VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

D REVANTH(1BM21CS047)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
JUN-2023 to SEP-2023

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "LAB COURSE **COMPUTER NETWORKS**" carried out by **D REVANTH(1BM21CS047),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks - (22CS4PCCON)** work prescribed for the said degree.

M Lakshmi Neelima
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

Sl. No.	Date	Experiment Title	Page No.
1	15/6/23	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.	3-4
2	22/6/23	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	5-7
3	13/7/23	Configure default route, static route to the Router	8-10
4	13/7/23	Configure DHCP within a LAN and outside LAN.	11-14
5	20/7/23	Configure RIP routing Protocol in Routers	15-18
6	27/7/23	Configure OSPF routing protocol	19-20
7	10/08/23	Demonstrate the TTL/ Life of a Packet	21-24
8	20/7/23	Configure Web Server, DNS within a LAN.	25-26
9	3/08/23	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	27-28
10	11/08/23	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	29-30
11	11/08/23	To construct a VLAN and make PC's communicate among a VLAN	31-33
12	11/08/23	To construct a WLAN and make the nodes communicate wirelessly	34-36
13	17/08/23	Write a program for error detecting code using CRC-CCITT (16-bits).	37-43
14	17/08/23	Write a program for congestion control using Leaky bucket algorithm.	44-48
15	24/08/23	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	49-52
16	24/08/23	Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	53-56
17	26/08/23	Tool Exploration -Wireshark	57-58

EXPERIMENT-1

Question: Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

Aim: To create a topology with switch/hub and PC's and ping PC's

Topology:

Procedure:

Procedure:

1. Place a generic hub.
2. Place three PC's from end devices
3. Connect pc's to hub using copper Straight through.
4. Place a generic Switch
5. Place three PC's from end devices
6. Connect PC's to Switch using Copper Straight through.
7. Connect Switch and hub using Copper Cross-over.

Output:

Observation:

Observation:

1. Switch broadcasts the packet to all the devices during first iteration & records IP address of intended destination & sends packet to specified destination.
2. Hub broadcasts packet to all devices and the device which is intended to receive packet discards & intended device recieves packet & send the acknowledgement.

EXPERIMENT-2

Question: Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply

Aim: To understand working with Routers

Topology:

Procedure:

Procedure:

1. Connect 2 PC's and 3 routers using copper cross over cable for PC to router & Serial DCE cable to connect router to router.
2. Set IP address of both PC's and their gateway numbers.
3. Now for setting IP address & gateway numbers to routers. Select a router and perform following commands:
   (i) type no & press enter.
   (ii) type enable & press enter.
   (iii) type config t & press enter.
   (iv) type interface fastethernet 0/0 & press enter.
   (v) type IP address 10.0.0.10 255.0.0.0 & press enter

(vi) type no shut & press enter.
(vii) type exit
(viii) type interface se 2/0 & press enter.
(ix) type IP address 20.0.0.10 255.0.0.0 & press enter
(x) type no shut and press enter.
(xi) type exit
(xii) type exit.

4. Repeat these commands for other two routers as well with their respective IP/gateway addresses.
5. Now to introduce other two IP addresses to the first router we follow these steps.
   (i) type config t and press enter.
   (ii) type IP route 30.0.0.0 255.0.0.0 20.0.0.20.
   (iii) type IP route 40.0.0.0 255.0.0.0 20.0.0.20.
   (iv) type exit
   (v) type exit.
6. Repeat same steps for other two routers with their appropriate addresses.
7. Go to Command prompt of PC 0 & config tab. and ping the Second PC, PC 1.

Output:

Observation:

Observation:

When the routers are not introduced with other two IP addresses and we ping then we get a message saying host unreachable. Once when we introduce routers with other two IP addresses and we ping, now we get message sent sucessfully with 0 % loss.

13/7/2023

EXPERIMENT-3

Question: Configure default route, static route to the Router

Aim: To work on default and static routing

Topology:

Procedure:

Procedure:

1. Connect 3 routers and 2 PC's using copper cross over Cables for PC to router and Serial DCE Cable to connect router to router.
2. Set the IP address of both PC's and respective gateway.
3. For all 3 routers set the respective IP addresses in CLI mode using following commands.
   - (i) Enable
   - (ii) Config t
   - (iii) Interface fastethernet 0/0.
   - (iv) IP address 10.0.0.10 255.0.0.0.
   - (v) No Shut.
   - (vi) Exit.
   - (vii) Interface se 2/0
   - (viii) IP address 20.0.0.10 255.0.0.0.
   - (ix) No Shut

(x) exit
(xi) exit

4. Repeat these commands for other two routers.
5. For Router 1, set IP route of its adjacent routers statically
   (i) Config t.
   (ii) ip route 10.0.0.0 255.0.0.0 20.0.0.10.
   (iii) ip route 40.0.0.0 255.0.0.0 30.0.0.20.
   (iv) exit
   (v) exit
   (vi) Show ip route (to view route table).
6. We set default ip routes to router 0 & router 2 which tells it can access any ip addresses with any Subnetmask address.
7. Set default ip route by following commands.
   (i) Config t.
   (ii) ip route 0.0.0.0 0.0.0.0 20.0.0.20 (Router 0)
   (iii) ip route 0.0.0.0 0.0.0.0 30.0.0.10. (Router 1)
8. Go to PC0 Command prompt & give ping to send message.

Output:

Observation:

Observation:

A default route is given to routers which takes effect when no other route is available for an ip address destination. if packet is received, defra first checks ip destination address, if it is not available it checks its routing table. packet is forwarded to next hop towards destination. Process repeats until packet is delivered.

19/7/2023

EXPERIMENT-4

Question: Configure DHCP within a LAN and outside LAN.

Aim: To assign IP addresses dynamically

Topology:

Procedure:

Procedure:

1. Add a router, a switch & 2 PC's to previous program & connect the router to both switches.
2. Set the server IP address of server and with help of server set the first 3 PC's IP address through DHCP.
3. Now set router IP address with following commands
   (i) NO
   (ii) enable.
   (iii) Config t
   (iv) interface fastethernet 4/0
   (v) IP address 10.0.0.20 255.0.0.0
   (vi) no shut.
   (vii) exit
   (viii) interface fastethernet 0/0
   (ix) ip address 20.0.0.20 255.0.0.0
   (x) no shut
   (xi) exit
   (xii) exit
   (xiii) show ip route.
4. Now, go to server & set gateway as 10.0.0.20
5. Again go to router CLI & follow commands.
   (i) Config t
   (ii) interface fastethernet 0/0
   (iii) ip helper-address 10.0.0.1
   (iv) no shut
   (v) exit.
6. Now, go to server services & add one more pool name as server pool 1, start IP address as 20.0.0.2 & default gateway as 20.0.0.20. Then add & save.
7. Now set other two PC's IP address by going to their. Desktop → IP configuration & select DHCP which will automatically generate its IP address.
8. Now the network is complete & ready to send packets from PC to other by typing ping ipaddress in command prompt

Output:

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 20.0.0.3
Pinging 20.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

Observation:

Observation:

DHCP is used to assign IP addresses dynamically to different devices. To assign continuous IP address we create a server pool where assign the starting IP address and a default gateway number. For PC's under different switches we create a different server pool again and start. This takes care of delivering the packets to correct destination IP address & also sends back ack to initial device.

19/7/2023

EXPERIMENT-5

Question: Configure RIP routing Protocol in Routers

Aim: Understanding how RIP protocol is used and how it works

Topology:

Procedure:

Procedure:

1. Create a Network using 3 routers & 2 PC's. Routers are connected using Serial DCE Cable & connection between PC & router is using copper-cross over cable.
2. Set IP address & gateway of both PC's
   10.0.0.1 10.0.0.10 — PC 0.
   40.0.0.1 40.0.0.10 — PC1 respectively.
3. Go to CLI of Routers & follow these commands.
   (i) No
   (ii) Enable
   (iii) config t
   (iv) interface fa 0/0
   (v) ip address 10.0.0.10 255.0.0.0.
   (vi) no Shut
   (vii) Exit
   (viii) interface Serial 2/0

(ix) ~~interface~~ ip address 20.0.0.10 255.0.0.0.
(x) Encapsulation ppp.
(xi) clock rate 64000.
(xii) no shut.

4. Go to router 0 CLI & follow these steps:
   (i) config T
   (ii) router rip.
   (iii) network 10.0.0.0
   (iv) network 20.0.0.0
   (v) exit.
5. Repeat these to all routers.
6. Go to PC 0 command prompt and ping PC 1.

Output:

pc > ping 40.0.0.1
pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1 : bytes = 32 time : 8ms TTL = 125
Reply from 40.0.0.1 : bytes = 32 time : 5ms TTL = 125
Reply from 40.0.0.1 : bytes = 32. time : 10 ms TTL = 125

Ping Statistics for 40.0.0.1 :
Packets : sent 4 , Received = 3 , Lost = 1 (25% loss).
Approximate round trip times in milliseconds.
Minimum = 5 ms , Maximum = 10ms , Average = 7ms.

Output:

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=5ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 10ms, Average = 7ms
PC>

Observation:

Observation:

Routing Information Protocol (RIP) is a dynamic routing protocol that uses hop count as routing metric to find best path between source & destination.

It is a distance vector routing protocol.

Hop count is no. of routers between source & destination.

EXPERIMENT-6

Question: Configure OSPF routing protocol

Aim: Understanding how OSPFprotocol is used and how it works

Topology:

Procedure:

Procedure:

1. Configure routers & PCs with IP address and gateway according to topology shown above.
2. Configure each router according to topology.
3. Encapsulation ppp and clock rate need to be set as done in RIP protocol experiment.
4. To set protocol follow these steps:
   (i) ~~router ospf 1~~ R1(config) #router ospf 1
   (ii) R1(config-router) #router-id 1.1.1.1
   (iii) #network 10.0.0.0 0.255.255.255 area 3
   (iv) #network 20.0.0.0 0.255.255.255 area 1.
5. Repeat the same following steps to each individual routers with their respective addresses.
6. Set loopbacks to the router by following these steps.
   (i) R1(config-if) #interface loopback 0.
   (ii) R1(config-if) # ip add 172.16.1.252 255.255.0.0.
   (iii) R1(config-if) # no shut
7. Repeat the same for all other routers by changing ip address 172.16.1.253, 172.16.1.254 ....
8. Create a virtual link between R1, R2 as area 3 is not

Connected to the backbone

(i) R1(config)# router ospf 1

(ii) R1(config-router)# area 1 virtual-link 2.2.2.2 → id of router

9. Now execute commands in router 2.

(i) R2(config)# router ospf 1

(ii) R2(config-router)# area 1 virtual-link 1.1.1.1 → id of router

Output:

Observation:

Observation:

Ospf is link state routing protocol which used to find the best path between source and destination router using its own shortest path algorithm. This is divided into areas where area 0 is the backbone and after we make a virtual link between the area which is not directly connected to the backbone area. We can ping the destination successfully.

26/9/2023.

EXPERIMENT-7

Question: Demonstrate the TTL/ Life of a Packet

Aim: Observing how TTL changes with each hop

Topology:

Procedure:

Procedure:

1. Create the above topology.
2. Goto simulation mode, select simple PDU. and send. message from PC0 to PC1
3. Click on capture button each time & Check the packet header each time.
4. Each time when capture is clicked you can observe the simulation of packet from 1 device to another.
5. Click on the packet each time and observe the header file for TTL.

Output:

30.0.0.20
PDU Information at Device: PC0
OSI Model
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 0090.2137.34B0
SRC MAC: 00D0.BC33.1653
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x10
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
Fa0

PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xf
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x10
SEQ NUMBER: 15
Fa

40.0.0.10
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xf
0x0
0x0
TTL: 253
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x10
SEQ NUMBER: 15
Fa0

Observation:

Observation:

We can observe that when we simulate a ping message using simple PDU we can click on the packet and check the headers of packet when clicked on capture button you can see the transition of packet. For every hop we can observe that the value of TTL in IP header decreases by 1 value.

11/8/2023

EXPERIMENT-8

Question: Configure Web Server, DNS within a LAN

Aim: To change web server IP to a Domain Name

Topology:

Procedure:

Procedure:

1. Connect a switch, PC and a Server to form a LAN
2. Set PC's IP address to 10.0.0.1 and Server's IP address to 10.0.0.2. and also Subnet mask
3. Go to PC's Desktop & open web browser, in URL tab type 10.0.0.2, which is servers IP.
4. Make changes to the webpage.
5. Go to Server → services → HTTP → Index.html, click on edit make changes and save it.
6. Go to PC web browserer and type 10.0.0.2.
7. Go to server services → DNS & switch on DNS services. Now add a Domain name & its IP address which is 10.0.0.2. Press add & save it.
8. Go to PC web browser & type the domain name given & click go.

Output:

Observation:

Observation:

To reach a webpage you need to type its IP address but what we actually type is its DNS which is called domain name. Since we cannot remember IP address all the time DNS server will search through for a matching IP address for respective domain name & retrieves the webpage.

24/2/2023

EXPERIMENT-9

Question: To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Aim: To know how ARP Protocol works to obtain IP addresses

Topology:

Procedure:

Procedure:

1. Create a topology of 3 PC's and server.
2. Assign IP address to all devices.
3. Connect them through a switch.
4. Using inspect tool to click on PC & server to see ARP Table.
5. In command prompt of PC type arp -a.
6. In CLI of switch type arp show mac address-table.

Can be given on transaction, you can see a address-table.

Output:

Observation:

Observation:
When we ping one PC and server the address of server is known to PC and vice versa. When we ping between other two PC's similarly the addresses of each other are known. by ARP protocol.
~~When~~ As switch is the common between all, so when you see the mac address table you can see all the PC and server mac addresses.

3/8/2023

EXPERIMENT-10

Question: To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

Aim: To remotely access router by PC connected to it, using TELNET

Topology:

Procedure:

Procedure:

1. Create the above topology with a PC and Router.
2. Set gateway of PC to 10.0.0.1 and IP address of PC to 10.0.0.2.
3. Go to CLI mode of router and set the IP address of interface Fa 0/0 to 10.0.0.1.
4. After setting up all the topology.
5. Go to router and enable config mode.
6. Type hostname r1, then type enable secret p1 here p1 is password of your choice.
7. Then type IP address 10.0.0.1 & execute no shut command.
8. In interface mode itself execute line vty 0 5 command. which creates 6 lines and next execute login and in next command. password p0 is. the password for login of. router.
9. Go to command prompt of PC & execute telnet 10.0.0.1 which will get you administration access of router. Enter password & access router from a connected end device.

Output:

Observation:

Observation:

We can observe in this experiment that when a hostname & password of a router is created & lines of them is created. Upto mentioned number of devices can access it using the password & its gateway. Once access is received a PC here is remotely able to change the setting of a router.

11/8/2023

EXPERIMENT-11

Question: To construct a VLAN and make PC's communicate among a VLAN

Aim: To create a sub-network and PC's communicate among VLAN

Topology:

Procedure:

Procedure:

1. Create above topology with 2 pcs in one set & other 2 PC's in other set with IP addresses 192.168.1.2 , 192.168.1.3 of one set & 192.168.20.2 , 192.168.20.3 for other set.
2. Connect all 4 PC's to switch & now connect this router to switch.
3. Now create a VLAN database in switch with a VLAN number and a VLAN name.
4. Now goto routers CLI mode
5. Select the interface fastethernet 0/5 which is towards or between switch & router. and make VLAN trunk
6. Go to the PC's which are set of VLAN & set it ethernet to new VLAN.

RANKA
DATE / /
PAGE

7. Goto CLI of router, change it to config mode.
   Router(config)# interface fastEthernet 0/0.1
   Router(config-subif)# encapsulation dot1q 2
   Router(config-subif)# ip address 192.168.20.1 255.255.255.
   Router(config-subif)# no shut.
8. This will create a sub interface 0/0.1 setting 192.168.20.1 as its gateway.
9. Now ping a pc from one network to another

Output:

Observation:

Observation:
From a network which is connected to one interface can be further divided into sub networks called virtual LAN. When created a new sub interface devices connected to it act like a different network, now after setting up, we can communicate between two networks.

EXPERIMENT-12

Question: To construct a WLAN and make the nodes communicate wirelessly

Aim: To connect devices wirelessly and they communicate among themselves.

Topology:

Procedure:

Procedure:

1. Create the above topology.
2. Now set the gateway and ip address for all the PC's.
3. connect router, Access point and PC directly to a Switch.
4. Before Setting IP address of PC1 and Laptop0 switch them off and. Change the LAN port with wireless modem.
5. Configure Port 1 of Access point, provide SSID Name, channel size and password.
6. Now go to PC & Laptops wireless interface in config tab & Connect it to created WLAN.
7. Now the topology is ready. Go to Command prompt of directly connected PC and ping the wireless one

Output:

Observation:

Observation:

In this experiment we could observe that we can communicate between end devices wirelessly rather than traditional wired connection, after creating a WLAN using a Access point & connecting end devices wirelessly.

11/8/2023.

EXPERIMENT-13

Question: Write a program for error detecting code using CRC- CCITT (16-bits).

Program:

P/08/23

Lab - 13.

Aim: Write a program for error detection using CRC=CCITT (16-bits).

```
#Include < stdio.h>
int arr[17];

void xor (int x[], int y[])
{
    int k=0;
    for (int i=1; i<16; i++)
    {
        if(x[i]==y[i])
            arr[k++]=0;
        else
            arr[i]=1;
    }
}.

void main(){
int dd[17], div[33], ze[17], i, k;
printf("Enter dataword \n");
for (i=0; i<17; i++)
    scanf("%d", &div[i]);
for (i=17; i<33; i++)
    div[i]=0;
for (i=0; i<17; i++)
    ze[i]=0;
printf("Enter dividend\n");
    for (i=0; i<17; i++)
        scanf("%d", &dd[i]);
i=0;
k=0;
```

```
for (i=i; i<17; i++)
    arr[k++] = div[i];
while (i<33) {
    if(arr[0] = =0)
        xor(arr, ze);
    else
        xor(arr, dd);
    arr[16] = div[i++];
}
k=0;
for(i=17; i<33; i++)
    div[i]=arr[k++];
printf("codeword" );
  for(i=0; i<33 ;i++)
printf(" %d ",div[i]);
  for(i=0; i<17 ; i++)
      arr[i] = 0;
printf("\n At receiver end \n");
k=0;
for(i=i; i<17 ; i++)
    arr[k++]= div[i];
while(i<33) {
  if (arr[0]= =0)
      xor(arr, ze);
  else
      xor(arr, dd);
    arr[16]= div[i++];
}
k=0;
for (i=17; i<33; i++)
    div[i]=arr[k++];
```

```
printf("Codeword: ");
for (i=0;i<33;i++)
  printf("%d", div[i]);
}
```

Output:

Enter dataword

1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1

Enter dividend

1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1

Codeword: 10110011110010111000000000000011011

At receiver end.

Codeword: 10110011110010111 000000000000000000.

19/8/2023

Code:

```
#include<stdio.h>

int arr[17];


void xor(int x[], int y[])

{

    int k=0;

    for(int i=1;i<16;i++)

    {

        if(x[i]==y[i])

            arr[k++]=0;

        else

            arr[i]=1;

    }

}


void main()

{

    int dd[17],div[33],ze[17],i,k;


    printf("Enter the dataword \n");

    for(i=0;i<17;i++)

        scanf("%d",&div[i]);


    for(i=i;i<33;i++)
```

```
    div[i]=0;


for(i=0;i<17;i++)
    ze[i]=0;
printf("Enter dividend \n");
for(i=0;i<17;i++)
    scanf("%d",&dd[i]);


i=0;
k=0;
    for(i=i;i<17;i++)
        arr[k++]=div[i];
while(i<33)
{
    if(arr[0]==0)
        xor(arr,ze);
    else
        xor(arr,dd);


    arr[16]=div[i++];


}
k=0;
for(i=17;i<33;i++)
    div[i]=arr[k++];
```

```
printf("Codeword: ");
    for(i=0;i<33;i++)
        printf("%d",div[i]);

for(i=0;i<17;i++)
    arr[i]=0;

printf("\nAt receiver end \n");

 k=0;
    for(i=i;i<17;i++)
        arr[k++]=div[i];
while(i<33)
{
    if(arr[0]==0)
        xor(arr,ze);
    else
        xor(arr,dd);

    arr[16]=div[i++];

}
k=0;
for(i=17;i<33;i++)
    div[i]=arr[k++];
```

```
    printf("Codeword: ");
        for(i=0;i<33;i++)
            printf("%d",div[i]);
}
```

Output:

EXPERIMENT-14

Question: Write a program for congestion control using Leaky bucket algorithm.

Program:

17/08/23

Lab – 14.

Aim: Write a program for congestion control using leaky bucket algorithm.

Code:

```
#include <stdio.h>

void main() {
 int buckets, outlets, k=1, num, remaining;
printf("Enter Bucket Size & outstream size \n");
scanf("%d %d", &buckets, &outlets);
remaining = buckets;
while (k){
 num = rand() % 1000;
 if (num < remaining) {
   remaining = remaining - num;
 printf(" Packet of %d bytes accepted \n", num);
 }
 else
   printf(" Packet of %d bytes is discarded \n", num);
 if (buckets - remaining > outlets)
      remaining += outlets;
 else
      remaining = buckets;
 printf(" Remaining bytes : %d \n", remaining);
 printf(" If you want to Stop input . press 0, otherwise, \n");
 scanf("%d", &k);
}.
while (remaining < buckets) {
   if (buckets - remaing > outlets.
          remaining += outlets;
   else
```

```
        remaining = buckets;
    printf("Remaining bytes: %d \n", remaining);
    }
}
```

Output:

```
Enter Bucket size & outstream size  2000  100.
Packet of 41 bytes accepted.
Remaining bytes: 2000.
If you want to stop input, press 0, otherwise, press 1.
Packet of 467 bytes accepted.   1
Packet of 334 bytes accepted
Remaining bytes : 1399.   0.
Remaining bytes : 1499
Remaining bytes : 1599
Remaining bytes : 1699
Remaining bytes : 1799
Remaining bytes : 1899
Remaining bytes : 1999
Remaining bytes : 2000.
```

19/8/2023

Code:

```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function

int main()
{
   int buckets, outlets, k = 1, num, remaining;
   printf("Enter Bucket size and outstream size\n");
   scanf("%d %d", &buckets, &outlets);
   remaining = buckets;
   while (k)
   {
      num = rand() % 1000; // Generate a random number between 0 and 999
      if (num < remaining)
      {
         remaining = remaining - num;
         printf("Packet of %d bytes accepted\n", num); // Added missing variable
      }
      else
      {
         printf("Packet of %d bytes is discarded\n", num);
      }
      if (buckets - remaining > outlets)
      {
         remaining += outlets; // Fixed the calculation
      }
      else
         remaining = buckets;
      printf("Remaining bytes: %d \n", remaining);
      printf("If you want to stop input, press 0, otherwise, press 1\n");
      scanf("%d", &k);
   }

   while (remaining < buckets) // Fixed the condition
```

```
    {
        if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else
            remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
    }
    return 0; // Added a return statement to indicate successful completion
}
```

Output:

```
PS D:\VS Code> cd "d:\VS Code\OS\" ; if ($?) { gcc bucket.c -o bucket } ; if ($?) { .\bucket }
Enter Bucket size and outstream size
2000
100
Packet of 41 bytes accepted
Remaining bytes: 2000
If you want to stop input, press 0, otherwise, press 1
1
Packet of 467 bytes accepted
Remaining bytes: 1633
If you want to stop input, press 0, otherwise, press 1
1
Packet of 334 bytes accepted
Remaining bytes: 1399
If you want to stop input, press 0, otherwise, press 1
1
Packet of 500 bytes accepted
Remaining bytes: 999
If you want to stop input, press 0, otherwise, press 1
1
Packet of 169 bytes accepted
Remaining bytes: 930
If you want to stop input, press 0, otherwise, press 1
1
Packet of 724 bytes accepted
Remaining bytes: 306
If you want to stop input, press 0, otherwise, press 1
1
Packet of 478 bytes is discarded
Remaining bytes: 406
If you want to stop input, press 0, otherwise, press 1
1
Packet of 358 bytes accepted
Remaining bytes: 148
If you want to stop input, press 0, otherwise, press 1
1
Packet of 962 bytes is discarded
Remaining bytes: 248
If you want to stop input, press 0, otherwise, press 1
0
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
```

```
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remaining bytes: 1148
Remaining bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remaining bytes: 1548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Remaining bytes: 1948
Remaining bytes: 2000
PS D:\VS Code\OS>
```

EXPERIMENT-15

Question: Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

Program:

24/08/25

Lab – 15

Aim: Using TCP/IP sockets, write a client-server program to make client sending file name & server to send back the contents of requested file if present.

Code:

Client TCP.py.

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input ("Enter file name: ")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode().
print("\n From Server: \n")
print(fileContents)
clientSocket.close().
```

ServerTCP.py.

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000.
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode().

    file = open(sentence, "r"),
```

```
l= file . read (1024)

connectionSocket send (l.encode()) .
print("\n Sent Contents of " + Sentence)
file.close()
connectionSocket.close().
```

28/5/2023

Code:

Server.py

```
from socket import *
serverName= '127.0.0.1'
serverPort= 12000
serverSocket= socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print("Ther Server is ready to receive")
    conectionSocket,addr=serverSocket.accept()
```

```
    sentence=conectionSocket.recv(1024).decode()

    file=open(sentence,"r")
    l=file.read(1024)

    conectionSocket.send(l.encode())
    print("\nSent contets of "+sentence)
    file=close()
    connectionSocket.close()
```

Client.py

```
from socket import *
serverName='127.0.0.1'
serverPort=12000
clientSocket=socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence=input("\nEnter file name: ")

clientSocket.send(sentence.encode())
filecontents=clientSocket.recv(1024).decode()
print('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

Output:

clienttcp.py - C:/Users/Admin/Desktop/1BM21CS047/clienttcp.py (3.10.8)
File Edit Format Run Options Window Help
from socket import *
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/Admin/Desktop/1BM21CS047/clienttcp.py ==========
Enter file name: servertcp.py
From Server:
from socket import *
serverName= '127.0.0.1'
serverPort= 12000
serverSocket= socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
print("Ther Server is ready to receive")
conectionSocket,addr=serverSocket.accept()
sentence=conectionSocket.recv(1024).decode()
file=open(sentence,"r")
l=file.read(1024)
conectionSocket.send(l.encode())
print("\nSent contets of "+sentence)
file=close()
connectionSocket.close()
>>>
Ln: 29 Col: 0
Ln: 6 Col: 37
servertcp.py - C:/Users/Admin/Desktop/1BM21CS047/servertcp.py (3.10.8)
File Edit Format Run Options Window Help
from socket import *
serverName= '127.0.0.1'
serverPort= 12000
serverSocket= socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
print("Ther Server is ready to receive")
conectionSocket,addr=serverSocket.accept()
sentence=conectionSocket.recv(1024).decode()
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/Admin/Desktop/1BM21CS047/servertcp.py ==========
Ther Server is ready to receive
Sent contets of servertcp.py
Traceback (most recent call last):
File "C:/Users/Admin/Desktop/1BM21CS047/servertcp.py", line 17, in <module>
file=close()
TypeError: _socket.close() takes exactly one argument (0 given)
>>>
Ln: 10 Col: 48
26°C
Mostly cloudy
Search
ENG
IN
10:40
24-08-2023

EXPERIMENT-16

Question: Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

Program:

24/08/23. Lab – 16.

Aim: Using UDP sockets, write a client-server program to make client sending the file name & server to send back the contents of requested file if present.

Code:

ClientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence, "utf-8"), (serverName, serv
fileContents, serverAddress = clientSocket.recvfrom(2048)
print("\nReply from server.\n")
print(fileContents.decode("utf-8"))
clientSocket.close()
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048).
```

```
serverSocket.sendto(bytes(lon,"utf-8"), clientAddress).

print('\n Sent contents of ', end = ' ')
print(sentence)
file.close()
```

28/8/2023

Code:

Server.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('127.0.0.1', serverPort))
print ('The server is ready to receive')
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode('utf-8')
    file=open(sentence,"r")
    con=file.read(2048)

    serverSocket.sendto(bytes(con,'utf-8'),clientAddress)

    print ('\nSent contents of ', end = '')
    print (sentence)
    # for i in sentence:
        # print (str(i), end = '')
    file.close()
```

Client.py

```
from socket import *
serverName = '127.0.0.1'


serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input('\nEnter file name: ')
clientSocket.sendto(bytes(sentence,'utf-8'),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ('\nReply from Server:\n')
print (filecontents.decode('utf-8'))
clientSocket.close()
clientSocket.close()
```

Output:

EXPERIMENT-17

Tool Exploration -Wireshark

Lab - 17

Aim: Tool Exploration - Wireshark.

Wireshark is an open-source packet analyzer, which is used for education, analysis, software development communication protocol development and network troubleshooting. It is used to track packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer and network analyzer. It is also used by network security engineers to examine security problems.

Wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as free packet sniffer computer application. It puts the network card into an unselective mode i.e, to accept all packets which it receives.

Uses of Wireshark:

Wireshark can be used in the following ways:

1. It is used by network security engineers to examine security problems.
2. It allows the users to watch all the traffic being passed over the network.
3. It is used by network engineers to troubleshoot network issues.
4. It also helps to troubleshoot latency issues & malicious activities on your network.
5. It can also analyze dropped packets.
6. It helps us to know how all the devices like laptop, mobile phones, desktop, switch, routers etc. communicate in a local network & the rest of the world.

Functionality of Wireshark:

Wireshark is similar to tcpdump in networking. Tcpdump is a common packet analyzer which allows the user to display other packets and TCP/IP packets, being transmitted and received over a network attached to computer. It has a graphic end and some sorting & filtering functions. Wireshark users can see all traffic passing through the network.

Wireshark can also monitor the unicast traffic which is not sent to network's MAC address interface. But the switch does not pass all traffic to the port. Hence, the promiscuous mode is not sufficient to see all the traffic. Port mirroring is a method to monitor network traffic. When it is enabled, the switch sends the copies of all network packets present at one port to another port.

28/9/2020 Features of wireshark.

1. It is multi-platform software, i.e, it can run on Linux, Windows, OS X, Free BSD, Net BSD, etc.
2. It is a Standard three pane packet browser.
3. It performs deep inspection of hundreds of protocols.
4. It often involves live analysis, i.e, from different types of network like Ethernet, loopback, etc, we can read live data.
5. It has sort and filter options which makes ease to user to view the data.
6. It is also useful in VoIP analysis.
7. It can also capture raw USB traffic.