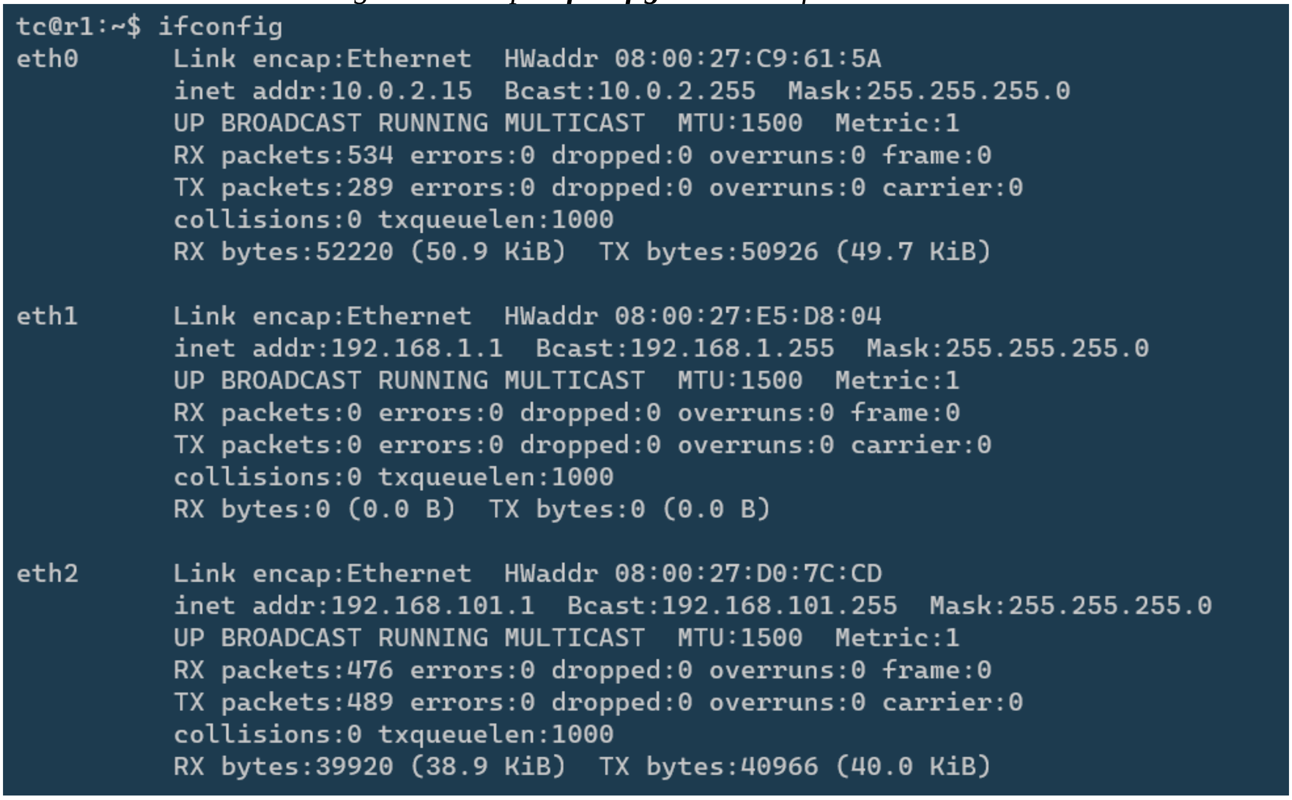
It is a packet sniffing and packet analyzing tool which can be used to analyze network traffic such as TCP/IP packets going through the system. It saves the information in a Packet Capture (pcap) file which can be used for further analysis.

1.7 netstat

It displays various network related information such as network connections, routing tables, interface statistics etc.

2 Setup

Figure 1: Sample ***ifconfig*** *command for router r1*



```
tc@r1:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:C9:61:5A
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:534 errors:0 dropped:0 overruns:0 frame:0
          TX packets:289 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:52220 (50.9 KiB)  TX bytes:50926 (49.7 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:E5:D8:04
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth2      Link encap:Ethernet  HWaddr 08:00:27:D0:7C:CD
          inet addr:192.168.101.1  Bcast:192.168.101.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:476 errors:0 dropped:0 overruns:0 frame:0
          TX packets:489 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:39920 (38.9 KiB)  TX bytes:40966 (40.0 KiB)
```

Running the **ifconfig** command lists all the interfaces along with other details such as *IP address, MAC address, Subnet mask and Broadcast address.* The IP address of the current *Virtual Machine* can be found in the

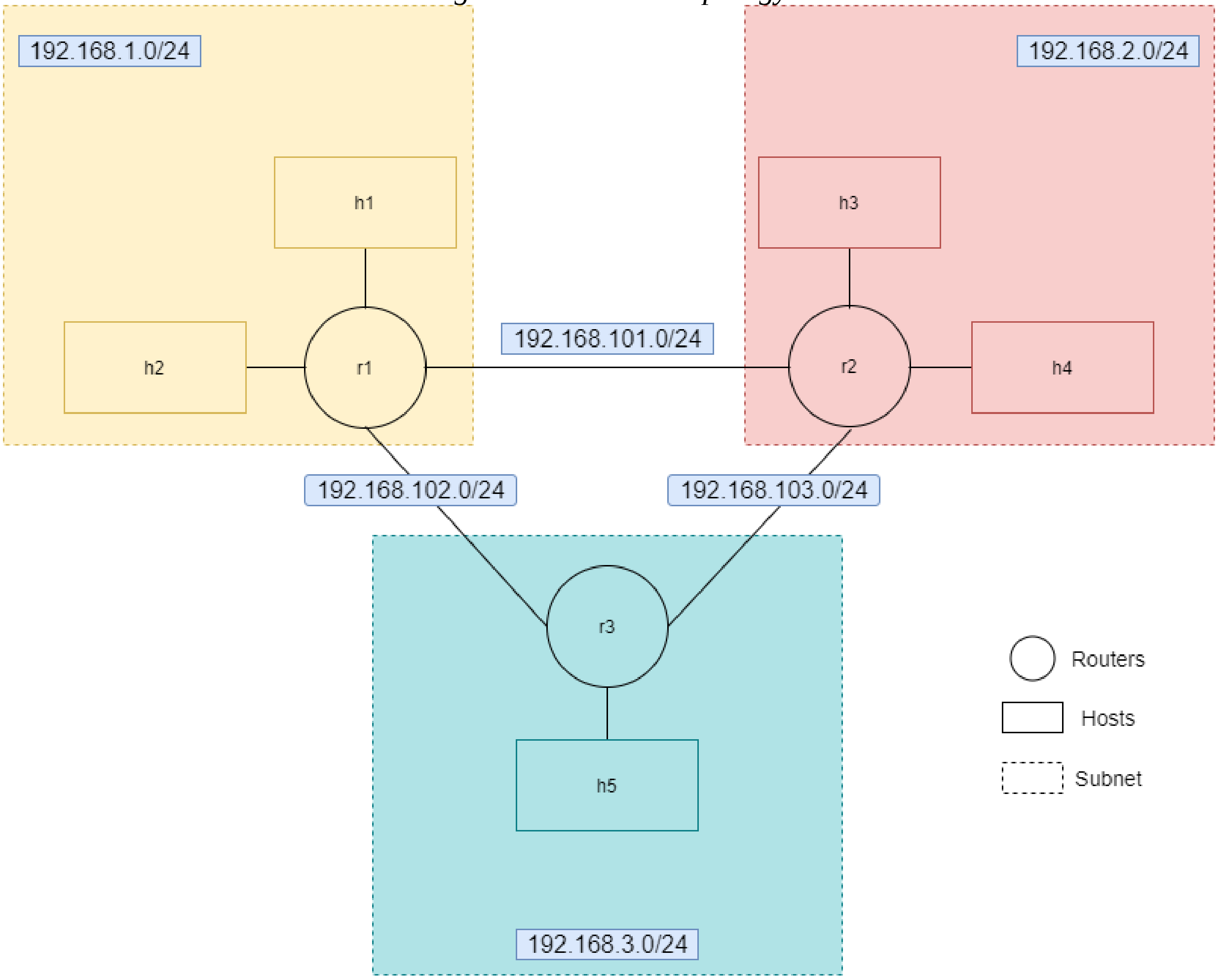
Running this command on all the machines gives the following IP addresses:

Table 1: IP address of each virtual machine

Virtual Machine	IP address (eth1)
r1	192.168.1.1
r2	192.168.2.1
r3	192.168.3.1
h1	192.168.1.2
h2	192.168.1.3
h3	192.168.2.2
h4	192.168.2.3
h5	192.168.3.2

3 Network Topology

Figure 2: Network Topology



- In the above figure, each colour represents a *subnet* and the blue coloured box represents the *address* assigned to that *subnet.*
- The *Mask* property in each interface has a value of 255.255.255.0, as shown in Figure 1. This means that the first 24 bits of each subnet is fixed.
- The network forms a ring topology due to the circular path formed by the routers *r1*, *r2 and r3.*
- There a total of 6 subnets:
  1. **192.168.1.0/24:** The hosts h1 and h2 are connected to the router r1.
  2. **192.168.2.0/24:** The hosts h3 and h4 are connected to the router r2.
  3. **192.168.3.0/24:** The host h5 is connected to the router r3.
  4. **192.168.101.0/24:** The routers r1 and r2 are connected.
  5. **192.168.102.0/24:** The routers r1 and r3 are connected.
  6. **192.168.103.0/24:** The routers r2 and r2 are connected.

Table 2: IP and MAC address of hosts (eth1)

Host	IP Address	MAC Address
h1	192.168.1.2	08:00:27:63:A5:D5
h2	192.168.1.3	08:00:27:FB:88:E4
h3	192.168.2.2	08:00:27:47:0D:B8
h4	192.168.2.3	08:00:27:7F:48:C9
h5	192.168.3.2	08:00:27:5D:FB:8B

Table 3: IP and MAC address of routers and their interfaces

Router	eth1	eth2	eth3
r1	192.168.1.1 08:00:27:E5:D8:04	192.168.101.1 08:00:27:D0:7C:CD	192.168.102.1 08:00:27:DB:3F:85
r2	192.168.2.1 08:00:27:03:03:21	192.168.101.2 08:00:27:A6:EF:5D	192.168.103.1 08:00:27:C4:F2:BE
r3	192.168.3.1 08:00:27:45:1B:1C	192.168.102.2 08:00:27:44:EE:79	192.168.103.2 08:00:27:C5:42:09

Each cell in Table 3 contains the IP address on the top and MAC address below it.

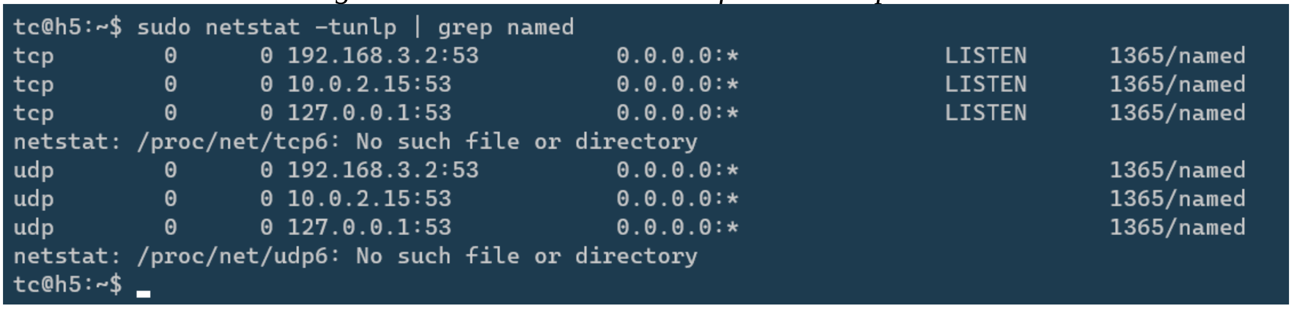
- Each host is connected to a router in its subnet by the interface *eth1*.
- Router *r1 (eth2)* is connected to router *r2 (eth2).*
- Router *r2 (eth3)* is connected to router *r3 (eth3).*
- *Router r1 (eth3) is connected to router r3 (eth2).*

4 Authoritative DNS Server

Yes, this network has an authoritative DNS server. The server is the host machine *h5* which is on the IP address *192.168.3.2* under the subnet *192.168.3.0/24.*

This can be confirmed from the fact that the host h5 runs a process ***named*** which is a Domain Name System (DNS) server as can be seen from Linux man pages.

Figure 3: Protocols and Ports of the named process.

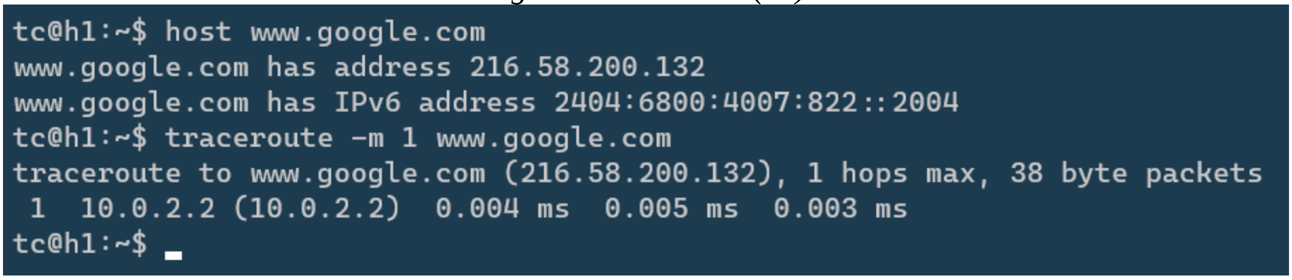


```
tc@h5:~$ sudo netstat -tunlp | grep named
tcp        0      0 192.168.3.2:53          0.0.0.0:*               LISTEN      1365/named
tcp        0      0 10.0.2.15:53            0.0.0.0:*               LISTEN      1365/named
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      1365/named
netstat: /proc/net/tcp6: No such file or directory
udp        0      0 192.168.3.2:53          0.0.0.0:*                           1365/named
udp        0      0 10.0.2.15:53            0.0.0.0:*                           1365/named
udp        0      0 127.0.0.1:53            0.0.0.0:*                           1365/named
netstat: /proc/net/udp6: No such file or directory
tc@h5:~$
```

5 Inferences on google.com

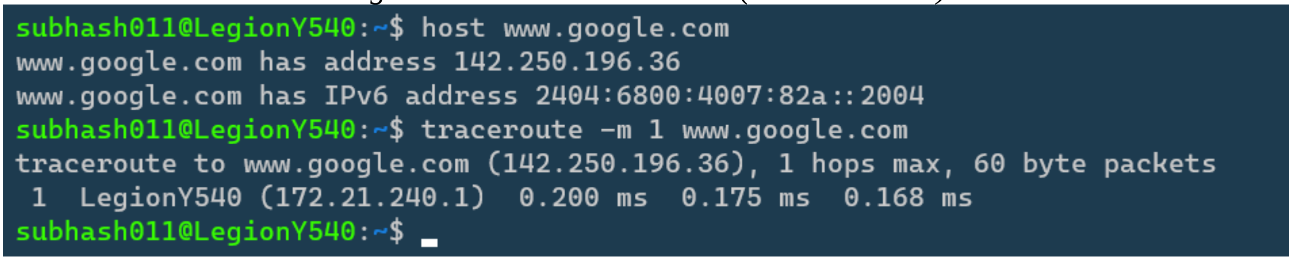
- Using the *host* command, we can see that the IPv4 address of *www.google.com* is **142.250.67.68**.
- The *traceroute* command is run with the *-m* flag which specifies maximum TTL (Time to live) to the value 1. The TTL value represents the hop counter.

Figure 4: Inside VM (h1)



```
tc@h1:~$ host www.google.com
www.google.com has address 216.58.200.132
www.google.com has IPv6 address 2404:6800:4007:822::2004
tc@h1:~$ traceroute -m 1 www.google.com
traceroute to www.google.com (216.58.200.132), 1 hops max, 38 byte packets
 1  10.0.2.2 (10.0.2.2)  0.004 ms  0.005 ms  0.003 ms
tc@h1:~$ _
```

Figure 5: From local machine (Ubuntu -20.04)



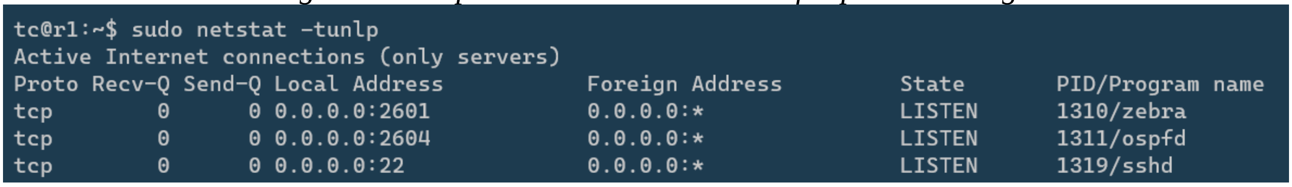
```
subhash011@LegionY540:~$ host www.google.com
www.google.com has address 142.250.196.36
www.google.com has IPv6 address 2404:6800:4007:82a::2004
subhash011@LegionY540:~$ traceroute -m 1 www.google.com
traceroute to www.google.com (142.250.196.36), 1 hops max, 60 byte packets
 1  LegionY540 (172.21.240.1)  0.200 ms  0.175 ms  0.168 ms
subhash011@LegionY540:~$ _
```

- From Figure 4 we can see that the traceroute command in host *h1* gives an IP address of *10.0.2.2* which corresponds to the host itself.
- From Figure 5 we can see that the traceroute command on my local machine (WSL-2 Ubuntu-20.04) gives an IP address of *172.21.240.1* which corresponds to the Ubuntu instance.
- From these figures, we can conclude that the first hop node is the sending node itself.

6 Port Scanning

6.1 r1, r2, r3

Figure 6: Sample netstat command used for port scanning



```
tc@r1:~$ sudo netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:2601            0.0.0.0:*               LISTEN      1310/zebra
tcp        0      0 0.0.0.0:2604            0.0.0.0:*               LISTEN      1311/ospfd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1319/sshd
```

Table 4: Open ports on routers

Port	Service
2601(TCP)	zebra
2604(TCP)	ospfd
22(TCP)	sshd

6.2 h1, h2, h3, h4

Table 5: Open ports on hosts (except h5)

Port	Service
22(TCP)	sshd

6.3 h5

Table 6: Open ports on h5

IP	Port	Service
192.168.3.2	53(TCP, UDP)	named
10.0.2.15	53(TCP, UDP)	named
127.0.0.1	53(TCP, UDP)	named
0.0.0.0	22(TCP)	sshd

6.4 Services and their purpose:

1. zebra: It is a routing manager that implements the zebra route engine.
2. ospfd: It is a routing component that works with the Quagga routing engine.
3. sshd: The SSH daemon which helps us to connect to the VM through SSH.
4. named: A DNS server.

7 Reverse DNS lookup

Table 7: Reverse DNS lookup on each VM

Host Machine	IP Address (eth1)	Domain name
r1	192.168.1.1	r1.virtnet.iitpkd
r2	192.168.2.1	r2.virtnet.iitpkd
r3	192.168.3.1	r3.virtnet.iitpkd
h1	192.168.1.2	h1.virtnet.iitpkd
h2	192.168.1.3	h2.virtnet.iitpkd
h3	192.168.2.2	h3.virtnet.iitpkd
h4	192.168.2.3	h4.virtnet.iitpkd
h5	192.168.3.2	h5.virtnet.iitpkd