

Aim

A client using a wireless device wants to communicate to the server located in an unknown location. Demonstrate the concept with relevant protocol communication.

Procedure

- 1) Firstly create a extension file in cocalc by name netwok.c.
- 2) Then create two new files by name server.c and client.c
- 3) Then in next step create two new term files by naming client.term and server.term.
- 4) Then type the code on client and server side then run the output in the server.

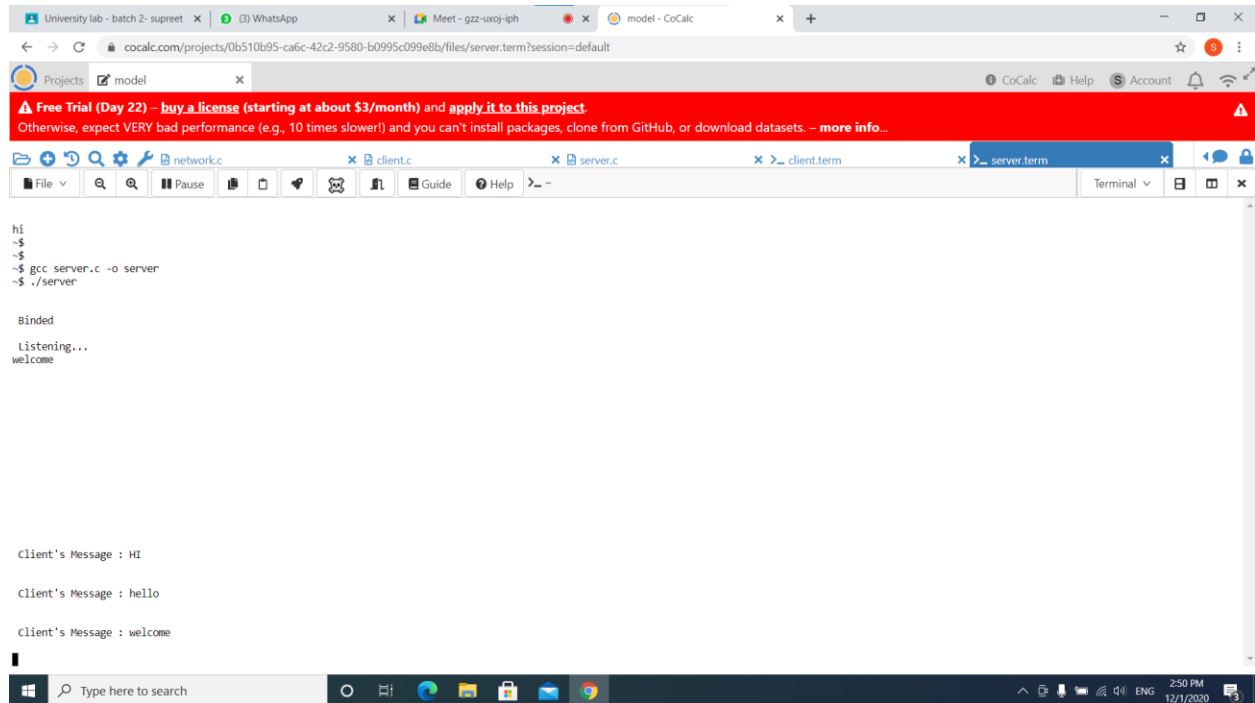
Server code:-

```
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<unistd.h>
#include<netdb.h>
```

```

#include<stdio.h>
#include<string.h>
#include<arpa/inet.h>
#define MAXLINE 1024
int main(int argc,char **argv)
{
    int sockfd;
    int n;
    socklen_t len;
    char msg[1024];
    struct sockaddr_in servaddr,cliaddr;
    sockfd=socket(AF_INET,SOCK_DGRAM,0);
    bzero(&servaddr,sizeof(servaddr));
    servaddr.sin_family=AF_INET;
    servaddr.sin_addr.s_addr=INADDR_ANY;
    servaddr.sin_port=htons(5035);
    printf("\n\n Binded");
    bind(sockfd,(struct sockaddr*)&servaddr,sizeof(servaddr));
    printf("\n\n Listening...");
    for(;;)
    {
        printf("\n ");
        len=sizeof(cliaddr);
        n=recvfrom(sockfd,msg,MAXLINE,0,(struct sockaddr*)&cliaddr,&len);
        printf("\n Client's Message : %s\n",msg);
        if(n<6)
            perror("send error");
        sendto(sockfd,msg,n,0,(struct sockaddr*)&cliaddr,len);
    }
    return 0;
}

```



The screenshot shows a web browser window with the CoCalc interface. A red banner at the top advertises a free trial and encourages buying a license. Below the banner, a file explorer shows several files: network.c, client.c, server.c, client.term, and server.term. The 'server.term' file is open in a terminal window. The terminal output shows the following commands and results:

```
hi
~$
~$
~$ gcc server.c -o server
~$ ./server

Binded
Listening...
welcome

Client's Message : HI

Client's Message : hello

Client's Message : welcome
```

The Windows taskbar at the bottom shows the search bar and system tray with the date 12/1/2020 and time 2:50 PM.

Client code :-

```
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<arpa/inet.h>
#include<string.h>
#include<arpa/inet.h>
#include<stdio.h>
#define MAXLINE 1024
int main(int argc,char* argv[])
{
int sockfd;
int n;
socklen_t len;
char sendline[1024],recvline[1024];
struct sockaddr_in servaddr;
strcpy(sendline,"");
printf("\n Enter the message : ");
scanf("%s",sendline);
```

```

sockfd=socket(AF_INET,SOCK_DGRAM,0);
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
servaddr.sin_port=htons(5035);
connect(sockfd,(struct sockaddr*)&servaddr,sizeof(servaddr));
len=sizeof(servaddr);
sendto(sockfd,sendline,MAXLINE,0,(struct sockaddr*)&servaddr,len);
n=recvfrom(sockfd,recvline,MAXLINE,0,NULL,NULL);
recvline[n]=0;
printf("\n Server's Echo : %s\n\n",recvline);
return 0;
}

```

The screenshot shows a web browser window with the CoCalc interface. The address bar shows a URL to a CoCalc project. A red banner at the top advertises a free trial. Below the banner, there are tabs for 'network.c', 'client.c', 'server.c', 'client.term', and 'server.term'. The 'client.term' tab is active, displaying a terminal window. The terminal shows the execution of a C program that sends and receives messages over a socket. The output shows the program successfully sending and receiving messages like 'welcome', 'Whatsupp', 'HI', 'hi', 'hello', and 'welcome'.

```

~$ gcc client.c -o client
~$ ./client
welcome
Whatsupp
~$
~$ gcc client.c -o client
~$ ./client
roger

Connection Failed
~$ gcc client.c -o client
~$ ./client
hi
welcome
~$ gcc client.c -o client
~$ ./client

Enter the message : HI
Server's Echo : HI

~$ gcc client.c -o client
~$ ./client

Enter the message : hello
Server's Echo : hello

~$ gcc client.c -o client
~$ ./client

Enter the message : welcome
Server's Echo : welcome

~$

```