DEPARTMENT OF COMPUTER SCIENCE & ENGG.
ANNA UNIVERSITY, CHENNAI 600 025

B.E. (Computer Science & Engineering)

V Semester – for P Batch

CS6111 – Computer Networks

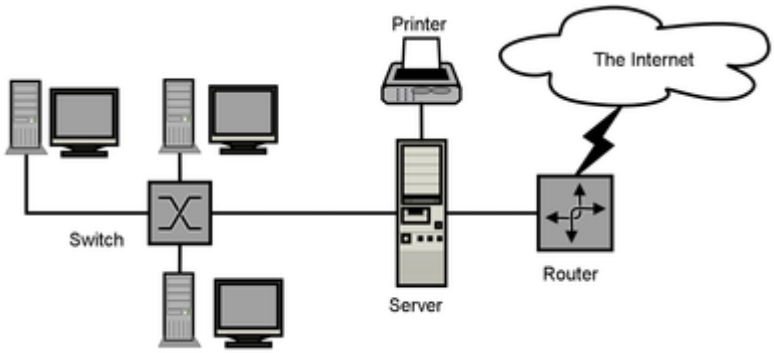
Lab Exercise: 01

(Dated 02/08/2024)

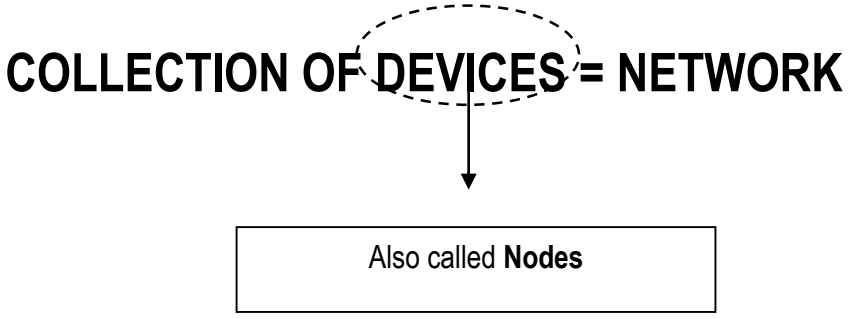
Preparatory Content:

What is a computer network?

A computer network is a group of interconnected nodes or computing devices that exchange data and resources with each other. A network connection between these devices can be established using cable or wireless media.



As you see, this is a simple network diagram which has server, router, switch and several other components which you will learn in the upcoming modules.



Now that we know nodes/Devices constitute a network, how do they communicate with each other??

“SOCKET PROGRAMMING”

Socket programming is a way of connecting two nodes on a network to communicate with each other.

It is a method and can be implemented in any language like C, C++ or Java etc but we generally prefer and follow C in this lab.

Before diving into the coding, lets understand a few terminologies that we use in programming.

In computer networks lab, what we tend to do is to establish a communication between two devices.

Of the two devices, we call one of them as client and the other one as server.

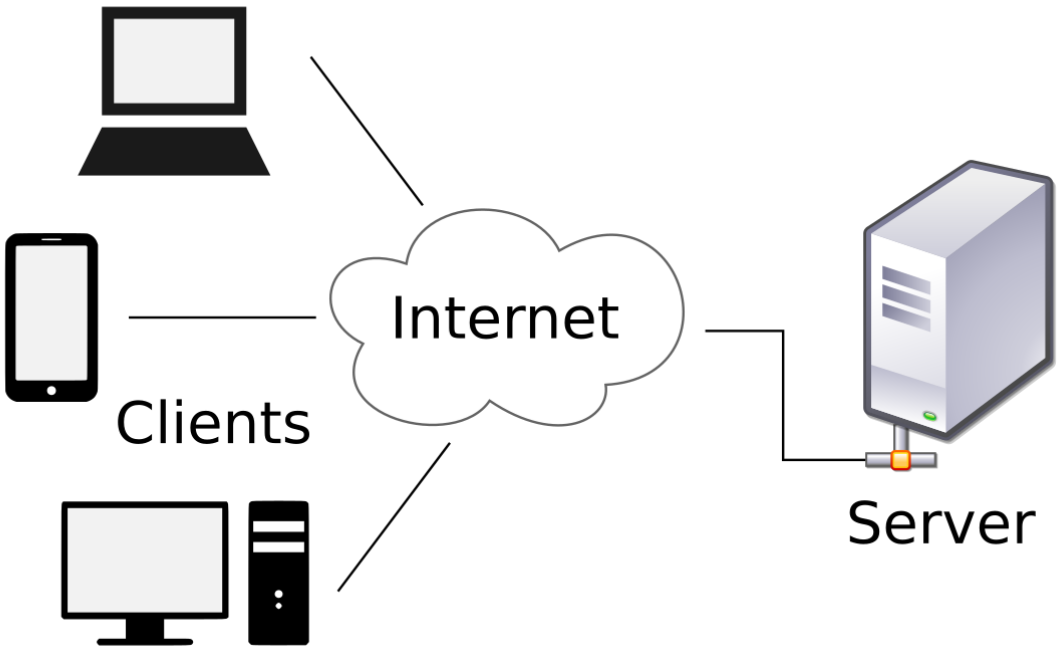
Why so??

There are many conventional reasons.

A **server** is a program that provides services to clients. It waits for client requests and responds to them.

A **client** is a program that requests services from a server. It initiates communication with the server and waits for responses.

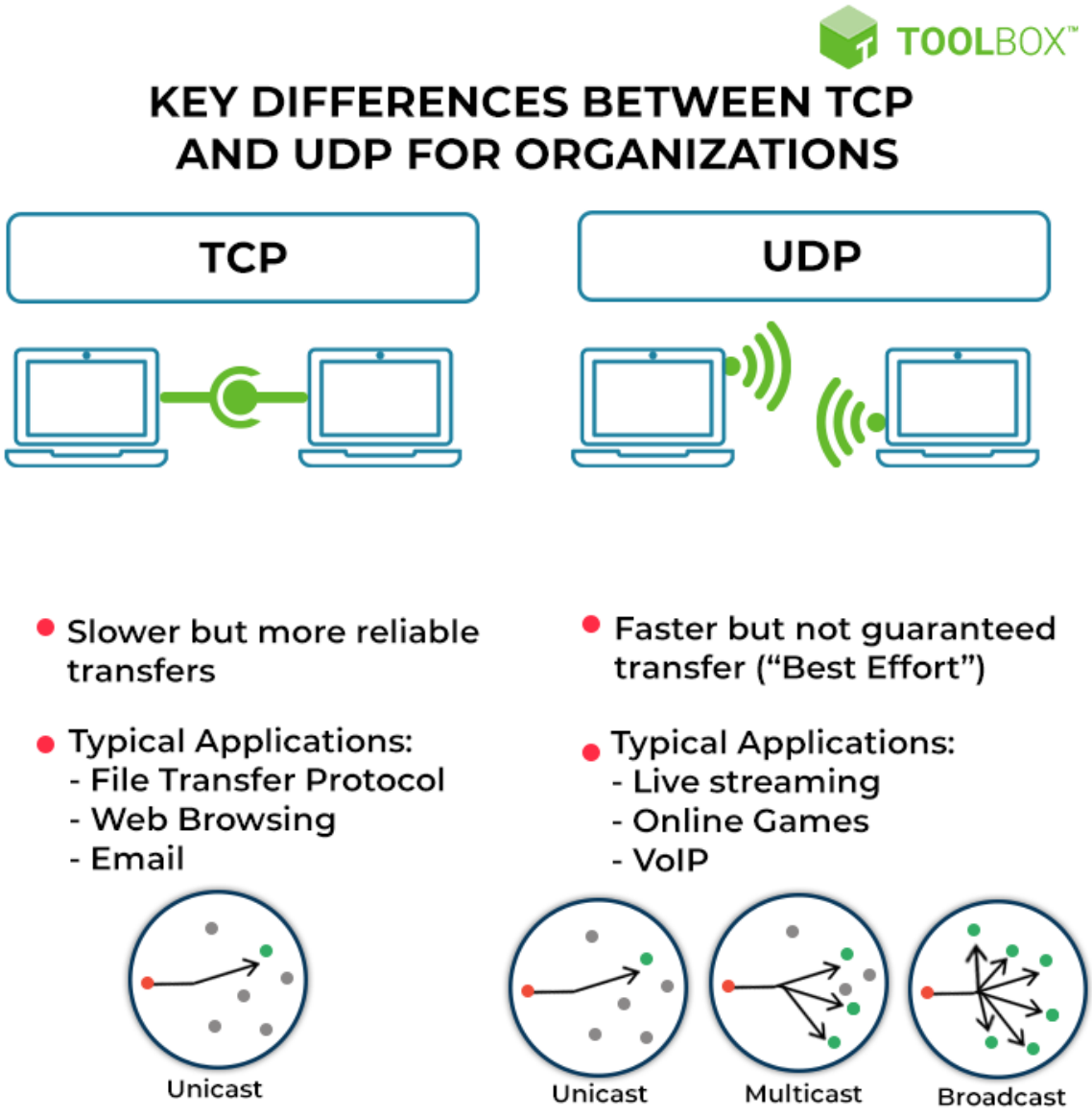
This may not be the case always. It is just a convention that we follow here.



Also there are two types of connections that we can establish between two nodes or devices.

1) TCP Connection
2) UDP Connection

You will learn a lot about these two type of connections in the further modules. For now, go through the diagram and you will get a picture of what the type of connection is.



To get a clearer understanding, go through this situation…

Imagine you want to order food from a restaurant. What do you do?? Well….If you are as lazy as me, then you will go for Swiggy or Zomato etc.

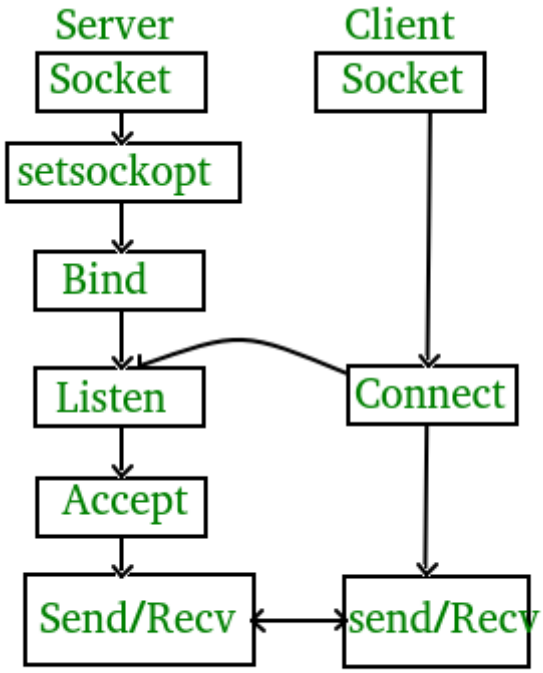
Just imagine, you (the client) are trying to order food from the restaurant (server) and for which you use Swiggy or Zomato (Socket programming).

For establishing the communication between you and the restaurant, you use Swiggy which helps in getting whatever tasty food you need from the restaurant.

This is the overall basic scenario that many of the real world situations are based.

Now that you know what a client, a server, a computer network and how a basic communication happens, lets dive into the coding part.

Let's try to establish a simple **TCP connection** between server and client.



This diagram at first will seem very vague.. but this is the overall steps that you will need to follow for establishing the connection.

"One interesting note": You will be writing two separate codes for client and the server and you will be compiling them individually in two individual terminals and you will be executing them parallelly.

Get ready to witness "Too Many logins" message in your terminal too many times ;)

The entire process can be broken down into following steps:

TCP Server –

1. using create(), Create TCP socket.
2. using bind(), Bind the socket to server address.
3. using listen(), put the server socket in a passive mode, where it waits for the client to approach the server to make a connection
4. using accept(), At this point, connection is established between client and server, and they are ready to transfer data.
5. Go back to Step 3.

TCP Client –

1. Create TCP socket.
2. connect newly created client socket to server.

You will have to learn the structure of the program for the client and the server.

The basic client and the server side program is given as script and has been put in the Dept site.

Refer to the department site for the client and server program.

Steps:

1) Open a terminal and create a file called “client.c” in one terminal of yours.
2) Open another new terminal and create a file called “server.c” in the same folder of your terminal.
3) Now compile both the codes individually with the following commands.

   Compilation –
   Server side:
   gcc server.c -o server
   ./server
   Client side:
   gcc client.c -o client
   ./client

4) Remember to run the server first and then the client side.
5) Get the output

You should get this output

Server side:

```
Socket successfully created..
Socket successfully binded..
Server listening..
server accept the client...
From client: hi
     To client : hello
From client: exit
     To client : exit
Server Exit...
```

Client side:

```
Socket successfully created..
connected to the server..
Enter the string : hi
From Server : hello
Enter the string : exit
From Server : exit
Client Exit...
```