1. Networking Commands

```
┌──(kali㉿kali)-[~]
└─$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.133.210          ether   f6:34:e7:6c:98:b6   C                     wlan0
```

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 24:b6:fd:4a:44:29  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 602  bytes 52828 (51.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 602  bytes 52828 (51.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.133.49  netmask 255.255.255.0  broadcast 192.168.133.255
        inet6 fe80::8420:5489:c592:d425  prefixlen 64  scopeid 0×20<link>
        inet6 2409:4073:4d85:a89f:6b0c:60a1:363a:3ebd  prefixlen 64 scopeid 0×0<global>
        ether e0:06:e6:0f:55:01  txqueuelen 1000  (Ethernet)
        RX packets 17715  bytes 15822952 (15.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14712  bytes 2920696 (2.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
┌──(kali㉿kali)-[~]
└─$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 0.0.0.0:55704           0.0.0.0:*
udp        0      0 0.0.0.0:52019           0.0.0.0:*
Udp6       0      0 [::]:56551              [::]:*
Udp6       0      0 [::]:45011              [::]:*
Udp6       0      0 [::]:45375              [::]:*
Udp6       0      0 [::]:49896              [::]:*
raw6       0      0 [::]:ipv6-icmp          [::]:*                  7
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     13135    Th/tmpLICE-unix/1851
unix  2      [ ACC ]     STREAM     LISTENING     12916    Amp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     18021    Amp/ssh-XXXXXXKoR6iY/agent.1851
unix  2      [ ACC ]     STREAM     LISTENING     13136    Amp/.ICE-unix/1851
unix  2      [ ACC ]     STREAM     LISTENING     12915    Th/tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     12978    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     12987    /run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     12989    /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM     LISTENING     12991    /run/user/1000/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     STREAM     LISTENING     12993    /run/user/1000/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     STREAM     LISTENING     12995    /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     12997    /run/user/1000/gnupg/S.gpg-agent
unix  2      [ ACC ]     STREAM     LISTENING     12999    /run/user/1000/pipewire-0
unix  2      [ ACC ]     STREAM     LISTENING     13001    /run/user/1000/pk-debconf-socket
unix  2      [ ACC ]     STREAM     LISTENING     13003    /run/user/1000/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     13117    /run/user/1000/at-spi/bus_0
unix  2      [ ACC ]     STREAM     LISTENING     26281    /run/user/1000/keyring/control
unix  2      [ ACC ]     STREAM     LISTENING     12234    /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     12236    /run/uuidd/request
unix  2      [ ACC ]     STREAM     LISTENING     29204    /run/user/1000/speech-dispatcher/speechd.sock
unix  2      [ ACC ]     STREAM     LISTENING     10697    /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     10699    /run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     10700    /run/systemd/io.system.ManagedOOM
unix  2      [ ACC ]     STREAM     LISTENING     10717    /run/lvm/lvmpolld.socket
unix  2      [ ACC ]     STREAM     LISTENING     10727    /run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     10729    /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     10015    /run/systemd/journal/io.systemd.journal
```

```
┌──(kali㉿kali)-[~]
└─$ nslookup google.com
Server:         192.168.133.210
Address:        192.168.133.210#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.206.142
Name:   google.com
Address: 2404:6800:4002:816::200e
```

```
┌──(kali㉿kali)-[~]
└─$ ping 8.8.8.8.
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from dns.google (8.8.8.8): icmp_seq=1 ttl=111 time=49.2 ms
64 bytes from dns.google (8.8.8.8): icmp_seq=2 ttl=111 time=48.1 ms
64 bytes from dns.google (8.8.8.8): icmp_seq=3 ttl=111 time=46.6 ms
64 bytes from dns.google (8.8.8.8): icmp_seq=4 ttl=111 time=59.5 ms
64 bytes from dns.google (8.8.8.8): icmp_seq=5 ttl=111 time=44.4 ms
64 bytes from dns.google (8.8.8.8): icmp_seq=6 ttl=111 time=52.5 ms
64 bytes from dns.google (8.8.8.8): icmp_seq=7 ttl=111 time=51.7 ms
^C
─── 8.8.8.8 ping statistics ───
7 packes transmitted, 7 recieved, 0% packet loss, time 6008ms
rtt min/avg/max/mdev = 44.423/50.294/59.490/4.556 ms
```

```
┌──(kali㉿kali)-[~]
└─$ dig google.com
; <<>> DiG 9.18.1-1-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; -->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23841
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             115     IN      A       142.250.192.174

;; Query time: 152 msec
;; SERVER: 192.168.133.210#53(192.168.133.210) (UDP)
;; WHEN: Wed Jun 29 10:27:01 UTC 2022
;; MSG SIZE  rcvd: 55
```

3. Transmission Control Protocol

Client Program

```
#include<stdio.h>
#include<sys/socket.h>
#include<string.h>
#include<arpa/inet.h>//inet_addr
int main(int argc,char *argv[])
{
	int socket_desc;
	char* message;
	struct sockaddr_in server;
	//Create socket
	socket_desc=socket(AF_INET,SOCK_STREAM,0);
	if(socket_desc==-1)
	{
		printf("Could not create socket");
	}
	server.sin_addr.s_addr=inet_addr("127.0.0.1");
	server.sin_family=AF_INET;
	server.sin_port=htons(8888);

	//connect to remote server
	if(connect(socket_desc,(struct sockaddr*)&server,sizeof(server))<0)
	{
		puts("connect error");
		return 1;
	}
	puts("connected");

	//Sending message to server
	message="Hai 456789\r\n\r\n";
	puts("Sending Message....");
	if(send(socket_desc,message,strlen(message),0)<0)
	{
		puts("Send failed");
		return 1;
	}
	//sleep(1);
	return 0;
}
```

Exp 3 Client Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp3$ gcc client.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp3$ ./a.out
connected
Sending Message....
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp3$
```

Server Program

```
#include<stdio.h>
#include<sys/socket.h>
#include<arpa/inet.h>//inet_addr
int main(int argc,char *argv[])
{
	int socket_desc,new_socket,c;
	char message[2000];
	struct sockaddr_in server,client;
	socket_desc=socket(AF_INET,SOCK_STREAM,0);
	if(socket_desc==-1)
	{
		printf("Could not create socket");
	}
	server.sin_family=AF_INET;
	server.sin_addr.s_addr=INADDR_ANY;
	server.sin_port=htons(8888);

	if(bind(socket_desc,(struct sockaddr*)&server,sizeof(server))<0)
	{
		puts("bind failed");
	}
	puts("bind done");
	//Listen
	listen(socket_desc,3);

	//Accept an incoming connection
	puts("Waiting for incoming connections.....");
	c=sizeof(struct sockaddr_in);
	new_socket=accept(socket_desc,(struct sockaddr*)&client,(socklen_t*)&c);
	if(new_socket<0)
	{
		perror("accept failed");
	}
	puts("Connection Accepted");
	if(recv(new_socket,message,2000,0)>0)
	{
		puts("recv failed");
		//puts(message);
	}
	puts(message);
	puts("MESSAGE RECIEVED\n");
	return 0;
}
```

Exp 3 Server Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp3$ gcc server.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp3$ ./a.out
bind done
Waiting for incoming connections......
Connection Accepted
recv failed
Hai 456789


MESSAGE RECIEVED

hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp3$
```

4A. User Datagram Protocol

Client Program

```
#include<stdio.h>
#include<sys/socket.h>
#include<string.h>
#include<stdlib.h>
#include<arpa/inet.h>//inet_addr
int main(int argc,char *argv[])
{
	struct sockaddr_in si_other;
	//Create Socket
	int s,i,slen=sizeof(si_other);
	char buf[2000];
	char message[2000];
	if((s=socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP))==-1)
	{
		perror("socket");
	}
	memset((char*)&si_other,0,sizeof(si_other));
	si_other.sin_family=AF_INET;
	si_other.sin_port=htons(8888);

	if(inet_aton("127.0.0.1",&si_other.sin_addr)==0)
	{
		fprintf(stderr,"inet_aton() failed\n");
		exit(1);
	}

	while(1)
	{
		printf("Enter Message\n");
		fscanf(stdin,"%s",message);

		//send the message
		if(sendto(s,message,strlen(message),0,(struct sockaddr*)&si_other,slen)==-1)
		{
			perror("sendto()");
		}
		//recieve a reply and print it
		//clear the buffer by filling it with null, it might have previously recieved data
		memset(buf,'\0',2000);
		//try to recieve some data, this is a blocking call
		if(recvfrom(s,buf,2000,0,(struct sockaddr*)&si_other,&slen)==-1)
		{
			perror("recvfrom()");
		}
		puts(buf);
	}
	return 0;
}
```

Exp 4a Client Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp4a$ gcc client.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp4a$ ./a.out
Enter Message
hello
hello
Enter Message
```

Server Program

```
#include<stdio.h>
#include<sys/socket.h>
#include<stdlib.h>
#include<string.h>
#include<arpa/inet.h>//inet_addr
int main(int argc,char *argv[])
{
	char message[2000];
	struct sockaddr_in si_me,si_other;
	int socket_desc,slen=sizeof(si_other),recv_len,i;
	socket_desc=socket(AF_INET,SOCK_DGRAM,0);
	if(socket_desc==-1)
	{
		printf("Could not create socket");
	}
	si_me.sin_family=AF_INET;
	si_me.sin_addr.s_addr=INADDR_ANY;
	si_me.sin_port=htons(8888);

	if(bind(socket_desc,(struct sockaddr*)&si_me,sizeof(si_me))<0)
	{
		puts("bind failed");
	}
	puts("bind done");

	//Keep listening for data
	while(1)
	{
		printf("Waiting for data.....");
		fflush(stdout);
		strcpy(message,"");
		//try to recieve some data, this is a blocking call
		if((recv_len=recvfrom(socket_desc,message,2000,0,(struct sockaddr*)&si_other,&slen))==-1)
		{
			perror("recvfrom()");
		}
		//print details of the client/peer and the data recieved
		printf("Recieved packet from %s:%d\n",inet_ntoa(si_other.sin_addr),ntohs(si_other.sin_port));

		for(i=0;i<recv_len;i++)
			putchar(message[i]);
		putchar('\n');
		//now reply the client with the same data
		if(sendto(socket_desc,message,recv_len,0,(struct sockaddr*)&si_other,slen)==-1)
		{
			perror("sendto()");
		}
	}
	//close(socket_desc);
	return 0;
}
```

Exp 4a Server Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp4a$ gcc server.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp4a$ ./a.out
bind done
Waiting for data......Recieved packet from 127.0.0.1:53211
hello
Waiting for data......
```

4B. Concurrent Time

Client Program

```
#include<stdio.h>
#include<sys/socket.h>
#include<string.h>
#include<stdlib.h>
#include<arpa/inet.h>//inet_addr
int main(int argc,char *argv[])
{
	struct sockaddr_in si_other;
	//Create Socket
	int s,i,slen=sizeof(si_other);
	char buf[2000];
	char message[2000];
	if((s=socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP))==-1)
	{
		perror("socket");
	}
	memset((char*)&si_other,0,sizeof(si_other));
	si_other.sin_family=AF_INET;
	si_other.sin_port=htons(8888);

	if(inet_aton("127.0.0.1",&si_other.sin_addr)==0)
	{
		fprintf(stderr,"inet_aton() failed\n");
		exit(1);
	}

	while(1)
	{
		printf("Enter Command:(date/time)\n");
		fscanf(stdin,"%s",message);

		//send the message
		if(sendto(s,message,strlen(message),0,(struct sockaddr*)&si_other,slen)==-1)
		{
			perror("sendto()");
		}
		//recieve a reply and print it
		//clear the buffer by filling it with null, it might have previously recieved data
		memset(buf,'\0',2000);
		//try to recieve some data, this is a blocking call
		if(recvfrom(s,buf,2000,0,(struct sockaddr*)&si_other,&slen)==-1)
		{
			perror("recvfrom()");
		}
		puts(buf);
	}
	return 0;
}
```

Exp 4b Client Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp4b$ gcc client.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp4b$ ./a.out
Enter Command:(date/time)
date
Date: 2022-5-30

Enter Command:(date/time)
time
Time: 6:11:50

Enter Command:(date/time)
```

4B. Concurrent Time

Server Program

```
#include<stdio.h>
#include<sys/socket.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>
#include<arpa/inet.h>//inet_addr
int main(int argc,char *argv[])
{
	char message[2000];
	struct sockaddr_in si_me,si_other;
	int socket_desc,slen=sizeof(si_other),recv_len,i;

	time_t t=time(NULL);
	struct tm tm=*localtime(&t);

	socket_desc=socket(AF_INET,SOCK_DGRAM,0);
	if(socket_desc==-1)
	{
		printf("Could not create socket");
	}
	si_me.sin_family=AF_INET;
	si_me.sin_addr.s_addr=INADDR_ANY;
	si_me.sin_port=htons(8888);

	if(bind(socket_desc,(struct sockaddr*)&si_me,sizeof(si_me))<0)
	{
		puts("bind failed");
	}
	puts("bind done");

	//Keep listening for data
	while(1)
	{
		printf("Waiting for data.....\n");
		fflush(stdout);
		strcpy(message,"");
		memset(message,'\0',2000);
		//try to recieve some data, this is a blocking call
		if((recv_len=recvfrom(socket_desc,message,2000,0,(struct sockaddr*)&si_other,&slen))==-1)
		{
			perror("recvfrom()");
		}

		//Checking Commands
		if(strcmp(message,"date")==0)
		{
			//sprintf(s,"%d",i);
			sprintf(message,"Date: %d-%d-%d\n",tm.tm_year+1900,tm.tm_mon+1,tm.tm_mday);
			//printf("Time: %d:%d:%d\n",tm.tm_hour,tm.tm_min,tm.tm_sec);
		}
		else if(strcmp(message,"time")==0)
		{
			//sprintf(s,"%d",i);
			//sprintf("Date: %d-%d-%d\n",tm.tm_year+1900,tm.tm_mon+1,tm.tm_mday);
			sprintf(message,"Time: %d:%d:%d\n",tm.tm_hour,tm.tm_min,tm.tm_sec);
		}
		puts(message);
		//now reply the client with the same data
		if(sendto(socket_desc,message,2000,0,(struct sockaddr*)&si_other,slen)==-1)
		{
			perror("sendto()");
		}
	}
//close(socket_desc);
return 0;
}
```

Exp 4b Server Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp4b$ gcc server.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp4b$ ./a.out
bind done
Waiting for data......
Date: 2022-5-30

Waiting for data......
Time: 6:11:50

Waiting for data......
```

Exp 5 Client Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp5$ ./a.out
number of frame is 10
sending frame is 1

ack for frame 1

sending frame is 2

ack for frame 2

sending frame is 3

ack for frame 3

sending frame is 4

ack for frame 4

sending frame is 5

ack for frame 5

sending frame is 6

ack for frame 6

sending frame is 7

ack for frame 7

sending frame is 8

ack for frame 8

sending frame is 9

ack for frame 9

sending frame is 10

ack for frame 10
```

5. Sliding Window Protocol

Client Program

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

int main(int argc,char *argv[])
{
	struct sockaddr_in si_other;
	//Create socket
	int s,slen=sizeof(si_other);
	char buf[2000];
	char message[2000];
	int i,j,noframe,x,x1=10,x2;
	noframe=10;
	i=1;
	j=1;
	if((s=socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP))==-1)
	{
		perror("socket");
	}
	memset((char*)&si_other,0,sizeof(si_other));
	si_other.sin_family=AF_INET;
	si_other.sin_port=htons(8888);
	if(inet_aton("127.0.0.1",&si_other.sin_addr)==0)
	{
		fprintf(stderr,"inet_aton() failed\n");
		exit(1);
	}
	printf("number of frame is %d\n",noframe);
	//getch();
	while(noframe>0)
	{
		printf("sending frame is %d\n",i);
		sprintf(message,"frame %d received\n",i);
		x=rand()%10;
		//message="frame received";
		if(x%10==0)
		{
			for(x2=1;x2<2;x2++)
			{
				printf("\nwaiting for %d seconds\n",x2);
				sleep(x2);
			}
			printf("\n sending frame%d\n",i);
			x=rand()%10;
		}
		noframe=noframe-1;
		i++;
		j++;
		if(sendto(s,message,strlen(message),0,(struct sockaddr*)&si_other,slen)==-1)
		{
			perror("sendto()");
		}
		if(recvfrom(s,buf,2000,0,(struct sockaddr*)&si_other,&slen)==-1)
		{
			perror("recvfrom()");
		}
		puts(buf);
	}
}
```

5. Sliding Window Protocol

Server Program

```
#include<stdio.h>
#include<sys/socket.h>
#include<stdlib.h>
#include<string.h>
#include<arpa/inet.h>//inet_addr
int main(int argc,char *argv[])
{
	char message[2000];
	char msg[2000];
	int i,j,noframe,x,x1=10,x2;
	noframe=10;
	i=1;
	j=1;
	struct sockaddr_in si_me,si_other;
	int socket_desc,slen=sizeof(si_other),recv_len;
	socket_desc=socket(AF_INET,SOCK_DGRAM,0);
	if(socket_desc==-1)
	{
		printf("Could not create socket");
	}
	si_me.sin_family=AF_INET;
	si_me.sin_addr.s_addr=INADDR_ANY;
	si_me.sin_port=htons(8888);
	if(bind(socket_desc,(struct sockaddr*)&si_me,sizeof(si_me))<0)
	{
		puts("bind failed");
	}
	puts("bind done");
	//keep listening for data
	while(1)
	{
		printf("Waiting for data...");
		fflush(stdout);
		strcpy(message,"");
		memset(message,'\0',2000);
		//try to receive some data, this is a blocking call
		if((recv_len=recvfrom(socket_desc,message,2000,0,(struct sockaddr*)&si_other,&slen))==-1)
		{
			perror("recvfrom()");
		}
		puts(message);
		while(noframe>0)
		{
			sprintf(msg,"\n ack for frame %d\n",j);
			if(sendto(socket_desc,msg,2000,0,(struct sockaddr*)&si_other,slen)==-1)
			{
				perror("sendto()");
			}
			noframe=noframe-1;
			j++;
		}
	}
	//close(socket_desc);
	return 0;
}
```

Exp 5 Server Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp5$ gcc server.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp5$ ./a.out
bind done
Waiting for data...frame 1 received

Waiting for data...frame 2 received

Waiting for data...frame 3 received

Waiting for data...frame 4 received

Waiting for data...frame 5 received

Waiting for data...frame 6 received

Waiting for data...frame 7 received

Waiting for data...frame 8 received

Waiting for data...frame 9 received

Waiting for data...frame 10 received

Waiting for data...
```

6. Distance Vector Routing Protocol

Program

```
#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);//Enter the 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][i]=0;
			rt[i].dist[j]=costmat[i][j];//initialise the distance equal to cost matrix
			rt[i].from[j]=j;
		}
	}
	do
	{
		count=0;
		for(i=0;i<nodes;i++)
		{//We choose arbitrary vertex k and
		 //we calculate the direct distance from the node i to k using the cost matrix
		//and add the distance from k to node j
			for(j=0;j<nodes;j++)
			{
				for(k=0;k<nodes;k++)
				{
					if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
					{	//We calculate the 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\nFor router %d\n",i+1);
		for(j=0;j<nodes;j++)
			printf("\t\nnode %d via %d Distance %d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
	}
	printf("\n\n");
	//getch();
}
```

Exp 6 Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp6$ gcc dvr.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp6$ ./a.out

Enter the number of nodes : 3

Enter the cost matrix :
0 1 4
1 0 10
4 10 0


For router 1

node 1 via 1 Distance 0
node 2 via 2 Distance 1
node 3 via 3 Distance 4

For router 2

node 1 via 1 Distance 1
node 2 via 2 Distance 0
node 3 via 1 Distance 5

For router 3

node 1 via 1 Distance 4
node 2 via 1 Distance 5
node 3 via 3 Distance 0

hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp6$
```

Client Program

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

int main(int argc,char *argv[])
{
	int socket_desc;
	char message[2000],filename[2000];
	struct sockaddr_in server;
	FILE *fp;
	//Create socket
	socket_desc=socket(AF_INET,SOCK_STREAM,0);
	if (socket_desc==-1)
	{
		printf("Could not create socket");
	}
	server.sin_addr.s_addr=inet_addr("127.0.0.1");
	server.sin_family=AF_INET;
	server.sin_port=htons(8888);

	//Connect to remote server
	if(connect(socket_desc,(struct sockaddr*)&server,sizeof(server))<0)
	{
		puts("connect error");
		return 1;
	}

	puts("Connected");

	do
	{
		//memset(message,'\n',2000);
		puts("FileName:");
		//fscanf(stdin,"%s",filename);
		scanf("%s",filename);
		if(send(socket_desc,filename,strlen(filename),0)<0)
		{
			puts("Send failed");
			return 1;
		}

		fp=fopen("backup","w");
		while(recv(socket_desc,message,2000,0))
		{
			if(strcmp(message,"EOF"))
			{
				fputs(message,fp);
				puts(message);
			}
			else
			{
				fclose(fp);
				break;
			}
		}
	}while(1);
//sleep(1);
return 0;
}
```

Server Program

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

int main(int argc,char *argvn)
{
	int socket_desc,new_socket,c;
	char filename[2000],message[2000];
	FILE *fp;
	struct sockaddr_in server,client;

	socket_desc=socket(AF_INET,SOCK_STREAM,0);
	if (socket_desc==-1)
	{
		printf("Could not create socket");
	}
	printf("socket created\n");
	server.sin_family=AF_INET;
	server.sin_addr.s_addr=INADDR_ANY;
	server.sin_port=htons(8888);

	if(bind(socket_desc,(struct sockaddr*)&server,sizeof(server))<0)
	{
		puts("bind failed");
		return(1);
	}
	puts("bind done");

	//Listen
	listen(socket_desc,3);

	//Accept and incoming connection
	puts("Waiting for incoming connections...");
	c=sizeof(struct sockaddr_in);
	new_socket=accept(socket_desc,(struct sockaddr*)&client,(socklen_t*)&c);
	if (new_socket<0)
	{
		perror("accept failed");
	}

	puts("Connection accepted");
	while(1)
	{
		recv(new_socket,filename,2000,0);
		fp=fopen(filename,"r");
		if(fp==NULL)
		{
			strcpy(message,"File Not Found\r\n");
			send(new_socket,message,2000,0);
			strcpy(message,"EOF");
		}
		else
		{
			while(fgets(message,2000,fp))
			{
				send(new_socket,message,2000,0);
				puts(message);
			}
			fclose(fp);
		}
		strcpy(message,"EOF");
		send(new_socket,message,2000,0);
	}
	return 0;
}
```

Exp 7 Client Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp7$ gcc client.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp7$ ./a.out
Connected
FileName:
hello.txt
Wilkommen

Gerngeschehen

FileName:
```

Exp 7 Server Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp7$ gcc server.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp7$ ./a.out
socket created
bind done
Waiting for incoming connections...
Connection accepted
Wilkommen

Gerngeschehen
```

File: backup.txt

```
Wilkommen
Gerngeschehen
```

8. Leaky Bucket

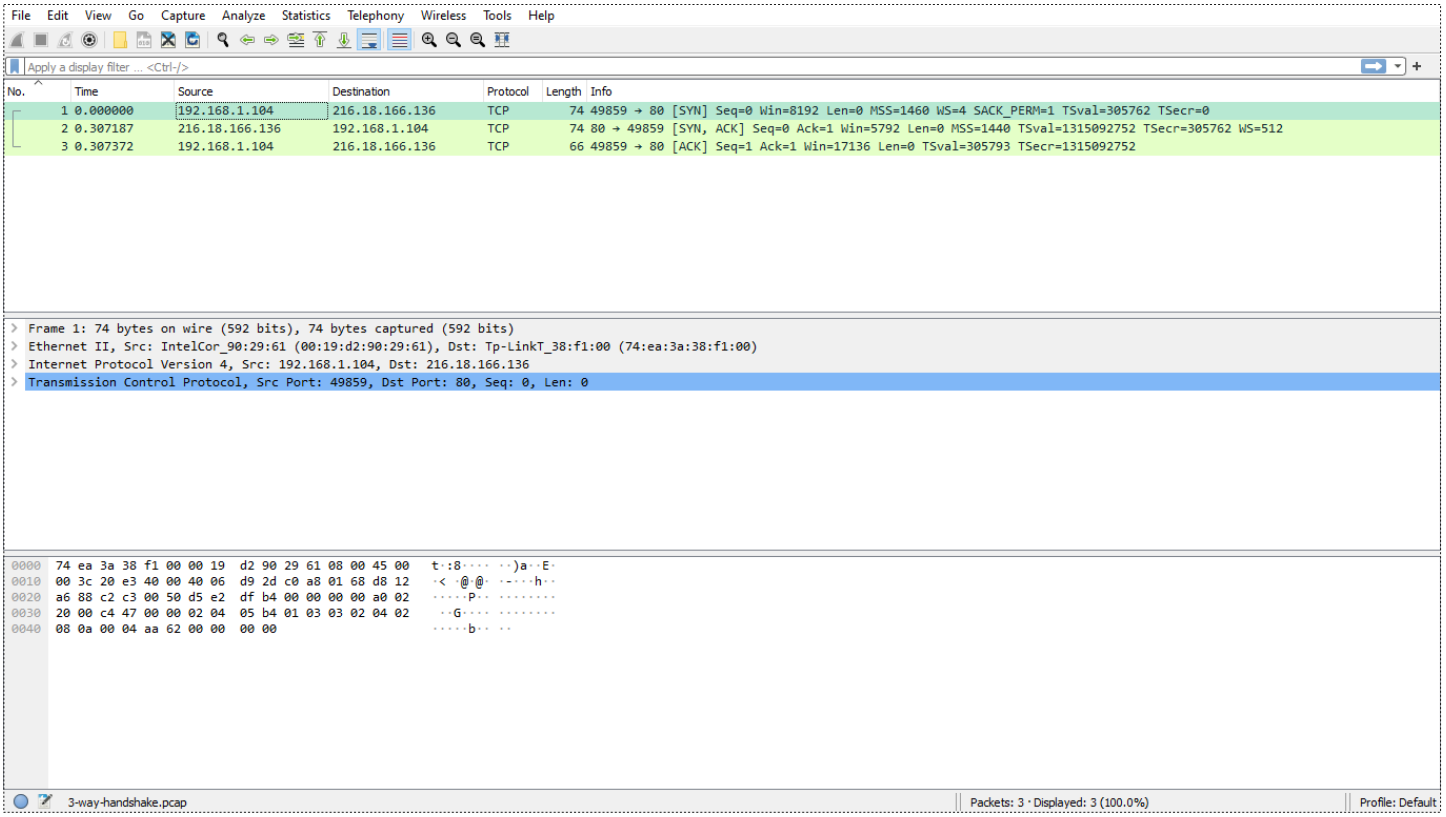
Program

```
#include<stdio.h>
#include<string.h>
int main()
{
   int no_of_queries,storage,output_pkt_size;
   int input_pkt_size,bucket_size,size_left;

   // initial packets in the bucket
   storage=0;
   // total no. of times bucket content is checked
   no_of_queries=4;

   // total no. of packets that can
   // be accommodated in the bucket
   bucket_size=10;

   // no. of packets that enters the bucket at a time
   input_pkt_size=4;

   // no. of packets that exits the bucket at a time
   output_pkt_size=1;
   for(int i = 0; i < no_of_queries; i++)//space left
   {
      size_left=bucket_size-storage;
      if(input_pkt_size<=size_left)
      {
         // update storage
         storage+=input_pkt_size;
         printf("Buffer size = %d out of bucket size = %d\n",storage,bucket_size);
      }
      else
      {
         printf("Packet loss = %d\n", (input_pkt_size-(size_left)));
         // full size
         storage=bucket_size;
         printf("Buffer size = %d out of bucket size = %d\n",storage,bucket_size);
      }
      storage-=output_pkt_size;
   }
return 0;
}
```

Exp 8 Output

```
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp8$ gcc lkybkt.c
hkc@hkc-HP-Pro-3090-MT:~/Desktop/riverrun/exp8$ ./a.out
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 3
Buffer size = 10 out of bucket size = 10
```

Exp 9 Output


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000 192.168.1.104 216.18.166.136 TCP 74 49859 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 TSval=305762 TSecr=0
2 0.307187 216.18.166.136 192.168.1.104 TCP 74 80 → 49859 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1440 TSval=1315092752 TSecr=305762 WS=512
3 0.307372 192.168.1.104 216.18.166.136 TCP 66 49859 → 80 [ACK] Seq=1 Ack=1 Win=17136 Len=0 TSval=305793 TSecr=1315092752
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Ethernet II, Src: IntelCor_90:29:61 (00:19:d2:90:29:61), Dst: Tp-LinkT_38:f1:00 (74:ea:3a:38:f1:00)
Internet Protocol Version 4, Src: 192.168.1.104, Dst: 216.18.166.136
Transmission Control Protocol, Src Port: 49859, Dst Port: 80, Seq: 0, Len: 0
0000 74 ea 3a 38 f1 00 00 19 d2 90 29 61 08 00 45 00
0010 00 3c 20 e3 40 00 40 06 d9 2d c0 a8 01 68 d8 12
0020 a6 88 c2 c3 00 50 d5 e2 df b4 00 00 00 00 a0 02
0030 20 00 c4 47 00 00 02 04 05 b4 01 03 03 02 04 02
0040 08 0a 00 04 aa 62 00 00 00 00
3-way-handshake.pcap
Packets: 3 · Displayed: 3 (100.0%)
Profile: Default