

HAND



A Project Submitted to Gangadhar Meher University in partial fulfilment of the requirements for the award of the degree of

Bachelor of science

In

Information science and telecommunication

School of Information science and telecommunication

Gangadhar Meher University, Amruta Vihar, Sambalpur, Odisha,
India – 768004

March 10, 2022

Supervised by:

Mr. Sujit Tripathy

Faculty in school

of

IST & ETC

Submitted by:

Balkishan padihar (PC21IST021)

Pradyumna Kumar Mirdha (PC21IST022)

Contact: balkishanpadihar@gmail.com

kumarmonpradyu2001@gmail.com

DECLARATION

I hereby declare that the thesis entitled “**HAND**” to Gangadhar Meher University, for the award of the degree of the Bachelor of Science in information science and telecommunication is the result of the project work carried out by me in the Gangadhar Meher University, Amruta Vihar, Sambalpur, Odisha-768004 under the guidance of Ms. Sujit Tripathi faculty in School of IST & ETC, Gangadhar Meher University, Amruta Vihar, Sambalpur, Odisha.

I further declare that the results of this research work have not been submitted previously to this or any other University for any degree, diploma, or fellowship.

Date: March 10, 2022,

Place: Sambalpur

Balkishan padihar

Pradyumna Kumar Mirdha

Nabin Kumar Sahu



SCHOOL OF INFORMATION SCIENCE

AND

TELECOMMUNICATION

GANGADHAR MEHER UNIVERSITY, AMRUTA VIHAR,

SAMBALPUR, ODISHA, INDIA-76800

Mr. Satyaban Behera

E-mail: sbehera@gmuniversity.ac.in

(Hod of school of IST & ETC)

phone: +91 98612 83124

CERETIFICATE

This is to certify that the work contained in the thesis entitled “HAND”, submitted by Mr. Balkishan padihar, Mr. Pradyumna Kumar Mirdha & Mr. Nabin Kumar Sahu for the award of the degree of Bachelor of Science to the Gangadhar Meher University, Amruta Vihar, Sambalpur, Odisha-768004, is a record of Bonafede project works carried out by them under my direct supervision and guidance. I considered that the thesis has reached the standards and fulfilling the requirements of the rules and regulations relating to the nature of the degree. The content embodied in the thesis have not been submitted for the award of any other degree or diploma in this or any other Institute or University.

I wish him all success in his life.

Date: March 10, 2022

Place: Sambalpur

Signature of Supervisor

ACKNOWLEDGEMENT

Foremost, thanks to God, the Almighty for the wisdom he bestowed upon me and his showers of blessings throughout my dissertation work to complete it successfully.

Gratitude and indebtedness to my guide, Mr. Sujit Tripathi, faculty, School of IST & ETC, Gangadhar Meher University, Amruta Vihar, Sambalpur, Odisha-768004. His timely direction, complete co-operation and minute observation have made my research work fruitful rewarding.

I am grateful to all my teachers and non-teaching staff members of the School of IST & ETC, Gangadhar Meher University, Amruta Vihar, Sambalpur, Odisha, for their encouragement to finish this project work. I am thankful to my friends for their help and co-operation till completion of this report.

Lastly, my higher education and career was a dream of my parents and family members. I am indebted to all of them for their good will, love, and affection. I pray for the blessings of my parents without whom I could not have dreamt anything in my life.

Date: March 10, 2022,

Balkishan padihar

Place: Sambalpur

Pradyumna Kumar Mirdha

Nabin Kumar Sahu

Abstract

Our Project is based on networking. A collection of computer devices that are connected in various ways to communicate and share resources.

Usually, connection between computers in a network are made using physical wires and Cables. However, some are wireless using radio wave and infrared signal.

A computer network is incomplete without a host/node. Wherever data packets are stopped that is the host. Client is the one from whom we want to request on network. Server is the one who is giving the service or fulfilling the request of client.

There are many types of networks:

LAN

MAN

WAN

This project works on LAN (Local Area Network) connection. Network that connects small number of machines in close geographical area or building. It is a private network. A dedicated environment. First of all, We are creating a network connection between users. To connect with different system, we are using socket programming.

CONTENT

<u>CHAPTER NO</u>	<u>TITLE</u>	<u>PAGE</u>
1	INTRODUCTION	
2	1.1 What is socket	
3	1.2 what is socket programming	
4	1.3 why we named the project HAND	
2	OBJECTIVES	
3	METHODOLOGIES	
4	TOOLS USED	
5	ALGORITHMS	
6	FLOWCHART	
8	BASIC CMD COMMANDS	
9	BASIC NETWORKING THEORY	
10	ADVANTAGES & APPLICATIONS	
11	LIMITATIONS	
12	FUTURE SCOPE	
13	CONCLUSION	
14	REFERENCE	

INTRODUCTION

What is a Socket?

Sockets allow communication between two different processes on the same or different machines. To be more precise, it's a way to talk to other computers using standard Unix file descriptors. In Unix, every I/O action is done by writing or reading a file descriptor. A file descriptor is just an integer associated with an open file and it can be a network connection, a text file, a terminal, or something else.

To a programmer, a socket looks and behaves much like a low-level file descriptor.

Just a file descriptor and can perform read/write over file descriptor. It creates end point to receive and send the information over the network.

What is socket programming?

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

Why we named the project HAND?

We are creating this project over TCP connection, and we all know that TCP uses 2–3-way handshake and handles the flow of control of data. We use two hands to have a better control on work like clapping, lifting. Like that we are creating a connection which will control the other Operating system.

OBJECTIVE

To exchange inform between two devices and to operate one device with another device. Sockets **allow you to exchange information between processes on the same machine or across a network, distribute work to the most efficient machine, and they easily allow access to centralized data.** Socket application program interfaces (APIs) are the network standard for TCP/IP.

A *socket* is a communications connection point (endpoint) that you can name and address in a network. Socket programming shows how to use socket APIs to establish communication links between remote and local processes.

The processes that use a socket can reside on the same system or different systems on different networks. Sockets are useful for both stand-alone and network applications. Sockets allow you to exchange information between processes on the same machine or across a network, distribute work to the most efficient machine, and they easily allow access to centralized data. Socket application program interfaces (APIs) are the network standard for TCP/IP. A wide range of operating systems support socket APIs. i5/OS sockets support multiple transport and networking protocols. Socket system functions and the socket network functions are threadsafe.

Programmers who use Integrated Language Environment® (ILE) C can refer to this topic collection to develop socket applications. You can also code to the sockets API from other ILE languages, such as RPG.

Methodology

1.First , We are including window's socket header file to create a socket.

Winsock2.h

- a. Winsock2.h is specifically created for socket programming for windows.
- b. We are going to use some functions inside our program.
Ex : htons, inet_addr, recv, WSStartup, WSAConnect, etc.
- c. Some of the structures we are going to use are:
SOCKADDR, Sockaddr_in, WSADATA

2.Then , We are including windows header file for calling windows API functions.

Windows.h

The header files for the Windows API enable you to create 32- bit and 64-bit applications. They include declarations for both Unicode and ANSI versions of the API.

3.We are asking for an TCPIP connection.

Ws2tcpip.h

This library is used for creating connection oriented service in our program.

4.Now we are trying to link the TCPIP header file to winsock header file .

#pragma comment(lib, "Ws2_32.lib")

For creating a winsock application We have to add Ws2_32.lib.The #pragma comment indicates to the linker that the Ws2_32.lib file is needed.

5. We need to declare a socket structure.

SOCKET

It is a predefined structure inside winsock header file.

This datatype structure is used to create a socket.

6. Initializing the environment for the sockaddr structure.

Sockaddr_in

The sockaddr structure varies depending on the protocol selected.

It contains the IPv4 socket address.

7. Initializing WSADATA .

8. Declaring a variable to hold the data.

9. Declaring a variable which holds the port number.

1. Now our main function starts where we will connect with the server and spawn a shell.

2. Before initializing a socket we are going to declare winsock version to our program.

3. Now , we will create a file descriptor which contains the family , type of connection , protocol, and the flags.

```
socket(  
    [in] int af,  
    [in] int type,  
    [in] int protocol  
);
```

[in] af

The address family specification. Possible values for the address family are defined in the *Winsock2.h* header file.

The values currently supported are AF_INET or AF_INET6, which are the Internet address family formats for IPv4 and IPv6.

[in] type

The type specification for the new socket. Possible values for the socket type are defined in the *Winsock2.h* header file.

SOCK_STREAM

A socket type that provides sequenced, reliable, two-way, connection-based byte streams with an OOB data transmission mechanism. This socket type uses the Transmission Control Protocol (TCP) for the Internet address family (AF_INET or AF_INET6).

SOCK_DGRAM

A socket type that supports datagrams, which are connectionless, unreliable buffers of a fixed (typically small) maximum length. This socket type uses the User Datagram Protocol (UDP) for the Internet address family (AF_INET or AF_INET6).

[in] protocol

The protocol to be used. The possible options for the *protocol* parameter are specific to the address family and socket type specified. Possible values for the *protocol* are defined in the *Winsock2.h* .

IPPROTO_TCP

6

The Transmission Control Protocol (TCP). This is a possible value when the *af* parameter is AF_INET or AF_INET6 and the *type* parameter is SOCK_STREAM.

IPPROTO_UDP

17

The User Datagram Protocol (UDP). This is a possible value when the *af* parameter is AF_INET or AF_INET6 and the *type* parameter is SOCK_DGRAM.

```
typedef struct sockaddr_in {
    short    sin_family;
    u_short  sin_port;
    struct in_addr sin_addr;
    char     sin_zero[8];
} SOCKADDR_IN, *PSOCKADDR_IN, *LPSOCKADDR_IN;
```

4. Declaring the family type to the socket Address.

5. sin_family

Declaring the IP address to the socket address.

sin_port

sin_addr

sin_zero

6. Declaring the port number to the socket address.

7. We are going to convert the host byte to the network byte format using the big endian.

8. Now, we will set WSAconnection to start connection to the server.

9. Some logics to the program is given.

If the connection to the routing do not matches, then it will return error.

10. If the socket connects with the server, then will receive a signal showing the connection is successful.

11. Now we will set the receiving data to zero which will reset after each reply.

12. Spawning of cmd through windows api to create a process.

13. Now we are piping the input, output, and errors to the socket.

14. For spawning up the process we must specify the start-up information and process information structure.

15. We will specify the size of info structure.

16. We will specify the flags to the info structure which will be related to the cmdline.

17. We will pass all of this to the handle of socket descriptor.

18. Create a process to spawn CMD.

19. We will call for single object function which will wait until the outputs or inputs are passed.
20. Now we will close the handles.
21. Now the programs will exit when the server side send exit command or receiver side close the connection.

TOOLS USED

A personal computer/laptop (having windows operating system)

Ease of use. Users familiar with earlier versions of Windows will probably also find the more modern ones easy to work with. This is ascribable to everything from the standardised look and feel of almost all programs written for Windows to the way the file system has been presented ever since the days of MS-DOS (disk A:\, disk C:\, etc.). This is one of the main reasons why Windows users are often reluctant to switch operating systems.

Available software. There is a huge selection of software available for Windows. This is both due to and the reason for Microsoft's dominance of the world market for PC computer operating systems and office software. If you're looking for an application to suit your business needs, chances are that if it exists there will be a Windows version of it available somewhere.

Backwards compatibility. If you're currently using an older version of Windows and need something more up to date, but you don't want to lose the use of some older programs that are only available for Windows and are critical to your business needs, the chances are good (although not a certainty) that those programs will also work with a newer version of Windows.

Support for new hardware. Virtually all hardware manufacturers will offer support for a recent version of Windows when they go to market with a new product. Again, Microsoft's dominance of the software market makes Windows impossible for hardware manufacturers to ignore. So, if you run off to a store today and buy some random new piece of computer hardware, you'll find that it will probably work with the latest version of Windows.

Compatibility with MS driven websites. After Windows had become the world's most popular desktop operating system, Internet Explorer (IE) became the world's most popular web browser soon after Microsoft began bundling it with Windows 95 in order to squash competition from rival Netscape's Navigator browser. Since Netscape's demise, Microsoft have introduced more and more proprietary features into their web servers that can only be taken advantage of with Internet Explorer. Obviously, these sites are less accessible with other browsers – sometimes not at all. This, coupled with the fact that the latest versions of IE are only available for Windows, has made Windows the only choice for those who want to take full advantage of those websites that use Microsoft's technology.

2. A personal computer / smartphone

3. Visual studio

Visual studio is open source and any one can use it for free. We use visual studio to do this program because doing programming with visual studio enhances our efficiency. Any type of program like cpp, program using graphics, etc are compile and run easily in visual studio. In visual studio prebuild Microsoft compiler is also present. So,

we use visual studio to do this program. We can also run the command line and command line in the visual studio which is not easy to execute in the other compiler.

Micro Focus recommends that you create and use Visual Studio projects for your source code. There is a number of benefits to using projects such as:

- Projects take full advantage of the numerous time-saving and productivity features of Visual Studio. Features such as IntelliSense suggestions and syntax checking in the editor are not fully available if you are using the files without a project.
- The Visual Studio project files store the details of any changes you make in the IDE.
- The Visual Studio project files are MS Build files. You can compile them at the command line using MS Build and integrate the build command in your build scripts or in a CI system.

The following sections include some recommendations about how to create and configure your projects, especially when you are working with many files.

4. Termux application

Termux is a free and open-source terminal emulator for Android which allows for running a Linux environment on an Android device. In addition, various software can be installed through the application's package manager.

The Power of Termux You're trying to create an program, and you want to do so with Python/C/C++/BASH/Php because it's simple to program with. This would be all well and good if the app were for a computer. As it is, you're trying to design an program for a mobile device. But, wait you're probably arguing that Python/C/C++/BASH/Php was not made to run on mobile devices.\

Termux is basically a programmer's dream. In itself, it is an application run as an emulator, which, in this case, is a mobile system made to act like a standard PC, on mainly Android devices. [Termux](#) uses a secure protocol, specifically OpenSSH, the open-source, open-community version of a standard secure networking Secure Shell (SSH). This means that anyone with programming experience can collaborate with each other, and may do so over a secure network, no matter where they physically are. This makes the platform perfect for programming on the go, whether it be for work, practice, or leisure. Additionally, for those inexperienced with programming, Termux is a simple-enough application to [learn and practice](#) Linux command lines. This is mainly because the Linux command

system is easy to understand; but its simplicity can also be attributed to Python's presence, should the programmer install its package. Termux already comes with a library of several packages, including but not limited to SQLMap – a vulnerability detection tool – and Wireshark – a Network analyser and trouble-shooter. All you need to do is command the install of the one you want. In Python's case, you would need to enter the command `pkg install python (pip)`. Once your desired system is established, you're ready to start programming.

5. Netcat software

Netcat is a utility capable of establishing a TCP or UDP connection between two computers, meaning it can write and read through an open port. With the help of the program, files can be transferred, and commands can be executed in some instances.

Netcat can be and is also used by server administrators. When a server is hacked, the hacker usually changes and infects the binary files on the system, and so even if the administrator starts cleaning the system it may not succeed because the hacker can track his work. Think about changing your passwords, for example. In these cases, if we have recorded on a cd the netcat program, it can connect to the server without the connection being compromised.

Another advantage of the program is the ability to copy files over the network without having an FTP server, HTTP or any other service that allows data to be transmitted. With netcat, both small and large files can be copied very easily, including full partitions.

Algorithm

Step 1: We are trying to create a connection between a user and the server.

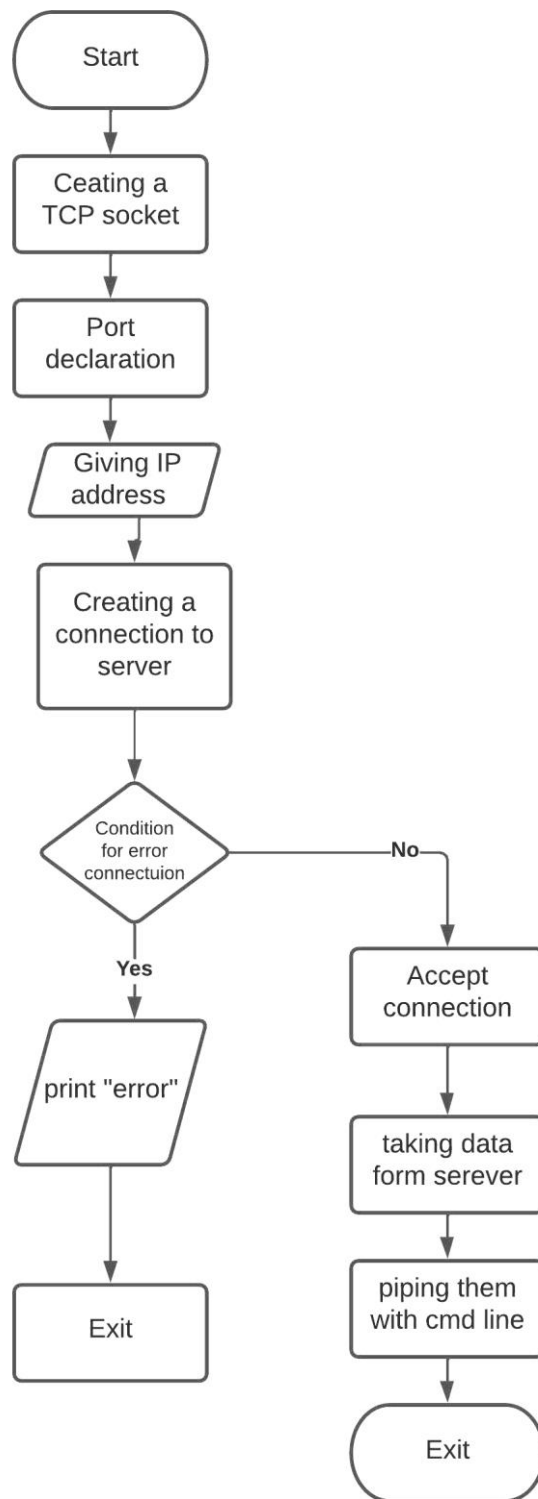
Step 2: Giving the IP address to the connection .

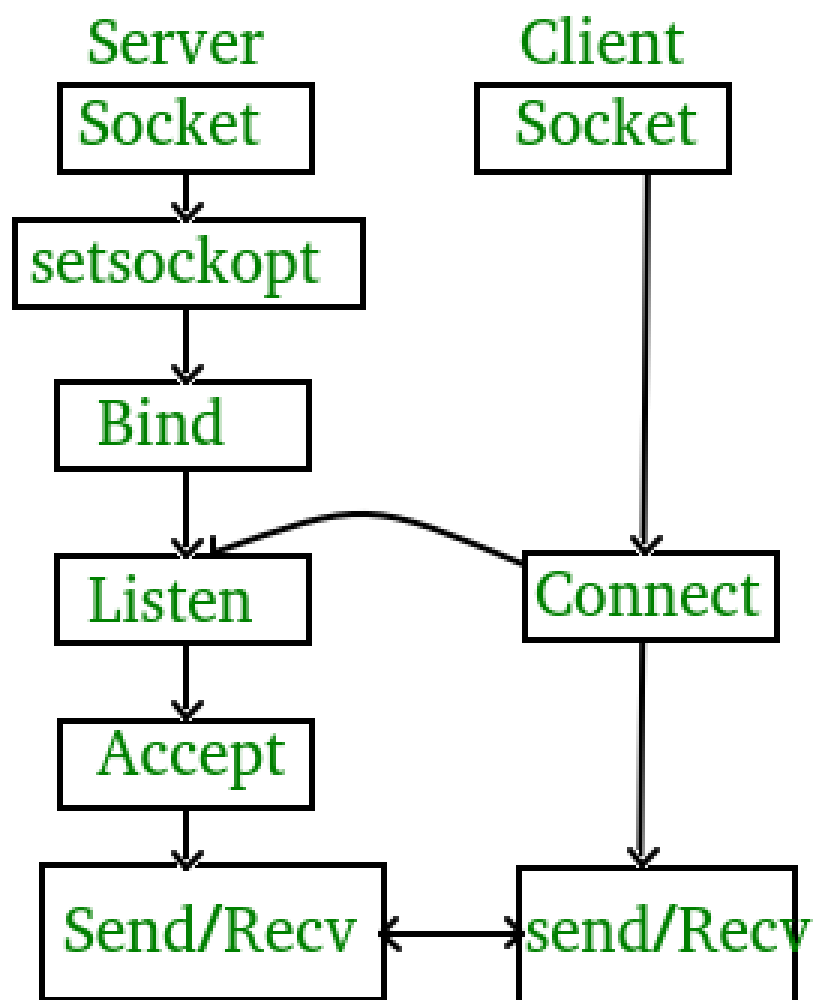
Step 3: Telling the connection which type of protocol to use

Step 4: After connection is made , now we will send our message to the destination address.

Step 5: Instead of receiving out the messages form the client we are replacing the output with the client's CLI.

Step 6: After the work done, we will exit out of the connection.





Basic CMD commands

cmd command	Description
Basics:	
call	calls a batch file from another one
cd	change directory
cls	clear screen
cmd	start command prompt
color	change console color
date	show/set date
dir	list directory content
echo	text output
exit	exits the command prompt or a batch file
find	find files
hostname	display host name
pause	pauses the execution of a batch file and shows a message

runas	start a program as another user
shutdown	shutdown the computer
sort	sort the screen output
start	start an own window to execute a program or command
taskkill	terminate a process or a application
tasklist	display applications and related tasks
time	display/edit the system time
timeout	wait any time
title	set title for prompt
ver	display operating system version
w32tm	setting time synchronisation/time server/time zone

Basic networking theory

A computer network is incomplete without a host/node. Wherever data packets are stopped that is the host.

Client is the one from whom we want to request on network .

Sever is the one who is giving the service or fulfilling the request of client.

There are many types of network:

LAN

MAN

WAN

This project works on LAN (Local Area Network) connection.

Network that connects small number of machines in close geographical area or building.

It is a private network. A dedicated environment.

First of all We are creating a network connection between users.

To connect with different system, we are using socket programming.

Switches, routers, and wireless access points are the essential networking basics. Through them, devices connected to your network can communicate with one another and with other networks, like the Internet. Switches, routers, and wireless access points perform very different functions in a network.

Switches

Switches are the foundation of most business networks. A switch acts as a controller, connecting computers, printers, and servers to a network in a building or a campus.

Switches allow devices on your network to communicate with each other, as well as with other networks, creating a network of shared

resources. Through information sharing and resource allocation, switches save money and increase productivity.

There are two basic types of switches to choose from as part of your networking basics: on-premises and cloud-managed.

- A managed on-premises switch lets you configure and monitor your LAN, giving you tighter control of your network traffic.
- Have a small IT team? A cloud-managed switch can simplify your network management. You get a simple user interface, multisite full-stack management, and automatic updates delivered directly to the switch.

Routers

Routers connect multiple networks together. They also connect computers on those networks to the Internet. Routers enable all networked computers to share a single Internet connection, which saves money.

A router acts a dispatcher. It analyses data being sent across a network, chooses the best route for data to travel, and sends it on its way.

Routers connect your business to the world, protect information from security threats, and can even decide which computers receive priority over others.

Beyond those basic networking functions, routers come with additional features to make networking easier or more secure. Depending on your security needs, for example, you can choose a router with a firewall, a virtual private network (VPN), or an Internet Protocol (IP) communications system.

We wanted to make a way to control the system , over other .

And for this reason we use the socket programming which is based only for Windows limiting this program only for Windows OS.

The reason is that , we want the control through CLI .

Command Line Interface is different on different System.

For windows it is Cmd or Powershell.

For Linux it is Terminals and so on.

So , If we had written a program with every OS compatibility then, It couldn't have found the CLI of those OS.

First of all We are creating a network connection between users.

To connect with different system we are using socket programming.

Socket programming is one way to connect two or more different system. But It only used to communicate with the client , server .

Why TCP connection?

For the connection , we choose TCP/IP model because first TCP/IP model is used in modern internet architecture.

The original TCP/IP portocol suite was defined as four software layers built upon hardware which was not thought of as a five-layer model .

TCP/IP Model contains:

Application layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

TCP/IP model is based on standard protocol around which the internet has developed. It is a communication protocol.

It follows horizontal approach.

When making TCP session from one session we could sent multiple request which means it is connection oriented.

Internet Protocol(IP)

Data is transferred in the form of packets via logical network paths in an ordered format controlled by network layer.

It does this by forwarding packets to network router, which relay on algorithms to determine the best path for data to travel. These paths are known as virtual circuits.

The Internet Protocol si the method or protocol by which data is sent from one computer to another on internet.

Each computer on the internet has at least one IP address that uniquely identifies it from all other computer on internet.

In IP a message is divided into a number of packets, each packets can be sent by different router across the internet.

The requirements for the packets are that it should be:

1. Independent of specifics of network
2. Independent of specific path
3. Understandable by any router along the way

The IP protocol makes a best effort to deliver the packets. It does not handle

- 1) Datagram duplication
- 2) Delay or out-of-order delivery

- 3) Corruption of data
- 4) Datagram loss

The reason for us to choose TCP over UDP because TCP is connection oriented Protocol. Means the connection will be transmitted until any one terminates the connection. Whereas , UDP is a Datagram or connection less Protocol .

TCP is more reliable than UDP because it guarantees the delivery of data to the destination router. The data flow of control is better in TCP connection .

Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in order at the receiver.

IP version Used

In our program we have only included the IPv4 format because IPv4 is in our present world many devices contain both IPv4 and IPv6 but in older devices IPv4 formats are available . So in order to make connection in each and every type of devices we used IPv4 type address.

IPv4 is a 32bit field . It contains the logical address of the senders or the receiver's datagram.

It contains checksum feild and it's message Transmission has broadcast scheme.

Its header file varies from 20-60 bytes.

It uses ARP to map the MAC address.

Its best feature is that it uses NAT devices which allows single NAT address can make thousand of non routable address, making end-to-end integrity achievable.

The Classes our ISP provides for IPv4 addressing are:

Class A

Class B

Class C

Functions

SOCKET:

It is a predefined structure inside winsock header file.

This datatype struture is used to create a socket.

sockaddr_in:

The SOCKADDR_IN structure specifies a transport address and port for the AF_INET address family.

sin_family:

The address family for the transport address. This member should always be set to AF_INET.

sin_port:

A transport protocol port number.

sin_addr:

An IN_ADDR structure that contains an IPv4 transport address.

sin_zero:

Reserved for system use. A WSK application should set the contents of this array to zero.

WSADATA:

The WSADATA structure contains information about the Windows Sockets implementation.

STARTUPINFO:

Specifies the window station, desktop, standard handles, and appearance of the main window for a process at creation time.

Cb:

The size of the structure, in bytes.

dwFlags:

bitfield that determines whether certain **STARTUPINFO** members are used when the process creates a window.

PROCESS_INFORMATION:

Contains information about a newly created process and its primary thread.

hProcess:

A handle to the newly created process. The handle is used to specify the process in all functions that perform operations on the process object.

hThread:

A handle to the primary thread of the newly created process. The handle is used to specify the thread in all functions that perform operations on the thread object.

WSASocket():

```
SOCKET WSAAPI WSASocketA(  
    [in] int          af,  
    [in] int          type,  
    [in] int          protocol,  
    [in] LPWSAPROTOCOL_INFOA lpProtocolInfo,  
    [in] GROUP         g,  
    [in] DWORD         dwFlags  
);
```

WSAConnect:

The WSAConnect function establishes a connection to another socket application, exchanges connect data.

```
int WSAAPI WSAConnect(  
    [in] SOCKET      s,  
    [in] const sockaddr *name,  
    [in] int          namelen,  
    [in] LPWSABUF     lpCallerData,  
    [out] LPWSABUF     lpCalleeData,  
    [in] LPQOS         lpSQOS,  
    [in] LPQOS         lpGQOS  
);
```

[in] s

A descriptor identifying an unconnected socket.

[in] name

A pointer to a `sockaddr` structure that specifies the address to which to connect. For IPv4, the `sockaddr` contains `AF_INET` for the address family, the destination IPv4 address, and the destination port. For IPv6, the `sockaddr` structure contains `AF_INET6` for the address family, the destination IPv6 address, the destination port, and may contain additional flow and scope-id information.

[in] namelen

The length, in bytes, of the `sockaddr` structure pointed to by the *name* parameter.

[in] lpCallerData

A pointer to the user data that is to be transferred to the other socket during connection establishment. See Remarks.

[out] lpCalleeData

A pointer to the user data that is to be transferred back from the other socket during connection establishment. See Remarks.

[in] lpSQOS

A pointer to the `FLOWSPEC` structures for socket *s*, one for each direction.

[in] lpGQOS

Reserved for future use with socket groups. A pointer to the `FLOWSPEC` structures for the socket group (if applicable). This parameter should be `NULL`.

Recv():

The **recv** function receives data from a connected socket or a bound connectionless socket.

Memset:

It converts the value `ch` to unsigned char and copies it into each of the first `n` characters of the object pointed to by `str[]`. If the object is not trivially-copyable (e.g., scalar, array, or a C-compatible struct), the behavior is undefined. If `n` is greater than the size of the object pointed to by `str`, the behavior is undefined.

CreateProcess:

To run the new process in the security context of the user represented by the impersonation token we use `createProcess`.

WaitforSingleObject:

Waits until the specified object is in the signaled state or the time-out interval elapses.

```
DWORD WaitForSingleObject(  
    [in] HANDLE hHandle,  
    [in] DWORD dwMilliseconds  
);
```

[in] hHandle:

A handle to the object. For a list of the object types whose handles can be specified, see the following Remarks section.

If this handle is closed while the wait is still pending, the function's behavior is undefined.

[in] dwMilliseconds:

The time-out interval, in milliseconds. If a nonzero value is specified, the function waits until the object is signaled or the interval elapses.

hProcess:

A handle to the newly created process. The handle is used to specify the process in all functions that perform operations on the process object.

hThread:

A handle to the primary thread of the newly created process. The handle is used to specify the thread in all functions that perform operations on the thread object.

STARTF_USESTDHANDLES:

The `hStdInput`, `hStdOutput`, and `hStdError` members contain additional information.

STARTF_USESHOWWINDOW:

The `wShowWindow` member contains additional information.

Advantage & application

Sockets are flexible and sufficient. Efficient socket-based programming can be easily implemented for general communications. It causes

low network traffic.

Socket based communications allows only to send packets of raw data between applications. Both the client-side and server-side must provide mechanisms to make the data useful in any way.

Networking is a concept of connecting two or more computing devices together so that we can share resources. The term network programming refers to writing programs that execute across multiple devices (computers), in which the devices are all connected to each other using a network.

- Provides flexible access to files and data over a network.
- Sharing resources.
- Security.
- Speed.
- Centralized software management.
- Provide security like sending sensitive (password protected) files and programs on a network.

Network Terminology

The following are networking terms:

- MAC Address
- IP address
- Protocol
- Port no.
- Connection-oriented and Connectionless protocol
- Socket

Protocol is a set of defined rules for communication between client and server.
For example:

- SMTP
- TCP
- FTP
- Telnet
- POP etc.

CONCLUSION

In this project work we created a local area Network (LAN) between two devices. by using several commands, we can operate client devices from the server.

Although we have some security concerns, but it can be modify in future.

After modification we can use single server to manipulate different system.

Future scope

This program can be modified in various ways. As the networking theories and the socket programming concepts are evolving day by day but in our programming, we have only declared the client as a window operated device.

But in future we can enhance the programme to some other operating system such as Linux, mac. DOCS operating system.

Right now, we are providing the private IP address of the server directly in the source code, but we can further develop it and we can ask the user to give the IP address at the run time.

This program currently requires a strong internet connection, but we can modify this program, so that it can be operated without an internet connection.

REFERENCES

<https://stackoverflow.com/questions/39314086/what-does-it-mean-to-bind-a-socket-to-any-address-other-than-localhost>

<https://docs.microsoft.com/en-us/windows/win32/api/winsock/ns-winsock-wsadata>

https://docs.microsoft.com/en-us/windows/win32/api/ws2def/ns-ws2def-sockaddr_in

<https://pubs.opengroup.org/onlinepubs/009695399/basedefs/netinet/in.h.html>

<https://stackoverflow.com/questions/19207745/htons-function-in-socket-programing>