Computer Networks and Internet Protocol
Prof. Soumya Kanti Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 07
Application Layer – III (Client Server, FTP)

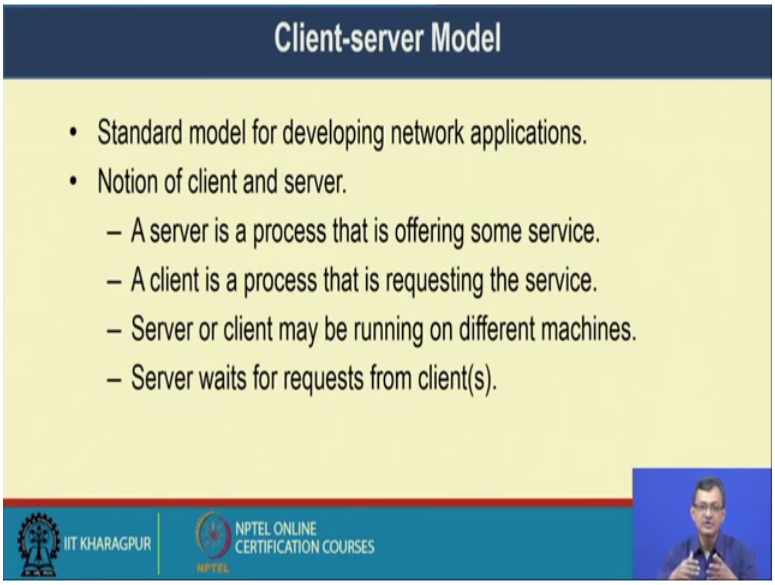
Hello, we will continue our discussion on Computer Networks and Internet Protocols. We were discussing on Application Layer protocol or say different protocols, which are prominent in the application layer and today's discussion will be primarily on FTP or, but before that we will have a quick overview of the client server system, right. Now this client server paradigm is I believe that well known to all of us, but it is for the sake of understanding, we will relook at the thing. Now this is a as we understand this client server paradigm is a predominant paradigm in our for application different application running over the internet, it helps us to application talking to each other across the network, right.

So, what is the basic philosophy? So, there is a server program and corresponding client program, right like we know that there should be if I am doing FTP, there should be a FTP server and corresponding FTP client similarly, if I am doing a say telnet. So, there should be a telnet server and there should be a telnet client and like this, right, the server and client can be on the same machine or in the different machine.

So, if it is a different machine, then the client server needs to know that the client needs to know where the server is and make a connection before establish a connection before the communication going on. So, what we are trying to at done at the basic at the underling level we have some applications which will work over the network.

And basically rely on this network typically TCPIP or SI or network models and the application can run over the this network, right. So, the later on in this course, we will look at some other things like web services service oriented architecture, but the predominant application layer processing or what we say application layer communication is will be done, we will be seeing the client server model.

(Refer Slide Time: 02:38)



Client-server Model

- Standard model for developing network applications.
- Notion of client and server.
  - A server is a process that is offering some service.
  - A client is a process that is requesting the service.
  - Server or client may be running on different machines.
  - Server waits for requests from client(s).

So, standard model for developing network application; so, as we are discussing and notion of client server a server is a process that is offering some service, right, as we normally know.

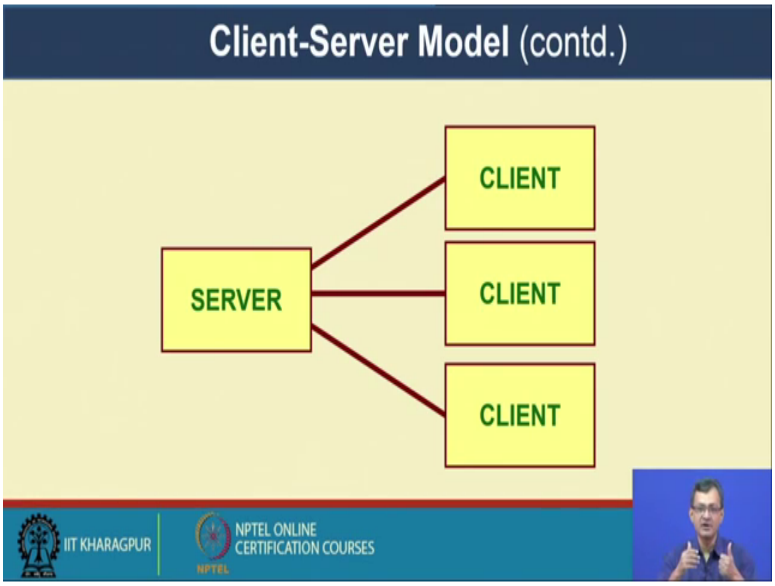
And a notion of a client is a process that is requesting for a service, right like if I have a print serverm a print client is requesting for the service in the for the printer, you many of you are accustomed with network level printer where over the network, we request for that service, even these days, we are using network where means a paradigm which is where you can connect or project something display something using a; using the underlining network, right. So, that is anything any such applications, which is giving service has to be this is typically known as the client server.

And the request requesting process is the client process right server or client may be running on the different machine or in the same and same machine, all right, if it is on the same machine or different machine, the way of handling the whole thing remains same, server waits for the request from the client. So, in other sense, if we look at server is always active waiting for the request from the client to happen like one of the very popularparadigm is our h t t p server, right, a any document you want to access over the internet, over the using our browser. So, what we see that h t t p colon slash, slash www say iitkgp dot is c dot in right.

So, there are two things that are the iitkgp server which is which is there in somewhere in IITK; Kharagpur network or somewhere in the internet we will respond back once the client this type of request is there. So, this browser my typical browser or your browser is acting as a h t t p client and the server machine is responding to that the server is always waiting for a kind to be request. So, it is waiting for the client to send their request and respond accordingly based on that the if the respond is successful and the format is correct it said other this is other part of the things, but it will respond to the nothing whereas, the.

So, I have h t t p server typically known as h t t p t or h t t p demon in terms of Linux or stuff like that and I have a h t t p client, which is h t t p i client or typically if it is http client, we this is manifested by our standard browsers web browsers.

(Refer Slide Time: 05:15)

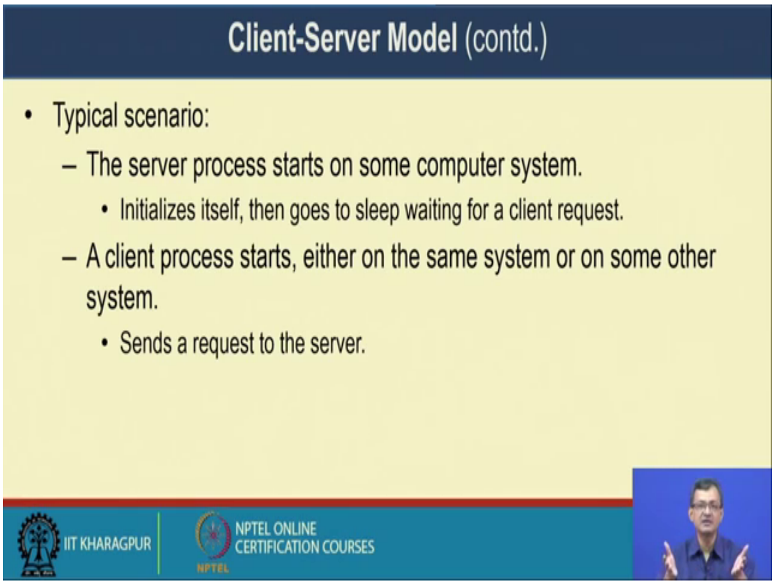


Client-Server Model (contd.)

So, what we see there is a server and there can be multiple clients. So, immediately two things pop up that whether the server will serve one client after another that is whether it is a iterative server one server sub seed second server, etcetera, etcetera, like that. So, it that can be iterative server or all the servers all the clients are served together. So, I have a concurrent server, right.

So, number of requests are being served together and the limit based on that the resource availability etcetera the number of servers can be served together right and this whether we will look at in couple of sites there, whether it is iterative or the concurrent based on the application requirement, right some resources where the server is handling maybe has to be done iterative way. So, the other things has to get or so, some of the most of the cases, it can be served concurrently like typically h t t p servers. So, that can be served concurrently.

(Refer Slide Time: 06:16)



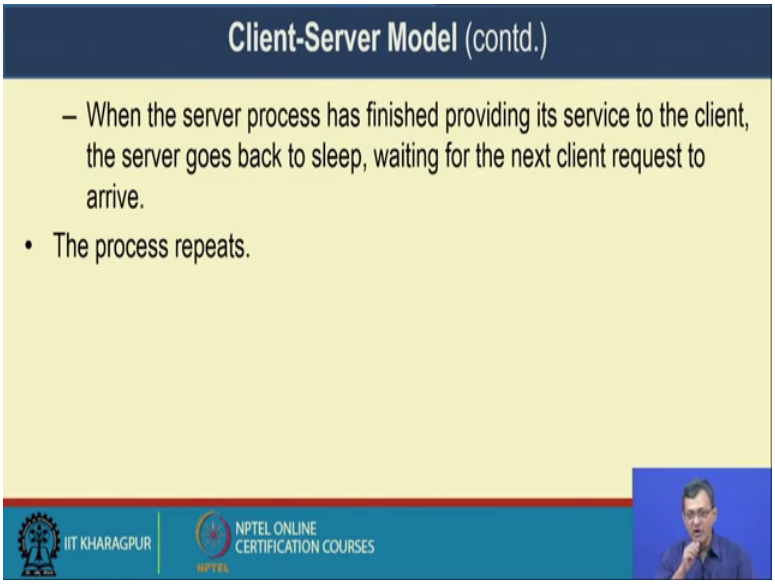
Client-Server Model (contd.)

- Typical scenario:
  - The server process starts on some computer system.
    - Initializes itself, then goes to sleep waiting for a client request.
  - A client process starts, either on the same system or on some other system.
    - Sends a request to the server.

So, typical scenario the server process starts on some computer system initialize itself and then goes to sleep waiting for the client to request, right. So, this is the thing a client process starts as an as the client needs it either on the same system or some other system right sends a request to the server.

So, this is the typical scenario and whenever whether whatever the client server paradigm things are there; that has to be the this sort of mechanism has to be there; there can be different way of handling some can have more than one connect is connection to be establish some single connection to be established that a protocol dependent, but nevertheless this has to be satisfied.

(Refer Slide Time: 07:00)



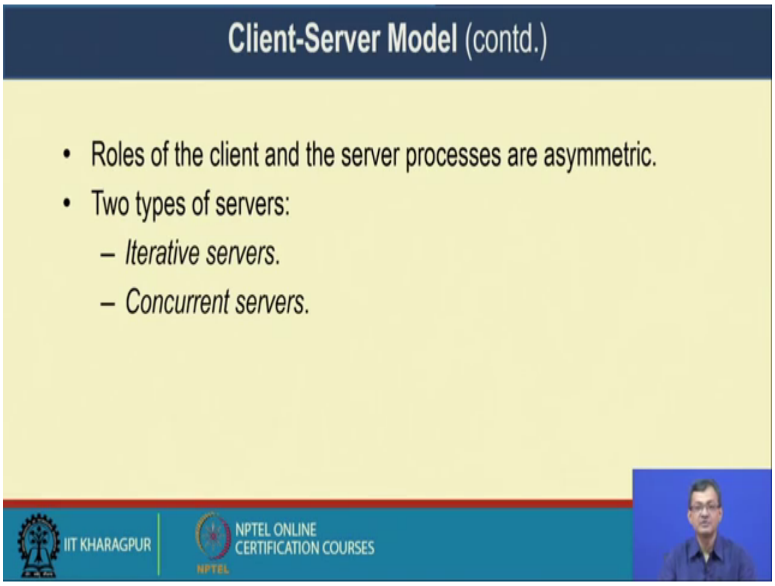
Client-Server Model (contd.)

- When the server process has finished providing its service to the client, the server goes back to sleep, waiting for the next client request to arrive.
- The process repeats.

When the server process finished providing its service to the client, the server goes back to sleep waiting for the next client request to arrive.

So, once it is the finished; it is the it will goes back to sleep the process repeats when the things are there, this is the very vanilla type of operation, but it describes the things how it works.

(Refer Slide Time: 07:23)

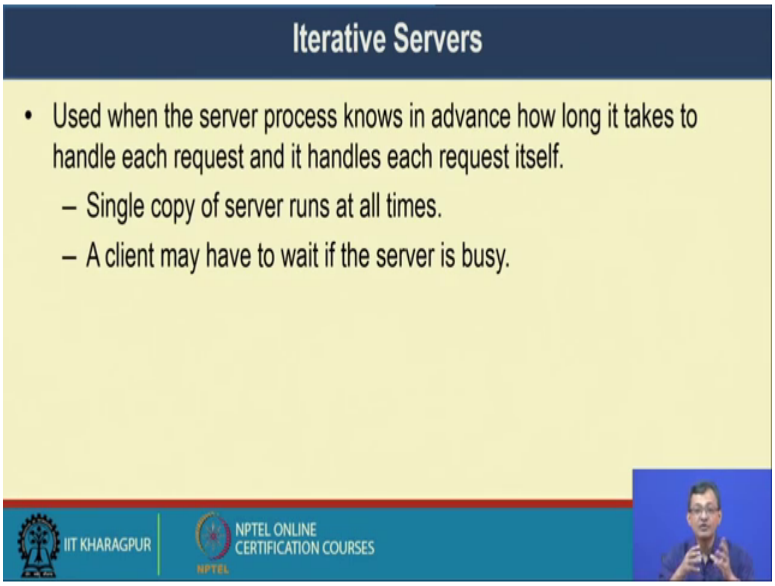


Client-Server Model (contd.)

- Roles of the client and the server processes are asymmetric.
- Two types of servers:
  - *Iterative servers.*
  - *Concurrent servers.*

So, as we are discussing the role of client and server processes are asymmetric, they are not there can there are may not be symmetric and two types of servers there, one is as we are discussing iterative server another category of server, what we called concurrent server which serves concurrently and iteratively one by one.

(Refer Slide Time: 07:44)



Iterative Servers

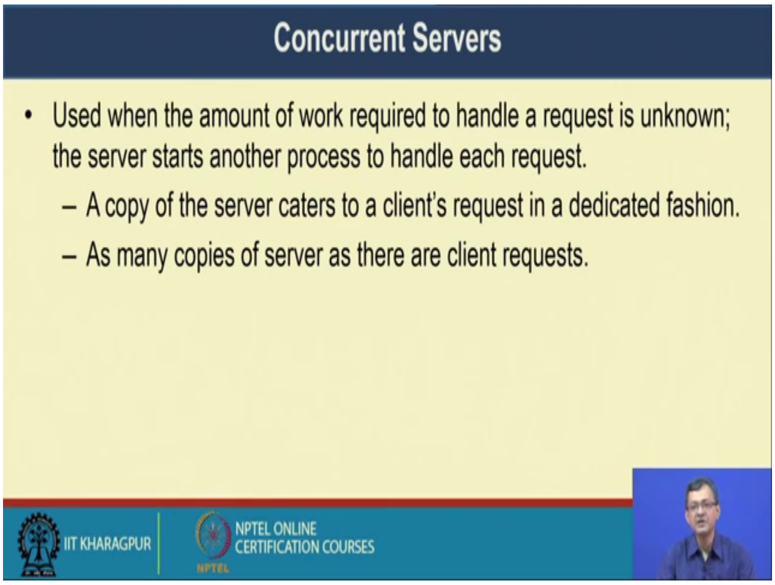
- Used when the server process knows in advance how long it takes to handle each request and it handles each request itself.
  - Single copy of server runs at all times.
  - A client may have to wait if the server is busy.

So, iterative server used when the server process knows in advance, how long it takes to handle each request and handle each request itself and type of things or more specifically when there is a requirement, which goes for an iterative things the resource allocation should be done one after another.

So, that all cannot bumped into the things like I have a some resource some say some sort of a resource to be reserved and type of thing and I cannot do concurrently maybe; I may have to do iteratively one by after one after another and in most of the cases we have some estimate that how much time it will take in working on it so, that I can have one step another. The single copy of the server runs all the time right and a client may have to wait if the server is busy right or in this case one server a one copy of the server or the server process is only one process.

So, it is running all the time it serves goes to the next, next, next, next. So, it is a iterative way of handling with it.

(Refer Slide Time: 08:52)



Concurrent Servers

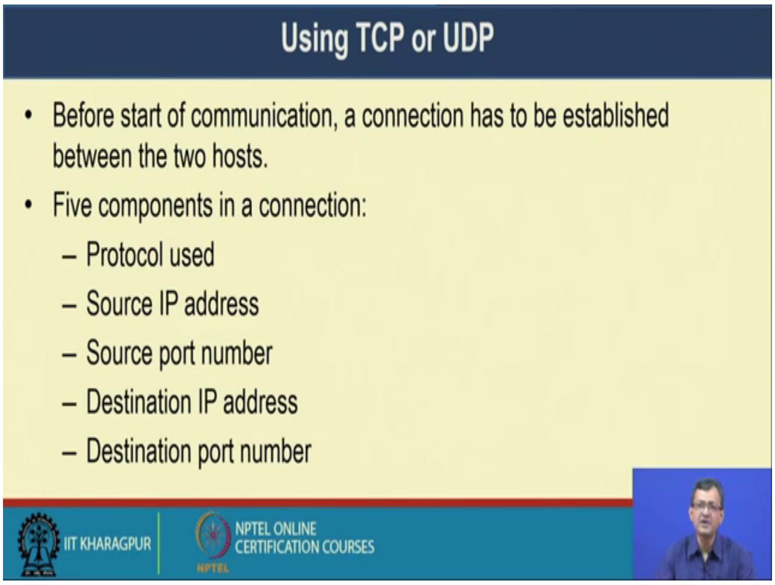
- Used when the amount of work required to handle a request is unknown; the server starts another process to handle each request.
  - A copy of the server caters to a client's request in a dedicated fashion.
  - As many copies of server as there are client requests.

Whereas, concurrent server were used when the amount of work required to handle a request is unknown right. So, if the concurrent server is required when the amount of work required to handle adequates is not known, right, the server starts another process to handle each request, right. So, the server starts other process or in other sense my requirement or my way of delivery is concurrent right like I have a say iitkgp website or my own website or something which I can serve concurrently.

So, there is no way. So, a copy of server caters to the client requests in a dedicated fashion. So, this is important, right; so, a copy of the server; so, what it does? It is those who are accustomed with some sort of a ways programming a there is a concept of forking, right; so, forking a child process. So, some sort of working a child process the server fork a child process, which goes on serving the request of that particular client and then it again comes back and listen to the start listening to the client request, right.

So, it makes a self copy of the things which goes on serving the things as many of the copies of the server, there can be many client requests. So, the as much as based on the resource availability, the amount of the number of copies will be going on serving the client request.

(Refer Slide Time: 10:23)



Using TCP or UDP

- Before start of communication, a connection has to be established between the two hosts.
- Five components in a connection:
  - Protocol used
  - Source IP address
  - Source port number
  - Destination IP address
  - Destination port number

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

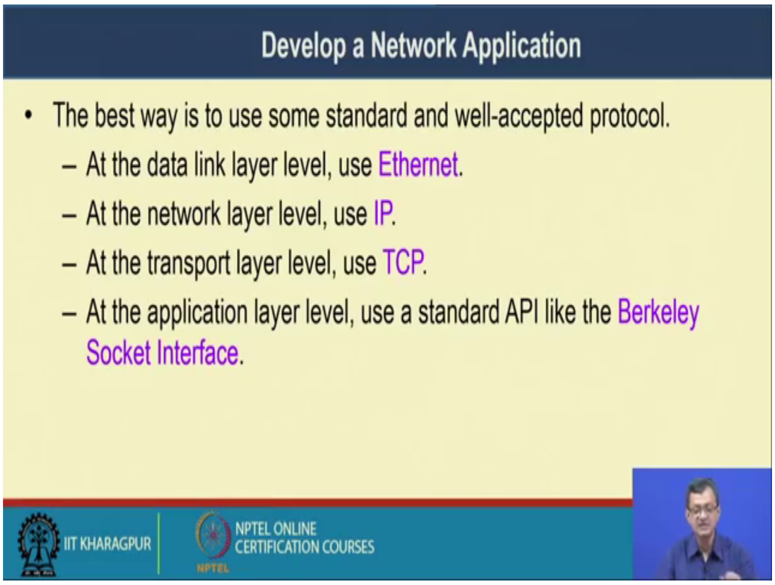
So, whether TCP or UDP, again, what is the requirement of the application before start of communication the connection has to be established between the host, right, it can be a connection oriented service like FTP type of things or it can be a connectionless service right either UDP like say DNA, sub DNA type of things where or DNAs resolution that can be a UDP type of services based on the what the application needs, right. So, if you see, what are we required to make a connection establish we require 5 things, right that IP of the server port of the server where the server is listening, right.

So, what so, if in our terminology; what we in network terminology what we see that what we see that to identify a system we require IP address to identify a process in the system we require a port. So, IP plus port combinely defines the process of the thing as the server process. So, I require the IP of the server and IP of the I port number of the server process where it is listening on the other hand, I require a client IP of the client right where the communication is other part of the communication and the port of the client, where to which it is the client process is communicating. So, this fourth thing apart from that we require that underlining protocol.

So, in most of our cases what we work on is the internet protocol is our predominant protocol. So, it is most of the cases in the IP protocol which you, but nevertheless it defines that thing this combination. Now if it is the same machine the IP will be same. So, server IP client IP will be same, but nevertheless the port number will not be same even the protocol is also same. So, this port will distinguish that two connection that is why as saying that if I open up a h t t p server, right I am requesting for I am multiple browser in my windows open.

And I am requesting say iitkgp page one and some other things say IIT, Delhi something IIT, Chennai, IIT, Madras and type of things and, but it is not like that request of these we will go to the thing, right. So, they are in they are these five triple distinguishes stuff distinguishes every connection or defines every connections now. So, what we require to develop a network application. So, at the data link layer we require internet at the network layer we require IP.

(Refer Slide Time: 13:05)



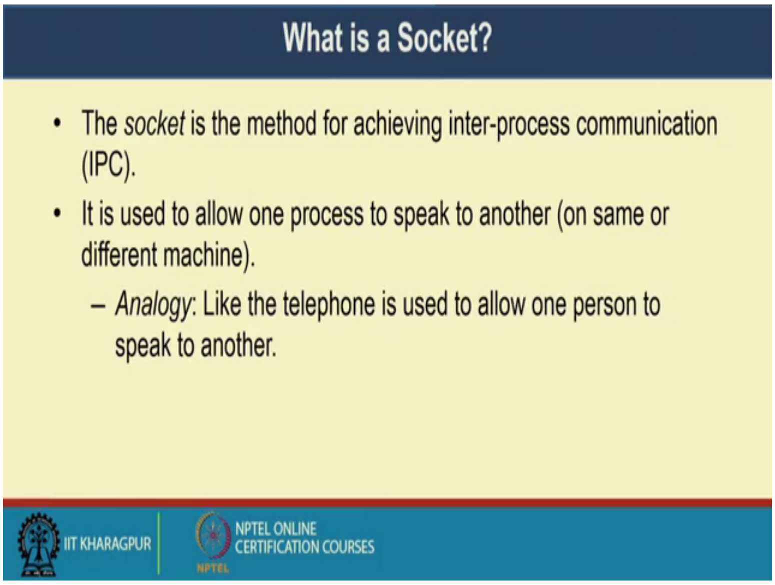
Develop a Network Application

- The best way is to use some standard and well-accepted protocol.
  - At the data link layer level, use Ethernet.
  - At the network layer level, use IP.
  - At the transport layer level, use TCP.
  - At the application layer level, use a standard API like the Berkeley Socket Interface.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

At the transport layer use of TCP or UDP and there is a concept of Berkeley socket, we will do some socket level programming at some part of this course show you that how things works, but nevertheless there is a Berkeley socket interface.

(Refer Slide Time: 13:24)



What is a Socket?

- The *socket* is the method for achieving inter-process communication (IPC).
- It is used to allow one process to speak to another (on same or different machine).
  - *Analogy*: Like the telephone is used to allow one person to speak to another.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, socket is a, what we can say it is a methodology or a mechanism by which inter process communication or IPC works, right. So, it us a mechanism by which this inter process communication works it is used to allow one process to speak to another on same or different machine, right. So, what we say that I establish a socket between these two processes and this IPC or Inter Process Communication works over this socket or those who are or all of you have some working experience on C or type of languages. So, what you see that if you open a file in a C language what we require a file ID to communicate right rest of the things right here also I get a socket id. So, I establish a comics communication like using this over five triple to be satisfied.

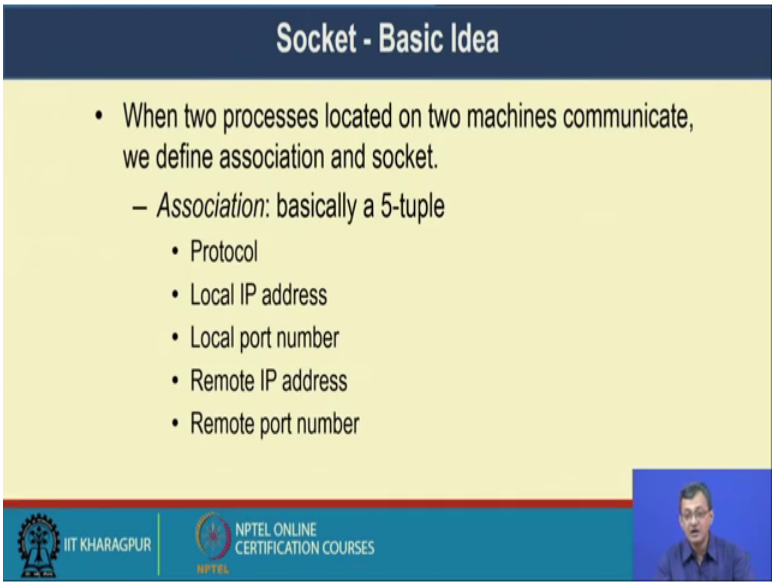
And once that is done I have that socket id which allows me to transfer traffic a transfer the data over the things, right. So, it is used to allow the process to speak to one another same or different machine some analogy like telephone is used to allow one person to another in that that is a very straightforward analogy that I it allows to this, but socket gives me a mechanism or method to have this IPC or inter process communication to happen, right. So, in order to establish a socket; so, what we do I the socket the socket mechanism to be supported by the system in most of the Linux system, they are supported. So, what you require you require a socket to be opened at the client end. So, if I serve end. So, server opens up a what we say some sort of a half socket, right.

So, it its own IP own port and the protocol right and wait on that port that is client to get the request client on its other end opens up another half socket like its IP port and the protocol and it knows that client server IP right that has to be known right. I if I want to do a FTP to a remote machine I need to know the IP or the u r l or the name of the things like you know that iiitk www dot iitkgp ac dot in or unless you know this name then you [con]- find the iitkgp page.

Now this name will not be applicable for any communication over the network right. So, network layer understands only the IP address. So, that has to be resolved by the d n s. So, d n s returns a ip. So, in other sense I should know the IP address in some way or other to of the destination. So, the client sent say some sort of a connection request to the server server on things, if it is find the format, etcetera everything protocol wise matching then establishes is 5 topping.

And that establishes a socket between these two client server client and server using this socket id the rest of the communication goes on like data transfer and others, etcetera, both way, etcetera.

(Refer Slide Time: 16:49)

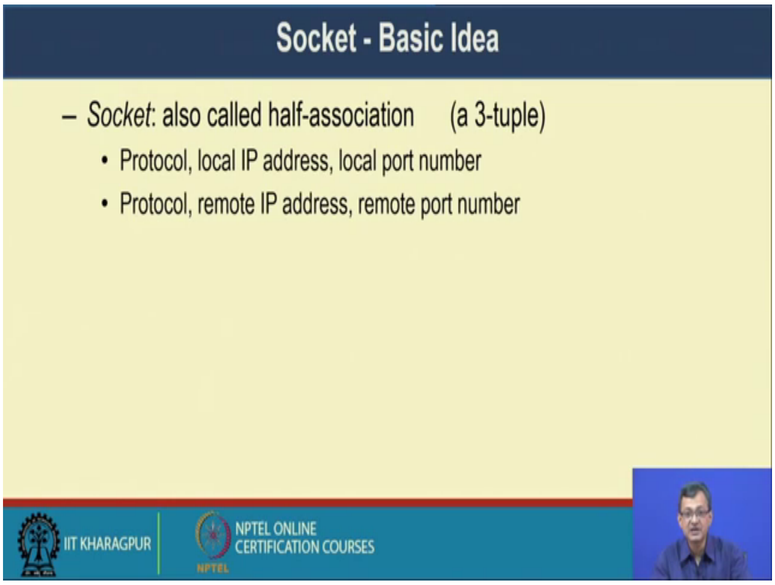


Socket - Basic Idea

- When two processes located on two machines communicate, we define association and socket.
  - *Association*: basically a 5-tuple
    - Protocol
    - Local IP address
    - Local port number
    - Remote IP address
    - Remote port number

So, when two process located on the same machine to communicate we defined a association defined a association and a socket. So, these are the as we have discussed; so which will have a protocol, IP local IP, or I say client IP client port server, IP server port.

(Refer Slide Time: 17:06)



Socket - Basic Idea

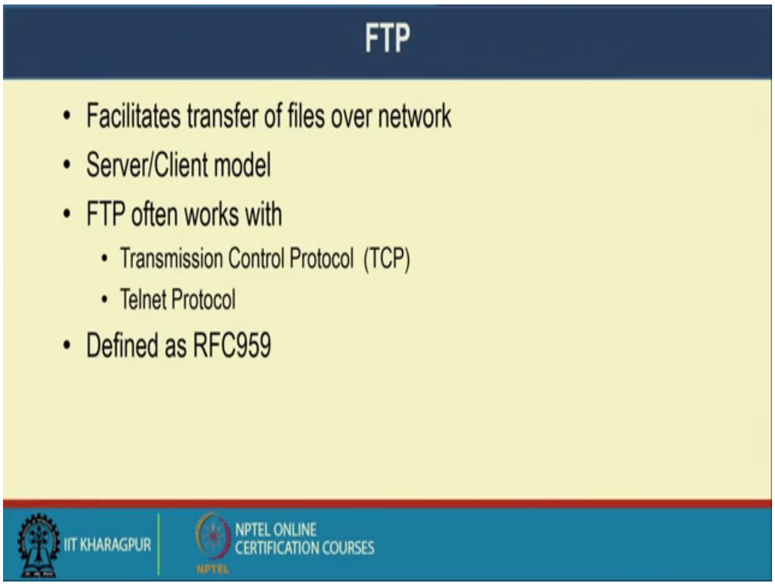
- *Socket*: also called half-association (a 3-tuple)
  - Protocol, local IP address, local port number
  - Protocol, remote IP address, remote port number

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, a typically also called a half association as we are discussing that at the things. So, protocol local IP local port or protocol client IP client port input or protocol remote there is server IP server port and it wants that communication path is there established and the goes on. So, that is in some sense defines that how this client server protocol works.

And all of our discussion what you are working on like most of the protocols are client server based as of now right; so, one of the protocol very predominant protocol is the FTP we are mostly used to that.So, it has a.

(Refer Slide Time: 17:48).



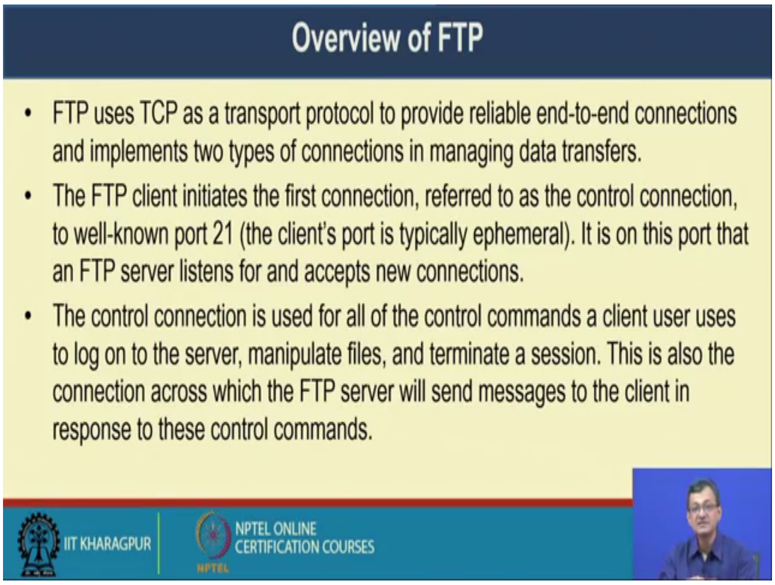
FTP

- Facilitates transfer of files over network
- Server/Client model
- FTP often works with
  - Transmission Control Protocol (TCP)
  - Telnet Protocol
- Defined as RFC959

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, what it facilitates transfer of files over network, it is a client server model often works with TCP or connection oriented reliable service and also telnet protocol the definition or the spec of the FTP is defined in RFC959 those who are interested can look into those RFCs; RFC.

(Refer Slide Time: 18:12)



Overview of FTP

- FTP uses TCP as a transport protocol to provide reliable end-to-end connections and implements two types of connections in managing data transfers.
- The FTP client initiates the first connection, referred to as the control connection, to well-known port 21 (the client's port is typically ephemeral). It is on this port that an FTP server listens for and accepts new connections.
- The control connection is used for all of the control commands a client user uses to log on to the server, manipulate files, and terminate a session. This is also the connection across which the FTP server will send messages to the client in response to these control commands.

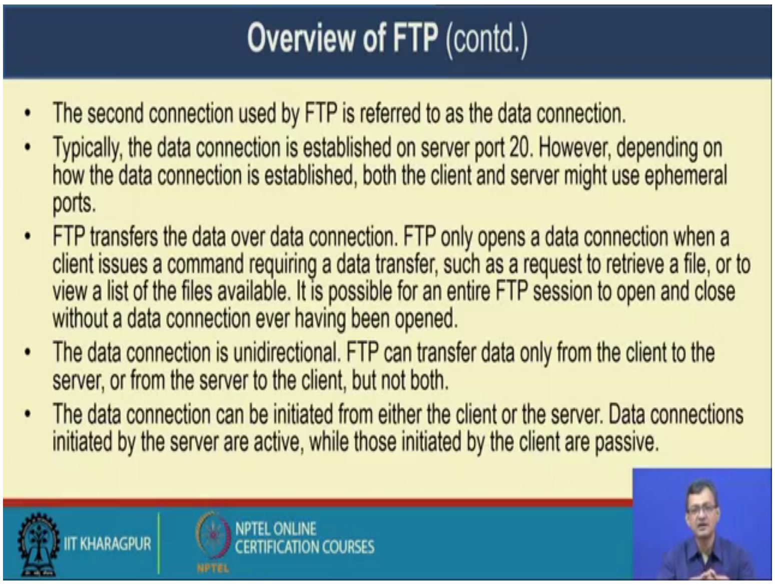
So, a FTP uses TCP at the transport layer. So, it is the application layer down the layer down layer is the transport is the TCP to provide reliable end to end connections and implements two type of connection managing the data transfer.

So, first of all it uses TCP layer in the transport moon and then it implements to connection. So, one for control and one for data will come to that the TCP client initiates the first connection referred to as control connection right and well known port 21. So, that initial connection is the port, it is it is on this port that the FTP server listens for the accepts of the new connection so; that means, FTP server if my linex time I say FTP d it once it initialization or in other our socket terms makes a half socket and listens to that port 21 is any FTP requests there sort of thing and the FTP client coming out from any port, it does not matter and then heats to that particular server at port 21, right. So, it is the default port of port 21.

So, when you give FTP by default, it knows that it goes to port 21 if you want to change the port 21 to some other port at the server end, if you have changed then the that particular port to be the connection request should come to that particular port suppose the port is instead of 21 it is say something 8 8 8 8 or something right. So, it has to be equal on 8 8 8 8.

So, that it says that you go for that service at that particular port, but anyway without going to that complicacies we see that its port 21 is the default port the control correction is used for all control commands a client server uses to log on to the server manipulates file terminates, ssn, etcetera, right. This is also connection across which FTP server will send messages to the client in response to this control command, etcetera. So, those are those are also defined in the things, we will see that some of the popular control commands data commands etcetera at the end of this lecture.

(Refer Slide Time: 20:20)



Overview of FTP (contd.)

- The second connection used by FTP is referred to as the data connection.
- Typically, the data connection is established on server port 20. However, depending on how the data connection is established, both the client and server might use ephemeral ports.
- FTP transfers the data over data connection. FTP only opens a data connection when a client issues a command requiring a data transfer, such as a request to retrieve a file, or to view a list of the files available. It is possible for an entire FTP session to open and close without a data connection ever having been opened.
- The data connection is unidirectional. FTP can transfer data only from the client to the server, or from the server to the client, but not both.
- The data connection can be initiated from either the client or the server. Data connections initiated by the server are active, while those initiated by the client are passive.

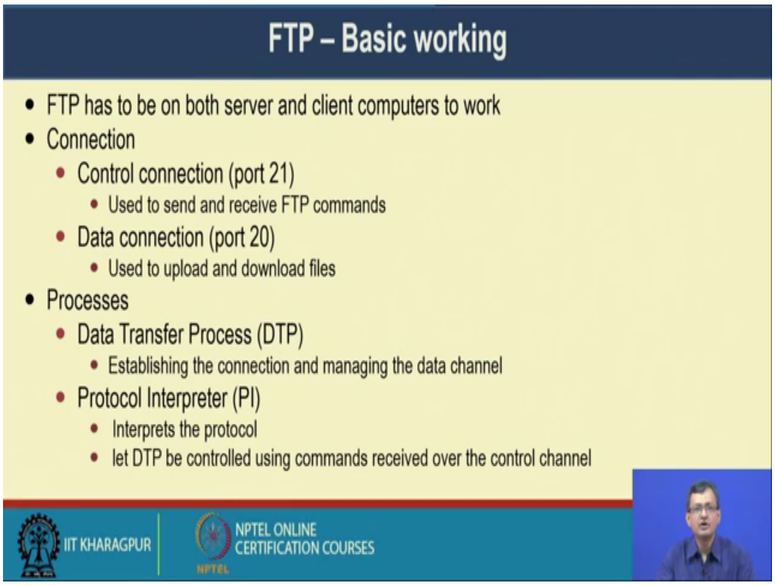
The second connection of the FTP is referred to the data connection. So, typically the data connection is established that port 20.

So, 21 is the control port 20 is the data connection; however, depending on how the how the data connection is established both the client server might be might use a ports. So, that can it may happen that 21 is the control panel, but the data client server can agree upon to use a the some other ephemeral port for that things FTP transfers data over the data connection FTP only opens a data connection, when the client issues a command requiring a data transfer, right such as request to retrieve a file or list the list of file, etcetera, different kinds of we will again we will has a has mentioning.

I will see will I will show you some standard commands anyway those are things available in any book or any over the network, but nevertheless we will popular commands data, data transfer commands also, we will show you the data connection is unilateral file can transfer data only from client to server or from server to client or not port.

So, that is one way either this or this. So, it is not the both can cannot go simultaneously, right, the data connection can be initiated either by the client or the server the data connection initiated by the server are active, while those initiated by the client are called passive, all right. So, it can be initiated by the both the things and the connection established by the server are called active connections or the initiated by the client or passive.

(Refer Slide Time: 21:59)



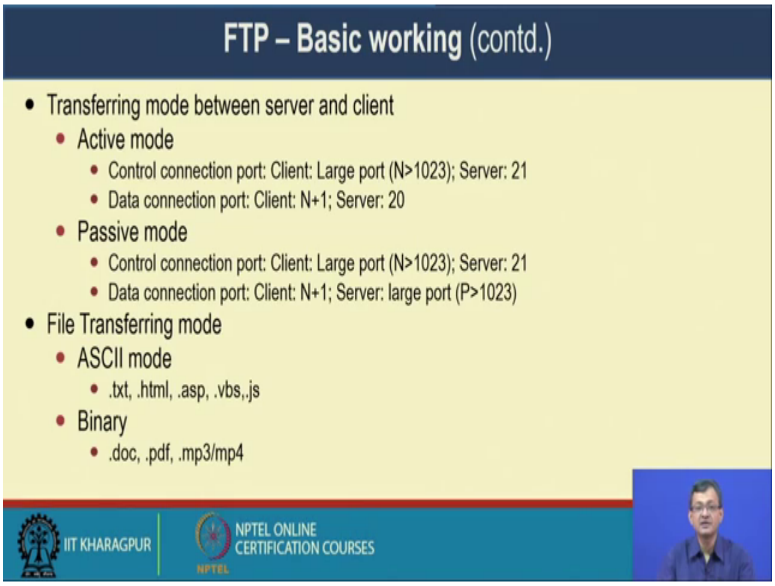
FTP – Basic working

- FTP has to be on both server and client computers to work
- Connection
  - Control connection (port 21)
    - Used to send and receive FTP commands
  - Data connection (port 20)
    - Used to upload and download files
- Processes
  - Data Transfer Process (DTP)
    - Establishing the connection and managing the data channel
  - Protocol Interpreter (PI)
    - Interprets the protocol
    - let DTP be controlled using commands received over the control channel

So, if we look at the basic operations though; so, it is a client server model. So, it is based on what; we have discussed at the initial part of this lecture. So, connection is control connection is typically port 21 uses to send and receive FTP commands data connection is typically port twenty used to upload.

And download files, right process the data transfer process two type of things are processes are there one is the data transfer process or let us say for it DTP establishes the connection and managing the data channel another what we say protocol interpreter or pi, right. So, interprets the protocol let us let DDP be controlled using command received from the control channel. So, the one is protocol interpreter to interprets the protocol and the DTPA data transfer protocol uses the command to transfer the files.

(Refer Slide Time: 22:59)



FTP – Basic working (contd.)

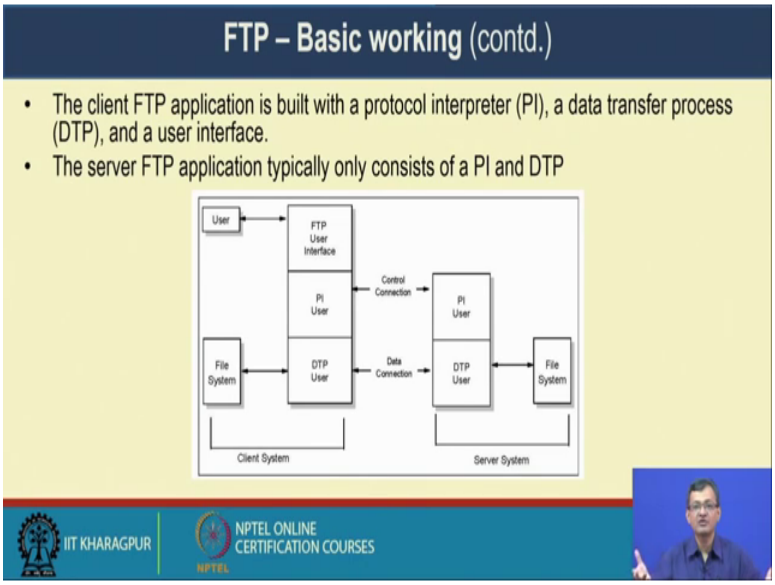
- Transferring mode between server and client
  - Active mode
    - Control connection port: Client: Large port (N>1023); Server: 21
    - Data connection port: Client: N+1; Server: 20
  - Passive mode
    - Control connection port: Client: Large port (N>1023); Server: 21
    - Data connection port: Client: N+1; Server: large port (P>1023)
- File Transferring mode
  - ASCII mode
    - .txt, .html, .asp, .vbs,.js
  - Binary
    - .doc, .pdf, .mp3/mp4

So, a to continue with the basic operation.

So, to as we had discussed there are two mode; one is active mode control connection port client, there are large port numbers server at port 21 data communication port is at the client is N plus 1 server is port 20, right. So, this is the in active mode in the passive mode control connection port. So, client a large port number should be more than 1023 and server is port 21 data connection client is again N plus 1 server, large port number of any greater than 1023; that it means that not those it is a airport or restricted port.

So, file transfer mode can be either ASCII that is txt, html and etcetera, etcetera or it can be binary like doc, pdf, some media file so on. So, you need to define I can define that the type of things whether it is ASCII or binary or bin can be defined and can be transferred.

(Refer Slide Time: 24:00)



FTP – Basic working (contd.)

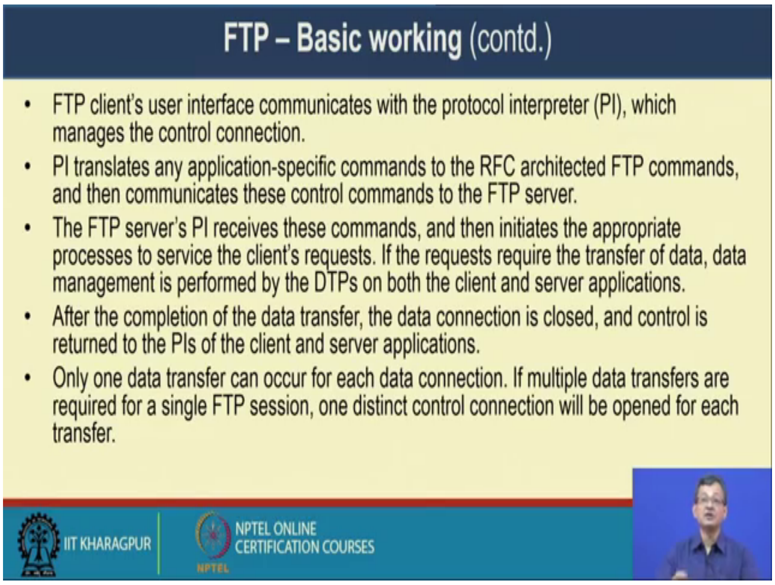
- The client FTP application is built with a protocol interpreter (PI), a data transfer process (DTP), and a user interface.
- The server FTP application typically only consists of a PI and DTP

So, the same thing if we try to look at that the client FTP is build with a protocol interpreter a data transfer process and a user interface. So, if you if you have your FTP client like there are very there are several open source client.

So, you it has a user interface and underlining it has a protocol interpreter and a data transfer process or PI and TTF, right a dtp data transfer process. So, what it does at the server end also there is a protocol interpreter and DTP.

So, it one is for the control connection one is the data connection and we have two file system right at the client side; once file system and also server sign to file system. So, either file data can be transfer from here to here or other way, but we have to file system. So, that the client end, client end has a user interface to do that; all right, there are command line things also those who are a custom can do FTP connection and do the command set the basic at the at the command line.

(Refer Slide Time: 25:11)



FTP – Basic working (contd.)

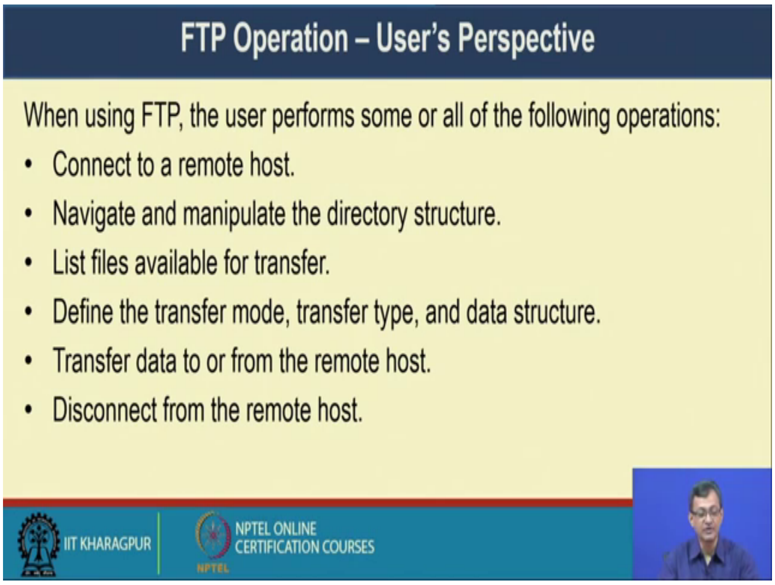
- FTP client's user interface communicates with the protocol interpreter (PI), which manages the control connection.
- PI translates any application-specific commands to the RFC architected FTP commands, and then communicates these control commands to the FTP server.
- The FTP server's PI receives these commands, and then initiates the appropriate processes to service the client's requests. If the requests require the transfer of data, data management is performed by the DTPs on both the client and server applications.
- After the completion of the data transfer, the data connection is closed, and control is returned to the PIs of the client and server applications.
- Only one data transfer can occur for each data connection. If multiple data transfers are required for a single FTP session, one distinct control connection will be opened for each transfer.

So, FTP clients user interface communicates with the protocol interpreter which manages the control connection the same thing PI translate any application specific command to the R RFC architect FTP command.

So, that there is a application specific command. So, it should be RFC architect FTP command it has to be there otherwise the server side will not understand or the same thing true for server to client side. So, processes command and then communicates these control commands to the server end the FTP server PI received this command and then initiates the appropriate processes to receive to service the client request right. If the request require the transfer of data if there is a data transferring involved the data management is performed by this DTPs, right both the end and both the client server applications, right.

So, first the PI is PI will take care of that control connection established and if there is a data transfer involve then the DTPs will come into play after the completion of the data transfer the data connection is closed control is returned to the PIs of the client and server applications and only one data transfer can occur at each connection if multiple data transfer are required on a single FTP session one distinct control connection will be opened for each transfer, right, either, it can be managed at the upper level that going on doing that at the or user interface or the FTP client program takes care all the individual data connection has to be established right for the each data transfer.

(Refer Slide Time: 26:55)



FTP Operation – User's Perspective

When using FTP, the user performs some or all of the following operations:

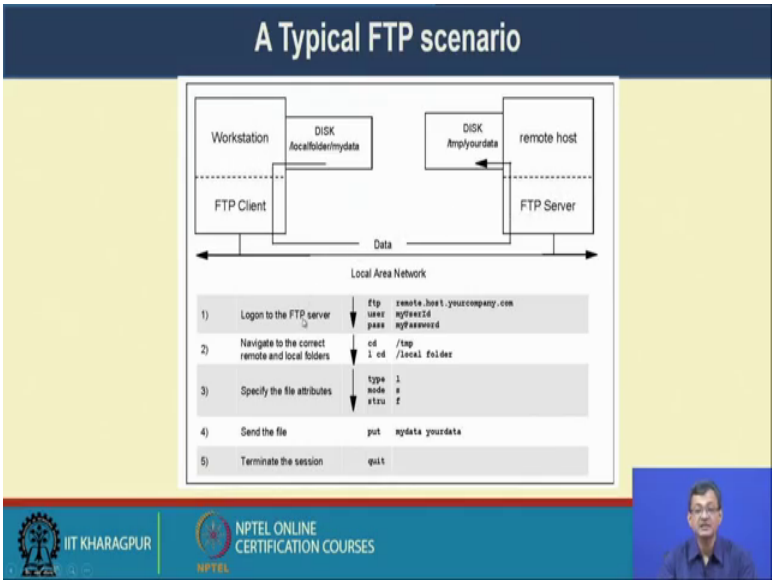
- Connect to a remote host.
- Navigate and manipulate the directory structure.
- List files available for transfer.
- Define the transfer mode, transfer type, and data structure.
- Transfer data to or from the remote host.
- Disconnect from the remote host.

IIT KHARAGPUR
NPTEL ONLINE
CERTIFICATION COURSES

So, if you look at the user perspective. So, what is the user perspective connect to the remote host navigate and manipulate the directory structure right. I can go to the directory structure of the remote host or there if there is a permission is there, I can manipulate least files available for transfer define the transfer mode transfer type and data structure, right. Once I want to transfer data to and from the remote host.

So, either it can be from to client to the server or server to the client disconnect the remote host whenever the whenever the work is over. So, that they can be in a typical way of looking at it.

(Refer Slide Time: 27:39)



A Typical FTP scenario

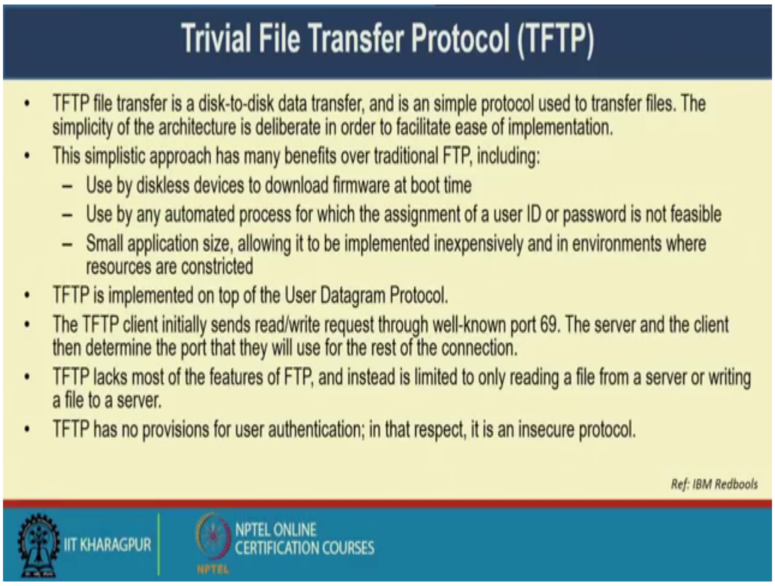
1)	Logon to the FTP server	ftp user pass	remote.host.yourcompany.com myUserId myPassword
2)	Navigate to the correct remote and local folders	cd l cd	/tmp /local folder
3)	Specify the file attributes	type mode stru	i s f
4)	Send the file	put	mydata yourdata
5)	Terminate the session	quit	

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, this is a typical scenario say workstation FTP client equates to a FTP server. So, it log on to the FTP server navigate to the correct remote directory right to where the data is there specify the file type that what sort of file is there saying; that means, whether you will require binary s key and type of things send the file put that is the data if I want to retrieve file then gate process they terminate the session by quitting it.

So, this is a typical way of looking at it when I put some data from this local host to this remote host or where it acts as a FTP client and this is a FTP server.

(Refer Slide Time: 28:24)



Trivial File Transfer Protocol (TFTP)

- TFTP file transfer is a disk-to-disk data transfer, and is an simple protocol used to transfer files. The simplicity of the architecture is deliberate in order to facilitate ease of implementation.
- This simplistic approach has many benefits over traditional FTP, including:
  - Use by diskless devices to download firmware at boot time
  - Use by any automated process for which the assignment of a user ID or password is not feasible
  - Small application size, allowing it to be implemented inexpensively and in environments where resources are constricted
- TFTP is implemented on top of the User Datagram Protocol.
- The TFTP client initially sends read/write request through well-known port 69. The server and the client then determine the port that they will use for the rest of the connection.
- TFTP lacks most of the features of FTP, and instead is limited to only reading a file from a server or writing a file to a server.
- TFTP has no provisions for user authentication; in that respect, it is an insecure protocol.

Ref: IBM Redbools

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, there is another concept called TFTP. So, it is I thought that it would be good to know. So, it is a for known as trivial FTP protocol. So, it is a low payload FTP protocol, right. So, typically widely used in say you want to upload a configuration file in a router or network device and type of things, where much resource are not level this TFTP plays a important role, this simplistic approach has many benefits over traditional FTP as because it is a very simple vennila approach to the things used by displace devices to download firmware at boot time, right.

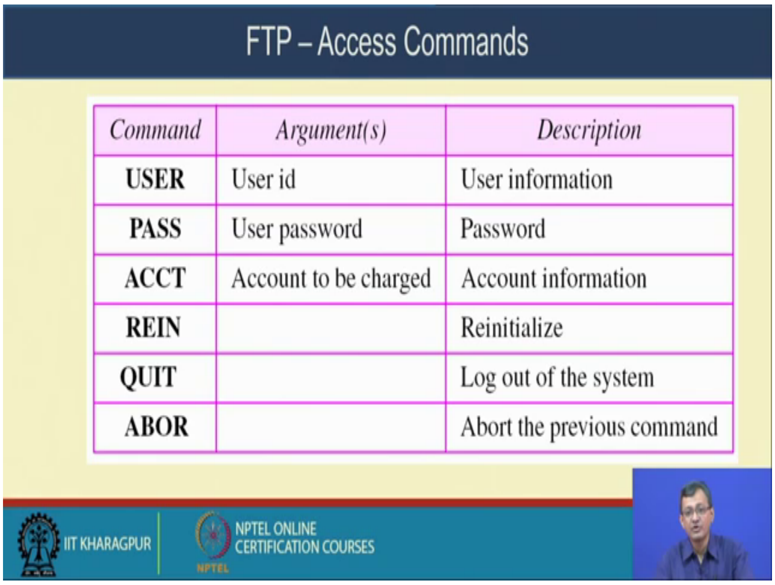
So, displays devices used by any automated process for v is the assignment of a user idea password is not feasible; that means, there are that is that much resource is not there model application size allowing it to be implemented in various devices right various low resource devices and in environmental resource there is resource requirement is not is much layer or lace or constrained resource requirement TFTP is implemented on the top of the UDP user data.

This is interesting right FTP is typically over TCP; TFTP is typically over UDP that it is not reliable transfer the TFTP client, initially send retried request through the well known port 69. So, it is not that our port 21 like that the server and the client then determine the port that will use for the rest of the connection.

So, initially that request 69 and then agreed upon a thing TFTP lacks most of the features of FTP and instead, it is limited only reading a file from a server or writing a file to a server.

So, it is more of a updating or reading or updating the server TFTP has no provision for user authentication in that respect it is in secure protocol, but the the places or the, but the situation where we use, we it plays the important role like up updating a firmware and those things where I can ensure security by different mechanisms, right, I that compromising that channel may be more difficult, because there may not be external connection like that another things.

(Refer Slide Time: 30:46)



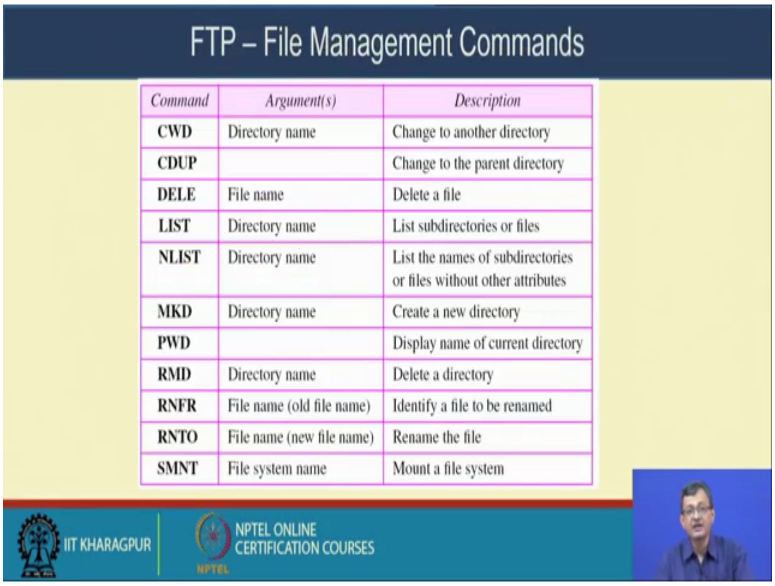
FTP – Access Commands

Command	*Argument(s)*	*Description*
USER	User id	User information
PASS	User password	Password
ACCT	Account to be charged	Account information
REIN		Reinitialize
QUIT		Log out of the system
ABOR		Abort the previous command

Then we have some quickly go to some commands these are available in the books these are these are taken from different sources including books.

And what I thought that it will be these are the typical command it will be good. So, user pass act, etcetera like user information password account information re installation logout aborting previous commands are some of the access commands.

(Refer Slide Time: 31:09)

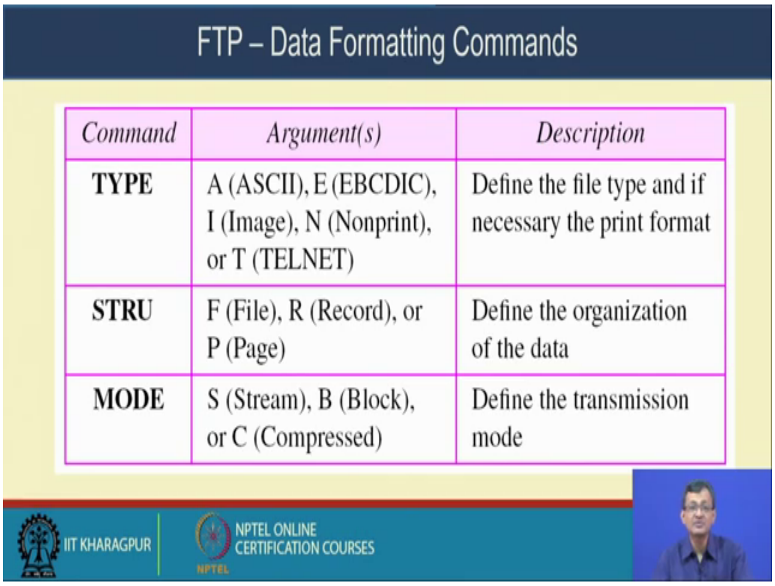


FTP – File Management Commands

Command	*Argument(s)*	*Description*
CWD	Directory name	Change to another directory
CDUP		Change to the parent directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
NLIST	Directory name	List the names of subdirectories or files without other attributes
MKD	Directory name	Create a new directory
PWD		Display name of current directory
RMD	Directory name	Delete a directory
RNFR	File name (old file name)	Identify a file to be renamed
RNTO	File name (new file name)	Rename the file
SMNT	File system name	Mount a file system

Some of the file management command like CWD change to another directory or delete to delete a file or MKD making it directory and so and so forth. So, these are some of the commands which are file management command.

(Refer Slide Time: 31:36)



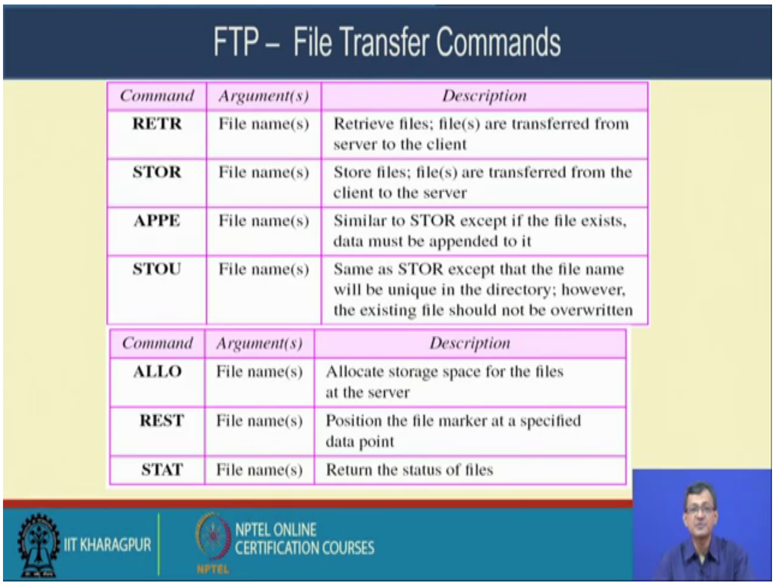
FTP – Data Formatting Commands

Command	*Argument(s)*	*Description*
TYPE	A (ASCII), E (EBCDIC), I (Image), N (Nonprint), or T (TELNET)	Define the file type and if necessary the print format
STRU	F (File), R (Record), or P (Page)	Define the organization of the data
MODE	S (Stream), B (Block), or C (Compressed)	Define the transmission mode

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

These are some of the data formatting commanded as we are discussing type to define whether it is ASCII or FD or what sort of file is there it structure; whether it is a file or record or page more whether it is stream block or compress mode and type of things. So, these are all definitions, you can define which are related to the data formatting.

(Refer Slide Time: 31:48)



FTP – File Transfer Commands

Command	*Argument(s)*	*Description*
RETR	File name(s)	Retrieve files; file(s) are transferred from server to the client
STOR	File name(s)	Store files; file(s) are transferred from the client to the server
APPE	File name(s)	Similar to STOR except if the file exists, data must be appended to it
STOU	File name(s)	Same as STOR except that the file name will be unique in the directory; however, the existing file should not be overwritten

Command	*Argument(s)*	*Description*
ALLO	File name(s)	Allocate storage space for the files at the server
REST	File name(s)	Position the file marker at a specified data point
STAT	File name(s)	Return the status of files

And there are file transfer command like RETR; retrieve files, STOR store files and so and so forth, right.

So, there are several sets of transfer file there is a command called stat to return the studies status of that set of files. So, with this what we see a in this particular today's lecture, we primarily discusses that how what is the basic philosophy of client server, we just introduced there socket program we will in some of the subsequent lectures or some of the session, we will try to will show you somehow; this programming can work and how we can write your own network label programs into using this socket.

So, that we will; so we discussed about FTP, one of the very predominant application layer protocol used for primarily for transferring data uses two ports one for control and the data port and also another variant of FTP, which is a low payload FTP or TFTP, which is used for several firmware updates and other configuration updates in devices with constrained resources. So, with this let us talk let us stop up a today's discussion.

Thank you.