ILLINOIS TECH

Application Software Design

ECE 528

Homework 1

Abhilash Kashyap Balasubramanyam

abalasubramanyam@hawk.iit.edu

A20566944

Spring 2025

Illinois Institute of Technology

Date: February 6, 2025

Table of Contents

List of Figures

List of Tables

List of Codes

Questions and Solutions

1 Question 1

Execute the commands ifconfig and route -n inside the course VM. Answer the following questions based on the output

```
ece448s25@iit:~$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.225.128  netmask 255.255.255.0  broadcast 172.16.225.255
        inet6 fe80::20c:29ff:fedb:e336  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:db:e3:36  txqueuelen 1000  (Ethernet)
        RX packets 738662  bytes 1076153910 (1.0 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 49752  bytes 5681228 (5.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 44  memory 0x3fe00000-3fe20000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9619  bytes 1795099 (1.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9619  bytes 1795099 (1.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 1: Execution of `ifconfig` command in the Linux terminal.

```
ece448s25@iit:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.225.2    0.0.0.0         UG    100    0        0 ens160
172.16.225.0    0.0.0.0         255.255.255.0   U     100    0        0 ens160
```

Figure 2: Execution of `route -n` command in the Linux terminal.

1.1 Sub-question 1

How many network interfaces are there and what are their names?

There are two network interfaces when the command `ifconfig` is run in the terminal as shown in Figure 1. These are:

1. ens160: This is the primary Ethernet interface which has the IP address 172.16.225.128.
2. lo: 127.0.0.14 is the IP address of the loopback interface. A unique virtual network interface called the loopback interface enables self-communication within the system.

1.2 Sub-question 2

List the MAC addresses of the network interfaces if there are any

There is only a single MAC address 00:0c:29:db:e3:36. This Ethernet interface is physical and has a valid MAC address.
Since loopback interfaces are utilized for internal communication, MAC addresses are not required.

1.3 Sub-question 3

What are the IP addresses and subnets associated with each interface?

The following are the IP address (IPv4 and IPv6) and subnet masks from Figure 1:

1. ens 160 Interface:
    a. IPv4 address - 172.16.225.128
    b. IPv4 Subnet Mask - 255.255.255.0
    c. IPv6 address - fe80::20c:29ff:fedb:e336/64

2. lo (Loopback Interface)
    a. IPv4 address - 127.0.01
    b. IPv4 Subnet Mask - 255.0.0.0
    c. IPv6 address - ::1/128

1.4 Sub-question 4

How many rules are there in the routing table? What is the address of the default gateway?

From the execution of command `route -n` as shown in Figure 2, the following information is gathered.

1. Routing Rules
    a. Default Route:
        i. Gateway: 172.16.225.2
        ii. Interface: ens160
        iii. If there are no other rules, this is the default gateway route for all traffic.
    b. Local Network Route:
        i. Gateway: 0.0.0.0 (direct connection)
        ii. Interface: ens160
        iii. Traffic inside the local subnet is managed by this rule.
2. Default Gateway

172.16.225.2 is the default gateway address, which is used to route traffic to locations outside the local network.

1.5 Sub-question 5

Which rule will apply if we need to send a packet to the default gateway?

The first rule in the routing table would be used to deliver a packet to the default gateway (172.16.225.2) is:

0.0.0.0 UG 100 0 0 ens160 172.16.225.2 0.0.0.0

With a destination of 0.0.0.0 and a netmask of 0.0.0.0, this rule is the default path. This is a gateway-based "up" route, as indicated by the UG flags. This default gateway will be used for any traffic that other routing rules do not specifically match.

2 Question 2

Research TCP and UDP header formats and explain why UDP is considered a lighter protocol than TCP in terms of packet size.

UDP and TCP header structure comparison:

Feature	TCP	UDP
Header Size	20-60 bytes	8 bytes (fixed)
Header Fields	Source Port, Destination Port, Sequence Number, Acknowledgment Number, Header Length, Reserved Fields, Code Bits, Window Size, Checksum, Urgent Pointer, Optional Fields	Source Port, Destination Port, Length, Checksum

Table 1: TCP and UDP comparison.

UDP is a lighter protocol than TCP primarily due to its simpler header structure and reduced overhead. This is because of its set 8-byte header size, which is substantially smaller than TCP's variable header size of 20–60 bytes, UDP can maintain a shorter footprint. The total packet size and processing demands are directly impacted by this size discrepancy.

TCP's three-way handshake and other connection formation procedures are superseded by UDP's connectionless nature. Furthermore, complicated control mechanisms like flow control, packet reordering, and acknowledgment systems are not present in UDP. Although reliability is sacrificed, UDP's streamlined structure makes it perfect for real-time applications where speed is essential.

3 Question 3

Why is DHCP required on a network? In what type of network would DHCP be beneficial? Research and explain how DHCP efficiently allocates IP addresses on a network.

Requirement of DHCP on a Network

Modern networks require DHCP (Dynamic Host Configuration Protocol) since it automates network configuration and IP address assigning. Without it, managers would have to individually set up the network settings on each device, which would take a lot of time and be prone to mistakes. By centralizing setup, avoiding IP conflicts, and dynamically managing address pools, the protocol makes network management easier. Large networks or settings with a lot of device connections and disconnections benefit greatly from this automation.

Benefits of DHCP

In public access points, huge enterprise networks, and educational institutions where many devices need network connectivity, DHCP is helpful. It is particularly useful in settings where there is a lot of device turnover, including coffee shops, hotels, and college campuses, where it would be time-consuming and difficult to manually configure IP addresses.

Allocation of IP addresses by DHCP on a network

By assigning temporary addresses from a predefined pool with predetermined lease times, DHCP effectively distributes IP addresses through dynamic allocation. Because IPs may be recovered and redistributed when devices disconnect, this guarantees effective address utilization. Clients broadcast discovery messages, receive offers from servers, request their chosen IP, and receive confirmation as part of the allocation process, which follows the DORA sequence (Discover, Offer, Request, Acknowledge). To guarantee dependable address allocation throughout the network, our automated procedure incorporates conflict detection.

4 Question 4

Assume that StringBuilder is unavailable for reversing the string (refer to the ReverseString class in Lecture 5). Write Java code to reverse the string without utilizing any libraries. Your function should identify the encoding

scheme of the string. Assume there are only two types: UTF-8 and UTF-16. You may need to research about how UTF-8 and UTF-16 encode strings.

The following is the Java code to reverse the string without utilizing any libraries:

```java
import java.nio.charset.StandardCharsets;
import java.lang.String;

public class ReverseString {
    public static String reverse(String input) {
        // Check null or empty string
        if (input == null) {
            return null;
        }
        if (input.isEmpty()) {
            return "";
        }

        // Convert string to character array
        char[] characters = input.toCharArray();
        int start = 0;
        int end = characters.length - 1;

        // Reverse character array
        while (start < end) {
            char temp = characters[start];
            characters[start] = characters[end];
            characters[end] = temp;
            start++;
            end--;
        }

        return new String(characters);
    }

    public static void main(String[] args) {
        String input = "Hello, World!";
        String reversed = reverse(input);
        System.out.println("Original: " + input);
        System.out.println("Reversed: " + reversed);
    }
}
```

Code 1: Java code for string reverse without library utilization.