**Department of Computer Engineering**

**NAME : TAUSIF KHAN SUBJECT: OSTL**

**STD: SE COMPS(B) ROLL NO: 12**

**EXPERIMENT NO: 10**

**Department of Computer Engineering**

**Experiment No: 10.**

**AIM:** Write a program to create a two-way communication between Server and Client in python

**TOOLS USED:** Python 3.4.3, Terminal

**THEORY:**

#### Explain the basics of network ?

Ans)

A network consists of two or more computers that are linked in order to share resources (such as printers and CDs), exchange files, or allow electronic communications. The computers on a network may be linked through cables, telephone lines, radio waves, satellites, or infrared light beams.

Two very common types of networks include:

## Local Area Network:

* + A Local Area Network (LAN) is a network that is confined to a relatively small area. It is generally limited to a geographic area such as a writing lab, school, or building.
  + Computers connected to a network are broadly categorized as servers or workstations. Servers are generally not used by humans directly, but rather run continuously to provide "services" to the other computers (and their human users) on the network. Services provided can include printing and faxing, software hosting, file storage and sharing, messaging, data storage and retrieval, complete access control (security) for the network's resources, and many others.
  + Workstations are called such because they typically do have a human user which interacts with the network through them. Workstations were traditionally considered a desktop, consisting of a computer, keyboard, display, and mouse, or a laptop, with with integrated keyboard, display, and touchpad. With the advent of the tablet computer, and the touch screen devices such as iPad and iPhone, our definition of workstation is

**Department of Computer Engineering**

quickly evolving to include those devices, because of their ability to interact with the network and utilize network services.

* + Servers tend to be more powerful than workstations, although configurations are guided by needs. For example, a group of servers might be located in a secure area, away from humans, and only accessed through the network. In such cases, it would be common for the servers to operate without a dedicated display or keyboard. However, the size and speed of the server's processor(s), hard drive, and main memory might add dramatically to the cost of the system. On the other hand, a workstation might not need as much storage or working memory, but might require an expensive display to accommodate the needs of its user. Every computer on a network should be appropriately configured for its use.
  + On a single LAN, computers and servers may be connected by cables or wirelessly. Wireless access to a wired network is made possible by wireless access points (WAPs). These WAP devices provide a bridge between computers and networks. A typical WAP might have the theoretical capacity to connect hundreds or even thousands of wireless users to a network, although practical capacity might be far less.
  + Nearly always servers will be connected by cables to the network, because the cable connections remain the fastest. Workstations which are stationary (desktops) are also usually connected by a cable to the network, although the cost of wireless adapters has dropped to the point that, when installing workstations in an existing facility with inadequate wiring, it can be easier and less expensive to use wireless for a desktop.
  + See the [Topology](https://fcit.usf.edu/network/chap5/chap5.htm), [Cabling](https://fcit.usf.edu/network/chap4/chap4.htm), and [Hardware](https://fcit.usf.edu/network/chap3/chap3.htm) sections of this tutorial for more information on the configuration of a LAN.

## Wide Area Network:

Wide Area Networks (WANs) connect networks in larger geographic areas, such as Florida, the United States, or the world. Dedicated transoceanic cabling or satellite uplinks may be used to connect this type of global network.

**Department of Computer Engineering**

Using a WAN, schools in Florida can communicate with places like Tokyo in a matter of seconds, without paying enormous phone bills. Two users a half- world apart with workstations equipped with microphones and a webcams might teleconference in real time. A WAN is complicated. It uses multiplexers, bridges, and routers to connect local and metropolitan networks to global communications networks like the Internet. To users, however, a WAN will not appear to be much different than a LAN.

##### NETWORKING DEVICES:

[**Switches**](https://www.cisco.com/c/en/us/solutions/small-business/switches.html)**:**

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: managed and unmanaged.

* An unmanaged switch works out of the box but can't be configured. Home- networking equipment typically offers unmanaged switches.
* A managed switch can be configured. You can monitor and adjust a managed switch locally or remotely, giving you greater control over network traffic and access.

# [Routers](https://www.cisco.com/c/en/us/solutions/small-business/routers.html):

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 analyzes data being sent across a network, chooses the best route for data to travel, and sends it on its way.

**Department of Computer Engineering**

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 needs, for example, you can choose a router with a firewall, a virtual private network (VPN), or an Internet Protocol (IP) communications system.

# Access points:

An access point allows devices to connect to the wireless network without cables. A wireless network makes it easy to bring new devices online and provides flexible support to mobile workers.

An access point acts like an amplifier for your network. While a router provides the bandwidth, an access point extends that bandwidth so that the network can support many devices, and those devices can access the network from farther away.

But an access point does more than simply extend Wi-Fi. It can also give useful data about the devices on the network, provide proactive security, and serve many other practical purposes.

Access points support different IEEE standards. Each standard is an amendment that was ratified over time. The standards operate on varying frequencies, deliver different bandwidth, and support different numbers of channels.

#### 2.What are the requirements to establish a network ?

**Ans)**

In order for a network to function, three basic requirements must be met: it must provide connections, communications, and services.

**Connections:**

Connections include the hardware (physical components) required to hook up a computer to the network. Two terms are important to network connections:

**Department of Computer Engineering**

* + **The network medium:** The network hardware that physically connects one computer to another. This is the cable between the computers.
  + **The network interface:** The hardware that attaches a computer to the network medium and acts as an interpreter between the computer and the network. Attaching a computer to a network requires an add-in board known as a network interface card (NIC).

### **Communications:**

Communications establish the rules concerning how computers talk and understand each other. Because computers often run different software, in order to communicate with each other they must speak a "shared language." Without shared communications, computers cannot exchange information, and remain isolated.

### **Services:**

A service defines those things a computer shares with the rest of the network. For example, a computer can share a printer or specific directories or files.

Unless computers on the network are capable of sharing resources, they remain isolated, even though physically connected.

### The basic elements of network connections:

Next we look at how the basic elements of connections, communications, and services work together to make networks function properly:

* + The connections must operate so that any computer can send or receive electrical signals (data) across the physical media that link them.
  + Communications must function so that when one computer sends a message, the receiving computer can listen and understand the message.
  + Computers on a network must either provide a service to other computers or make use of a service provided by other computers.
* Connectivity

**Department of Computer Engineering**

* Cost-Effective Resource Sharing
* Support for Common Services
* Performance

Requirements differ according to the perspective:

1. **Application programmer**

List the services that his or her application needs.

Example: A guarantee that each message it sends will be delivered without error within a certain amount of time.

1. **Network designer**

List the properties of a cost-effective design.

Example: The network resources efficiently utilized and fairly allocated to different users.

1. **Network provider**

List the characteristics of a system that is easy to administer and manage. Example: Fault can be easily isolated and it is easy to account for usage.

**1. Connectivity**

A network must provide connectivity among a set of computers

* Links and Nodes
* Types of Links or Connections
* Direction of Data Flow
* Unicast, Broadcast and Multicast
  1. **Links and Nodes**

A network consists of two or more computers directly connected by some physical medium, such as a coaxial cable or an optical fiber. Such a physical medium is called as **links.**

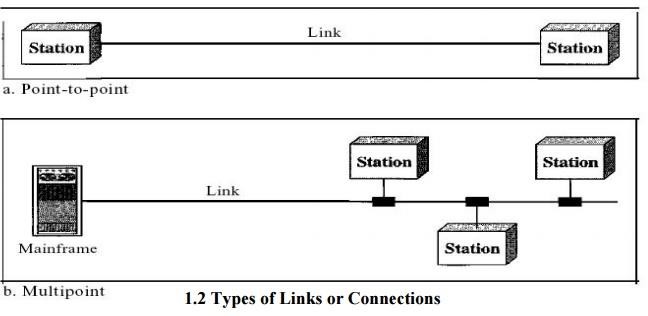
The links are connected to the computers named as **nodes.**

* 1. **Types of Links or Connections Point-to-Point**

A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices.

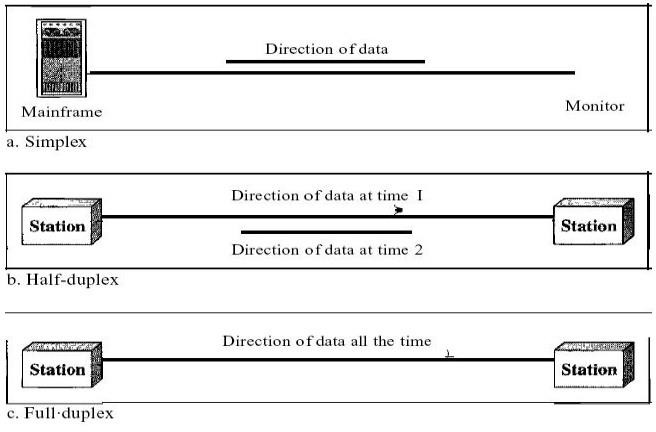
**Multipoint**

A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link. In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a *spatially shared* connection. If users must take turns, it is a *timeshared* connection.



* 1. **Direction of Data Flow**

Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure



**Simplex:**

In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive (Fig a). Keyboards and traditional monitors are examples of simplex devices.

**Half-Duplex:**

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa(Fig b). The half-duplex mode is like a one-lane road with traffic allowed in both directions.

**Full-Duplex:**

In full-duplex both stations can transmit and receive simultaneously (Fig c)

**1.3 Unicast, Broadcast and Multicast Unicast**

Unicast is the term used to describe communication where a piece of information is sent from one point to another point. In this case there is just one sender, and one receiver.

**Broadcast**

Broadcast is the term used to describe communication where a piece of information is sent from one point to all other points. In this case there is just one sender, but the information is sent to all connected receivers.

**Multicast**

Multicast is the term used to describe communication where a piece of information is sent from one or more points to a set of other points. In this case there is may be one or more senders, and the information is distributed to a set of receivers (theer may be no receivers, or any other number of receivers).

**2.Cost-Effective Resource Sharing**

Multiplexing is a way that a system resource is shared among multiple users.



Two or more simultaneous transmissions on a single circuit. Transparent to end user. Multiplexing cost less.

Multiple telephone channels may share a transmission link by means of multiplexing – this sharing is static

FDM (Frequency Division Multiplexing) is used in analogue systems (a telephone analogue channel has a nominal bandwidth of 4 kHz)

STDM (Synchronous Time Division Multiplexing) is used in digital systems (the basic telephone digital channel has a capacity of 64 kbit/s)

**3.Support for Common Services**

A computer network provides more than packet delivery between nodes. We don’t want application developers to rewrite for each application higher layer networking services.

The channel is a pipe connecting two applications. How to fill the gap between the underlying network capability and applications requirements? a set of common services– Delivery guarantees, security, delay.

* 1. **Types of Applications**

Interactive terminal and computer sessions:– Small packet length, small delay, high reliability.

1. File transfer:– High packet length, high delay, high reliability
2. Voice application:– Small packet length, small delay, small reliability, high arrival rate
3. Video-on-demand:– Variable/high packet length, fixed delay, small reliability.

**2.State the different protocols used in networking and explain them in detail ?**

**Ans)**

Definition - What does *Network Protocols* mean?

Network protocols are formal standards and policies comprised of rules, procedures and formats that define communication between two or more devices over a network.

They ensure that computer network devices can transmit and receive data using a common language regardless of their different designs, hardware or infrastructures.

Network protocols govern the end-to-end processes of timely, secure and managed data or network communication.

**Department of Computer Engineering**

They can be built into hardware or software, and they’re so important that, in practice, every network use rely on network protocols for communications and connectivity.

Network protocols incorporate all the processes, requirements and constraints of initiating and accomplishing communication between computers, servers, routers and other network-enabled devices.

They must be confirmed and installed by the sender and receiver to ensure network/data communication and apply to software and hardware nodes that communicate on a network.

Network protocols are developed and published by several groups according to certain industry standards.

The most important ones include:

* World Wide Web Consortium (W3C)
* International Telecommunication Union (ITU)
* International Organization for Standardization (ISO)
* Internet Engineering Task Force (IETF)
* The Institute of Electrical and Electronics Engineers (IEEE)

A protocol suite is a set of different network protocols that cooperate to exchange information and enable internet connectivity across various layers.

For example: the TCP/IP suite comprehends the Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Protocol (IP), and Hypertext Transfer Protocol (HTTP) or File Transfer Protocol (FTP).

Each one of them allow the transmission of data between devices at the information packet level (mainly TCP, and UDP as well to add an additional layer of stability), Internet address level (IP), and display level (HTTP and FTP).

##### Broad Types of Networking Protocols:

**Network communication protocols**

Communication protocols allow basic data communication between network devices. Their purposes range from transferring files between computers or via the internet, to exchange text-based messages, and establishing communication between routers and external or IoT devices.

Examples of communication protocols are:

* Bluetooth protocols
* FTP

**Department of Computer Engineering**

* TCP/IP
* And HTTP

**Network security protocols**

Security protocols implement security over network communications by guaranteeing that data transferred over a network cannot be accessed by unauthorized users.

Security is established through various means, such as via the use of passwords, authentication protocols, or data encryption.

Data can be cryptographed and protected during transportation between devices, and access can be denied to unknown or unauthorized devices or users.

Examples of security protocols include:

* HTTPS
* SSL
* SSH
* And SFTP

**Network management protocols**

Network management protocols provide network governance and maintenance by defining the procedures required to operate a network.

They are applied on all devices operating in a given network — such as servers, routers, and computers — to coordinate them in an efficient way.

Network management protocols ensure that each device is connected to the others and to the network itself, and guarantee the stability of these connections.

They are often used for troubleshooting purposes and to assess the quality of the network connection.

Examples of network management protocols include:

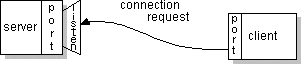
* SNMP
* and ICMP

**2.What are Sockets and explain them in detail ?**

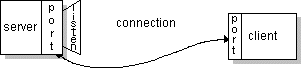
**Ans)**

Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

**Department of Computer Engineering**

On the client-side: The client knows the hostname of the machine on which the server is running and the port number on which the server is listening. To make a connection request, the client tries to rendezvous with the server on the server's machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system.

If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. It needs a new socket so that it can continue to listen to the original socket for

connection requests while tending to the needs of the connected client.

On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server.

The client and server can now communicate by writing to or reading from their sockets.

**Definition:**

A *socket* is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

**Socket Programming in Python**

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.

They are the real backbones behind web browsing. In simpler terms there is a server and a client.

Socket programming is started by importing the socket library and making a simple socket.

**Department of Computer Engineering**

import socket

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

Here we made a socket instance and passed it two parameters. The first parameter is AF\_INET and the second one is SOCK\_STREAM. AF\_INET refers to the address family ipv4. The SOCK\_STREAM means connection oriented TCP protocol.

Now we can connect to a server using this socket. Connecting to a server:

Note that if any error occurs during the creation of a socket then a socket.error is thrown and we can only connect to a server by knowing it’s ip. You can find the ip of the server by using this :

$ ping [www.google.com](http://www.google.com/)

You can also find the ip using python:

import socket

ip = socket.gethostbyname('www.google.com') print ip

**Department of Computer Engineering**

#### 2.Explain IP Address and their formats in detail ? Ans)

An **Internet Protocol address** (**IP address**) is a numerical label assigned to each device connected to a [computer network](https://en.wikipedia.org/wiki/Computer_network) that uses the [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) for communication[.[1](https://en.wikipedia.org/wiki/IP_address#cite_note-rfc760-1)[][2]](https://en.wikipedia.org/wiki/IP_address#cite_note-rfc791-2) An IP address serves two main functions: host or network interface [identification](https://en.wikipedia.org/wiki/Identification_(information)) and

location [addressing](https://en.wikipedia.org/wiki/Network_address).

[Internet Protocol version 4](https://en.wikipedia.org/wiki/IPv4) (IPv4) defines an IP address as a [32-bit](https://en.wikipedia.org/wiki/32-bit) number[.[2]](https://en.wikipedia.org/wiki/IP_address#cite_note-rfc791-2) However, because of the growth of the Internet and the [depletion of available IPv4 addresses](https://en.wikipedia.org/wiki/IPv4_address_exhaustion), a new version of IP ([IPv6](https://en.wikipedia.org/wiki/IPv6)), using 128 bits for the IP address, was standardized in 1998[.[3](https://en.wikipedia.org/wiki/IP_address#cite_note-rfc1883-3)[][4]](https://en.wikipedia.org/wiki/IP_address#cite_note-rfc2460-4)[[5]](https://en.wikipedia.org/wiki/IP_address#cite_note-rfc8200-5) [IPv6 deployment](https://en.wikipedia.org/wiki/IPv6_deployment) has been ongoing since the mid-2000s.

IP addresses are written and displayed in [human-readable](https://en.wikipedia.org/wiki/Human-readable) notations, such as 172.16.254.1 in IPv4, and 2001:db8:0:1234:0:567:8:1 in IPv6. The size of the routing prefix of the address is designated in [CIDR notation](https://en.wikipedia.org/wiki/CIDR_notation) by suffixing the address with the number of significant bits,

e.g., 192.168.1.15/24, which is equivalent to the historically used subnet mask 255.255.255.0.

The IP address space is managed globally by the [Internet Assigned Numbers Authority](https://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority) (IANA), and by five [regional Internet registries](https://en.wikipedia.org/wiki/Regional_Internet_registry) (RIRs) responsible in their designated territories for assignment to [local Internet registries](https://en.wikipedia.org/wiki/Local_Internet_registry), such as [Internet service providers](https://en.wikipedia.org/wiki/Internet_service_providers), and other end users. IPv4 addresses were distributed by IANA to the RIRs in blocks of approximately 16.8 million addresses each, but have been exhausted at the IANA level since 2011. Only one of the RIRs still has a supply for local assignments in Africa[.[6]](https://en.wikipedia.org/wiki/IP_address#cite_note-6) Some IPv4 addresses are reserved for [private](https://en.wikipedia.org/wiki/Private_network) [networks](https://en.wikipedia.org/wiki/Private_network) and are not globally unique.

Network administrators assign an IP address to each device connected to a network. Such assignments may be on a static (fixed or permanent) or dynamic basis, depending on network practices and software features.

**Department of Computer Engineering**

**IP Address Formats**

Originally IP addresses were divided into five *classes* as shown below. Classes A, B and C are the most important: the initial bits determine which class an address belongs to, and the classes differ in how much of the address is taken up with the *network address* and how much with the *host address*.

Offsets

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 |  | 8 |  | 16 |  | 24 |

Class A

|  |  |  |
| --- | --- | --- |
| 0 Network |  | Host |

Addresses 1.0.0.0 to 127.255.255.255

Class B

|  |  |  |
| --- | --- | --- |
| 10 Network |  | Host |

Addresses 128.0.0.0 to 191.255.255.255

Class C

|  |  |  |
| --- | --- | --- |
| 110 Network |  | Host |

Addresses 192.0.0.0 to 223.255.255.255

Class D

1110 Multicast address

Addresses 224.0.0.0 to 239.255.255.255

Class E

11110 Reserved for future use

Addresses 240.0.0.0 to 255.255.255.255

**Department of Computer Engineering**

**Exclusions**

Some address ranges are excluded from the above classes. All addresses of the form 127.xx.yy.xx are used for diagnostics. The following address ranges are designated as "private": meaning thay can only be used on networks which are not connected to the global Internet:

10.0.0.0 - 10.255.255.255

172.16.0.0.0 - 172.31.255.255

192.168.0.0 - 192.168.255.255

**Examples**

Find the class, network and host addresses for:

a. 4.23.145.90

b. 227.34.78.7

c. 246.7.3.8

d. 129.6.8.4

e. 198.76.9.23

## Subnetting, Supernetting and Network Address Translation

The classfull addressing scheme has proved to be too inflexible to accommodate the demand for IP addresses; specifically, the convention that the part of the address which identifies a physical network must come on an 8-bit boundary meant that the address space could not be allocated efficiently and a significant proportion of addresses would be wasted. For instance, a Class A network can in theory have 16 million host IDs, which is way more than would be possible for a single physical network (subnet).

Many enterprises have been allocated Class B addresses, giving them a theoretical maximum of 65,535 hosts, and may have no more than a few hundred hosts or less: under the classfull scheme the remaining host IDs for that network ID would be unused.

The lack of inflexibility in the addressing scheme is particulalrly undesirable because according to many authorities there is a real danger of the Internet

**Department of Computer Engineering**

running out of addresses, as a result of several factors which were not envisaged when the Internet protocols were originally designed, such as :

* + take-up by business and domestic users
  + increasing demand for "always-on" connections requiring a dedicated IP address, so that ISPs have less ability to share out a limited number of IP addresses by reallocating them as customers disconnect
  + convergence of computing, communications and entertainment industries could mean that every TV set, mobile phone, playstation and DVD player will want to have an IP address.

**Subnetting**

The basic idea behind subnetting is that we take away some bits from the host portion of the address and use them to identify the *subnet*. This will only be visible to hosts and routers on the local network; from the point of view of the Internet at large only the network ID portion will have any particular meaning. The way this works is that the network administrator defines a *subnet mask*, a string of 32 binary digits indicating the boundary between the subnet and host sections of the "local" portion of an IP address. Some examples should help make this clear:

1. Take the Class A address 9.67.38.1, where 9 is the network address and 67.38.1 identifies a particular host on that network. Suppose the network administrator wants to use bits 8 to 25 to identify the subnet, leaving 26 to 31 for host addresses. This is done by means of a *mask* which is all 1's from bits 0 to 25 with the remainder set to 0's:

11111111 11111111 11111111 11000000

The subnet can then be indentified by ANDing the complete IP address with the mask giving the result shown below:

00001001 01000011 00100110 00000001 = class A

address 9.67.38.1

00001001 01000011 00100110 00000000 = subnet base

address 9.67.38.0

**Department of Computer Engineering**

1. Suppose a class C network has the subnet mask

11111111 11111111 11111111 11100000

This means that the first three bits from the host ID are used to identify the subnet, and the remaining 5 bits indicate the host itself. Take a host with the IP address

11010100 01110010 00010000 01110001 =

212.114.32.113

Applying the subnet mask gives us

11010100 01110010 00010000 01100000 =

212.114.32.96

The host ID proper comes out as 00010001 = 17.

For compatibility, routers include masks for unsubnetted addresses:

A. 255.0.0.0

B. 255.255.0.0

C. 255.255.255.0

1. N/A
2. N/A

**Supernetting and Classless Inter-Domain Routing (CIDR)**

CIDR is in a sense the converse of subnetting, in that bits are taken from the network ID to identify a cluster of network addresses which are to be treated as a single entity. The main reasons for this are:

* + As a consequence of the address exhaustion problem, Class B addresses are only assigned to organisations that can show a clear need for them: organisations which do not meet the requirements are instead given a contiguous block of Class C addresses.
  + Having a series fo router table entries for networks which all map onto the same route is wasteful of space and degrades routing algorithms.

CIDR uses a *supernet mask* along with the *lowest* network address in the assigned block to identify the range of assigned network addresses.

**Network Address Translation and IPv6**

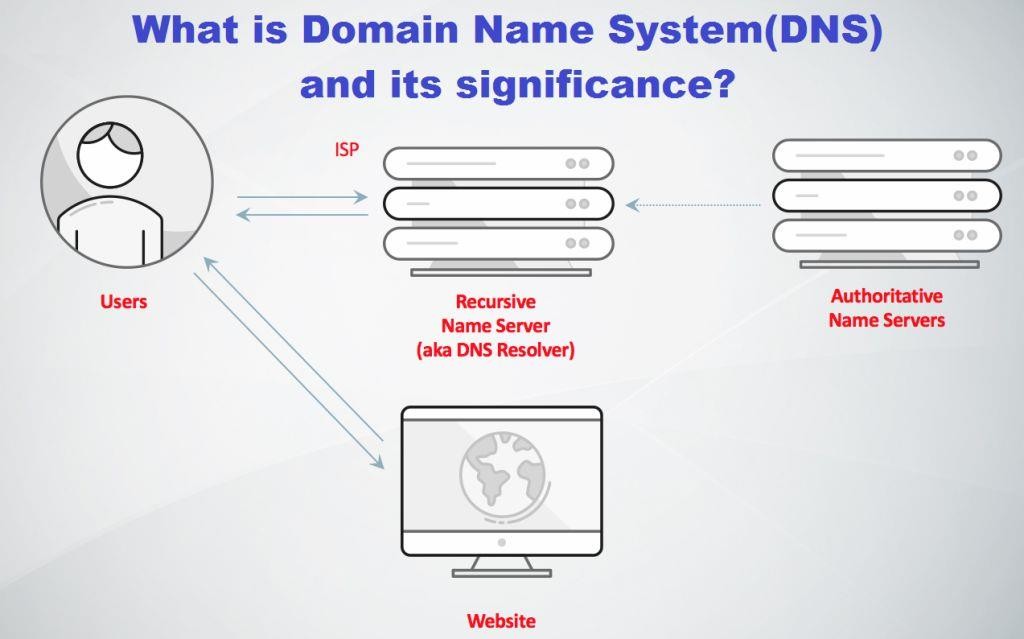
Network Address Translation is a technique which allows for the composition of a network to be completely hidden from the outside world, with the entire network identified by a single IP address. Within the network, hosts and routers have addresses which are unique to that network, typically taked from the ranges designated as "private" (see above). In order to make sure that responses get back to the right hosts when packets are sent out into the Internet, the router will construct a table associating outgoing packets with private IP addresses; the address of the relevant table entry will be stored in the packet itself. This technique is controversial however; for one thing the field in the packet where the index into the table is stored is part of the TCP header, which violates the principles of modularity and encapsulation on which the "protocol stack" models are based.

#### Explain what is DNS and its significance ? Ans)

Short for D*omain* N*ame* S*ystem* (or S*ervice* or S*erver*),

an [Internet](https://www.webopedia.com/TERM/I/Internet.html) service that translates [*domain names*](https://www.webopedia.com/TERM/D/domain_name.html) into IP addresses. Because domain names are alphabetic, they're easier to remember. The Internet however, is really based on [IP addresses](https://www.webopedia.com/TERM/I/IP_address.html). Every time you use a domain name, therefore, a DNS service must translate the name into the corresponding IP address. For example, the domain name [*www.example.com*](http://www.example.com/) might translate to *198.105.232.4*.

The DNS system is, in fact, its own [network](https://www.webopedia.com/TERM/N/network.html). If one DNS server doesn't know how to translate a particular domain name, it asks another one, and so on, until the correct IP address is returned.



**Significance of DNS server:**

It will enhance security to discourage cyber attacks and a quicker site loading time. If a user inadvertently accesses a malicious or phishing site, the new public system would immediately open up a page or popup to alert the user of such potential threat so that the suspicious resource could be avoided, the official who is aware of the initiative. The security features are created by the National Informatics Centre (NIC) and has a capability to host as many as 5 million users that can be scaled up further if needed.

No doubt, DNS is an important tool that requires to be fool-proof and has a major role in browsing the internet. The new DNS will be placed across the country to minimise outage and would be available round the clock.

Users simply can use by typing the IP number in to the internet browser.

So, now you may have understood about the DNS technology as an effective cyber security measures and policies to stop cyber attackers.

#### Write in short client and server in networking ?

**Ans)**

A **client-server network** is designed for end-users, called **clients**, to access resources such as files, songs, video collections, or some other service from a central computer called a **server**. A server's sole purpose is to do what its name implies - serve its clients! You may have been using this configuration and not even have known it. Have you ever played Xbox Live or used the PlayStation Network? Your Xbox One is the client, and when it logs into the network, it contacts the Xbox Live servers to retrieve gaming resources like updates, video, and game demos.

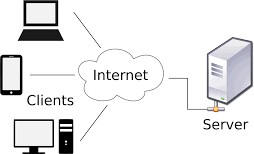
**How Does It Work?**

Imagine a customer sitting at a restaurant. He is waiting for the server to come by and take his order. The same rules apply in a client-server network; the client, which can be a laptop, desktop, a smartphone, or pretty much any computerized device, can make a request from the server.

The client uses the **network** as a way to connect with and speak to the server. Just as the customer speaks to his server, the client uses the network to send and receive communications about its order, or request. The server will take the request and make sure that the request is valid. If everything checks out okay, then the server will fetch the request and serve the client.

The server can make a request from the client as well. It may want to check up on the status of the client, or ask if it has received any security patches, or if it still needs resources from the server. If not, the server will close the connection in order to free up network traffic.

Can you imagine a server standing next to a customer who just stares at the menu without ordering anything? After 15 minutes, it would be a good idea for the server to leave and check on other customers. In both cases, the server moves on to other clients as needed.



**Advantages:**

Centralization of control: access, resources and integrity of the data are controlled by the dedicated server so that a program or unauthorized client cannot damage the system. This centralization also facilitates task of updating data or other resources (better than the networks P2P).

Scalability: You can increase the capacity of clients and servers separately. Any element can be increased (or enhanced) at any time, or you can add new nodes to the network (clients or servers).

Easy maintenance: distribute the roles and responsibilities to several standalone computers, you can replace, repair, upgrade, or even move a server,

while customers will not be affected by that change (or minimally affect). This independence of the changes is also known as encapsulation.

There are technologies sufficiently developed, designed for the paradigm of C / S to ensure security in transactions, interface friendliness, and ease of use.

**Disadvantages:**

Traffic congestion has always been a problem in the paradigm of C / S. When a large number of simultaneous clients send requests to the same server might cause many problems for this (to more customers, more problems for the server). On the contrary, P2P networks each node in the network server also makes more nodes, the better bandwidth you have.

The paradigm of C / S Classic does not have the robustness of a network P2P. When a server is down, customer requests cannot be met. In most part, P2P networks resources are usually distributed across multiple nodes of the network. Although some out or abandon download, others may still end up getting data download on rest of the nodes in the network.

The software and hardware of a server are usually very decisive. A regular computer hardware staff may not be able to serve a certain number of customers. Usually you need specific software and hardware, especially on the server side, to meet the work

. Of course, this will increase the cost.

The client does not have the resources that may exist on the server. For example, if the application is a Web, we cannot write the hard disk of the client or print directly on printers without taking before the print preview window of the browser.

**CODE:**

**#Server.py**

import socket

host = '127.0.0.1'

port = 8000

#create server side socket

s= socket.socket()

s.bind((host,port))

#let maximum number of connections are 1 only

s.listen(1)

#wait till a client connects

c, addr = s.accept()

print("A Client Connected")

#server runs continously

while True:

#receive data from client

data= c.recv(1024)

#if client sends empty string, come out

if not data:

break

print("from client: "+str(data.decode()))

#enter response data from server

data1 = input("Enter response: ")

#send that data to client

c.send(data1.encode())

#close connection

c.close()

**#Client.py**

import socket

host = '127.0.0.1'

port = 8000

#create a client side socket

s= socket.socket()

s.connect((host,port))

#enter data at client

str= input("Enter data: ")

#continue as long as exit not entered by user

while str!='exit':

#send data from client to server

s.send(str.encode())

#receive the response data from server

data = s.recv(1024)

data = data.decode()

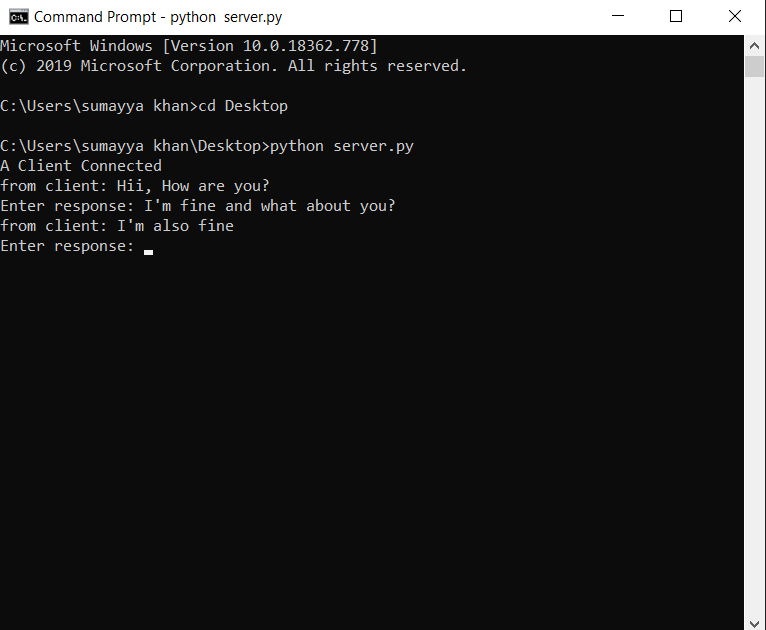
print("from server: "+data)

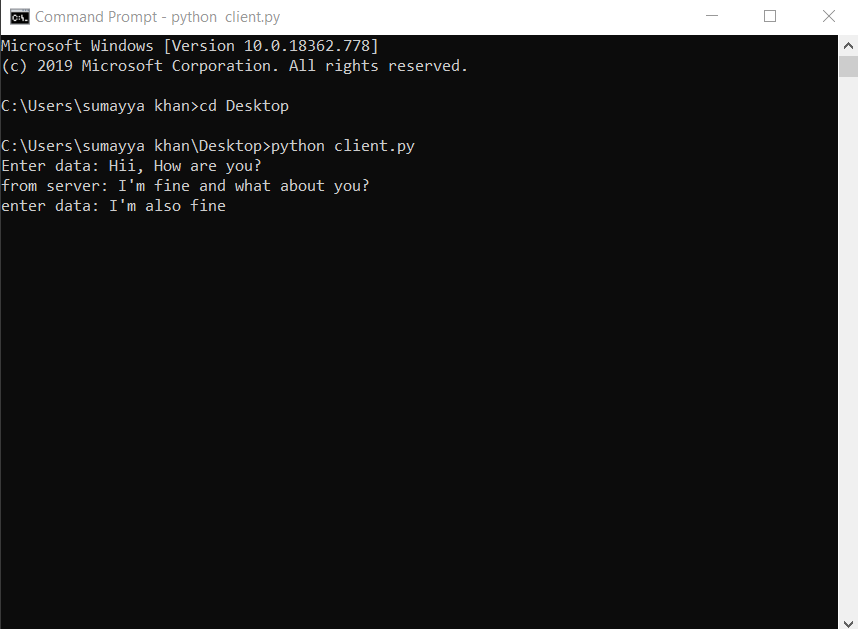
#enter data

str=input("enter data: ")

s.close()

**OUTPUT:**



\*

**CONCLUSION:**

Thus, we have successfully implemented the concept of networking in python.