School of Computer and Information Sciences
University of Hyderabad

NETWORK LAB MANUAL

1 Learn to Use *ifconfig* Command

To do: Run the command *ifconfig* at the prompt.

Output: The output will consist of all the interfaces in the system and the IP addresses assigned to the interface (if any), the MAC addresses of the interfaces and the characteristics such as MTU as well as statistics on each interface.

What to look for: Check how many Ethernet/WiFi cards are present in your system by looking physically for the interfaces and also verifying using the following command:

```
lspci -vv
```

Verify that all these are shown up in the output of the *ifconfig* command. In addition, there should be a logical interface called *lo* which is the loopback interface.

Answer the following :

1. What is the IP address assigned to the loopback interface and what is its MTU?
2. What is the MTU of an Ethernet interface?
3. What is the MTU of a WiFi interface on your laptop?
4. What are the MAC addresses of the systems? If the vendor is the same for the Ethernet card of two systems, are the first 3 bytes equal?
5. Are MAC addresses of systems equal in any of the systems you have checked?

2 Learning Wireshark

To learn to use *wireshark* to capture and analyze packets.

To do: Start *wireshark* on one interface that the machine has. Go to the capture menu and select the start option. Wait for some time and after enough packets have been captured, stop the capture.

Output: *wireshark* output will show three frames.

- The first one will consist of one line messages for all packets captured stating the basic protocol type of the packet.
- The second frame will show the packet headers present in each packet such as Ethernet, IP, UDP etc. Each of these headers can be expanded and the values of the various fields in these headers should be observed.
- The third frame consists of the hex dump of the packet. This can be related to the headers in the second frame

What to look for:

- For each different protocol type seen in the *wireshark* output, such as ARP/IP at the data link layer, identify what is the demultiplexing field in the ethernet frame. Similarly look for the protocol field in the IP header to determine whether the datagram is of ICMP, UDP or TCP.
- Identify how *wireshark* is able to classify the output as belonging to a particular protocol

Answer the following :

1. Why do you see many ARP Requests but not many ARP replies in the output?

3 Configure IP addresses

To learn how to configure IP addresses in Linux and understand rules of IP addressing.

To do: Disconnect the machines from the general LAN. Using the switches provided, connect the machines into different switches two or three per switch as shown in Fig. 1. Configure any random IP address to each of the machines as shown in the example below:

```
ifconfig eth0 172.16.88.1/24
```

Output: The IP address of the machine should be the configured value. The output can be verified by just typing *ifconfig* at the prompt and checking that the IP address in the output matches the value given.

What to look for: Verify that the machine is now reachable on the network by typing the following command from the same machine and from other machines:

```
ping 172.16.88.1
```

Answer the following :

1. What happens if two machines are given the same IP address?
2. Are you able to reach other machines connected to your switch? If you are able to, what was the address of the other machine? If not, why?
3. Are you able to reach machines connected to other switches? If not, why?

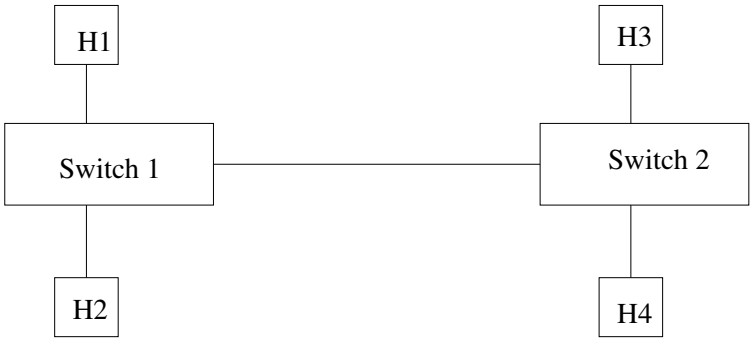

Figure 1: Simple Network Topology to understand IP addressing

4 Understanding ARP

To understand ARP packets.

To do: When you *ping* a machine, use *Wireshark* to capture the packets before the start of *ping*.

Output: You should see ARP REQUESTS and RESPONSES in the output. Run the following command to see the contents of the ARP cache:

```
arp -an
```

What to look for:

1. Expand the header of Ethernet header and verify that the type field contains the value of 0x806 signifying ARP for the ARP request and response messages.
2. *ping* messages are ICMP messages encapsulated in IP header. Hence, for all ICMP messages, the Ethernet header type field should contain the value 0x800 to signify IP packets are being carried in these frames.
3. Verify that the ARP cache contains the IP address – MAC address mapping for all the recently pinged machines.

Answer the following :

1. What is the MAC address of the frame carrying ARP REQUEST messages? What about ARP RESPONSE messages?
2. Wait for a while after a ping and verify if the ARP entry vanishes if no further packets are sent between the machines. Do you see sometimes a value of "<incomplete>" instead of MAC address in the ARP cache? When do you see this and why?

5 Creating LANs

To learn to create multiple independent LANs and test connectivity between them.

To do: Connect machines once again as in Fig. 1. When configuring IP addresses, however, configure unique addresses that belong to the same network ID.

Output: The machines should be able to *ping* each other.

Answer the following :

1. Why do you need to configure all addresses with the same IP network address?
2. If you configure machines of the same switch as follows: 10.1.0.1/24, 10.1.1.1/16 and 10.1.2.2/8, what is the reachability of these machines that are on the same switch? Explain what you observed.
3. Are you able to reach all machines belonging to different switches from all machines? Why or why not?
4. What do you need to reach all machines from all machines?

6 Routing Table Entries

To observe the routing table entries on the machines – both hosts and routers. Create a topology as shown in Fig. 2 where the machine marked $R1$ is a machine with at least 2 network interfaces, where one of each is connected to a switch.

To do: In each system, run either of the two commands given here:

```
route -n
netstat -rn
```

Output: The routing table entries are shown.

What to look for: Verify that there is an entry for the network of the IP address you have assigned to the system.

Answer the following :

1. What are the important fields of the routing table entries you observed?
2. Do the hosts have a default route entry?
3. What are the entries that are common to all machines? Why?
4. What hosts/routers are you able to reach via the *ping* command?

7 Adding Routes Manually

To understand how to add static route entries into hosts and routers.

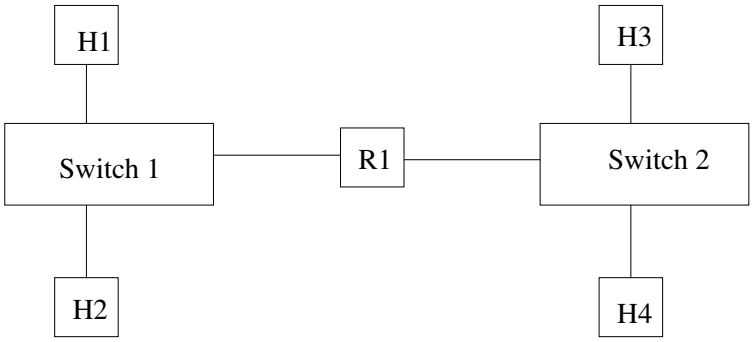

Figure 2: Simple Network with two LANs connected via a Router

To do: Add a default route to the routing table as follows:

```
route add default gw 172.16.88.1 eth0
```

where the IP address to give is the IP address of the router which is connected to your LAN and interface (eth0 in the example) is the interface on your computer that is shown in the output of *ifconfig*.

To add routes to networks not directly attached to a router, do the following:

```
route add -net 12.0.0.0 netmask 255.0.0.0 gw 172.16.8.1 eth1
```

where the network ID is of a network to which a route is being established via the "gw" and interface specified.

Output: The default route should be created when you see the output of the command *route -n*.

Answer the following :

1. Is there any change in the machines you are able to reach after adding the default router entry in the routing table?

8 Enable Forwarding

To convert a multi-homed host into a router. A system with two interfaces does not automatically become a router. Forwarding needs to be enabled to convert a multi-homed host to a router.

To do: One way to enable forwarding (a portable way across all Linux flavours) is to do the following:

```
cd /proc/sys/net/ipv4
echo 1 > ip_forward
```

The file ip_forward contains the value 0 by default which indicates to the kernel that forwarding is disabled. When it is set to 1, this tells the kernel to enable forwarding in IP - effectively converting the system with more than one interface to a router as packets are now forwarded from one interface to the other.

Output: Verify that the file ip_forward contains the value 1 by doing a *cat* on it.

What to look for: Start pinging from a machine connected to one interface of the router to a machine connected to the other interface on the router – for example, ping from $H1$ to $H3$ in Fig. 2.

Answer the following :

1. Try pinging from $H1$ to $H3$ in Fig. 2 before enabling forwarding. Is this working? What about after?

9 The *ping* Command

To do: Start capture on wireshark and then ping between two systems that are reachable from each other - preferably after creating the LANs as discussed in Section 5 to avoid extraneous traffic that will make it difficult to observe carefully the required data. This should be done for all the remaining exercises.

What to look for: Expand the ICMP header contents of the wireshark output. The messages exchanged for *ping* command should be ICMP ECHO REQUEST and ICMP ECHO REPLY.

Answer the following :

1. What is the value of the *protocol* field in the IP header? Is this the expected value for an ICMP message?
2. What are the values of **type and code** fields of the ICMP header for the ECHO REQUEST and REPLY messages? Verify these against the expected values as given in the textbook.

10 IP Fragmentation and Reassembly

To do: Start capture of traffic on wireshark and do the following. Type the following command at the shell prompt after creating LANs with the size specified after the -s option can be any value greater than 1500:

```
ping -s 3000
```

Do this simultaneously to a single machine from multiple source machines.

Output: Wireshark output should show fragmented datagrams.

What to look for: Expand the IP header in the wireshark output. Look for the ID, MF, offset, IP length fields in the various packets exchanged and identify the fragments.

Answer the following :

1. Why should the value after -s be greater than 1500?
2. How do you identify the fragments of a single datagram?
3. How did you distinguish between fragments of different datagrams?
4. Did the sum of the fragments add up to the length you specified in your *ping* command?

11 Multi-homed Host Issues

To do: Take a multi-homed host and configure unique IP addresses on its interfaces but all addresses belonging to the same network.

Output: The routing table should contain multiple entries for the same network ID but with different interfaces.

What to look for: Connect only one of the interfaces at a time to the switch via the cable. Ping other systems in the same LAN and verify the connectivity with other machines in the LAN.

Answer the following :

1. Were you able to ping machines in the LAN with both interfaces when only one of them was connected to the switch? If not, which interface succeeded and which did not? Make a note of the interface from which it succeeded.
2. What is the order of the routing table entries? Which interface is the first entry in the table? Did this interface match the interface noted in the first question?

12 ICMP Time Exceeded

To do: Set up the LANs as in Section 5, configure IP addresses on all the systems and set up the static routes properly in all the routers. Then, type the following:

```
ping -t 1 <ipaddr>
```

where $< ipaddr >$ is the IP address of a system in the same LAN as the source machine and in the LAN connected to the source by exactly one router. The "-t" option allows you to change the TTL of the IP header.

Another method of changing the TTL value set in the IP header is to do the following:

```
cd /proc/sys/net/ipv4
echo 1 > ip_default_ttl
```

Output: *ping* should succeed for the system in the same LAN but fail for the system in the other LAN.

What to look for: In the wireshark output, you should see an ICMP TIME EXCEEDED message when trying to *ping* the system from the other LAN. Verify that the TTL value in the IP header is indeed the value set using either the "-t" option or the value set in the file ip_default_ttl.

Answer the following :

1. Why does the second *ping* generate an ICMP_TIME_EXCEEDED msg?
2. What should be the value for the second *ping* to succeed?

13 ICMP Destination Unreachable/Host Unreachable

To do: Connect systems in different LANs as specified in Section 5. Start capture on Wireshark. After IP addresses have been given to all machines in each LAN, make a note of the addresses assigned. Now, do the following:

1. *ping* an IP address that does not exist in your LAN.
2. *ping* an IP address that does not exist in another LAN.

Output: In the Wireshark there should be ICMP Destination Unreachable/Host unreachable messages should be seen.

What to look for:

1. When exactly are the ICMP messages seen out of the two experiments given above?
2. Expand the ICMP header and verify that the type and code fields of the messages match that of DEST_UNREACH and HOST_UNREACH respectively.
3. Expand the body of the ICMP message and verify that it does contain the first 64 bytes of the original ICMP message (i.e., the ping) which caused the DEST_UNREACH error.

Answer the following :

1. Why dont we see ICMP Destination Unreachable message in the first experiment above?
2. What are the values carried in the first 64 bytes of the original datagram that caused an ICMP error message? Will this help in identifying the various information such as the original source and destination machines, the message being sent etc. for a network administrator to troubleshoot the network?

14 ICMP Destination Unreachable/Network Unreachable

To do: Do the experiment exactly as in Section 13 except that

1. *ping* a machine with a network ID that has not been configured to any system.
2. *ping* a machine with a network ID that exists but some router on the path has not been configured a route to this when adding static routes on the routers.

Output: ICMP Destination Unreachable/Network Unreachable message should be seen from a router on the path to the source machine.

Answer the following :

1. Which router returns the error to the source machine?
2. What are the **type and code** fields in the ICMP header? Verify that the **type** is the same as observed in Section 13 but **code** is different.

15 ICMP Destination Unreachable/Port Unreachable

To do: Set up the LANs and configure the IP addresses and static routes properly in all the routers. Then type the following command from any machine to any valid IP address in any other LAN:

```
traceroute <ipaddr>
```

Output: You should see the IP addresses of all routers on the path from the source machine to the destination machine.

What to look for: In the wireshark output,

1. Check what is the message generated by *traceroute*.
2. Look for the ICMP messages that are generated as a result of traceroute.
3. Run the command *netstat -aun* in the final destination. It shows all the UDP ports open in the destination. Check this against the UDP port value carried in the *traceroute* message by expanding the UDP header contents.

Answer the following :

1. What are the various ICMP error messages seen?
2. What is the ICMP error message seen from routers as compared to that from the final destination?
3. Is the UDP port carried in the *traceroute* message open in the final destination machine as seen from the *netstat* output?

16 ICMP Redirect

To do: Connect 3 LANs using two routers as shown in Fig. 3. Please note that the hosts H1, H2 and R1's one interface are on LAN 1, H3, H4, R1's other interface and R2's one interface are on LAN 2 and R2's second interface, H5 and H6 are on LAN 3. In the host H3, have the default router as R1 and in H4 as R2. Then, do the following:

1. *ping* H5 from H3.

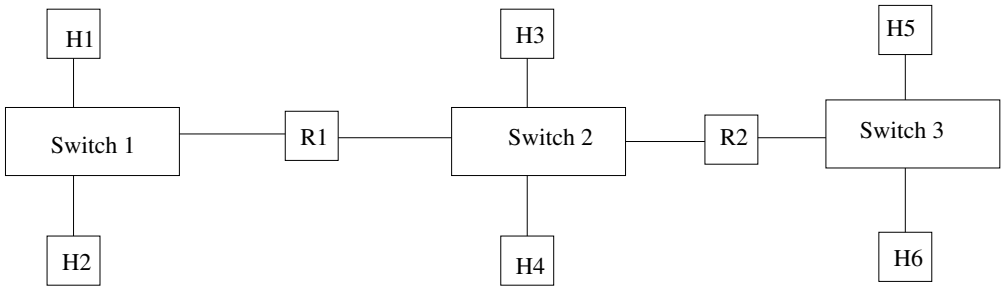

Figure 3: Simple Network with two routers to demonstrate ICMP Redirect

2. *ping* H5 from H4.
3. *ping* H1 from H4.
4. *ping* H1 from H3.

Output: When pinging some of the machines, you should see the output ICMP Redirect in addition to ping reply messages.

What to look for: Examine the output of wireshark to see that there are ICMP redirect messages coming to the source machines by expanding the ICMP header and verifying that the type and code fields match that of Redirect.

Answer the following :

1. Which of the above four *ping* commands results in ICMP redirect messages? Why is it not seen in other commands?
2. Is the redirect seen with every reply to all the ping messages or only once in a while? Why?
3. What are the contents of the body of the ICMP Redirect message?
4. What happens if you give a gap and repeat the messages?

17 TCP 3-way SYN Handshake

Install the telnet-server software on the system. Once this is installed, you should find an executable called **in.telnetd** in your system. Then, to actually run the telnet daemon, do the following (it may change a little with each Linux version):

```
 cd /etc/xinetd.d
 create a file called telnet (if it does not exist) by copying one of the
 other files.
 service daytime
{
        id              = telnet
```

```
        socket_type     = stream
        protocol        = tcp
        user            = root
        disable         = no
        server = /usr/sbin/in.telnetd
        log_on_failure += USERID
        FLAGS           = IPv4
}
```

To do: Confirm that the telnet server is running by running the following command:

```
netstat -atn | grep 23
```

There should be a server running with port 23 open. Once this is confirmed, start capture on Wireshark. Then, run telnet client from another machine to the machine where the server software is running as follows:

```
telnet <ipaddr>
```

This shows the login: prompt followed by password: prompt. Enter the user ID and password of users on the server system. This should result in a login on the remote system with the shell prompt being displayed.

Output: In the wireshark output, you should be able to see lot of TCP messages exchanged. The first 3 messages exchanged will be the 3-way SYN handshake messages.

What to look for: Expand the TCP header and look at the SEQ no sent for the first message, the bits set in TCP header: ony the SYN bit must be set for the first message. The Ack number will be 0 as there is nothing to ack yet.

The second message, should have the ACK number set to SEQ no + 1 and should have a SEQ number of the server system. Both SYN and ACK bits must be set.

The third message should have ACK bit only set.

In Wireshark, you can use the "Analyze" menu and select "Follow TCP stream" to get the entire data exchanged between Telnet client and server.

Answer the following :

1. When you do TCP follow stream, why is every letter entered as part of user ID and password shown twice in the output?
2. What is the size of the Window sent by the client and server to the other side of the connection?
3. After logging into the server, what is the output of *netstat -atn*?
4. Is the MSS option exchanged in the SYN handshake? If so, what is the value?

18 TCP 3-way FIN Handshake

To do: After logging into the remote system using telnet as in Section 17, type the command "netstat -atn" on both the client and server systems and observe the connection details.

Then, type the command "exit" at the shell prompt. Now, re-run the netstat command and see the relevant information.

What to look for: In Wireshark, you should see an exchange of the FIN messages as the connection is closed on both the client and server ends.

Answer the following :

1. What are the flag bits set in the FIN messages?
2. Is there any change in the value of the "STATE" field in the netstat output as the messages are exchanged?

19 Understanding Retransmissions in TCP

To do: After logging into the Telnet server as specified in Section 17, type the command "ls" at the remote system's shell prompt after disconnecting the server from the switch. Wait for a while and reconnect the server back to the switch.

Output: You should see that the same message is sent multiple times.

What to look for: The retransmitted packets in Wireshark output are marked clearly. Look at the timestamp of time of sending for the retransmitted packets.

Answer the following :

1. What is the timestamp value for each of the retransmitted messages? Do you see an exponential backoff?
2. If you keep the server disconnected long enough, what happens?
3. What is the output of "netstat -atn" in both client and server if the server is disconnected – both for a short time and long time?
4. What is the value of the column "STATE" in the netstat output?

20 TCP half-close connections

To do: As in the experiment in Section 19, disconnect the server after logging into the remote system.

Output: The client should return to the shell prompt with the message that the connection timed out.

What to look for: In the wireshark output on both client and server, look for the FIN handshake.

Answer the following :

1. What is the output for "netstat -atn" before the disconnection of the server from the switch?
2. What is the output for "netstat -atn" after the disconnection of the server from the switch?
3. What is the value of the "STATE" column in "netstat -atn" output on the server after the client has returned to the prompt?

21 TCP RST bit if server is rebooted

To do: Log into the remote system as in Section 17. Verify that the connection exists on both client and server using the *netstat -atn* command.

Disconnect the server from the switch and reboot the system. Once the server is rebooted, verify the connection information on both client and server once again.

Now, type the command "ls" at the remote shell prompt on the client.

Output: The client will return to the local shell prompt with the message: "Connection reser by the peer".

What to look for: In the wireshark output, look for the TCP message with the RST bit set.

Answer the following :

1. Who sends the message with the RST bit set to whom?
2. What is the sequence number of the message prior to RST coming in from the remote side?
3. What is the sequence number of the RST message and is there an ACK? Why or why not?

22 TCP Keepalive Timer

This is similar to the experiment in Section 21 except that disconnect the client system from the switch and reboot it.

To do: Before disconnecting the client and rebooting it do the following on the server system and note down the values:

```
cd /proc/sys/net/ipv4
cat tcp_keepalive_time
cat tcp_keepalive_probes
cat tcp_keepalive_intvl
```

Then, do the following:

```
echo 300 > tcp_keepalive_time
echo 3 > tcp_keepalive_probes
echo 30 > tcp_keepalive_intvl
```

At this point of time, the keepalive timer value is set to 5 minutes and 3 probes will be sent at the end of that time for an idle connection with an interval between the probes of 30 seconds.

When the client has rebooted, connect the client back to the switch.

What to look for: The wireshark output should show some TCP messages after some time even if you dont do anything.

Answer the following :

1. Change the values of the keepalive timer as shown above and without rebooting the client, observe the behaviour with an idle connection - i.e., do not do anything after logging into the server. Are there any messages sent? If so, when?

2. What are the contents of the TCP messages seen to be exchanged with no interference from a user after client is rebooted and reconnected to the switch? Why?