



UNIVERSITY OF DHAKA

Department of Computer Science and Engineering

CSE-3111 : Computer Networking Lab

Lab Report 1 : An Exercise on LAN Configuration and Troubleshooting Tools

Submitted By :

Name: Md. Emon Khan

Roll No : 30

Name: Mahmudul Hasan

Roll No : 60

Submitted To :

Dr. Md. Abdur Razzaque

Dr. Md. Mamun Or Rashid

Dr. Muhammad Ibrahim

Md. Redwan Ahmed Rizvee

Submitted On : January 26, 2024

1 Introduction

The primary objective of Lab Experiment-1 is to get a quick introduction to LAN configuration and troubleshooting tools using command line with tools like PING, Traceroute, ARP, Static routing, netstat, ifconfig, nslookup, whois, etc.

1.1 Objectives

Some of the specific objectives of the lab experiment are:

- List a few commands recommended by the teacher and try them out in the cmd
- Understand how and what information each of the commands give, or what tasks may be done by them
- Note how the given information may be beneficial in the context of computer networking

2 Theory

Devices receive local addresses within their LANs, and routers connect local networks to broader networks using public addresses. These processes adhere to specific protocols. Troubleshooting and configuring LANs and connected devices involve using various tools.

3 Methodology

During the lab session, we systematically executed each command to explore their functionalities. Some commands offered different options, prompting us to experiment with variations. We tested the addresses of other devices within the LAN and also probed various internet websites. Notably, we extracted and scrutinized diverse network configurations and usage statistics using the employed commands.

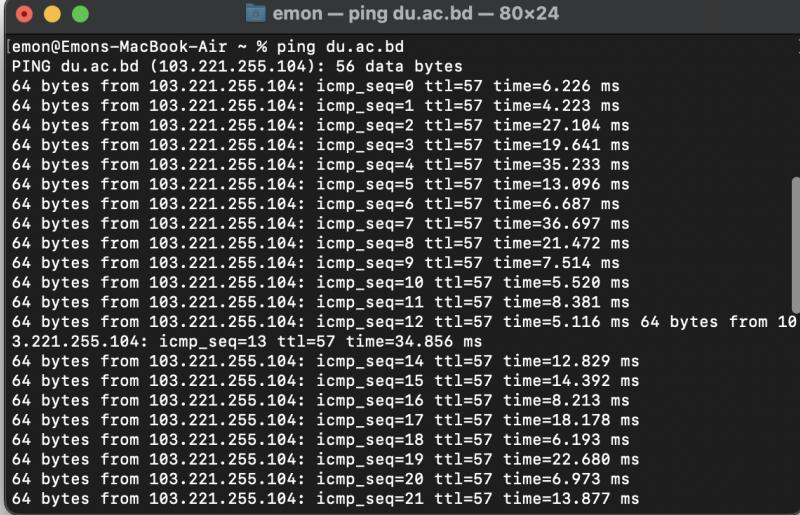
4 Experimental result

Some Snapshots of the terminal output for each of these tools.

4.1 Ping

The `ping` command is commonly used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination computer.

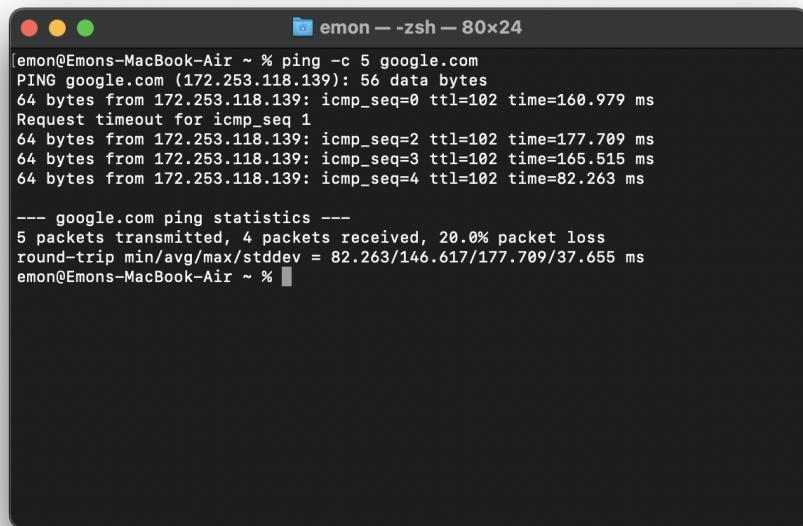
4.1.1 Basic Ping: `ping du.ac.bd`

A screenshot of a macOS terminal window titled "emon — ping du.ac.bd — 80x24". The window displays the output of a ping command. The text shows multiple ICMP Echo Request messages being sent to the IP address 103.221.255.104. Each message is 64 bytes long and has a sequence number (icmp_seq) ranging from 0 to 21. The output includes the source IP (103.221.255.104), sequence number, TTL (57), and round-trip time in milliseconds (ms). The times range from 6.193 ms to 34.856 ms.

```
[emon@Emons-MacBook-Air ~ % ping du.ac.bd
PING du.ac.bd (103.221.255.104): 64 data bytes
64 bytes from 103.221.255.104: icmp_seq=0 ttl=57 time=6.226 ms
64 bytes from 103.221.255.104: icmp_seq=1 ttl=57 time=4.223 ms
64 bytes from 103.221.255.104: icmp_seq=2 ttl=57 time=27.104 ms
64 bytes from 103.221.255.104: icmp_seq=3 ttl=57 time=19.641 ms
64 bytes from 103.221.255.104: icmp_seq=4 ttl=57 time=35.233 ms
64 bytes from 103.221.255.104: icmp_seq=5 ttl=57 time=13.096 ms
64 bytes from 103.221.255.104: icmp_seq=6 ttl=57 time=6.687 ms
64 bytes from 103.221.255.104: icmp_seq=7 ttl=57 time=36.697 ms
64 bytes from 103.221.255.104: icmp_seq=8 ttl=57 time=21.472 ms
64 bytes from 103.221.255.104: icmp_seq=9 ttl=57 time=7.514 ms
64 bytes from 103.221.255.104: icmp_seq=10 ttl=57 time=5.520 ms
64 bytes from 103.221.255.104: icmp_seq=11 ttl=57 time=8.381 ms
64 bytes from 103.221.255.104: icmp_seq=12 ttl=57 time=5.116 ms 64 bytes from 103.221.255.104: icmp_seq=13 ttl=57 time=34.856 ms
64 bytes from 103.221.255.104: icmp_seq=14 ttl=57 time=12.829 ms
64 bytes from 103.221.255.104: icmp_seq=15 ttl=57 time=14.392 ms
64 bytes from 103.221.255.104: icmp_seq=16 ttl=57 time=8.213 ms
64 bytes from 103.221.255.104: icmp_seq=17 ttl=57 time=18.178 ms
64 bytes from 103.221.255.104: icmp_seq=18 ttl=57 time=6.193 ms
64 bytes from 103.221.255.104: icmp_seq=19 ttl=57 time=22.680 ms
64 bytes from 103.221.255.104: icmp_seq=20 ttl=57 time=6.973 ms
64 bytes from 103.221.255.104: icmp_seq=21 ttl=57 time=13.877 ms
```

Figure 1: This sends a series of ICMP Echo Request messages to the specified domain and displays the round-trip time for each message.

4.1.2 Ping with Specific Count: `ping -c 5 google.com`



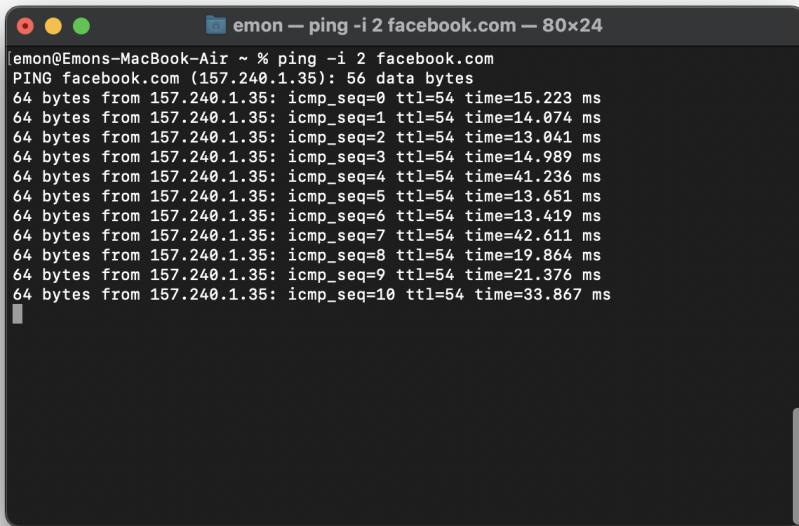
A screenshot of a terminal window titled "emon -- zsh - 80x24". The window shows the command "ping -c 5 google.com" being run. The output displays five ICMP Echo Request messages sent to google.com (172.253.118.139) with sequence numbers 0, 1, 2, 3, and 4. The times for these requests range from 160.979 ms to 177.709 ms. A request timeout for icmp_seq 1 is also mentioned. The statistics show 5 packets transmitted, 4 received, and 20.0% packet loss. The round-trip time is 82.263 ms. The terminal prompt "emon@Emons-MacBook-Air ~ %" is visible at the bottom.

```
[emon@Emons-MacBook-Air ~ % ping -c 5 google.com
PING google.com (172.253.118.139): 56 data bytes
64 bytes from 172.253.118.139: icmp_seq=0 ttl=102 time=160.979 ms
Request timeout for icmp_seq 1
64 bytes from 172.253.118.139: icmp_seq=2 ttl=102 time=177.709 ms
64 bytes from 172.253.118.139: icmp_seq=3 ttl=102 time=165.515 ms
64 bytes from 172.253.118.139: icmp_seq=4 ttl=102 time=82.263 ms

--- google.com ping statistics ---
5 packets transmitted, 4 packets received, 20.0% packet loss
round-trip min/avg/max/stddev = 82.263/146.617/177.709/37.655 ms
emon@Emons-MacBook-Air ~ %
```

Figure 2: This sends only 5 ICMP Echo Request messages to google.com and then stops

4.1.3 Ping with Interval: `ping -i 2 facebook.com`

A screenshot of a terminal window titled "emon — ping -i 2 facebook.com — 80x24". The window shows the command being run and its output. The output consists of 10 ICMP Echo Request messages sent to the IP address 157.240.1.35. Each message is 64 bytes long, has a ttl of 54, and is timestamped. The sequence numbers range from 0 to 9. The times between messages are approximately 2 seconds each.

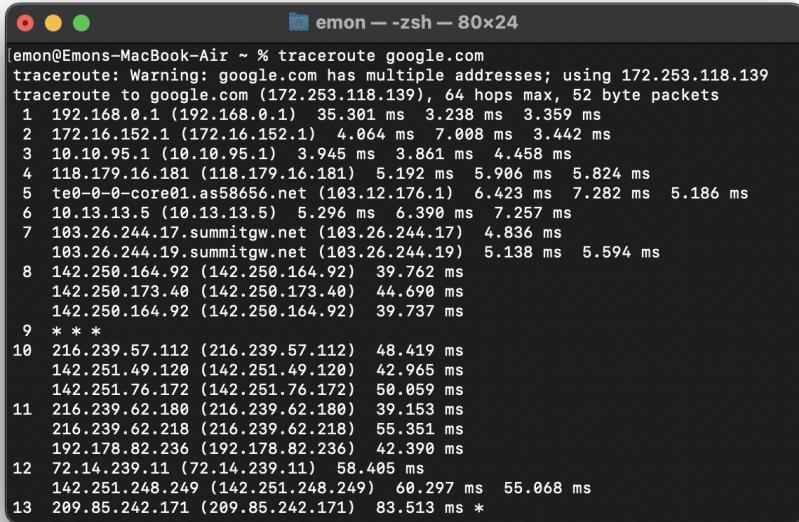
```
[emon@Emons-MacBook-Air ~ % ping -i 2 facebook.com
PING facebook.com (157.240.1.35): 64 data bytes
64 bytes from 157.240.1.35: icmp_seq=0 ttl=54 time=15.223 ms
64 bytes from 157.240.1.35: icmp_seq=1 ttl=54 time=14.074 ms
64 bytes from 157.240.1.35: icmp_seq=2 ttl=54 time=13.041 ms
64 bytes from 157.240.1.35: icmp_seq=3 ttl=54 time=14.989 ms
64 bytes from 157.240.1.35: icmp_seq=4 ttl=54 time=41.236 ms
64 bytes from 157.240.1.35: icmp_seq=5 ttl=54 time=13.651 ms
64 bytes from 157.240.1.35: icmp_seq=6 ttl=54 time=13.419 ms
64 bytes from 157.240.1.35: icmp_seq=7 ttl=54 time=42.611 ms
64 bytes from 157.240.1.35: icmp_seq=8 ttl=54 time=19.864 ms
64 bytes from 157.240.1.35: icmp_seq=9 ttl=54 time=21.376 ms
64 bytes from 157.240.1.35: icmp_seq=10 ttl=54 time=33.867 ms
```

Figure 3: This sends ICMP Echo Request messages to facebook.com with a 2-second interval between each message.

4.2 Traceroute

The `traceroute` is a network diagnostic tool used to track the route taken by packets in an IP network from the source to the destination. It provides information about the number of hops, round-trip times

4.2.1 traceroute google.com



A screenshot of a terminal window titled "emon -- zsh - 80x24". The command "traceroute google.com" is run, and the output shows the path to Google's servers. The output includes the hop number, IP address, and round-trip time for each hop.

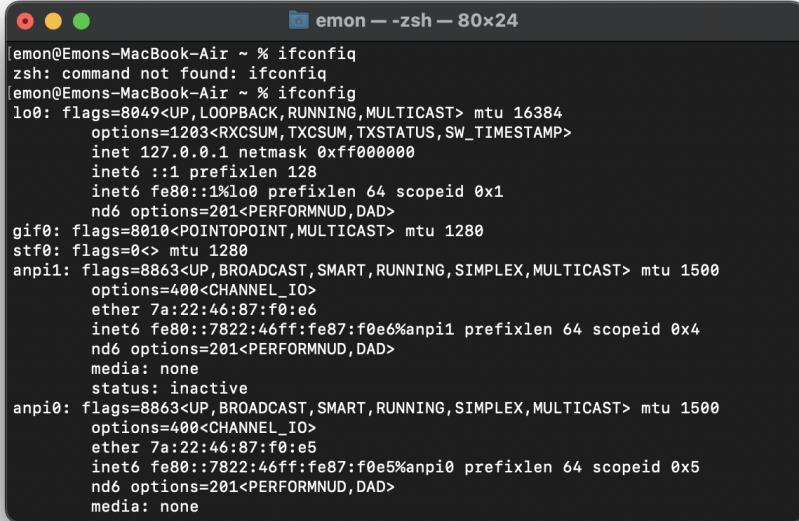
```
[emon@Emons-MacBook-Air ~ % traceroute google.com
traceroute: Warning: google.com has multiple addresses; using 172.253.118.139
traceroute to google.com (172.253.118.139), 64 hops max, 52 byte packets
 1  192.168.0.1 (192.168.0.1)  35.301 ms  3.238 ms  3.359 ms
 2  172.16.152.1 (172.16.152.1)  4.064 ms  7.008 ms  3.442 ms
 3  10.10.95.1 (10.10.95.1)  3.945 ms  3.861 ms  4.458 ms
 4  118.179.16.181 (118.179.16.181)  5.192 ms  5.906 ms  5.824 ms
 5  te0-0-0-core01.as58656.net (103.12.176.1)  6.423 ms  7.282 ms  5.186 ms
 6  10.13.13.5 (10.13.13.5)  5.296 ms  6.390 ms  7.257 ms
 7  103.26.244.17.summitgw.net (103.26.244.17)  4.836 ms
 103.26.244.19.summitgw.net (103.26.244.19)  5.138 ms  5.594 ms
 8  142.250.164.92 (142.250.164.92)  39.762 ms
 142.250.173.40 (142.250.173.40)  44.690 ms
 142.250.164.92 (142.250.164.92)  39.737 ms
 9  * * *
10  216.239.57.112 (216.239.57.112)  48.419 ms
 142.251.49.120 (142.251.49.120)  42.965 ms
 142.251.76.172 (142.251.76.172)  50.059 ms
11  216.239.62.180 (216.239.62.180)  39.153 ms
 216.239.62.218 (216.239.62.218)  55.351 ms
 192.178.82.236 (192.178.82.236)  42.390 ms
12  72.14.239.11 (72.14.239.11)  58.405 ms
 142.251.248.249 (142.251.248.249)  60.297 ms  55.068 ms
13  209.85.242.171 (209.85.242.171)  83.513 ms *
```

Figure 4: This command traces the route to the server of google.com, displaying the IP addresses and round-trip times for each hop.

4.3 IfConfig

ifconfig (interface configuration) is a command-line tool that allows users to configure and display information about network interfaces on a system. It enables users to perform tasks such as configuring IP addresses, creating aliases, setting hardware (MAC) addresses, and enabling or disabling interfaces.

4.3.1 ifconfig

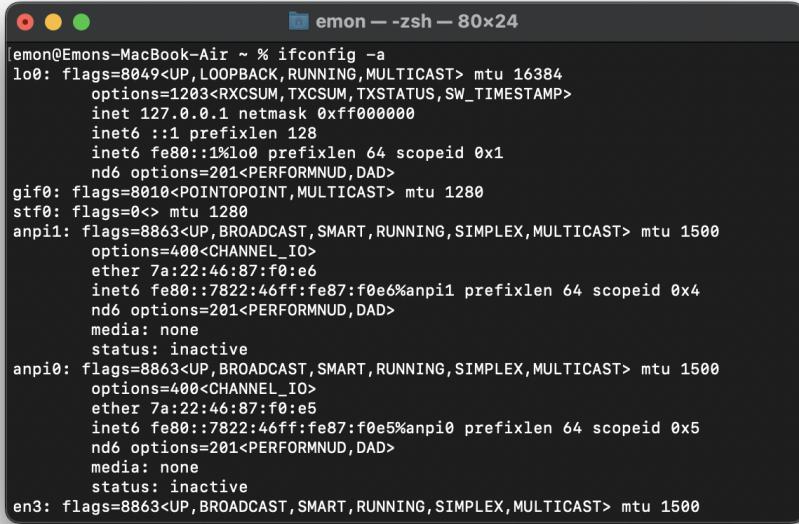


The screenshot shows a terminal window titled "emon -- -zsh -- 80x24". The command "ifconfig" is run, but "zsh: command not found: ifconfig" is displayed, indicating the command is available in a different shell. The output then lists various network interfaces:

```
[emon@Emons-MacBook-Air ~ % ifconfig
zsh: command not found: ifconfig
[emon@Emons-MacBook-Air ~ % ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
            inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
                nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
anpii: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether 7a:22:46:87:f0:e6
    inet6 fe80::7822:46ff:fe87:f0e6%anpii prefixlen 64 scopeid 0x4
        nd6 options=201<PERFORMNUD,DAD>
        media: none
        status: inactive
anpi0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether 7a:22:46:87:f0:e5
    inet6 fe80::7822:46ff:fe87:f0e5%anpi0 prefixlen 64 scopeid 0x5
        nd6 options=201<PERFORMNUD,DAD>
        media: none
```

Figure 5: The **ifconfig** command with no arguments will display all the active network interface configuration details. This includes loopback interfaces, software interfaces, network interfaces with hardware addresses (MAC), and Ethernet interfaces

4.3.2 ifconfig -a



A terminal window titled "emon -- zsh -- 80x24" displays the output of the "ifconfig -a" command. The output lists various network interfaces with their flags, MTU, options, MAC addresses, and IP configurations.

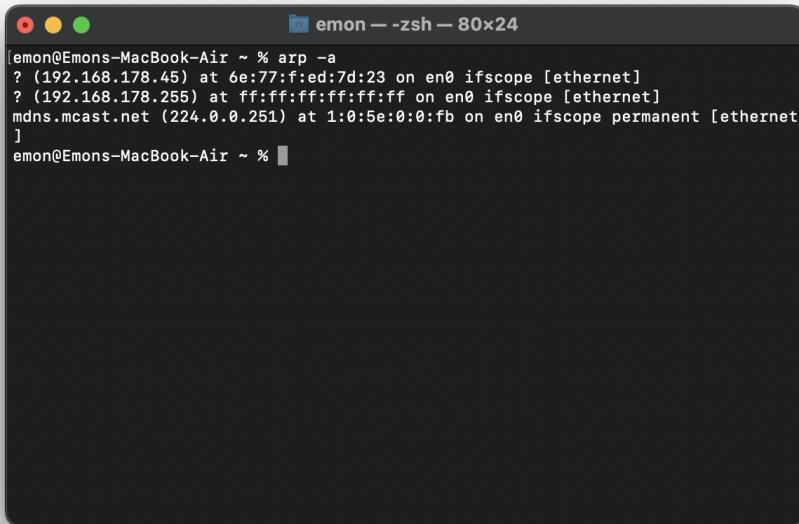
```
[emon@Emons-MacBook-Air ~ % ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
            inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
                nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
anpi1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether 7a:22:46:87:f0:e6
    inet6 fe80::7822:46ff:fe87:f0e6%anpi1 prefixlen 64 scopeid 0x4
        nd6 options=201<PERFORMNUD,DAD>
        media: none
        status: inactive
anpi0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether 7a:22:46:87:f0:e5
    inet6 fe80::7822:46ff:fe87:f0e5%anpi0 prefixlen 64 scopeid 0x5
        nd6 options=201<PERFORMNUD,DAD>
        media: none
        status: inactive
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
```

Figure 6: The `ifconfig` command with the `-a` argument will display information on all active or inactive network interfaces on the server

4.4 arp

`arp` command is used to display and manage the Address Resolution Protocol (ARP) cache. ARP is a protocol used to map an IP address to a physical (MAC) address on a local network.

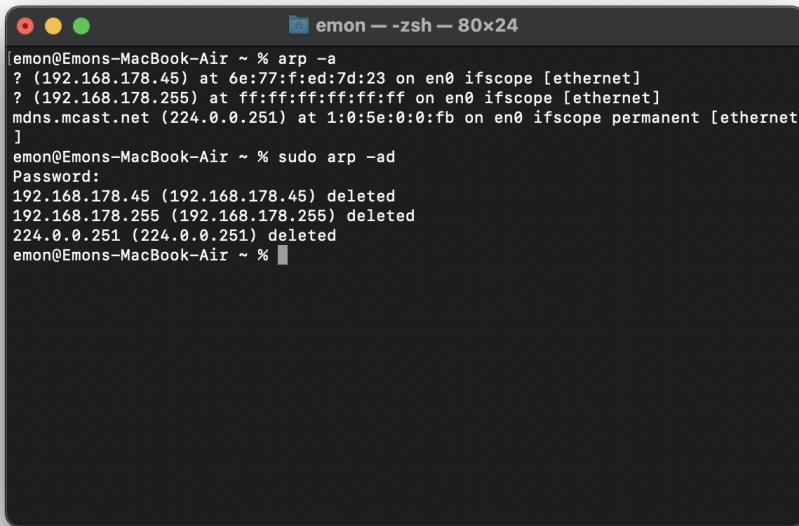
4.4.1 Display ARP Cache: `arp -a`

A screenshot of a macOS terminal window titled "emon -- zsh - 80x24". The window shows the command "arp -a" being run and its output. The output lists three entries: a broadcast entry for 192.168.178.255, a permanent entry for mdns.mcast.net (224.0.0.251), and a temporary entry for 192.168.178.45. Each entry shows the IP address, MAC address, interface (en0), and scope information.

```
[emon@Emons-MacBook-Air ~ % arp -a
? (192.168.178.45) at 6e:77:f:ed:7d:23 on en0 ifscope [ethernet]
? (192.168.178.255) at ff:ff:ff:ff:ff:ff on en0 ifscope [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
]emon@Emons-MacBook-Air ~ % ]
```

Figure 7: This will display a list of IP addresses and their corresponding MAC addresses.

4.4.2 Flush ARP Cache: `sudo arp -ad`



A screenshot of a terminal window titled "emon -- zsh -- 80x24". The window shows the command "arp -a" being run, followed by the output of the ARP cache entries. Then, the command "sudo arp -ad" is run, which prompts for a password. The output shows that three entries have been deleted: "192.168.178.45 (192.168.178.45) deleted", "192.168.178.255 (192.168.178.255) deleted", and "224.0.0.251 (224.0.0.251) deleted". The terminal prompt "emon@Emons-MacBook-Air ~ %" is visible at the bottom.

```
[emon@Emons-MacBook-Air ~ % arp -a
? (192.168.178.45) at 6e:77:f:ed:7d:23 on en0 ifscope [ethernet]
? (192.168.178.255) at ff:ff:ff:ff:ff:ff on en0 ifscope [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:fb on en0 ifscope permanent [ethernet]
]
emon@Emons-MacBook-Air ~ % sudo arp -ad
Password:
192.168.178.45 (192.168.178.45) deleted
192.168.178.255 (192.168.178.255) deleted
224.0.0.251 (224.0.0.251) deleted
emon@Emons-MacBook-Air ~ %
```

Figure 8: This command use to clear (flush) all entries in the ARP cache

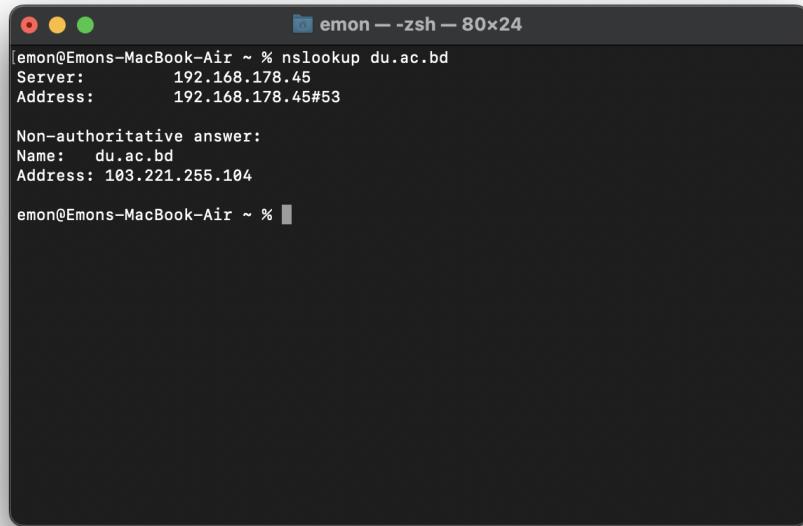
4.5 rarp

rarp, or Reverse Address Resolution Protocol, is an obsolete networking protocol used in early computer networks. Its primary purpose was to map a device's physical hardware address (usually its MAC address) to its corresponding IP address.

4.6 Nslookup

`nslookup` (stands for “Name Server Lookup”) is a useful command for getting information from the DNS server. It is used for querying DNS (Domain Name System) servers to obtain domain name or IP address mapping, DNS records, and other information related to domain names.

4.6.1 nslookup du.ac.bd

A screenshot of a terminal window titled "emon -- -zsh -- 80x24". The window shows the command "nslookup du.ac.bd" being run. The output indicates that the query was sent to a local server at 192.168.178.45 on port 53. The domain "du.ac.bd" was resolved to the IP address 103.221.255.104. The terminal prompt "emon@Emons-MacBook-Air ~ %" is visible at the bottom.

```
[emon@Emons-MacBook-Air ~ % nslookup du.ac.bd
Server:      192.168.178.45
Address:     192.168.178.45#53

Non-authoritative answer:
Name:   du.ac.bd
Address: 103.221.255.104

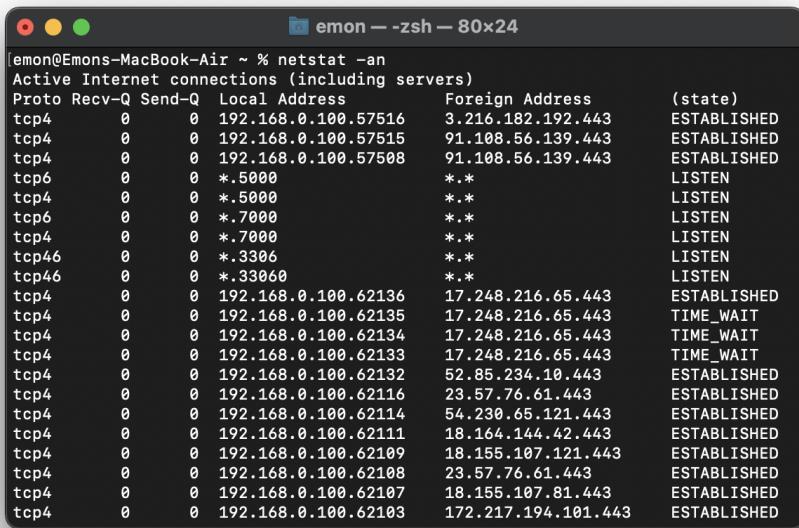
emon@Emons-MacBook-Air ~ % ]
```

Figure 9: The DNS server being used for the query is at the IP address 192.168.178.45, and it operates on port 53, the standard port for DNS. The domain ”du.ac.bd” resolves to the IP address 103.221.255.104. This is the information obtained from the DNS server.

4.7 Netstat

The `netstat` command is a network utility tool used to display information about network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.

4.7.1 `netstat -an`

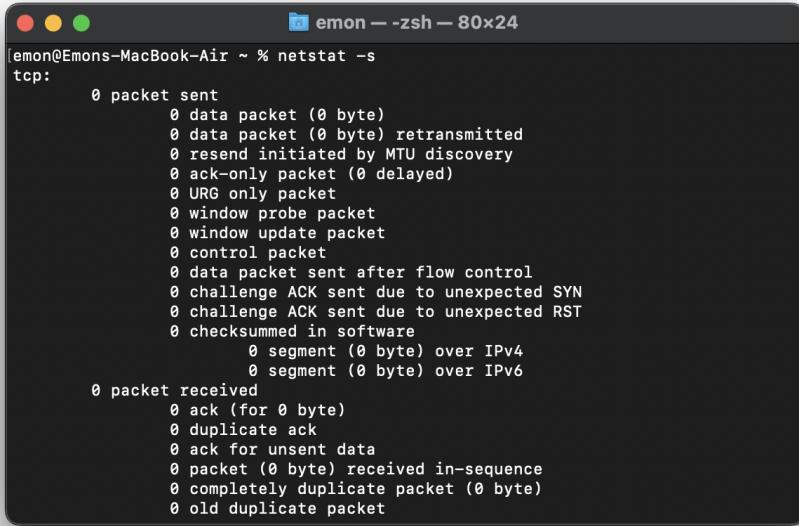


A terminal window titled "emon -- zsh -- 80x24" displaying the output of the "netstat -an" command. The output lists active Internet connections with columns for Proto, Recv-Q, Send-Q, Local Address, Foreign Address, and state. The connections are mostly ESTABLISHED, with some in TIME_WAIT or LISTEN states.

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp4	0	0	192.168.0.100.57516	3.216.182.192.443	ESTABLISHED
tcp4	0	0	192.168.0.100.57515	91.108.56.139.443	ESTABLISHED
tcp4	0	0	192.168.0.100.57508	91.108.56.139.443	ESTABLISHED
tcp6	0	0	*.5000	.*.*	LISTEN
tcp4	0	0	*.5000	.*.*	LISTEN
tcp6	0	0	*.7000	.*.*	LISTEN
tcp4	0	0	*.7000	.*.*	LISTEN
tcp46	0	0	*.3306	.*.*	LISTEN
tcp46	0	0	*.33060	.*.*	LISTEN
tcp4	0	0	192.168.0.100.62136	17.248.216.65.443	ESTABLISHED
tcp4	0	0	192.168.0.100.62135	17.248.216.65.443	TIME_WAIT
tcp4	0	0	192.168.0.100.62134	17.248.216.65.443	TIME_WAIT
tcp4	0	0	192.168.0.100.62133	17.248.216.65.443	TIME_WAIT
tcp4	0	0	192.168.0.100.62132	52.85.234.10.443	ESTABLISHED
tcp4	0	0	192.168.0.100.62116	23.57.76.61.443	ESTABLISHED
tcp4	0	0	192.168.0.100.62114	54.230.65.121.443	ESTABLISHED
tcp4	0	0	192.168.0.100.62111	18.164.144.42.443	ESTABLISHED
tcp4	0	0	192.168.0.100.62109	18.155.107.121.443	ESTABLISHED
tcp4	0	0	192.168.0.100.62108	23.57.76.61.443	ESTABLISHED
tcp4	0	0	192.168.0.100.62107	18.155.107.81.443	ESTABLISHED
tcp4	0	0	192.168.0.100.62103	172.217.194.101.443	ESTABLISHED

Figure 10: This command displays a list of all active network connections, along with their local and remote IP addresses and port numbers

4.7.2 netstat -s

A screenshot of a macOS terminal window titled "emon -- zsh - 80x24". The window shows the output of the "netstat -s" command. The output is as follows:

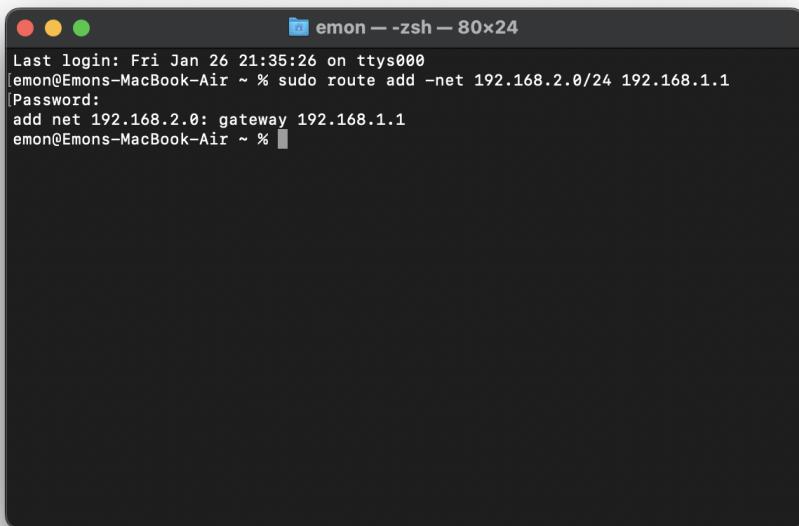
```
[emon@Emons-MacBook-Air ~ % netstat -s
tcp:
  0 packet sent
    0 data packet (0 byte)
    0 data packet (0 byte) retransmitted
    0 resend initiated by MTU discovery
    0 ack-only packet (0 delayed)
    0 URG only packet
    0 window probe packet
    0 window update packet
    0 control packet
    0 data packet sent after flow control
    0 challenge ACK sent due to unexpected SYN
    0 challenge ACK sent due to unexpected RST
    0 checksummed in software
      0 segment (0 byte) over IPv4
      0 segment (0 byte) over IPv6
  0 packet received
    0 ack (for 0 byte)
    0 duplicate ack
    0 ack for unsent data
    0 packet (0 byte) received in-sequence
    0 completely duplicate packet (0 byte)
    0 old duplicate packet
```

Figure 11: This command displays statistics for each protocol, including the number of packets sent and received.

4.8 Static Routing

Static routing is a method of configuring a network by manually adding routes to the routing table. In a static routing setup, network administrators define specific paths for data to travel from its source to its destination

4.8.1 sudo route add -net 192.168.2.0/24 192.168.1.1



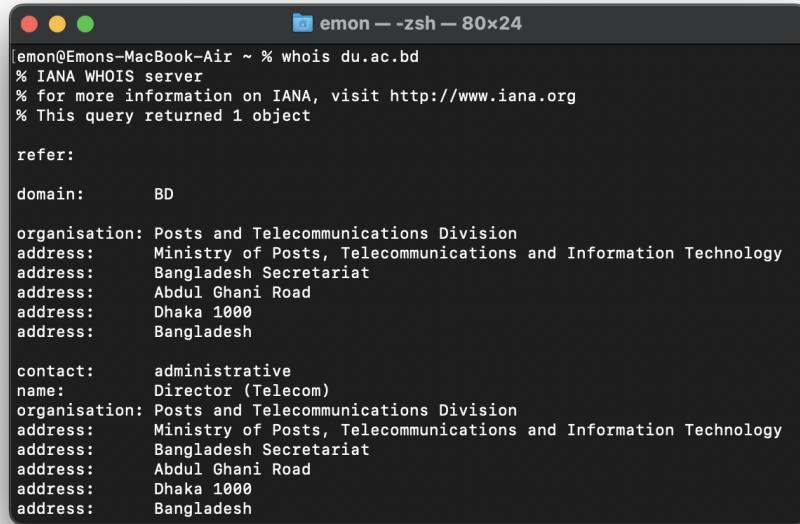
A screenshot of a macOS terminal window titled "emon -- zsh -- 80x24". The window shows the command "sudo route add -net 192.168.2.0/24 192.168.1.1" being run. The user is prompted for a password, which is entered. The command is executed successfully, and the output shows the route being added: "add net 192.168.2.0: gateway 192.168.1.1". The terminal window has a dark background and light-colored text.

Figure 12: This command adds a static route for the network 192.168.2.0/24 with the next-hop router's IP address 192.168.1.1.

4.9 WHOIS

WHOIS is a query and response protocol used to obtain information about domain names, IP addresses, and autonomous systems, registration Authorities. The term "WHOIS" can refer to both the protocol itself and the databases that store the information.

4.9.1 whois du.ac.bd



A terminal window titled "emon -- zsh - 80x24" displays the output of the "whois du.ac.bd" command. The output shows the following details:

```
[emon@Emons-MacBook-Air ~ % whois du.ac.bd
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:

domain:      BD

organisation: Posts and Telecommunications Division
address:      Ministry of Posts, Telecommunications and Information Technology
address:      Bangladesh Secretariat
address:      Abdul Ghani Road
address:      Dhaka 1000
address:      Bangladesh

contact:     administrative
name:        Director (Telecom)
organisation: Posts and Telecommunications Division
address:      Ministry of Posts, Telecommunications and Information Technology
address:      Bangladesh Secretariat
address:      Abdul Ghani Road
address:      Dhaka 1000
address:      Bangladesh
```

Figure 13: This command will retrieve and display information about the domain "du.ac.bd"

5 Experience

1. We had to see some examples of how to use the tools in the command line
2. We used these commands for the first time to actually find the LAN configurations

References

- [1] <https://pimylifeup.com/>
- [2] <https://cloudinfrastructureservices.co.uk/>
- [3] <https://www.tecmint.com/>
- [4] <https://www.geeksforgeeks.org/>