

## 1.COMMANDS IN LINUX

### ip addr

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 6c:4b:90:c1:88:ea brd ff:ff:ff:ff:ff:ff
    inet 192.168.15.168/23 brd 192.168.15.255 scope global noprefixroute enp1s0
        valid_lft forever preferred_lft forever
    inet6 fe80::661e:6771:1a19/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:30:a7:58 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:30:a7:58 brd ff:ff:ff:ff:ff:ff
student@H68:~/Downloads$
```

### ping

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ ping mindmajix.com
PING mindmajix.com (108.158.251.3) 56(84) bytes of data.
64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=1 ttl=249 time=16.0 ms
64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=2 ttl=249 time=16.0 ms
64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=3 ttl=249 time=15.9 ms
64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=4 ttl=249 time=15.9 ms
64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=5 ttl=249 time=15.9 ms
64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=6 ttl=249 time=15.9 ms
64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=7 ttl=249 time=15.7 ms

64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=8 ttl=249 time=15.7 ms
64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=9 ttl=249 time=16.2 ms
64 bytes from server-108-158-251-3.maa51.r.cloudfront.net (108.158.251.3): icmp_seq=10 ttl=249 time=15.9 ms
```

## dig

```
student@H68: ~/Downloads
student@H68:~/Downloads$ dig www.geeksforgeeks.org

; <>> DiG 9.11.3-lubuntu1.11-Ubuntu <>> www.geeksforgeeks.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21098
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.geeksforgeeks.org.      IN      A

;; ANSWER SECTION:
www.geeksforgeeks.org.  245      IN      CNAME   www.geeksforgeeks.org.edgesuite.net.
www.geeksforgeeks.org.edgesuite.net. 7199  IN  CNAME  a1991.dscr.akamai.net.
a1991.dscr.akamai.net. 19      IN      A       49.44.130.8
a1991.dscr.akamai.net. 19      IN      A       49.44.130.32

;; Query time: 61 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Wed Jul 26 12:22:57 IST 2023
```

## netstat

```
student@H68: ~/Downloads
student@H68:~/Downloads$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp        0      0 H68:51556              239.237.117.34.bc:https ESTABLISHED
tcp        4      0 H68:34090              192.168.15.:netbios-ssn ESTABLISHED
tcp        0      0 H68:55242              55.65.117.34.bc.g:https ESTABLISHED
tcp        0      0 H68:35352              maa05s12-in-f5.1e:https ESTABLISHED

Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type      State           I-Node Path
unix    2      [ ]        DGRAM               35952  /run/user/1003/system
d/notify
unix    2      [ ]        DGRAM               29821  /run/user/120/systemd
/notify
unix    3      [ ]        DGRAM               16057  /run/systemd/notify
unix    9      [ ]        DGRAM               16067  /run/systemd/journal/
socket
unix    2      [ ]        DGRAM               16085  /run/systemd/journal/
syslog
unix   22      [ ]        DGRAM               16109  /run/systemd/journal/
dev-log
unix    3      [ ]        STREAM   CONNECTED     46464  @/dbus-vfs-daemon/soc
ket-nsu21Elj
unix    3      [ ]        STREAM   CONNECTED     33769
unix    3      [ ]        STREAM   CONNECTED     32445  /run/systemd/journal/
stdout
unix    3      [ ]        STREAM   CONNECTED     24560
unix    3      [ ]        STREAM   CONNECTED     34964  /var/run/dbus/system_
bus_socket
unix    3      [ ]        STREAM   CONNECTED     37329
unix    3      [ ]        STREAM   CONNECTED     36230
unix    3      [ ]        STREAM   CONNECTED     33167
unix    3      [ ]        STREAM   CONNECTED     30572
unix    3      [ ]        STREAM   CONNECTED     28478
```

## host

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ host geeksforgeeks.org
geeksforgeeks.org has address 34.218.62.116
geeksforgeeks.org mail is handled by 1 aspmx.l.google.com.
geeksforgeeks.org mail is handled by 10 alt3.aspmx.l.google.com.
geeksforgeeks.org mail is handled by 10 alt4.aspmx.l.google.com.
geeksforgeeks.org mail is handled by 5 alt1.aspmx.l.google.com.
geeksforgeeks.org mail is handled by 5 alt2.aspmx.l.google.com.
student@H68:~/Downloads$
```

## arp

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ arp
Address          HWtype  HWaddress          Flags Mask   Iface
_gateway         ether    00:1a:8c:6b:54:cc  C      enp1s
0               ether    40:f2:e9:d7:1d:95  C      enp1s
0               ether    9c:b6:54:0d:2c:38  C      enp1s
puppet          ether    00:0c:29:4f:4d:90  C      enp1s
0               ether    00:0c:29:4f:4d:90  C      enp1s
student@H68:~/Downloads$
```

## iwconfig

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ iwconfig
enp1s0    no wireless extensions.

lo       no wireless extensions.

virbr0    no wireless extensions.

virbr0-nic  no wireless extensions.

student@H68:~/Downloads$
```

## route

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
default         gateway        0.0.0.0       UG    100    0      0 enp1s0
link-local      0.0.0.0       255.255.0.0   U     1000   0      0 virbr0
192.168.14.0   0.0.0.0       255.255.254.0 U     100    0      0 enp1s0
192.168.122.0  0.0.0.0       255.255.255.0 U     0      0      0 virbr0
student@H68:~/Downloads$
```

## hostname

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ hostname
H68
student@H68:~/Downloads$
```

## mtr

```
student@H68:~/Downloads$ mtr google.com
student@H68:~/Downloads$
```

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
My traceroute [v0.92]
H68 (192.168.15.168) 2023-07-26T12:27:46+0530
Keys: Help Display mode Restart statistics Order of fields quit
      Packets          Pings
      Host           Loss%   Snt    Last    Avg  Best Wrst StDev
1. gateway          0.0%   18     0.4    0.4   0.3   0.6   0.1
2. 136.232.57.109  0.0%   18     1.5    1.7   1.5   2.2   0.2
3. 172.20.97.57   0.0%   17    14.1   14.4  14.1  15.1   0.2
4. 172.27.9.126   0.0%   17    15.5   15.9  15.5  17.9   0.5
5. 172.27.9.125   0.0%   17    15.6   15.9  15.3  20.5   1.2
6. 172.17.97.221  0.0%   17    14.2   14.3  14.0  15.0   0.2
7. 172.16.5.90    0.0%   17    15.4   15.4  15.2  15.7   0.1
8. maa05s14-in-f14.1e100.net 0.0%   17    15.3   15.5  15.2  15.7   0.1
```

### **3.TRANSMISSION CONTROL PROTOCOL**

SERVER PROGRAM:

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <arpa/inet.h>

int main(void)
{
    // initialize variables
    int socket_desc, client_sock, client_size;
    struct sockaddr_in client_addr, server_addr;
    char client_msg[2000], server_msg[2000], temp;
    // clean buffer that had been initialized
    memset(client_msg, '\0', sizeof(client_msg));
    memset(server_msg, '\0', sizeof(server_msg));
    // create socket
    socket_desc = socket(AF_INET, SOCK_STREAM, 0);
    if (socket_desc < 0)
    {
        printf("Unable to create socket\n");
        return -1;
    }
    printf("Socket created successfully\n");
    // set port and ip
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    server_addr.sin_port = htons(3000);
    int bvalue = 1020;
    bvalue = bind(socket_desc, (struct sockaddr *)&server_addr, sizeof(server_addr));
    // bind to set port and ip
    if (bvalue < 0)
    {
        printf("Binding is not possible\n");
        return -1;
    }
    // listen for client
    if (listen(socket_desc, 1) < 0)
    {
        printf("Error\n");
```

```

        return -1;
    }
while (1)
{
    // accept an incoming connection
    client_size = sizeof(client_addr);
    client_sock = accept(socket_desc, (struct sockaddr *)&client_addr, &client_size);
    if (client_sock < 0)
    {
        printf("Can't accept");
        return -1;
    }
    // receive and send message (client <-> server)
    if (recv(client_sock, client_msg, sizeof(client_msg), 0) < 0)
    {
        printf("Couldn't receive\n");
        return -1;
    }
    printf("Message from client: %s\n", client_msg);
    printf("Enter message: ");
    gets(server_msg);
    if(send(client_sock, server_msg, strlen(server_msg), 0)<0) {
        printf("Can't send\n");
        return -1;
    }
    memset(client_msg, '\0', sizeof(client_msg));
}
close(client_sock);
close(socket_desc);
return 0;
}

```

### **3.TRANSMISSION CONTROL PROTOCOL**

CLIENT PROGRAM:

```

#include<stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <arpa/inet.h>
int main(void) {
    // initialize variables
    int client_sock;

```

```

struct sockaddr_in client_addr, server_addr;
char client_msg[2000], server_msg[2000];
while(1) {
    // clean buffer that had been initialized
    memset(client_msg, '\0', sizeof(client_msg));
    memset(server_msg, '\0', sizeof(server_msg));
    // create socket
    client_sock = socket(AF_INET, SOCK_STREAM, 0);
    if(client_sock < 0) {
        printf("Unable to create socket\n");
        return -1;
    }
    // set port and IP
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    server_addr.sin_port = htons(3000);
    // connect to server
    if(connect(client_sock, (struct sockaddr*)&server_addr, sizeof(server_addr))<0)
    {
        printf("Unable to connect\n");
        return -1;
    }
    // send message to server
    printf("Enter message: ");
    gets(client_msg);
    if(send(client_sock, client_msg, strlen(client_msg), 0)<0){
        printf("Unable to send the message\n");
        return -1;
    }
    // receive message from the server
    if(recv(client_sock, server_msg, sizeof(server_msg), 0) < 0)
    {
        printf("Couldn't receive\n");
        return -1;
    }
    printf("Message from server: %s\n", server_msg);
    memset(client_msg, '\0', sizeof(client_msg));
}
close(client_sock);
return 0;
}

```

### 3.TRANSMISSION CONTROL PROTOCOL

#### OUTPUT:

##### Server

```
student@H68:~$ gcc server.c -o server
server.c: In function 'main':
server.c:69:9: warning: implicit declaration of function 'gets'; did you mean 'f
gets'? [-Wimplicit-function-declaration]
    gets(server_msg);
    ^
    fgets
/tmp/cca0Fd28.o: In function `main':
server.c:(.text+0x1e6): warning: the `gets' function is dangerous and should not
be used.
student@H68:~$ ./server
Socket created successfully
Message from client: Hello Server(-
Enter message: Hello Client!

```

##### Client

```
student@H68:~$ gcc client.c -o client
client.c: In function 'main':
client.c:40:9: warning: implicit declaration of function 'gets'; did you mean 'f
gets'? [-Wimplicit-function-declaration]
    gets(client_msg);
    ^
    fgets
/tmp/ccYW6mmg.o: In function `main':
client.c:(.text+0x108): warning: the `gets' function is dangerous and should not
be used.
student@H68:~$ ./client
Enter message: Hello Server(-
Message from server: Hello Client!
Enter message: 
```

#### 4.USER DATAGRAM PROTOCOL

SERVER PROGRAM:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#define MAXLINE 1024
#define PORT 4000
int main (void)
{
    int sockfd;
    struct sockaddr_in serveraddr , clientaddr;
    char buffer[MAXLINE], hello[20];
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if(sockfd < 0)
    {
        printf("Socket cannot be created");
        return -1;
    }
    printf("Socket created successfully\n");
    memset(&serveraddr, '\0', sizeof(serveraddr));
    memset(&clientaddr, '\0', sizeof(clientaddr));
    serveraddr.sin_family= AF_INET ;
    serveraddr.sin_port= htons(PORT);
    serveraddr.sin_addr.s_addr= INADDR_ANY ;
    if(bind(sockfd, (struct sockaddr*)&serveraddr, sizeof(serveraddr))<0)
    {
        printf("Cannot Bind");
        return -1;
    }
    printf("Binding Successful\n");
    int len, n;
    len = sizeof(clientaddr);
    n = recvfrom(sockfd, (char*) buffer,MAXLINE, MSG_WAITALL,(struct
    sockaddr*)&clientaddr, &len);
    buffer[n]='\0';
    printf("Message received : %s\n",buffer);
    printf("Enter the message: ");
    gets(hello);
```

```

sendto(sockfd, (const char*) hello, strlen(hello), MSG_CONFIRM,(const struct
sockaddr*)&clientaddr, len);
    printf("Message sent\n");
    return 0;
}

4.CLIENT PROGRAM:
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#define MAXLINE 1024
#define PORT 4000
int main(void)
{
    int sockfd;
    struct sockaddr_in serveraddr;
    char buffer[MAXLINE], hello[20];
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if (sockfd < 0)
    {
        printf("Socket cannot be created");
        return -1;
    }
    printf("Socket created successfully\n");
    memset(&serveraddr, '0', sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_port = htons(PORT);
    serveraddr.sin_addr.s_addr = INADDR_ANY;
    int len, n;
    len = sizeof(serveraddr);
    printf("Enter the message: ");
    gets(hello);
    sendto(sockfd, (const char *)hello, strlen(hello), MSG_CONFIRM, (const struct sockaddr *)
&serveraddr, len);
    printf("Message sent\n");
    n = recvfrom(sockfd, (char *)buffer, MAXLINE, MSG_WAITALL, (struct sockaddr *)
&serveraddr, &len);
    buffer[n] = 0;
    printf("Message received : %s\n", buffer);
    return 0; }

```

#### 4.USER DATAGRAM PROTOCOL

##### OUTPUT:

###### Server

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ gcc udpserver.c -o server
udpserver.c: In function 'main':
udpserver.c:50:5: warning: implicit declaration of function 'gets'; did you mean
  'fgets'? [-Wimplicit-function-declaration]
    gets(hello);
    ^
    fgets
/tmp/cc0PcshK.o: In function `main':
udpserver.c:(.text+0x190): warning: the `gets' function is dangerous and should
not be used.
student@H68:~/Downloads$ ./server
Socket created successfully
Binding Successful
Message received : Hi UDPServer(-
Enter the message: Hi UDPClient!
Message sent

```

###### Client

```
File Edit View Search Terminal Help
student@H68:~/Downloads
student@H68:~/Downloads$ gcc udpclient.c -o client
udpclient.c: In function 'main':
udpclient.c:37:5: warning: implicit declaration of function 'gets'; did you mean
  'fgets'? [-Wimplicit-function-declaration]
    gets(hello);
    ^
    fgets
/tmp/ccHCaBYV.o: In function `main':
udpclient.c:(.text+0xe0): warning: the `gets' function is dangerous and should n
ot be used.
student@H68:~/Downloads$ ./client
Socket created successfully
Enter the message: Hi UDPServer(-
Message sent
Message received : Hi UDPClient!
Enter the message: 
```

## 5.STOP AND WAIT ARQ

PROGRAM:

```
#include <stdio.h>
#include <stdlib.h>
typedef struct frame {
    int info, seq;
}frame;
int ack; // to mark acknowledgement
int t=5, k; // t to represent timer and k
int disconnect=0; // to disconnect the connection
frame p;
char turn = 's'; // to check if send or receive
int errorframe=1, errorack=1;
void sender();
void receiver();
int main() {
    p.info = 0;
    p.seq = 0;
    while(!disconnect) {
        sender();
        for(k=1; k<=10000000; k++) {
            receiver();
        }
    }
}
void sender() {
    static int flag=0;
    if(turn=='s') {
        if (errorack==0)
        {
            printf("Sender: sent packet with seq no.: %d\n", p.seq);
            errorframe = rand()%4;
            printf("%s\n", (errorframe==0?"Error while sending packet": ""));
            turn = 'r';
        }
        else {
            if(flag==1) printf("Sender: Received ACK for packet %d\n", ack);
            if(p.seq==5) {
                disconnect = 1;
                return;
            }
        }
    }
}
```

```

    p.info += 1;
    p.seq += 1;
    printf("Sender: sent packet with seq no.: %d\n", p.seq);
    errorframe = rand()%4;
    printf("%s\n", (errorframe==0?"Error while sending packet":"")); 
    turn = 'r';
    flag = 1;
}
}
else {
    t--;
    printf("Sender time reducing \n");
    if(t==0) {
        turn='s';
        errorack = 0;
        t = 5;
    }
}
}

void receiver() {
    static int frexp = 1;
    if(turn=='r') {
        if(errorframe!=0) {
            if(p.seq==frexp) {
                printf("Receiver: received packet with seq no.: %d\n", p.seq);
                ack=p.seq;
                frexp+=1;
                turn='s';
                errorack=rand()%4;
                printf("%s\n", (errorack==0?"Error while sending ACK":"")); 
            }
        }
        else {
            printf("Receiver: Duplicated packet with seq %d\n", frexp-1);
            ack = frexp-1;
            turn = 's';
            errorack = rand()%4;
            printf("%s\n", (errorack==0?"Error while sending ACK":"")); 
        }
    }
}
}

```

## 5.STOP AND WAIT ARQ

### OUTPUT:

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ gcc stopandwait.c
student@H68:~/Downloads$ ./a.out
Sender: sent packet with seq no.: 1

Receiver: received packet with seq no.: 1

Sender: Received ACK for packet 1
Sender: sent packet with seq no.: 2

Receiver: received packet with seq no.: 2

Sender: Received ACK for packet 2
Sender: sent packet with seq no.: 3

Receiver: received packet with seq no.: 3

Sender: Received ACK for packet 3
Sender: sent packet with seq no.: 4

Receiver: received packet with seq no.: 4
Error while sending ACK
Sender: Received ACK for packet 4
Sender: sent packet with seq no.: 5

Receiver: received packet with seq no.: 5

Sender: Received ACK for packet 5
student@H68:~/Downloads$
```

## 6.GO BACK N ARQ

SERVER PROGRAM:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>
#include <ctype.h>
#include <arpa/inet.h>

#define W 5
#define P1 50
#define P2 10

char a[10];
char b[10];

void alpha9(int);

int main() {
    struct sockaddr_in ser, cli;
    int s, n, sock, i, j, c=1, f;
    unsigned int s1;
    s = socket(AF_INET, SOCK_STREAM, 0);
    ser.sin_family = AF_INET;
    ser.sin_port = 6500;
    ser.sin_addr.s_addr = inet_addr("127.0.0.1");
    bind(s, (struct sockaddr *)&ser, sizeof(ser));
    listen(s,1);
    n = sizeof(cli);
    sock = accept(s, (struct sockaddr *)&cli, &n);
    printf("\nTCP Connection Established.\n");
    s1 = (unsigned int)time(NULL);
    srand(s1);
    strcpy(b, "Time Out");
    recv(sock, a, sizeof(a), 0);
    f = atoi(a);
    while (1)
    {
        for (i = 0; i < W; i++)
        {
            recv(sock, a, sizeof(a), 0);
            if(strcmp(a,b)==0) {
                break;
            }
        }
    }
}
```

```

        }
    }
i=0;
while(i<W) {
    j=rand()%P1;
    if(j<P2) {
        send(sock, b, sizeof(b), 0);
        break;
    }
    else {
        alpha9(c);
        if(c<=f) {
            printf("\nFrame %s Received", a);
            send(sock, a, sizeof(a), 0);
        }
        else {
            break;
        }
        c++;
    }
    if(c>f) {
        break;
    }
    i++;
}
}
close(sock);
close(s);
return 0;
}
void alpha9(int z) {
    int k, i=0, j, g;
    k=z;
    while(k>0) {
        i++;
        k=k/10;
    }
    g=i;
    i--;
    while(z>0) {
        k=z%10;
        a[i]=k+48;
        i--;
        z=z/10;
    }
    a[g]='\0';
}

```

## 6.GO BACK N ARQ

CLIENT PROGRAM:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>
#include <ctype.h>
#define W 5
#define P1 50
#define P2 10
char a[10];
char b[10];
void alpha9(int);
int main() {
    int s, f, wl, c=1, i=0, p=0, e=0;
    struct sockaddr_in ser;
    s=socket(AF_INET, SOCK_STREAM, 0);
    ser.sin_family = AF_INET;
    ser.sin_port = 6500;
    ser.sin_addr.s_addr = inet_addr("127.0.0.1");
    if(connect(s, (struct sockaddr *)&ser, sizeof(ser))==0) {
        printf("\nTCP Connection Established.\n");
    }
    else {
        printf("Connection establishment failed\n");
        return 0;
    }
    printf("\n Enter the number of Frames:");
    scanf("%d", &f);
    alpha9(f);
    send(s,a,sizeof(a),0);
    strcpy(b, "Time Out");
    while(1) {
        for ( i = 0; i < W; i++)
        {
            alpha9(c);
            send(s,a,sizeof(a),0);
            if(c<=f) {
                printf("\nFrame %d Sent", c);
                c++;
            }
        }
        i=0;
        wl=W;
    }
}
```

```

while(i<W) {
    recv(s,a,sizeof(a),0);
    p=atoi(a);
    if(strcmp(a,b)==0){
        e=c-wl;
        if(e<f) {
            printf("\nTime Out, Resent Frame %d onwards", e);
        }
        break; }
    else {
        if(p<=f) {
            printf("\nFrame %s Acknowledged", a);
            wl--;
        }
        else {
            break;
        }
    }
    if(p>f) {
        break;
    }
    i++;
}
if(wl==0 && c>f) {
    send(s,b,sizeof(b), 0);
    break; }
else {
    c=c-wl;
    wl=W;
}
close(s);
return 0;
} }

void alpha9(int z) {
int k, i=0, j, g;
k=z;
while(k>0) {
    i++;
    k=k/10;
}
g=i;
i--;
while(z>0) {
    k=z%10;
    a[i]=k+48;
    i--;
    z=z/10; }
a[g]='\0';
}

```

## 6.GO BACK N ARQ

### OUTPUT:

#### Server

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ gcc gobacknserver.c -o server
student@H68:~/Downloads$ ./server
TCP Connection Established.
Frame 1 Received
Frame 2 Received
student@H68:~/Downloads$ 
```

#### Client

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ gcc gobacknclient.c -o client
gobacknclient.c: In function 'main':
gobacknclient.c:26:27: warning: implicit declaration of function 'inet_addr'; did you mean 's6_addr'? [-Wimplicit-function-declaration]
    ser.sin_addr.s_addr = inet_addr("127.0.0.1");
                           ^
                           s6_addr
student@H68:~/Downloads$ ./client
TCP Connection Established.
Enter the number of Frames:10
Frame 1 Sent
Frame 2 Sent
Frame 3 Sent
Frame 4 Sent
Frame 5 Sent
Frame 1 Acknowledged
student@H68:~/Downloads$ 
```

## 7.SELECTIVE REPEAT ARQ

SERVER PROGRAM:

```
#include <stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#include<arpa/inet.h>
#define W 5
#define P1 50
#define P2 10
char a[10];
char b[10];
void alpha9(int);
void alp(int);

int main()
{
    struct sockaddr_in ser,cli;
    int s,n,sock,i,j,c=1,f;
    unsigned int s1;
    s=socket(AF_INET,SOCK_STREAM,0);
    ser.sin_family=AF_INET;
    ser.sin_port=6500;
    ser.sin_addr.s_addr=inet_addr("127.0.0.1");
    bind(s,(struct sockaddr*)&ser,sizeof(ser));
    listen(s,1);
    n[sizeof(cli)];
    sock=accept(s,(struct sockaddr*)&cli,&n);
    printf("\nTCP Connection Established.\n");
    s1=(unsigned int) time(NULL);
    srand(s1);
    strcpy(b,"Time out");
    recv(sock,a,sizeof(a),0);
    f=atoi(a);
    while(1)
    {
        for(i=0;i<W;i++)
        {
```

```

        recv(sock,a,sizeof(a),0);
        if(strcmp(a,b)==0)
        {
            break;
        }
    }

i=0;
while(i<W)
{
    L:
    j=rand()%P1;
    if(j<P2)
    {
        alp(c);
        send(sock,b,sizeof(b),0);
        goto L;
    }
    else
    {
        alpha9(c);
        if(c<=f)
        {
            printf("\nFrame %s Received",a);
            send(sock,a,sizeof(a),0);
        }
        else
        {
            break;
        }
        c++;
    }
    if(c>f)
    {
        break;
    }
    i++;
}
}

close(sock);
close(s);
return 0;
}

void alpha9(int z)
{
    int k,i=0,j,g;
    k=z;
    while(k>0)
    {

```

```

        i++;
        k=k/10;
    }
    g=i;
    i--;
    while(z>0)
    {
        k=z%10;
        a[i]=k+48;
        i--;
        z=z/10;
    }
    a[g]='\0';
}
void alp(int z)
{
    int k,i=1,j,g;
    k=z;
    b[0]='N';
    while(k>0)
    {
        i++;
        k=k/10;
    }
    g=i;
    i--;
    while(z>0) {
        k=z%10;
        b[i]=k+48;
        i--;
        z=z/10;
    }
    b[g]='\0';
}

```

## **7.SELECTIVE REPEAT ARQ**

CLIENT PROGRAM:

```

#include <stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#define W 5

```

```

char a[10];

```

```

char b[10];
void alpha9(int);
int con();
int main()
{
    int s,f,w1,c=1,x,i=0,j,n,p=0,e=0;
    struct sockaddr_in ser;
    s=socket(AF_INET,SOCK_STREAM,0);
    ser.sin_family=AF_INET;
    ser.sin_port=6500;
    ser.sin_addr.s_addr=inet_addr("127.0.0.1");
    connect(s,(struct sockaddr*)&ser,sizeof(ser));
    printf("\nTCP Connection Established.\n");
    printf("\nEnter the number of Frames: ");
    scanf("%d",&f);
    alpha9(f);
    send(s,a,sizeof(a),0);
    strcpy(b,"Time Out ");
    while(1) {
        for(i=0;i<W;i++)
        {
            alpha9(c);
            send(s,a,sizeof(a),0);
            if(c<=f)
            {
                printf("\nFrame %d Sent",c);
                c++;
            }
            i=0;
            w1=W;
        }
        while(i<W) {
            recv(s,a,sizeof(a),0);
            p=atoi(a);
            if(a[0]=='N') {
                e=con();
                if(e<f)
                {
                    printf("\nNAK %d",e);
                    printf("\nFrame %d sent",e);
                    i--;
                }
            }
            else {
                if(p<=f) {
                    printf("\nFrame %s Acknowledged",a);
                    w1--;
                }
            }
        }
    }
}

```

```

        break;
    }
    if(p>f) {
        break;
    }
    i++;
}
if(w1==0 && c>f) {
    send(s,b,sizeof(b),0);
    break;
}
else {
    c=c-w1;
    w1=W;
}
close(s);
return 0;
}
void alpha9(int z)
{
    int k,i=0,j,g;
    k=z;
    while(k>0) {
        i++;
        k=k/10;
    }
    g=i;
    i--;
    while(z>0) {
        k=z%10;
        a[i]=k+48;
        i--;
        z=z/10;
    }
    a[g]='\0';
}
int con()
{
    char k[9];
    int i=1;
    while(a[i]!='\0') {
        k[i-1]=a[i];
        i++;
    }
    k[i-1]='\0';
    i=atoi(k);
    return i;
}

```

## 7.SELECTIVE REPEAT ARQ

### OUTPUT:

#### Server

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~$ cd Downloads
student@H68:~/Downloads$ gcc srn.c -o server
student@H68:~/Downloads$ ./server

TCP Connection Established.

Frame 1 Received
Frame 2 Received

```

#### Client

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
srnclient.c: In function 'main':
srnclient.c:27:27: warning: implicit declaration of function 'inet_addr'; did yo
u mean 's6_addr'? [-Wimplicit-function-declaration]
    ser.sin_addr.s_addr = inet_addr("127.0.0.1");
               ^
               s6_addr
student@H68:~/Downloads$ ./client

TCP Connection Established.

Enter the number of Frames:3

Frame 1 Sent
Frame 2 Sent
Frame 3 Sent
Frame 1 Acknowledged
Frame 2 Acknowledged

```

## 8.DISTANCE VECTOR ROUTING PROTOCOL

```
#include <stdio.h>
struct node {
    unsigned dist[20];
    unsigned from[20];
} rt[10];
int main()
{
    int costmat[20][20];
    int nodes,i,j,k,count=0;
    printf("\nEnter the number of nodes:");
    scanf("%d",&nodes);
    printf("\nEnter the cost matrix:\n");
    for(i=0;i<nodes;i++)
    {
        for(j=0;j<nodes;j++)
        {
            scanf("%d",&costmat[i][j]);
            costmat[i][j]=0;
            rt[i].dist[j]=costmat[i][j];
            rt[i].from[j]=j;
        }
    }
    do
    {
        count=0;
        for(i=0;i<nodes;i++)
            for(j=0;j<nodes;j++)
                for(k=0;k<nodes;k++)
                    if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
                    {
                        //calculate minimum distance
                        rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
                        rt[i].from[j]=k;
                        count++;
                    }
    }while(count!=0);
    for(i=0;i<nodes;i++)
    {
        printf("\n\n For router %d\n",i+1);
        for(j=0;j<nodes;j++)
        {
            printf("\t\n node %d Distance %d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
        }
    }
    printf("\n\n");
}
```

## 8.DISTANCE VECTOR ROUTING PROTOCOL

### OUTPUT:

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ ./a.out

Enter the number of nodes:3

Enter the cost matrix:
0
1
4
2
0
6
5
6
0

For router 1

node 1 Distance 1
node 2 Distance 2
node 3 Distance 3

For router 2

node 1 Distance 1
node 2 Distance 2
node 3 Distance 3

For router 3

node 1 Distance 1
node 2 Distance 2
node 3 Distance 3

student@H68:~/Downloads$
```

## 9.FILE TRANSFER USING TCP

SERVER PROGRAM:

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <arpa/inet.h>
int main()
{
    int socketfd,client_socket,client_size;
    struct sockaddr_in server_addr,client_addr;
    char server_message[2000],client_message[2000];
    memset(server_message,'\0',sizeof(server_message));
    memset(client_message,'\0',sizeof(client_message));
    socketfd=socket(AF_INET,SOCK_STREAM,0);
    if(socketfd<0)
    {
        printf("Error while creating socket\n");
        return -1;
    }
    printf("Socket created successfully\n");

    server_addr.sin_family=AF_INET;
    server_addr.sin_port=htons(2000);
    server_addr.sin_addr.s_addr=inet_addr("127.0.0.1");
    if(bind(socketfd,(struct sockaddr*)&server_addr,sizeof(server_addr))<0)
    {
        printf("Couldn't bind to the port\n");
        return -1;
    }
    printf("Done with binding\n");

    if(listen(socketfd,0)<0)
    {
        printf("Cannot listen");
        return -1;
    }

    client_size=sizeof(client_addr);

    client_socket=accept(socketfd,(struct sockaddr*)&client_addr,&client_size);
    if(client_socket<0)
```

```

{
printf("Can't accept\n");
return -1;
}
printf("Client connected at IP:%s and
port:%i\n",inet_ntoa(client_addr.sin_addr), ntohs(client_addr.sin_port));

if(recv(client_socket,client_message,sizeof(client_message),0)<0)
{
printf("Couldn't receive\n");
return -1;
}
printf("%s",client_message);
FILE* fileptr=fopen(client_message,"r");
if(fileptr==NULL)
{
    strcpy(server_message,"NOT A VALID FILE");
    if(send(client_socket,server_message,strlen(server_message),0)<0)
    {
        printf("Can't send\n");
        return -1;
    }
    return -1;
}

int i=0;
while(!feof(fileptr))
{
server_message[i++]=fgetc(fileptr);
}
server_message[i]='\0';
printf("%s",server_message);

if(send(client_socket,server_message,strlen(server_message),0)<0)
{
    printf("Can't send\n");
    return -1;
}
fclose(fileptr);
close(client_socket);
close(socketfd);
return 0;
}

```

## 9.FILE TRANSFER USING TCP

CLIENT PROGRAM:

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <arpa/inet.h>
int main(void)
{
    int socket_desc;
    struct sockaddr_in server_addr;
    char server_message[4000];
    char filename[2000];
    memset(server_message, '\0', sizeof(server_message));
    socket_desc=socket(AF_INET,SOCK_STREAM,0);
    if(socket_desc<0)
    {
        printf("Unable to create socket\n");
        return -1;
    }
    printf("Socket created successfully\n");
    server_addr.sin_family =AF_INET;
    server_addr.sin_port =htons(2000);
    server_addr.sin_addr.s_addr =inet_addr("127.0.0.1");
    if(connect(socket_desc,(struct sockaddr*)&server_addr,sizeof(server_addr))<0)
    {
        printf("Unable to connect\n");
        return -1;
    }
    printf("Connected with server successfully\n");
    printf("Enter a filename:");
    scanf("%s",filename);
    if(send(socket_desc,filename,strlen(filename),0)<0)
    {
        printf("Unable to send message\n");
        return -1;
    }
    if(recv(socket_desc,server_message,sizeof(server_message),0)<0)
    {
        printf("Error while receiving server's msg\n");
        return -1;
    }
    printf("File contents: %s\n",server_message);
    close(socket_desc);
    return 0;
}
```

## 9.FILE TRANSFER USING TCP

### OUTPUT:

#### Server

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~$ cd Downloads
student@H68:~/Downloads$ gcc Serverftp.c
Serverftp.c: In function 'main':
Serverftp.c:33:10: error: expected ';' before 'to'
    bind to set port and ip
      ^
student@H68:~/Downloads$ gcc Serverftp.c
student@H68:~/Downloads$ ./a.out
Socket created successfully
Unable to connect
student@H68:~/Downloads$ gcc Serverftp.c -o s
student@H68:~/Downloads$ ./s
Socket created successfully
Done with binding
Client connected at IP:127.0.0.1 and port:53868
t1.txtHello
world..
Good
Morning!
student@H68:~/Downloads$
```

#### Client

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~$ cd Downloads
student@H68:~/Downloads$ gcc Clientftp.c
student@H68:~/Downloads$ gcc Clientftp.c
student@H68:~/Downloads$ ./a.out
Socket created successfully
Connected with server successfully
Enter a filename:t1.txt
File contents: Hello
world..
Good
Morning!
student@H68:~/Downloads$
```

## 10.LEAKY BUCKET ALGORITHM

```
#include <stdio.h>
#include<stdlib.h>
#include<unistd.h>
#define NOF_PACKETS 10
int rnd(int a)
{
    int rn = random() % a;
    return rn == 0 ? 1 : rn;
}
int main()
{
    int packet_sz[NOF_PACKETS], i, clk, b_size, o_rate,
    p_sz_rm=0, p_sz, p_time, op;
    for(i = 0; i<NOF_PACKETS; ++i)
    {
        packet_sz[i] = rnd(6)* 10;
    }
    for(i = 0; i<NOF_PACKETS; ++i)
    {
        printf("\npacket[%d]:%d bytes\t", i, packet_sz[i]);
    }
    printf("\nEnter the Output rate:");
    scanf("%d", &o_rate);
    printf("Enter the Bucket Size:");
    scanf("%d", &b_size);
    for(i = 0; i<NOF_PACKETS; ++i)
    {
        if( (packet_sz[i] + p_sz_rm) > b_size)
            //compare the packet size with bucket size
            if(packet_sz[i] > b_size)
                printf("\n\nIncoming packet size (%dbytes) is Greater than bucket
capacity (%dbytes)-PACKET REJECTED",packet_sz[i], b_size);
            else
                printf("\n\nBucket capacity exceeded-PACKETS REJECTED!!");
        else
        {
            p_sz_rm += packet_sz[i];
            printf("\n\nIncoming Packet size: %d",packet_sz[i]);
            printf("\nBytes remaining to Transmit: %d",p_sz_rm);
            p_time = rnd(4) * 10;
            printf("\nTime left for transmission: %d units",p_time);
        }
    }
}
```

```

for(clk = 10; clk <= p_time; clk += 10)
{
    sleep(1);
    if(p_sz_rm)
    {
        //Compare remaining packet size with output rate
        if(p_sz_rm <= o_rate)
        {
            op = p_sz_rm;
            p_sz_rm = 0;
        }
        else
        {
            op = o_rate;
            p_sz_rm -= o_rate;
        }
        printf("\nPacket of size %d Transmitted",op);
        printf("----Bytes Remaining to Transmit:%d", p_sz_rm);
    }
    else
    {
        printf("\nTime left for transmission: %d units", p_time-clk);
        printf("\nNo packets to transmit!!");
    }
} //close for
}
}
}

```

## 10.LEAKY BUCKET ALGORITHM

### OUTPUT

```
student@H68: ~/Downloads
File Edit View Search Terminal Help
student@H68:~/Downloads$ gcc lb.c
student@H68:~/Downloads$ ./a.out

packet[0]:10 bytes
packet[1]:40 bytes
packet[2]:30 bytes
packet[3]:10 bytes
packet[4]:50 bytes
packet[5]:10 bytes
packet[6]:40 bytes
packet[7]:10 bytes
packet[8]:30 bytes
packet[9]:10 bytes
Enter the Output rate:10
Enter the Bucket Size:40

Incoming Packet size: 10
Bytes remaining to Transmit: 10
Time left for transmission: 20 units
Packet of size 10 Transmitted---Bytes Remaining to Transmit:0
Time left for transmission: 0 units
No packets to transmit!!

student@H68: ~/Downloads
File Edit View Search Terminal Help
Incoming Packet size: 40
Bytes remaining to Transmit: 40
Time left for transmission: 30 units
Packet of size 10 Transmitted---Bytes Remaining to Transmit:30
Packet of size 10 Transmitted---Bytes Remaining to Transmit:20
Packet of size 10 Transmitted---Bytes Remaining to Transmit:10

Incoming Packet size: 30
Bytes remaining to Transmit: 40
Time left for transmission: 20 units
Packet of size 10 Transmitted---Bytes Remaining to Transmit:30
Packet of size 10 Transmitted---Bytes Remaining to Transmit:20

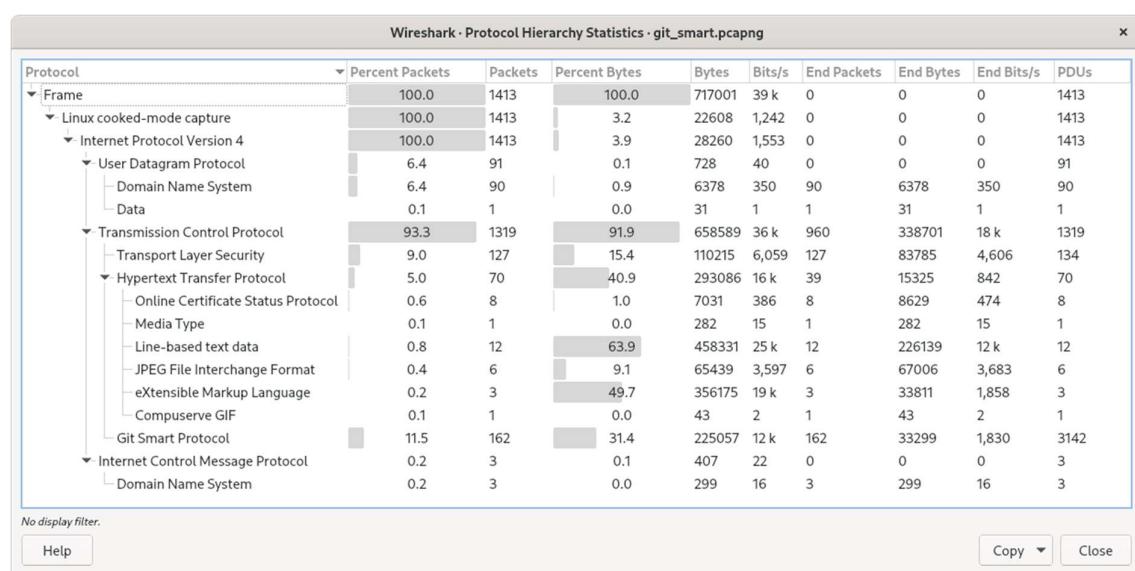
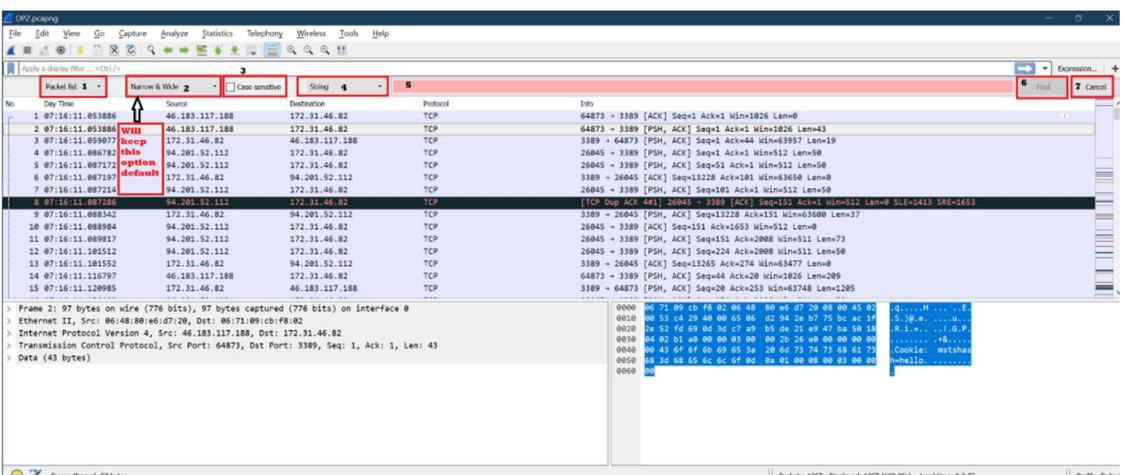
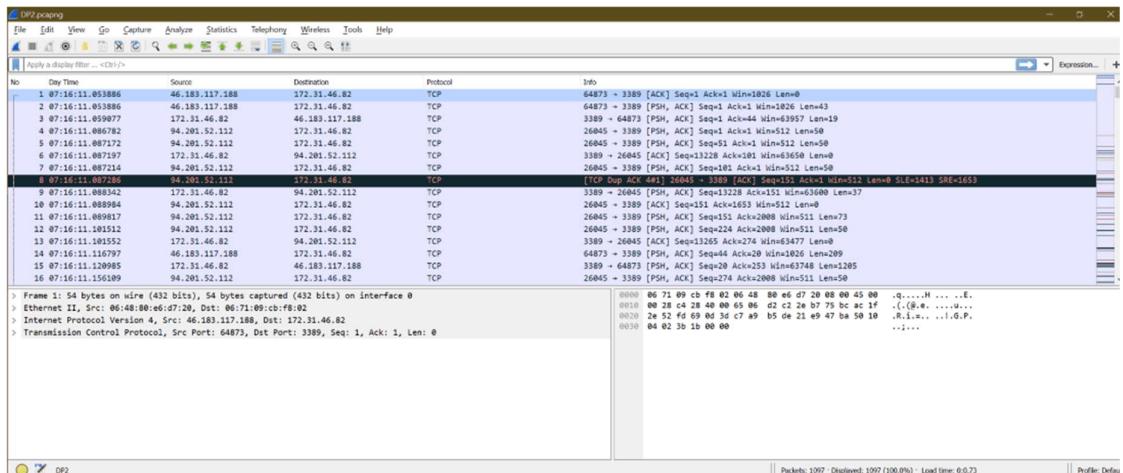
Incoming Packet size: 10
Bytes remaining to Transmit: 30
Time left for transmission: 30 units
Packet of size 10 Transmitted---Bytes Remaining to Transmit:20
Packet of size 10 Transmitted---Bytes Remaining to Transmit:10
Packet of size 10 Transmitted---Bytes Remaining to Transmit:0

Incoming packet size (50bytes) is Greater than bucket capacity (40bytes)-PACKET REJECTED

Incoming Packet size: 10
Bytes remaining to Transmit: 30
Time left for transmission: 10 units
Packet of size 10 Transmitted---Bytes Remaining to Transmit:20

Bucket capacity exceeded-PACKETS REJECTED!!
```

## 11.WIRESHARK



tcp\_trace\_feb\_2019.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
22	9.762431	2405:204:d208:92bd:..	2001:1af8:4100:b100..	TCP	74	52998 + 443 [FIN, ACK] Seq=1 Ack=1 Win=4110 Len=0
23	9.763068	192.168.43.15	128.119.245.12	TCP	66	53000 + 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
24	9.763493	192.168.43.15	128.119.245.12	TCP	66	53001 + 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
25	10.014400	192.168.43.15	128.119.245.12	TCP	66	53002 + 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
26	10.061033	2001:1af8:4100:b100..	2405:204:d208:92bd:..	TCP	74	443 + 52998 [RST] Seq=1 Win=0 Len=0
27	10.141249	128.119.245.12	192.168.43.15	TCP	66	80 + 53001 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1370 SACK_PERM=1 WS=128
28	10.141436	192.168.43.15	128.119.245.12	TCP	54	53001 + 80 [ACK] Seq=1 Ack=1 Win=16440 Len=0
29	10.141701	128.119.245.12	192.168.43.15	TCP	66	80 + 53000 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1370 SACK_PERM=1 WS=128
30	10.141793	192.168.43.15	128.119.245.12	TCP	54	53001 + 80 [ACK] Seq=1 Ack=1 Win=16440 Len=0
31	10.144161	192.168.43.15	128.119.245.12	TCP	713	53001 + 80 [PSH, ACK] Seq=1 Ack=1 Win=16440 Len=659 [TCP segment of a reassembled PDU]
32	10.144718	192.168.43.15	128.119.245.12	TCP	1424	53001 + 80 [ACK] Seq=660 Ack=1 Win=16440 Len=1370 [TCP segment of a reassembled PDU]
33	10.397573	128.119.245.12	192.168.43.15	TCP	66	80 + 53002 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1370 SACK_PERM=1 WS=128
34	10.397750	192.168.43.15	128.119.245.12	TCP	54	53002 + 80 [ACK] Seq=1 Ack=1 Win=16440 Len=0

> Frame 1: 714 bytes on wire (5712 bits), 714 bytes captured (5712 bits) on interface 0

> Ethernet II, Src: IntelCor\_ad:6b:23 (b8:03:05:ad:6b:23), Dst: XiaomiCo\_fa:19:07 (04:b1:67:fa:19:07)

> Internet Protocol Version 6, Src: 2405:204:d208:92bd:5da8:c5c1:16a8:6678, Dst: 2001:1af8:4100:b100::102

> Transmission Control Protocol, Src Port: 52997, Dst Port: 443, Seq: 1, Ack: 1, Len: 640

> Secure Sockets Layer

tcp\_trace\_feb\_2019.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
198	13.712773	128.119.245.12	192.168.43.15	TCP	54	80 + 53001 [ACK] Seq=1 Ack=148172 Win=274304 Len=0
199	13.712907	128.119.245.12	192.168.43.15	TCP	54	80 + 53001 [ACK] Seq=1 Ack=149542 Win=277248 Len=0
200	13.717823	128.119.245.12	192.168.43.15	TCP	54	80 + 53001 [ACK] Seq=1 Ack=150912 Win=280064 Len=0
201	13.718117	128.119.245.12	192.168.43.15	TCP	54	80 + 53001 [ACK] Seq=1 Ack=152976 Win=284288 Len=0
202	13.718172	128.119.245.12	192.168.43.15	HTTP	831	HTTP/1.1 200 OK (text/html)
203	13.918034	192.168.43.15	128.119.245.12	TCP	54	53001 + 80 [ACK] Seq=152976 Ack=778 Win=15660 Len=0
204	19.025755	128.119.245.12	192.168.43.15	TCP	54	80 + 53001 [FIN, ACK] Seq=778 Ack=152976 Win=284288 Len=0
205	19.025904	192.168.43.15	128.119.245.12	TCP	54	53001 + 80 [ACK] Seq=152976 Ack=779 Win=15660 Len=0
206	20.002288	2405:204:d208:92bd:..	2001:1af8:4100:b100..	TLSv1.2	714	Application Data
207	20.167977	XiaomiCo_fa:19:07	IntelCor_ad:6b:23	ARP	42	Who has 192.168.43.15? Tell 192.168.43.1
208	20.168021	IntelCor_ad:6b:23	XiaomiCo_fa:19:07	ARP	42	192.168.43.15 is at b8:03:05:ad:6b:23
209	20.306533	2001:1af8:4100:b100..	2405:204:d208:92bd:..	TLSv1.2	332	Application Data
210	20.506108	2405:204:d208:92bd:..	2001:1af8:4100:b100..	TCP	74	52997 + 443 [ACK] Seq=2561 Ack=1033 Win=4045 Len=0

> Frame 1: 714 bytes on wire (5712 bits), 714 bytes captured (5712 bits) on interface 0

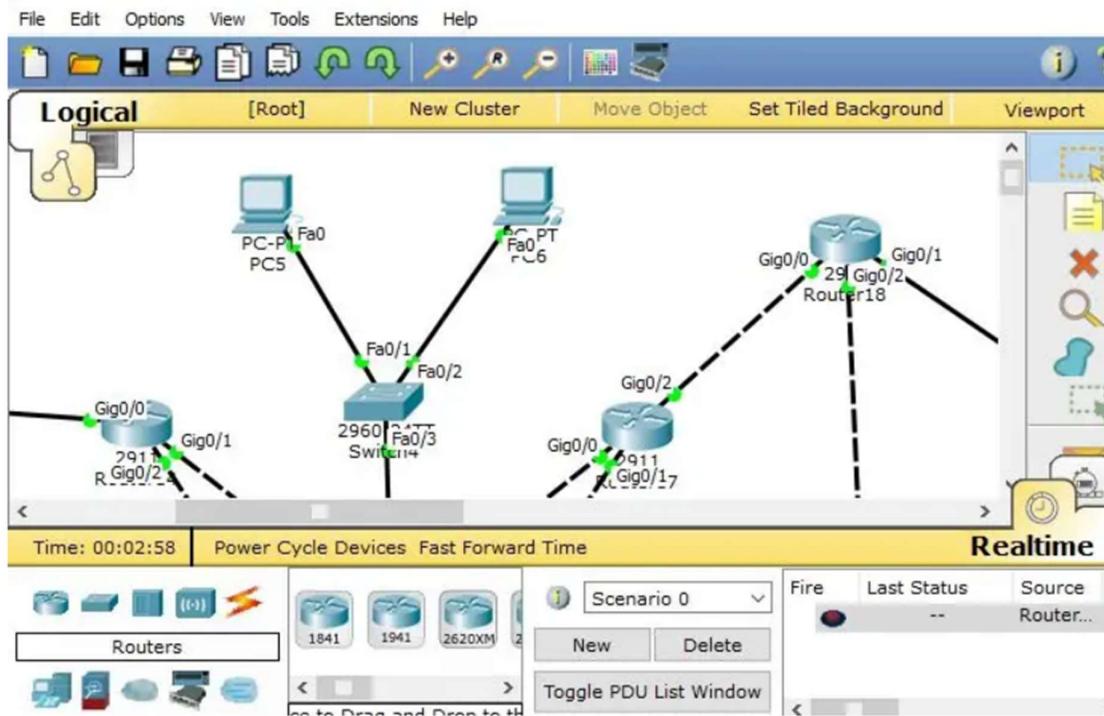
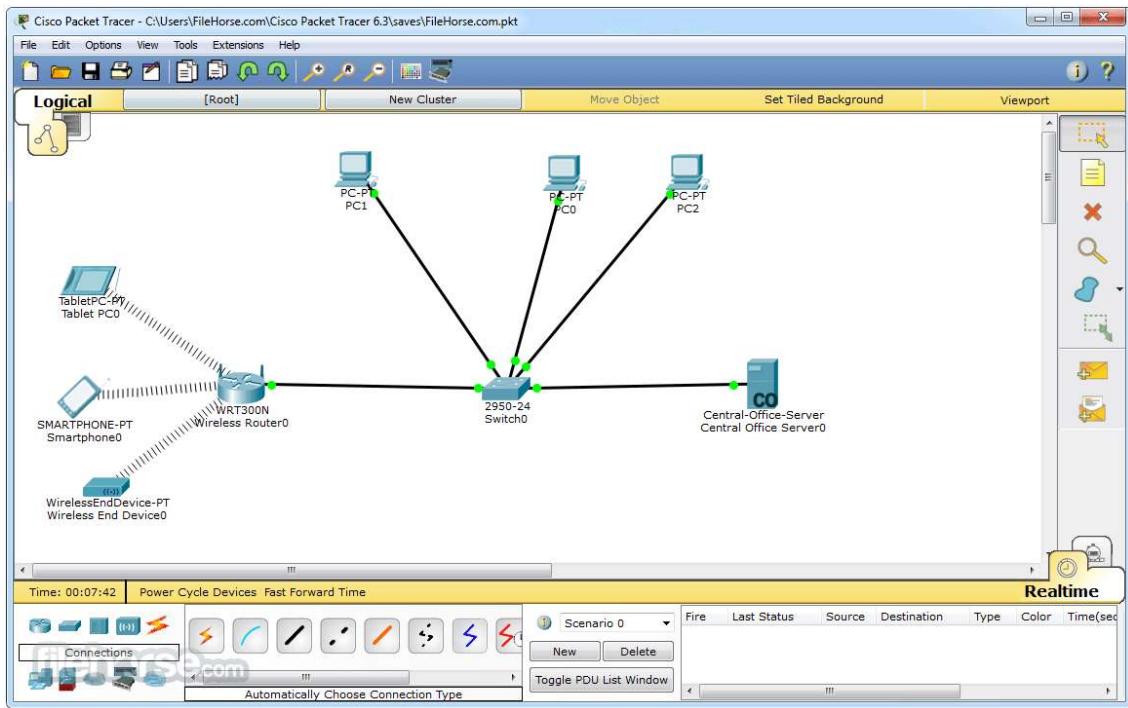
> Ethernet II, Src: IntelCor\_ad:6b:23 (b8:03:05:ad:6b:23), Dst: XiaomiCo\_fa:19:07 (04:b1:67:fa:19:07)

> Internet Protocol Version 6, Src: 2405:204:d208:92bd:5da8:c5c1:16a8:6678, Dst: 2001:1af8:4100:b100::102

> Transmission Control Protocol, Src Port: 52997, Dst Port: 443, Seq: 1, Ack: 1, Len: 640

> Secure Sockets Layer

## 12.PACKET TRACER



### 13.NS2

