Spring Semester 2020-2021

A Report for CSN-361: (Computer Networks Laboratory)

Submitted by

Subhash Suryawanshi (18114076)

ssuryawanshi@cs.iitr.ac.in

Department of Computer Science and Engineering

Indian Institute of Technology (IIT) Roorkee

May 27, 2021

Contents

1 Problem Statement 1

1.1 Question

Write a C++ program to print the MAC address of your computer, the host name and the IP address of your computer.

1.2 Theory

The MAC Address is a physical address that uniquely identifies each device on a network.It is a 6-byte-long (48bit) binary number. For the most part, computers list MAC addresses as 12-digit hexadecimal numbers.
Imagine MAC addresses are like phone numbers. So they are very unique as no two persons can have the same number as no two computers can have the same MAC address.
example of MAC ADDRESS 01-1A-3F-F1-4C-C6
IPV4 ADDRESS EXAMPLE 69.89.31.226
IPV6 ADDRESS EXAMPLE 2002:4559:1FE2::4559:1FE2
IP addresses are utilised as a number for labelling any internet-connected device utilised as a communication channel by Internet Protocol.
IPv4 is an abbreviation for Internet Protocol Version 4. The majority of the internet's networks and systems are presently set on IPv4.
IPv6 is an abbreviation for Internet Protocol Version 6. Because there is a chance that we will run out of IP addresses on the internet,
We utilised Hostnames to differentiate between devices connected to the network. This is the network device.

1.3 Implementation Details

A class named p1 contains the main function. for IP and host name in this main function, a 'try' and 'catch' block is defined. Within the 'try' block, an occasion of InetAddress is created and the 'getLocalHost' function is utilized to get the Host address and host name of the InetAddress instance. In case one of the attributes isn't found, the 'catch' piece characterizes catching the exception and printing the significant message on the comfort. For getting MAC address I created a method to urge the MAC addresses of the in that we to begin with make enumeration of Network Interface and store the values. call getHardwareAddress() method on each network and store the returned value in a byte array in the event that there are so numerous values I have given a while loop will store all the MAC address.

1.4 Results

```
PS C:\Users\91977\Desktop\CNL\18114076> cd P1
PS C:\Users\91977\Desktop\CNL\18114076\P1> javac p1.java
PS C:\Users\91977\Desktop\CNL\18114076\P1> java p1
Current MAC address : 9A-22-EF-75-6A-55
Current MAC address : 98-29-A6-46-3C-45
Current MAC address : 98-22-EF-75-6A-55
Current MAC address : AA-22-EF-75-6A-55
System IP Address : 192.168.43.221
The IP address is : 192.168.43.221
The host name is : LAPTOP-KV27U5BM
PS C:\Users\91977\Desktop\CNL\18114076\P1>
```

Results of the program we can see all the different MAC address and IP address with the hostname

2 Problem Statement 2

2.1 Question

Write a socket program in Java for PING command.

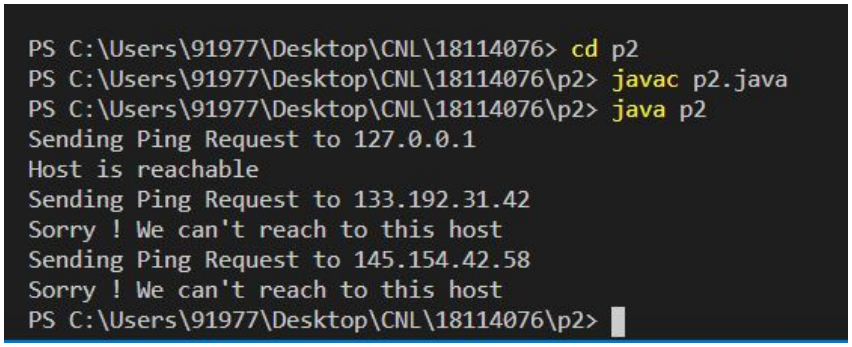
2.2 Theory

PING is a command-line application that can be found on almost every operating system that has network connectivity and is used to detect whether or not a networked device is accessible. The ping command sends a network request to a specific device. A successful ping sends a response back to the originating computer from the machine that was pinged. When you issue a ping command, an echo request packet is sent to the IP address you choose. When a remote host receives an echo request, it responds with an echo reply packet.

2.3 Implementation Details

In this program, we implement a java program utilizing sockets to make program that's able to ping ip addresses. Address in CIDR Notation - Using the CIDR notation 192.168.0.15/24, IP address 192.168.0.15 is associated with netmask 255.255.255.0. This indicates that the first 24 bits of an IP address are crucial for network routing. The attachment's true job is done by an occurrence of the SocketImpl java lesson. An application can design itself to produce attachments that are compatible with the local firewall by modifying the attachment plant that produces the attachment implementation.

2.4 Results



Results are for 127.0.0.1 ping we have a success full have so it prints out the message Host is reachable. and for the other ip 133.192.31.42 and 145.154.42.58 we will send the ask but the Host is unreachable.

3 Problem Statement 3

3.1 Question

Implement an error detection mechanism using the standard CRC algorithm. Write two programs: generator and verifier. The generator program reads from standard input an n-bit message as a string of 0's and 1's as a line of ASCII text. The second line is the k-bit polynomial, also in ASCII. It first checks that the polynomial is not divisible by x and x+1. If it is divisible by x or x+1, it outputs error else it outputs to standard output a line of ASCII text with n+k-1 0's and 1's representing the message to be transmitted. Then it outputs the polynomial, just as it read it in. The verifier program reads in the output of the generator program (if output is not error) and outputs a message indicating whether it is correct or not. Finally write a program, alter, that inverts one bit in the first line of the output of the generator depending on its argument, but copies rest of the first line and second line correctly. Now type the following and report the outcome.

1. generator ¡ file — verifier
2. generator ¡ file — alter arg — verifier

3.2 Theory

CRC is a checksum algorithm used to identify data inconsistencies, such as bit mistakes during data transfer. CRC is a division-based organisation. The real input data is read as a single lengthy binary stream (divident) split by a fixed binary integer (divisor). The balance The checksum value is the result of this division. Divident and divisor are modelled as binary polynomials, with the actual bits serving as coefficients. The fundamental mathematics employed for CRC computation is the XOR operation.

- A generator polynomial is a (n+1)-bit binary polynomial.
- Before performing the actual CRC computation, n zero bits are inserted to the input data.
- The value of the CRC checksum is specified as divident percent divisor.

3.3 Implementation Details

First, I developed the p3.cpp file. This programme will accept two binary arguments as input: the first is a non-binary message, The message character array is then added with K-0 bits. The programme will now check to see if the polynomial which can be divided by x and x+1. If a polynomial is divisible by any of them, the message "Error: divisor is divisible by x or x+1" is written, and the programme is terminated. If the polynomial is not divisible by any of them, the programme continues. First, we make the remaining equal to the dividend (message). We get the final remaining by iterating through the message, taking k bits each time. If the first bit of these k bits in the remainder is '1', we XOR these k bits with the divisor polynomial; otherwise, we XOR these k bits with the k zero bits.Each repetition, the matching bits of residual are updated. The checksum is the difference between the payout and the remainder. Print the polynomial as well as the resulting checksum. Using the same method as stated previously, calculate the final remaining. If this remainder is a binary array with a beginning n bit of zero, the message is properly delivered. Otherwise, the message has a 1 bit mistake and is not successfully delivered.

3.4 Results

```
$g++ -o main *.cpp

$main
Enter dataword and generator polynomial
Enter 1 to generate Codeword or 2 to verify or 3 to alter a bit of dataword or 4 to exit
Codeword : 100100
Enter 1 to generate Codeword or 2 to verify or 3 to alter a bit of dataword or 4 to exit
Remainder : 000
verified
Enter 1 to generate Codeword or 2 to verify or 3 to alter a bit of dataword or 4 to exit
Enter bit position to be altered
Bit at position 1 is altered
Enter 1 to generate Codeword or 2 to verify or 3 to alter a bit of dataword or 4 to exit
Remainder : 010
error in transmission
Enter 1 to generate Codeword or 2 to verify or 3 to alter a bit of dataword or 4 to exit
```

Result of Program we used the test case 100100,1001,1,2,3,1,2,3

4 Problem Statement 4

4.1 Question

Write a C++/Java program that accepts an IP address and subnet mask in CIDR notation, and print the following information about the sub-network:

1. Subnet Mask in dotted decimal notation (Example 255.0.0.0)
2. Network Address in dotted decimal notation: (Example 103.0.0.0)
3. Usable Host IP Range: Starting IP 103.0.0.1 — Ending IP 103.255.255.254

4.2 Theory

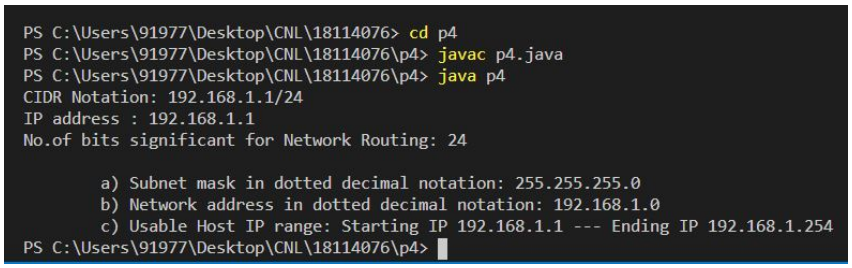
addresses allow network resources to be accessed through network interfaces. In this context, we're discussing IPv4 addresses, which are 32-bit numbers. Each byte is separated by a period and is represented as an integer between 0 and 255. 192.168.0.5 looks to be a prevalent IPv4

address.
IP Address in CIDR Notation - Using the CIDR notation 192.168.0.15/24, IP address 192.168.0.15 is associated with netmask 255.255.255.0. This indicates that the first 24 bits of an IP address are crucial for network routing..
IP Address in CIDR Notation - Using the CIDR notation 192.168.0.15/24, IP address 192.168.0.15 is associated with netmask 255.255.255.0. This indicates that the first 24 bits of an IP address are crucial for network routing.

4.3 Implementation Details

A method was made for conversion of IP address to binary form named it as (binary) that takes a String str (IP address). Same a function (decimal) was made for binary to decimal changes.

4.4 Results



Result of the Program P4

5 Problem Statement 5

5.1 Question

Write a program that an instructor can use to demonstrate the method of calculating IPv4 checksum. Your program should ask the user to enter the values of different fields of an IPv4 header. It should then calculate IPv4 checksum. Your program should not only show the final result but should also demonstrate the method (each step) to calculate the checksum.

5.2 Theory

The Internet Protocol version 4 (IPv4) is the fourth version of the Internet Protocol (IP). It is one of the foundational protocols for standards-based internetworking solutions on the Internet and other packet-switched networks. In 1982, IPv4 was the principal adaptation communicated for generation on the SATNET, and in January 1983, it was the principal adaptation communicated for generation on the ARPANET. Despite the continuing transmission of a successor convention, IPv6, which is utilised for the following step after 1 more bit is dropped down to make it n bits long, it nevertheless routes most Web traffic today[1]. We get a result when there are no more parts to draw down. The (n-1)-bit remainder inserted on the sender side

5.3 Implementation Details

The IP header checksum is the one's complement sum of the all the 16 bit words in the IP header in 16 bits. A example IP Header is "4500 003c 1c46 4000 4006 b1e6 ac10 0a63 ac10 0a0c", which would also be made up of 16 bit words. This header is interpreted in the binary system as:
0100010100000000 0000000000111100 0001110001000110 0100000000000000 0100000000000110 0000000000000000 1010110000010000 0000101001100011 1010110000010000 0000101000001100.
each of these 16 bit words are first generated using a function that converts the input word from hex to binary, Now these are added one by one such that te result is a 16bit word and any carryovers are added again to the result.now one's complement operation is implemented by switching one's to zeros and vice versa. Finally the generated checksum is printed out.

5.4 Results

```
PS C:\Users\91977\Desktop\CNL\18114076> cd p5
PS C:\Users\91977\Desktop\CNL\18114076\p5> javac p5.java
PS C:\Users\91977\Desktop\CNL\18114076\p5> java p5
Enter IP version, header length and TOS in hexadecimal format : 4500
IP version : 4, Header Length : 5, TOS : 00

Enter Total length Field of IP header : 0073
Adding above two 16 bit words gives result : 4573

Enter Identification field : 0000
Adding last result and next input gives result : 4573

Enter flags and fragment offset : 4000
Adding last result and next input gives result : 8573

Enter TTL field and protocol field : 4011
Adding last result and next input gives result : c584

Enter Header Checksum field (which is 0000 at destination end) : b861
Adding last result and next input gives result : 7de6

Enter Source IP Address : c0a8
Adding last result and next input gives result : 3e8f

Enter Destination IP Address : 0001
Adding last result and next input gives result : 3e90

Enter Options and Padding Fields :
First 16 bits : 00c7
Adding last result and next input gives result : 3f57

Next 16 bits : 00c7
Final result is : 401e

Complement the final result to get the Checksum

Checksum generated is bfe1
PS C:\Users\91977\Desktop\CNL\18114076\p5>
```

Result of Program P5

6 Problem Statement 6

6.1 Question

Write a C++/Java program for the Decibel (dB) calculator. You program should perform the following operations:
1) If transmit power of a network device is given in watts (Example: '20 W') then the program should print the transmit power in Decibel Watts (dBW) and Decibel Milliwatts (dBm).
2) Else if transmit power of a network device is given in Decibel Watts (dBW) or Decibel Milliwatts (Example: '20 dBW' or '20 dBm') then the program should print the transmit power in

Watts.

6.2 Theory

What we should do is make a fundamental power modification, such as Decibel Watts, Decibel Milliwatts, and Watts.

Equation to convert Watts into dBW

$$P_{dBW} = 10 * log_{10}(P_W) \tag{1}$$

where P_{dBW} is power in dBW and P_W is power in Watts.

Equation to convert Watts into dBm

$$P_{dBm} = 10 * log_{10}(P_W + 30) \tag{2}$$

where P_{dBm} is power in dBW and P_W is power in Watts.

Equation to convert dBW into Watts

$$P_W = 10^{(\frac{P_{dBW}}{10})} \tag{3}$$

where P_{dBW} is power in dBW and P_W is power in Watts.

Equation to convert dBm into Watts

$$P_W = 10^{(\frac{P_{dBW} - 30}{10})} \tag{4}$$

where P_{dBm} is power in dBm and P_W is power in Watts.

6.3 Implementation Details

Standard numerical formulae were utilized to change over values of signal power from one unit to another. The units Watt and decibal watts are related with a straightforward logarithmic equation

6.4 Results

```
PS C:\Users\91977\Desktop\CNL\18114076> cd p6
PS C:\Users\91977\Desktop\CNL\18114076\p6> javac p6.java
PS C:\Users\91977\Desktop\CNL\18114076\p6> java p6

 Unit of input power?
1.watt (W)
2.decibel watts (dBW)
3.decibel milliwatts (dBm)
1
enter transmit power Watt (W).....
10
 Watt (W) = 10.0 Decibel Watts (dBW)
 Watt (W) = 40.0 Decibel Milliwatts (dBm)
PS C:\Users\91977\Desktop\CNL\18114076\p6>
```

Result of the Program P4 we selected 1 for giving the input 10 Watts.