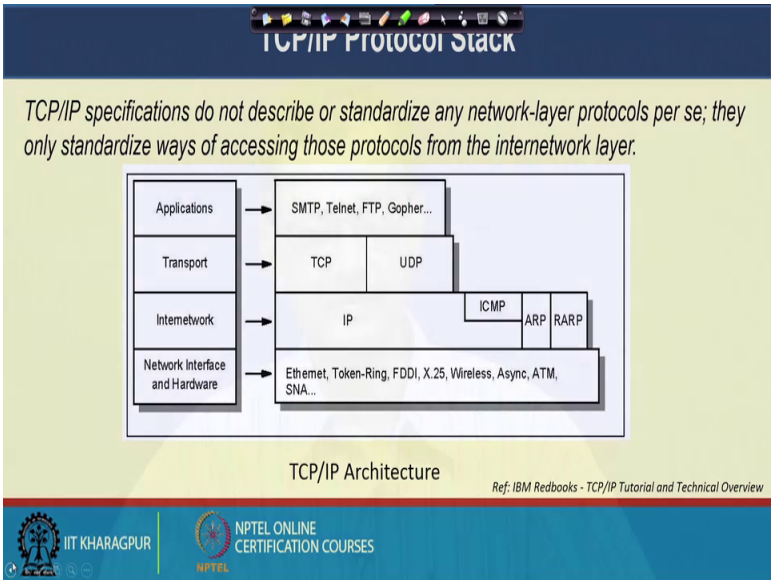
Computer Networks and Internet Protocol
Prof. Soumya Kanti Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 05
Application Layer - I

Hi. So, we will be discussing today on application layer of our TCP IP protocol or network protocol part say. So, what are the so, as we have discussed in our earlier lectures, that we will be initially we were discussing about the overall protocol stack. Now, we will start discussing layer by layer and their basic properties etcetera, and what are the different features of those layers. So, this application layer as we all understand plays the more most vital role as far as the inducer of the clients are concerned, right more concerned about the application which is running over the things, right not the underlining thing.

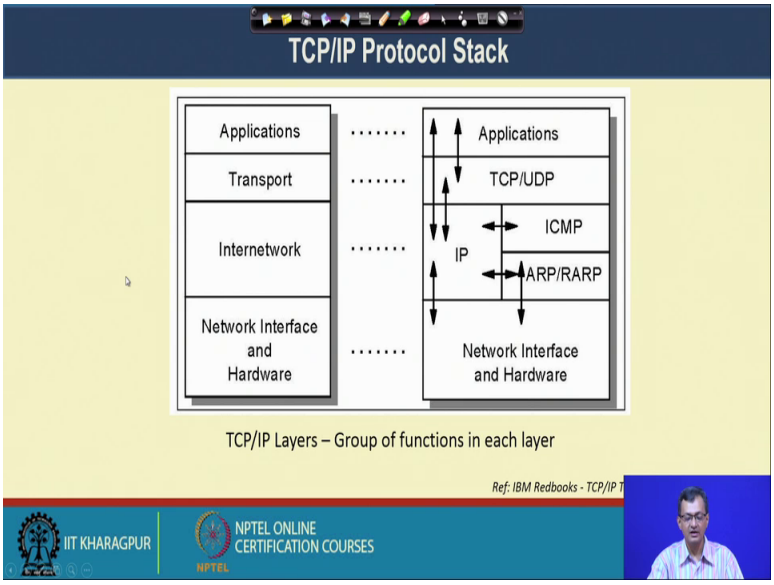
So, when we open a web page or transfer a file open the mail server or download mail or look at the mail server. So, we are these are the different applications which are at the front of us, right which the end user or is bothered or the client is bothered. Whereas, where as there are a lot of under underlining things goes on down the layer which are equally important we will go through the things. So, application layer has a direct connection or manifestation to our, this end user perspective.

(Refer Slide Time: 01:34)



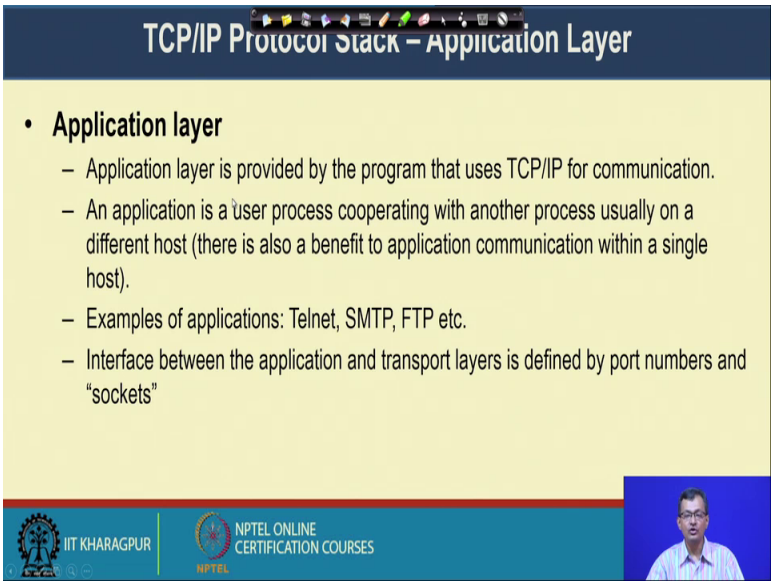
So, if we look at our typical protocol stack as we have discussed, that it has a application transport internetworking or layer tree. And this network interfaces an hardware as repeatedly, we are telling that there are there can be data link layer and the physical layer which constitute this. So, there and there is a underlining physical layer underlining there. So, if you look at the applications there the probing predominant things are like Telnet, FTP, SMTP, HTTP and DNS and so and so forth. So, these are the predominant functionalities. We will be for next couple of lectures we will be concentrating on these application layers.

(Refer Slide Time: 02:21)



So, again if we if you recollect what I have seen that these application layer protocols, primarily talks with the next layer of the transport layer which provides either a connection oriented service or connection less service which in turns talks to the down the layer like layer 3 IP layer and so on so forth. There can be application layer which directly talks with that IP layer and like that. So, these application layer can primarily talks with these transport layer or in some cases in some of the applications it can talk with the directly to the IP layer.

(Refer Slide Time: 03:00)



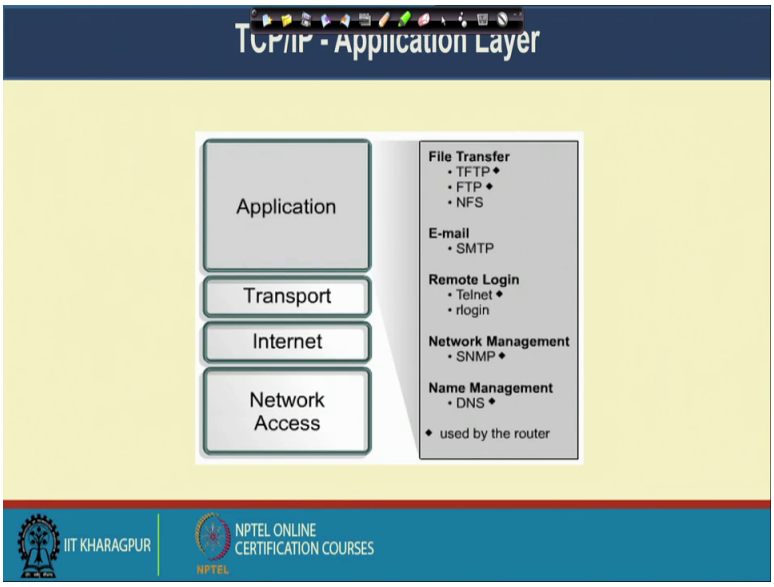
So, again what are the basic philosophy that application layer is provided by the program that uses TCP IP communication. So, it is the, what we say that programming quote unquote programming interface to this whole networking paradigm. An application layer is a user process cooperating with another process usually on the different host. Typically, we call this as client server things. So, when I run an application it is a client to some application at the server so, at the other end of the network.

So, though the client server do not say that it should be other internal network it can be the same network itself, but nevertheless I have a application server client which connects to the server like when we download www, or link to www iitkgp ac dot in, then what we do? We basically connect to that iitkgp web server and my browser at my end acts as a client, client to the things which is connect to the server.

So, this HTTP client to HTTP server then there can be FTP client to FTP server and so and so forth. So, it is a so, user process cooperate with another process. So, is that in turn can have other type of things. So, there are popular example like Telnet SMTP SNMP, FTP HTTP type protocol DNS and so on and so forth. Interface between the application and transport layer is defined by port numbers, right like how do I identify a system? By a IP address how do I identify a process in the system by a IP plus a port number which is which is a mostly as a transport layer phenomena.

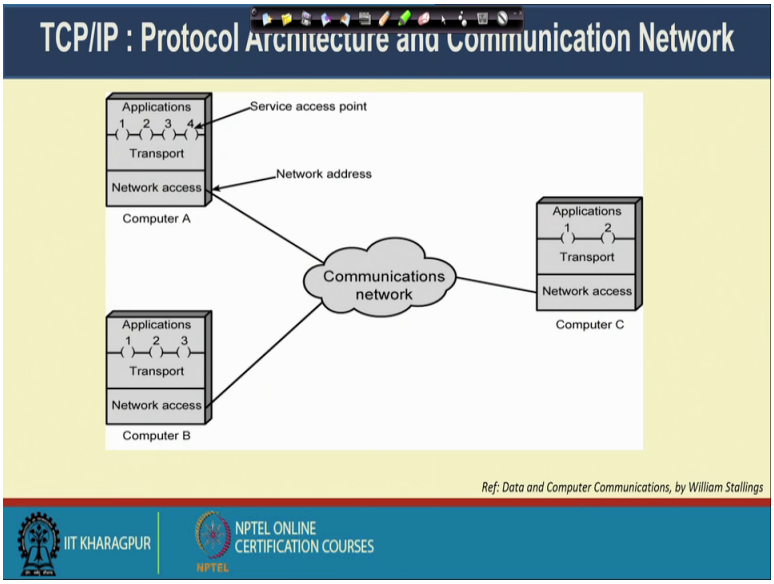
So, that identifies a process in the system. And there is there are a concept called sockets. We will be discussing on those sockets a little bit of socket programming, also we will be looking into, but this is what we say socket. So, it is a socket which is establish a socket interface with establish between the source and destination. And popularly what we use the term like socket programming and so and so forth, which allows me to communicate between each other.

(Refer Slide Time: 05:11)



Now, same thing this if application layer there is a variety of things some of the file transfer label things, some are email type of thing that use SNM SMTP, remote login Telnet a login network management, name there are name management like DNS, and used by routers. There are different applications which are at the this at different level.

(Refer Slide Time: 05:38)



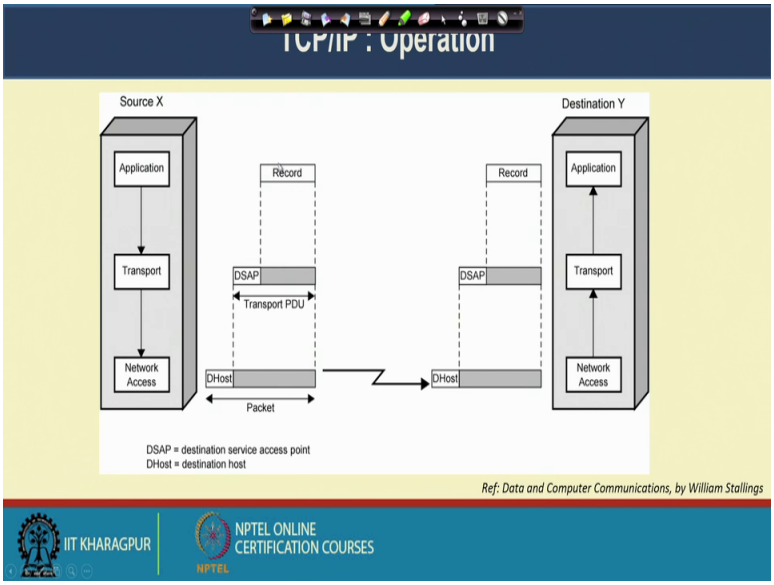
So, if we have a more holistic mix, where there are applications different applications. So, there are different service points of the things. So, this transport layer defines that, along with that we have a network access layer. So, IP plus the process allows me to run that application. So, that means, I may have a server which acts as a say FTP server, a SSI server or a Telnet server or different type of server. So, where the different things are define at different port.

There are popular ports like what we know that Telnet like port 23, FTP is port 21, rather there are 2 ports data and control port 21, 22 then HTTP port 8 and so and so forth these are popular port, but you can define your own port where the server is running client can connect through any other port, right. Similarly, so, there may be computer 1 computer 2 computer 3 and there are there can be several applications.

So, if you look at the at the top level view so, these different applications are talking to each other, right. So, that that actually what gives us that realization of how a process can communicate to process, or a application talks to each other, and that has a with the with the underlining network. So, the beauty of the things that this, the underlining network is not exposed to the client or the server, right. When I access a through a when you do HTTP www, iitkgp ac dot in. So, you are basically bothered about the page to be described like. So, neither the how this protocols stag or intermediate routers come into

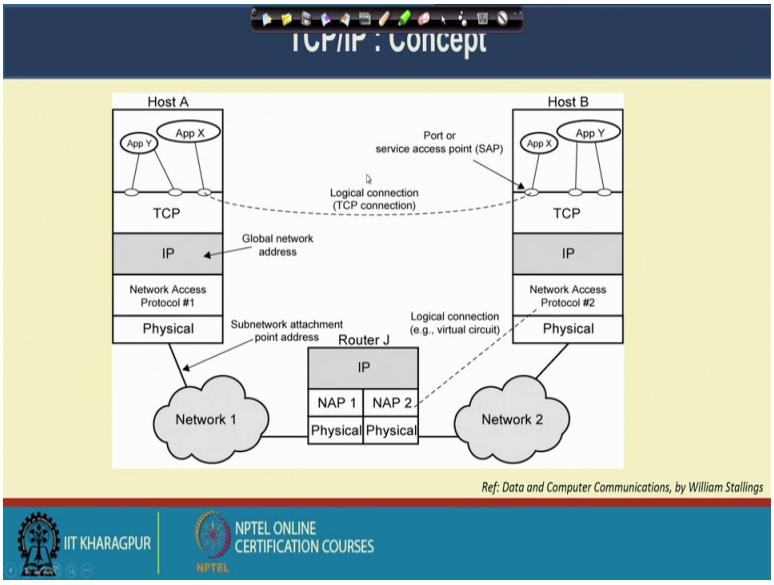
play that we are not looking at. So, what we look at is the intermediate a communication networking is in place, right.

(Refer Slide Time: 07:27)



So, similarly so if you as we discuss so, if you look at so, I have application transport and network access. So, to say that means, one defines the port and IP, and then whatever the data is there, it is being a payload to this next layer; that is for the transport video and it being becomes a payload this whole thing becomes a payload to this network access layer. And at the destination it gets deciphered.

(Refer Slide Time: 07:54)

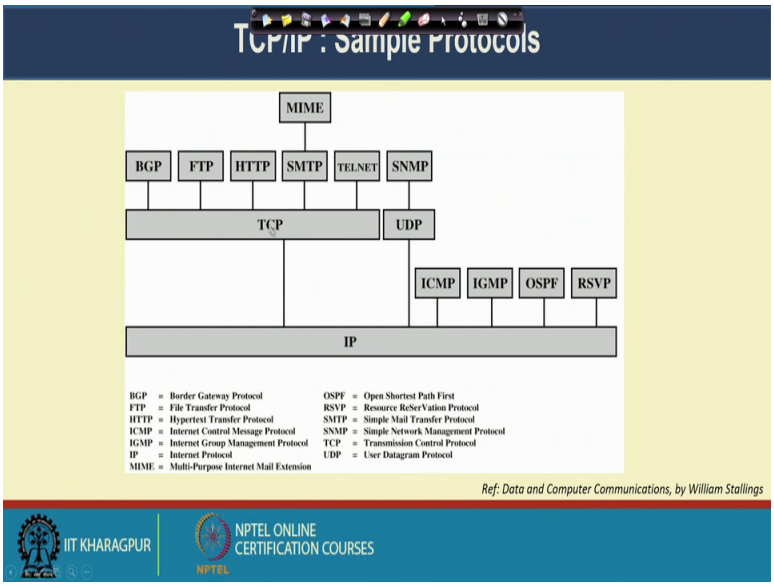


Or look at some another viewer if you look at. So, I have different applications. So, it is logically connected to different applications like as the p TCP is the predominant protocol that is which is a connection oriented protocol. So, it has a it finds a logical connection between the things. We as we discussed that that it is this whole thing runs on IP IP is again a connectionless best effort protocol. It does not guarantee that the packet will be delivered or not.

So, there should be some mechanisms. Here at the TCP end which will allow this logical connectivity, and what we say reliable connectivity on the over this unreliable layer right. So, we will discuss when we go into those lecture series, that how it is feasible to do that and how in respective of that underlining layer giving some services I can have a upper layer services on the things. Now down the line I can have different network access level protocol by which the network is access. And it goes on through the physical layer and the type of thing so, it goes on routed.

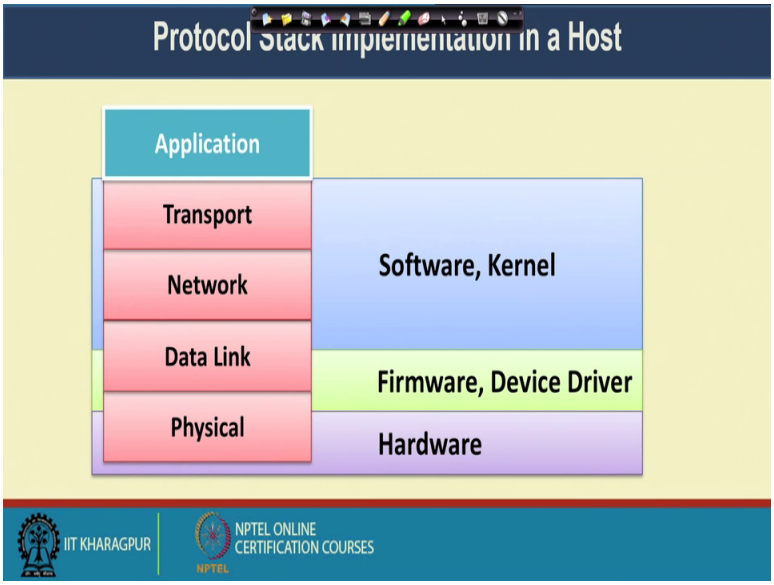
So, there can be n number of router in between, again we will be discussing those things when we go at different layer that how these routings are made possible. Nevertheless, this application X talks to the application x, right or if we little bit look on if the, this is a application X client talks with the application X server, right. Or application Y client while talks to the application server right two applications are talking to communicating with each other.

(Refer Slide Time: 09:30)



And I can have different type of things there are predominant applications which are the TCP. There are few applications which are UDP. Like, SNMP is one of that for the network management. There are several companion protocol for the ICMP, IGMP, OSPF, RSVP and so and so forth that the IP somewhere in between transport and IP. And so, there is a bunch of protocols which are they are in the within this protocol stack.

(Refer Slide Time: 10:02)



So, if we look at so if these are the layers. So, applications and then we have something which is controlled by the software and kernel. This data link some part and the physical is primarily controlled by this firmware devices or device drivers and hardware at the physical layer; that means, you require a network interface card and network interface card to have a physical connectivity, like when you put ARJ 45 cable into your laptop or desktop.

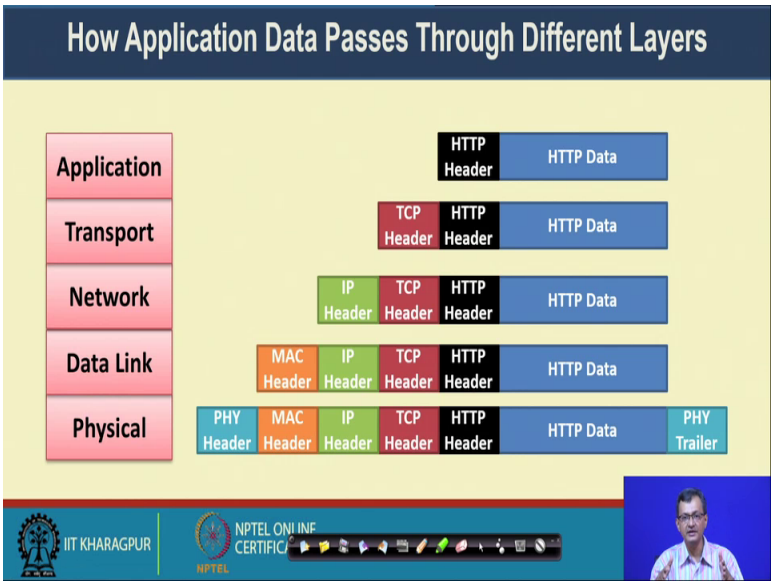
So, you there should be nic card which takes care. You will have a re you do require another interface card like wireless interface to work on those looking at those that there is also a hardware, over that I require a farmer and device driver which runs the things like if I have a nic net inter card there should be a particular device driver to work on if you have wireless or Wi-Fi interface.

So, there should be a device driver support from the operating system to that, right. Over that primarily it is handle by the machine and by software and kernel of the particular ways, and the device diverse to takes care of this network and protocol; they network and

transport layer and over that the application runs, right. So, if you look at if a when we are running FTP or HTTP. So, underlining we are using some of the things which are defined at the net at the stack level transport and IP layer which primarily if you see that if you in a normal windows pc or even Linux level or Unix system. What we do we basically this define the TCPIP property, right.

So, we provide those information to look at and there are some of the things are taken care by the OS or the kernel itself. Say, when a pack a when a when a client is going out it gets a port address to go out of the serve out of that particular interface. So, this gives a overview of the stack. And if we look at again some again that stack view

(Refer Slide Time: 12:15)

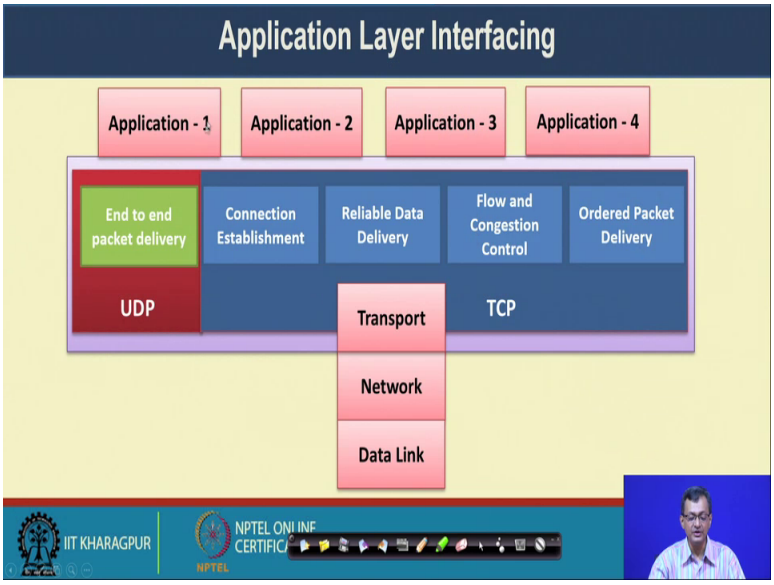


So, I have say there is a HTTP data. So, HTTP data along with the HTTP header which creates the application it becomes a payload for the transport layer, it becomes a payload for IP layer adds the IP header and it becomes a payload for this data link layer and finally, it goes to the physical layer and where the things are being transmitted, alright. So, this way it goes on and it goes it gets unpacked or extracted at that different level if there is a router it gets extracted up to the network layer. If it is a other end system, it gets extracted up to the application layer right. So, it gets unfolded as a as far this as the device level whatever the support is there.

So, in other sense what we what we try to see that it supports interoperable in a bigger sense right. So, what we say that I do not care about what intermediate router is there. So

now, I am following the protocol right it goes on hop to router it the overall routing business to finding the best path between the source, and destination is not primarily dependent on the system level things right. So, these are the things we look at it.

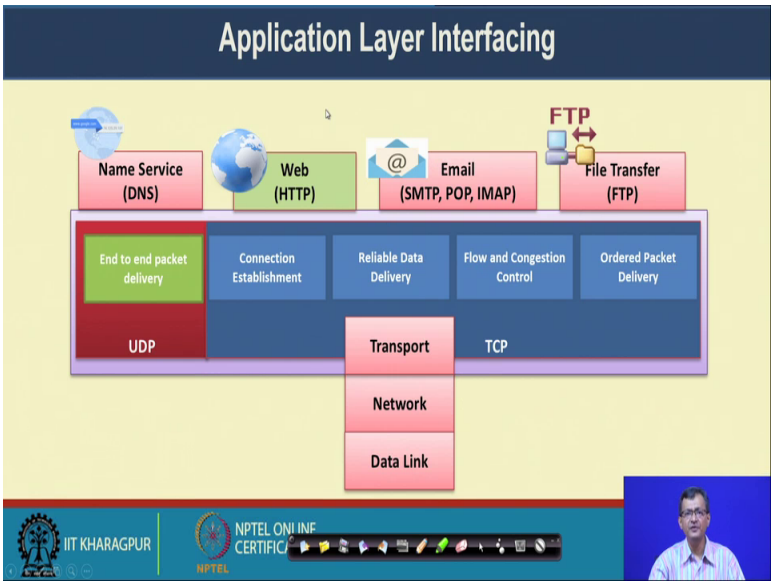
(Refer Slide Time: 13:33)



So, if we look at the anti-application layer interfacing in some other way. So, we have different applications, and they have different type of requirement, right. So, one may be the end to end packet delivery which may not be that reliable service required we can push it to UDP; whereas, there are some of the things where we require a reliable connection oriented service TCP.

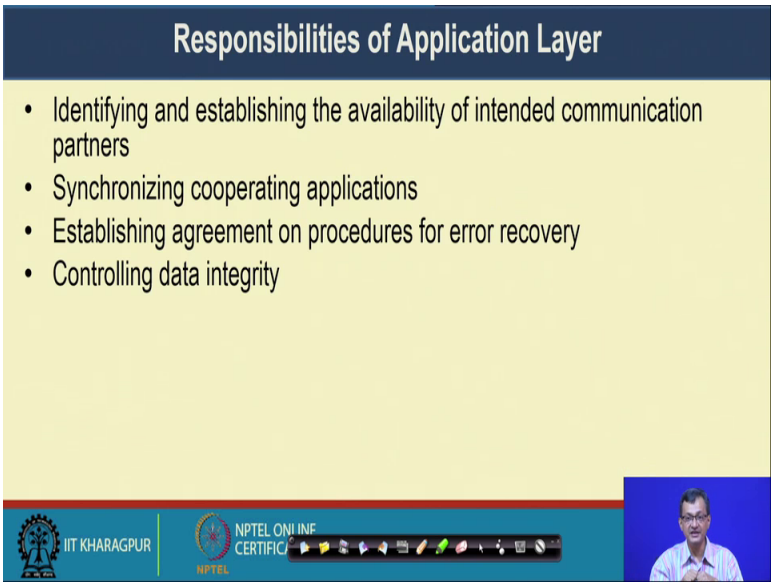
So, we require different type of things. Connection establishment reliable data transfer flow and congestion control order packet delivery and then the rest of the stack.

(Refer Slide Time: 14:04)



And different types of services are like typically DNS types of services are over UDP. Whereas, HTTP email file transfer are over this type of TCPIP TCP type of thing connection oriented service. So, what we see that there are several application which has their different kind of need, and based on that either they are pushed through the TCP type of things or UDP type of things, UDP type of connections.

(Refer Slide Time: 14:35)



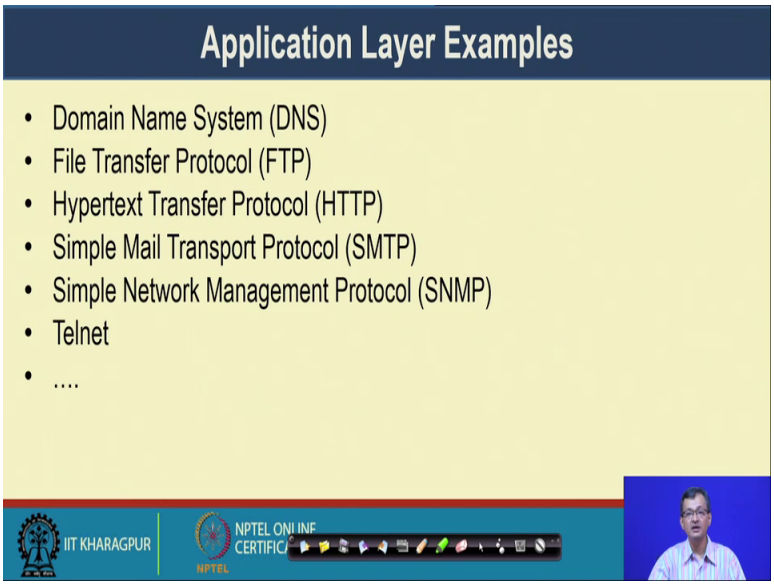
So, again if we come back we if we try to again look at it so one is what are the typical responsibilities of the application layer, identifying establish establishing availability of

intended, communication partners like if I am doing a HTTP iitkgp ac dot in www HTTP ac dot in; that means, my intended other partner is the iitkgp web server it need to connect somewhere other it should take care synchronizing cooperating applications.

If there is a cooperative applications that like say I have a chat server request response I have multiple applications, where I require orchestration or synchronization of the applications right one data goes there then their response income in some other face in that need to be taken care by this at the application level. It establishing agreement on procedures for error recovery, if there is a error how to recover from the reporting recovery from the error. So, there should be some established procedure for that right controlling data integrity. So, I need to have again mechanism procedure for handling data integrity.

So, these are the things need to be, they are basic responsibilities there can be based on the application, then several other properties or responsibilities of the application, but nevertheless if you try to fit in somewhere other they fit in into the this overall structure.

(Refer Slide Time: 16:08)

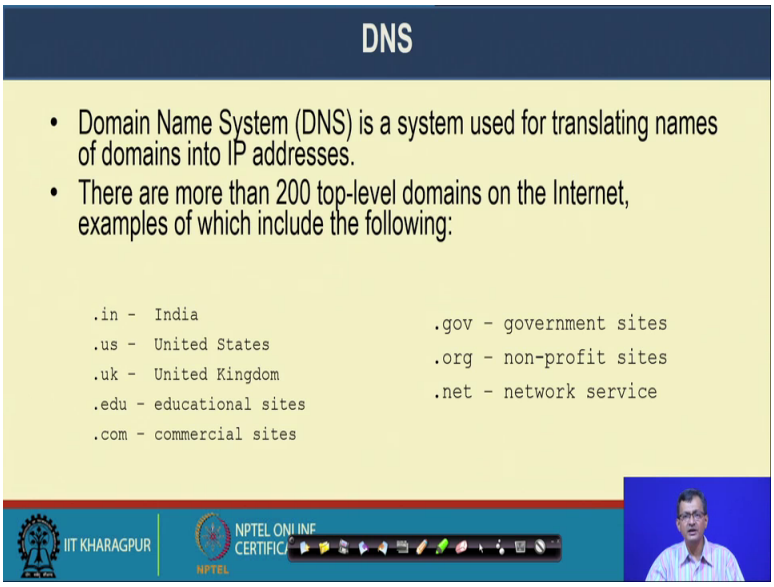


And if we look at the examples, there is a (Refer Time: 16:12) number of examples what we use directly or indirectly day to day, one of the major thing is the DNS or domain name systems. There are file transfer protocols or FTP, hypertext transfer protocol or what we say HTTP which is which is the predominant applications which is which has mostly used across the world. There are simple mail transfer protocols or SMTP which

takes care of our mailing system. There are simple network management protocol basic management of the overall network SMTP Telnet.

And there are there can be any host of applications some of the applications can be used as defined applications which are defined by that particular user; some of the applications are do whatever the applications we are talking about these are mostly generic applications, mostly available in systems and like that.

(Refer Slide Time: 17:18)



So, if we look at the DNS so, what is DNS? Domain Name Systems so, it is primarily major job is to translate name to IP like. So, when we because if you see the; if we I say that I want to find out www iitkgp ac dot in. Now www this name does not have any meaning at the IP layer or the down the layer right say whenever I want a routing the router requires a IP right. So, there should be there is either I give the IP at the top so that in understand this is the thing.

Or there should be someone which resolve that id IP right. So, what we do? That when I give a name I send a resolver that you resolve it. So, during DNS so I send to a DNS server we dissolve and send me back this IP, right. And based on this IP the rest of the things goes on. So, name to IP conversant it is the resolve.

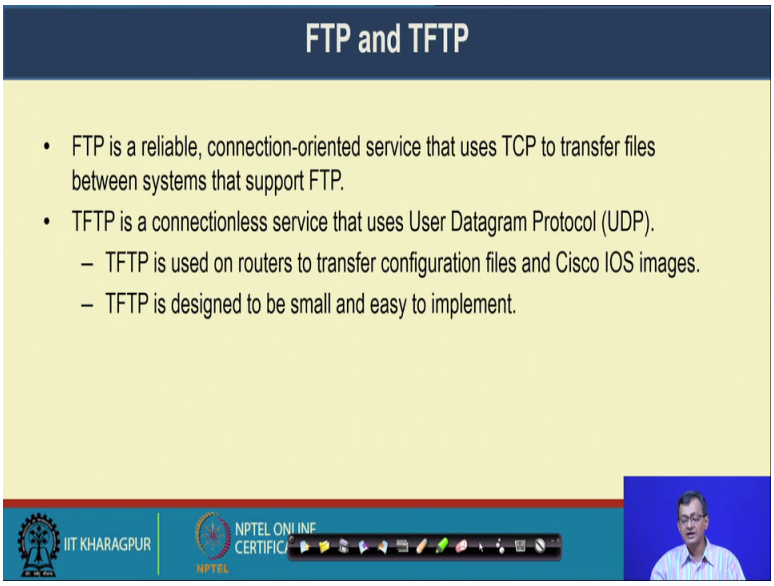
So, I my system say by particular this system or my laptop acts as a primarily a it has a DNS client, which requests to the DNS server which is revert back. How do I know that

DNS server? Either you while network configuration somebody has put the DNS server here or from the network administrator you got it in the DCP IP protocol stack etcetera. Or that automatically done if you have a DCP type of things which when you put the systems it loads the things.

But nevertheless it should know where the server is right. And there are as it is and here is domain based so, that a domain definition there should be a particular protocol to follow. So, there are more than 200 top level domains right in the internet will come to those things more in detail. Some of the example like dot in is India dot u s is us dot edu is educational sites like dot com is combined size dot net is network services and so and so forth.

So, these are top level domains. So, when I say iitkgp dot ac dot in is my domain, then India is the top level domain or sometimes what we say that is a TLD, and below that there is a sub domain called ac; that is, which primarily represent academics below that we have a sub domain call iitkgp, right ?

(Refer Slide Time: 19:39)

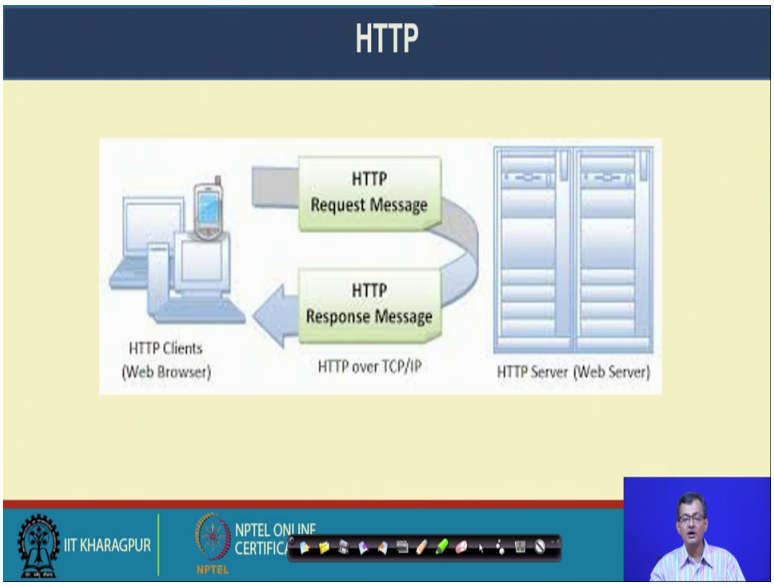


So, when we take the TCP FTP and TFTP. FTP is a reliable connection oriented service that uses TCP to transfer files between systems that support FTP, right. So, TCP FTP is a reliable connection oriented service right. Whereas, TFTP is a connectionless that uses UDP to transfer.

So, there are different places where we need this type of things. Once you have to do for a connection oriented the resource requirement may be high. So, you may have some of the cases whether those resources are not there. Secondly, if there is a failure you can easily retransmit right that may not be a big deal to retransmit, right.

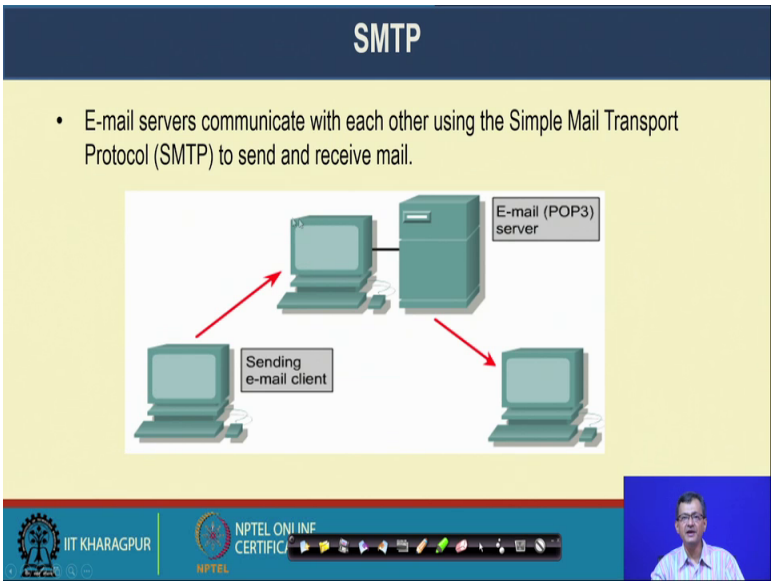
I may not be requiring connection things like I say I want to send a something from one place to another. I one is that I want a reliable service I do not want failure to occur other things if there is a failure I against end up at things to again right then I do not mind sending it again. So that means, based on the requirement like typically TCP is used for routers figures like, like typically for example, some of the iOS images or the router images TFTP is designated for small and easy to implement. So, it is a less payload so it is easy to implement.

(Refer Slide Time: 20:56)



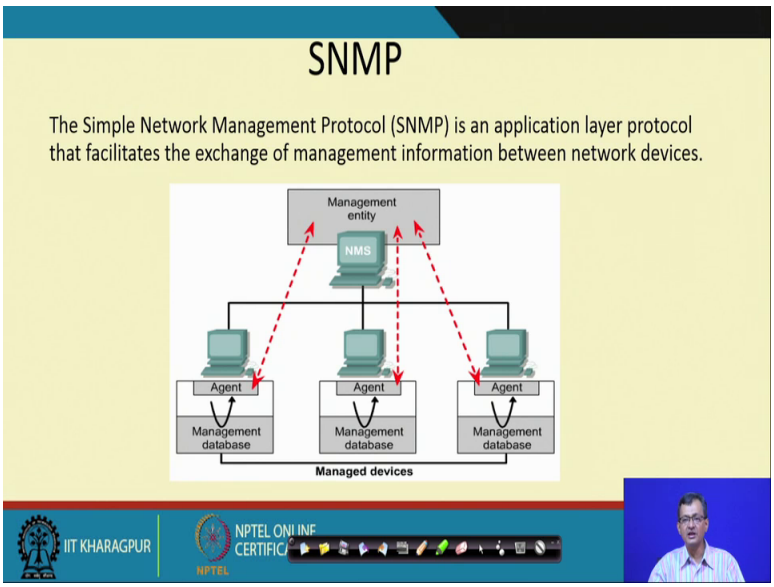
So, another is the HTTP hypertext transfer protocol which is our as we are telling the predominant protocol, and not only that as HTTP is extensively used HTTP is also allowed in most of the routers and firewalls they allow to cross if at all they are allowing the crossing of the thing HTTP is the first thing. That is why I will later on we will see if time permits that if we have this web services etcetera which predominated piggyback on the HTTP. Because these are anyway that part we will see later on. So, it goes for HTTP request and the HTTP server responses which is a HTTP response message. So, this way it goes on communicating between the things, right.

(Refer Slide Time: 21:35)



Then we have a SMTP email server communicate other using simple mail transfer protocol to send and receive mails. So, this is SMTP protocol is for as for the for mail transport protocols. There are other back copies or the front end protocol like pop 3 entire things likes that.

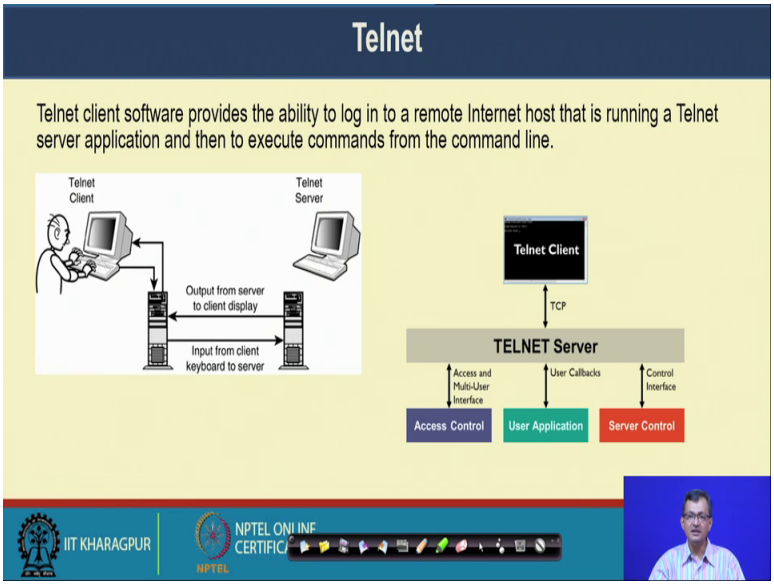
(Refer Slide Time: 21:58)



So, other another protocol which is there is a simple network management protocol SNMP is an application layer protocol, that facilitate exchange of management information between the 2 devices. So, SNMP is a application layer protocol, and it

facilitates that exchange of management information, right. Network level management like it goes on, there are SNMP agents which reports the different status of the network, where this SNMP there is your network management system takes that SNMP data and do. So, it is not may not be directly used by us, but nevertheless it is required for network management right. So, there is a important protocol for network management.

(Refer Slide Time: 22:54)



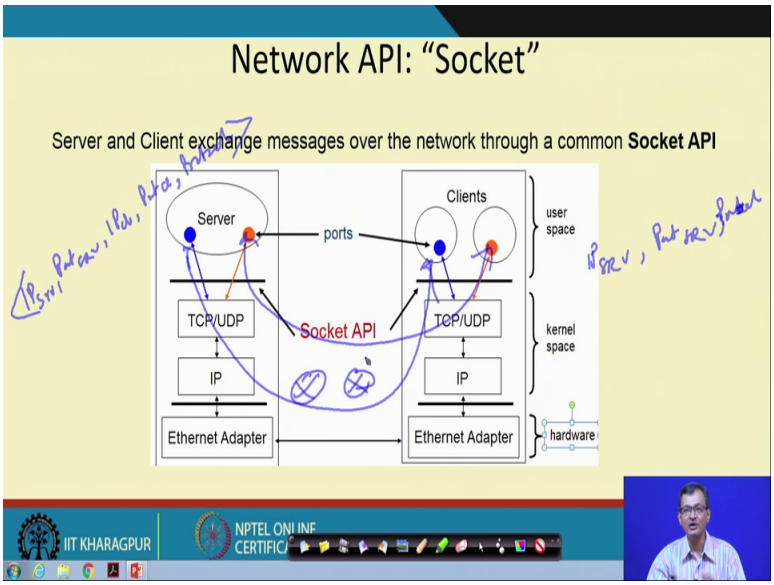
There is another protocol which is called Telnet, right. So, that which allows me which allows us to connect to a remote system or site. So, Telnet client provides ability to log into a remote internet host that is running a Telnet server. So, any client should have a corresponding server at the other end. So, and then to execute commands from the command line so, I have a Telnet client and server and a allowed to the comment to the command line.

So, we have Telnet client and Telnet server. And it goes on communicating between each other. Or if you look at it that Telnet client basically does a TCP connection to the Telnet server, which in turn can have different type of things. One is that can it can be running a user applications or some sort of access control mechanisms, or there is a server control for some other things, right.

So, these are the things which are possible with the things. So, it is a way that I can do a remote login to another system, right. So, I can have a remote access to the systems by, and then we can have applications, we can have server control, we can have access

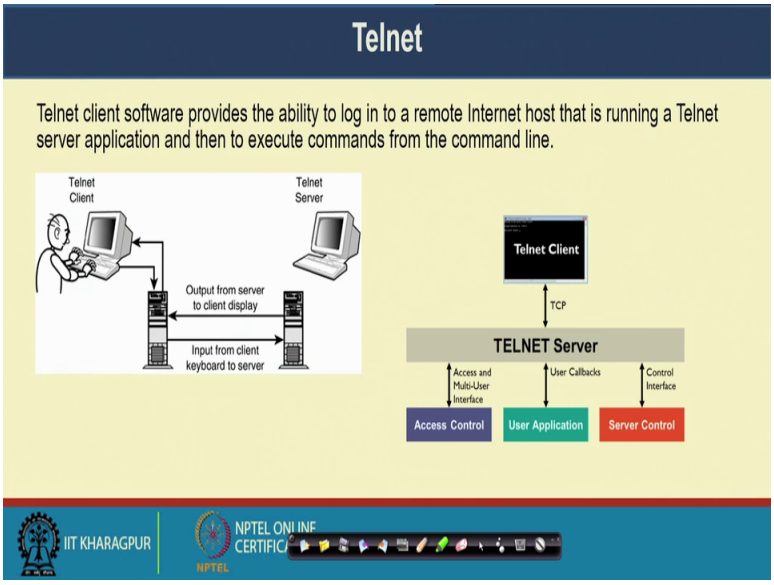
control type of things. Again it is a client server things. So, if you have a Telnet client the other end Telnet server will be there and there should be some credential check that IP login password and so and so forth to go there further matter in FTP also we require those things to be there.

(Refer Slide Time: 24:22)



Now, we come to another thing which is to see that what underlining things how things work what we see a network socket, right. So, it is we hear about network programming or socket API's and like this.

(Refer Slide Time: 24:53)



So, like if we if we try to little look back so, say if this Telnet server, what it is doing? It is basically opening up or it is running a Telnet server things or in sometimes what we say it is a daemon it is running right; that means, which is running and listening to a particular a it is running and listening to a particular port, right. So, always active, right say like if I if I do when I do a say HTTP.

So, this HTTP server what it is doing? So, say if I consider this is a www iitkgp server then what it is doing. So, they are at some port say port 80, that is the standard port 80, port 80 that it is always listening. And listening for what? Is there any request from the client? If there is a request from the client, when I went from these I kean so, it absorb it takes it if it is a concurrent server, it creates a child process or it is 4 k child process and go on serving that like.

So, whether whatever may be whatever is that any kind server protocol, whatever is the applications if the corresponding server end it creates a server process to look at it. So,. So, while connecting from the source, what we require? I require that IP right, of that where I want to connect, like in our case when we d www iitkgp ac dot in what we are looking at. We are basically resolving it and going to a IP address of the iitkgp web server. And then I want to know that where it is available; that means, I want to identify the machine and also identify the process in that machine, right.

How do I identify the process is by the port number. How do I get the port number? That is either it is known or for the popular HTTP things what we have the port is the port 80. So, it is listening to a port 80, right. So, what we have from the client end it is a sending a particular say application. So, it goes on that particular port and goes on that port. So, this is suppose this application is running at port say x this application is running as port y, and then when it goes it goes to the port. So, it may be the same server same IP address, but I have a different port numbers, right. So, if you so, client what is what it goes on. It basically take a the client what it, what is does? It goes to the IP of the server port of the server, right.

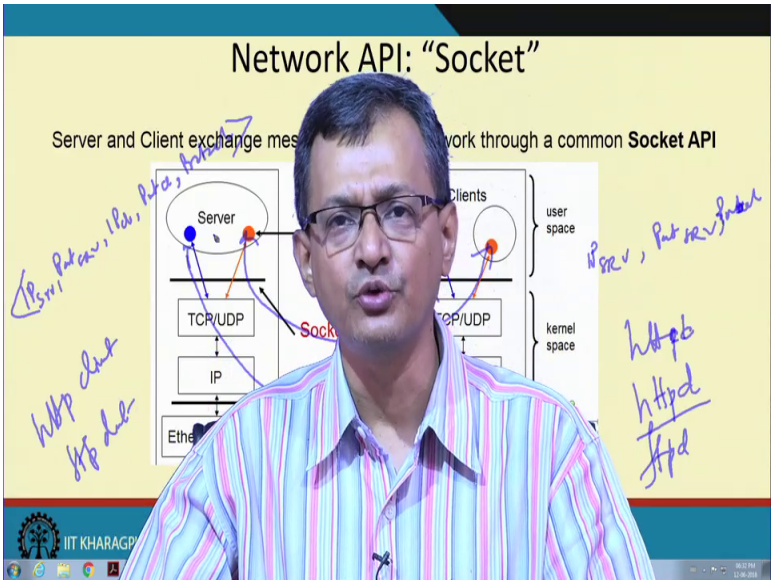
And it goes to the other end, right? When it goes out this it knows that IP of the server, port server IP of the client because it coming from where and port of the client. Port of the client is automatically provided by the system if there is no more thing. Another thing I require is that the protocol. So, if we know this 5 stop allows me to connect to the

things. Like here also we if we consider protocol so, this is it is a what we did once this is there.

So, there is a path is established between these two, right. I am not bother that in between there may be lot of routers etcetera and so and so forth. That is underlining network, but there is a path established. Now I can basically talk like a chat server or a FTP file transfer or any other applications like that. Similarly, if I have another this applications so, we can have another like this, right. So, another things established.

So, this things what we say they a basically a socket is established. So, that the communication path is established. We will be dealing little detail with this type of programming aspect, what we say socket programming how we can write my own socket programming that how what are the things required. So, by this if we if we logically see like the server when that things are running that http daemon is studying what popularly what we say that http daemon is running.

(Refer Slide Time: 30:04)



So, at the server end so, for the http and here we have that http client which is our browser; so, this daemon what it is doing it is listening to port 80 or that server thing and always alive. Like, it is something is anybody there. So, when the client is a request comes, it takes it right it gets the client IP etcetera, and the protocol though our for our case the predominant protocol is the IP protocol.

So, it takes that thing, and it based on its resource and etcetera it responses that whether it accept this protocol. So, once that is there the connection is established, and then goes on thing based on the whether is a state food statelets etcetera based on the protocol thing like it is like for these it is the page is displayed for FTP. So, for the same like for FTP what we have FTPD and here we have FTP client and so and so forth.

Other thing is that in some cases the, it can be a server other case it can access a client and so and so forth. Like, I can have a print server which has a FTP client right. So, it is all that process level things what we want to do. See by doing this we are giving a interface to the inducer to write program one network which can communicate with the other processes, right. And the beauty of the thing is that you are not bothered, but say with the underlining network it takes care of the things, because we are falling some protocols some particular rules which is guided by the things.

Not only that, it is device independent we can have different type of devices and different levels right we can have different mix and type of things. So, long they are fitting into the protocol we are not able to connect, that we are not bothered about the communication path. It can be fiber, it can be wireless, it can be wired and anything, right. So, long again following the protocol, but end of the day your page get displayed, or you write a program which can communicate to each other each other.

So, this is the ubiquitousness of this whole network level communication things, right. Which that is why it is so popular because now it is heterogeneous, it is what we say no centralized control, but say you whatever you are doing at you know rate only you are following the protocols, right. There are guidelines, there are protocol guidelines, there are authorities which takes care of those things, but say what you develop at your end and flowed at the things is a business for your means it is your own business, right.

So, that means, that that allows me to do that. So, what we will do in the subsequent classes will see some of this more application layer things. And slowly go into the transport and network and data link and so on and so forth right. So, with this we let us end our today's class.

Thank you.