

Internet Technology
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture No #12
World Wide Web
Part – II

We will be continuing our discussion on World Wide Web. Now if you recall in your last class we are talking about the http protocol. And we had mentioned that this http protocol provides the heart of the World Wide Web. All transactions that go over the internet today, whenever a client wants to contact a web server and the responses are sent back, they are done through the http protocol. So today we will be continuing our discussion from there.

(Refer Slide Time: 01:18)



So World Wide Web part two is the topic of our discussion today.

(Refer Slide Time: 01:23)



So we start with by looking at the requirements of a web server. For example if someone asks you to design a web server, so we try to address the issues that will be faced by you. As a designer what are things needed that needs to be taken care of? So let us look at the basic requirement of a web server first.

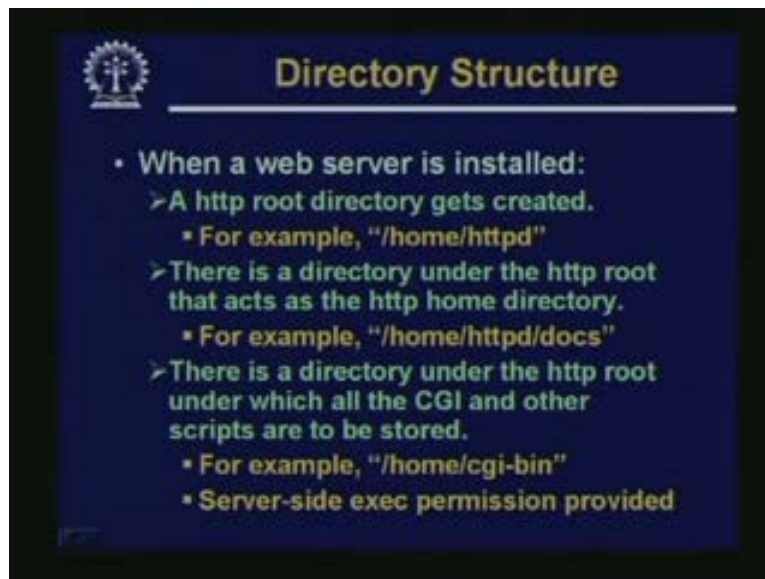
(Refer Slide Time: 01:49)



Now of course the most basic of the requirements is that your web server must be able to accept http request because a web is based on the basis of http protocol. So the first point is absolutely mandatory. And for the majority of requests the request type will be either

GET where you are possibly trying to fetch a web page or the head where you are trying to fetch the information about a page. And if you are supporting forms then possibly also POST. There are other http commands also you recall. But these three are the most basic there are PUT, DELETE this kind of commands are also there but these three are the most basic ones. Now if you want to have GET and POST facility for that you can submit a query string. Then in addition you will have to have the facility for handling so called server side scripts. Now server side scripts are nothing but programs these are executable which are residing on the server machine. This executable can be written in any languages. They may be written in some scripting language Pearl or may written in languages like C or Java. And whenever you submit a query string using GET or POST, these executables will GET executed. Of course which programs you are trying to GET executed that must be mentioned in GET or POST command. And after execution the outputs of this program will be sent back to the browser or client. Typically this will be in the form html page. So this is a simple requirement that a web server must confirm to.

(Refer Slide Time: 03:55)



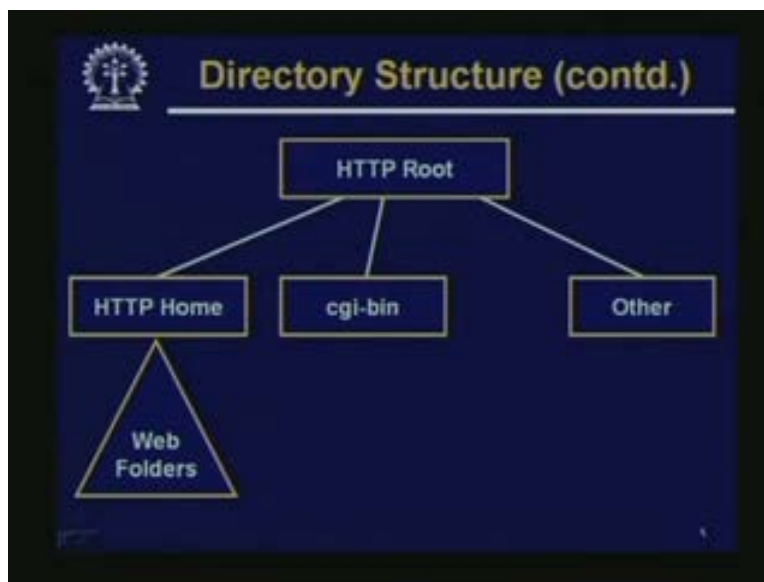
Now in addition there are few other things well when a typical web server is getting installed is being installed. You will find there are some typical conventions for the directory structures which are followed. For instance there is something called the root directory. So all http files and folders are created under this root directory. So as an example this root directory may be slash home slash httpd for some web server like apache this is the default web server root directory. Similarly there will be a root tree which will be located under the root that will be called the http home directory.

Well home directory is something where all your web pages are located. So whenever you are putting in some web pages on the server it must be under the home directory. In contrast the root directory is something where all the files related to the web server will be stored. Some of them may be your actually web pages some may be other miscellaneous files also. So typically the home directory can be something as home httpd

docs may be and optionally there can be another directory under the root under which all the so called common gateway interface or cgi script programs and other scripts can be stored.

Well again as an example this home cgi bin. This is a typical name not for the files which is storing under the cgi bin. Normally server side execution is provided for them because the idea is that the files that are stored under the cgi bin directory they are meant to be executed. They will GET executed if you have a GET or POST command which is actually specifying a query sting and a script which is located under the cgi bin directory for example. So all these files must have the execute permission whenever they are specified the programs will start executing.

(Refer Slide Time: 06:21)



So the directory structure looks something like this. So on the top we will be having http root under which all the http directories and folders would be created. There would be the home directory, cgi directory, there could be other directory some may be simple documentations some may be some other library files etcetera. And your web page or web site that you are designing that will be under this http home. So these web folders which are shown here they will contain the directory tree structure of your so called web site. This is how the files or the directory folders are located or organized.

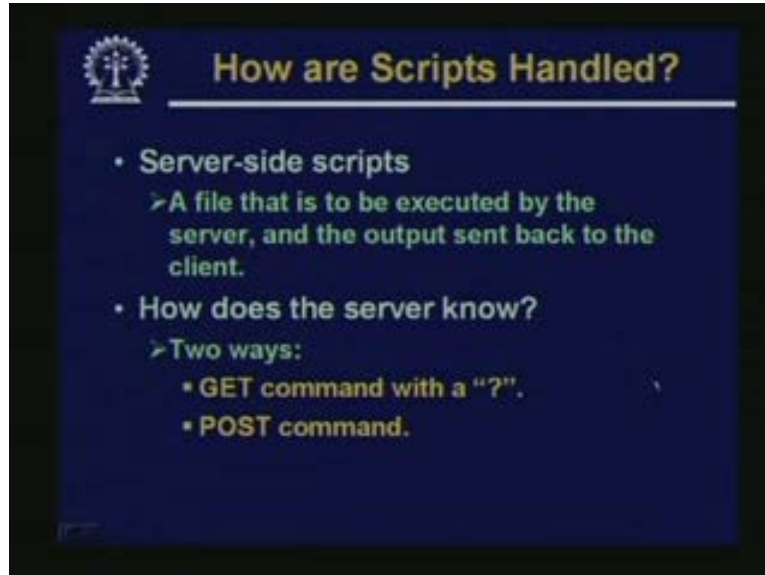
(Refer Slide Time: 07:07)



In addition there is something called a default web page which is supported by all web servers. There is some default web page that will get returned by the server. If you are not specifying the document explicitly. For example you consider a request like this, GET you are simply specifying a name of the site you are not specifying the name of any file or the path name of that file directory structure. So just as specifying `www.xyz.com`, what this command will actually mean is that the request will be sent to a web server of course and the web server what it will do?

It will search in its own directory structure under the web root and you try to find out whether there is a file with that default name present in it. If it is found that default file name is returned to the requesting client and usually the default file name is `index.htm` or `index.html`. They refer to a html document. Now it is possible to change the name of this default file directory because there is something called server script configuration. There is a configuration file where a number of options are there. You can simply open that file and edit. So in that file there is one line which specifies - What is the name of your default page? So instead of `index dot htm` you can change it to for example `default.htm` so that will be your default page in that case.

(Refer Slide Time: 08:51)



Now talking about the scripts, the first thing about scripts is that they are residing on the side of the server. They are server side scripts and as I mentioned these scripts they are nothing but a program. This program may be written in any language you want. So this actually refers to a file that will be executed by the server and the output of execution will be sent back to the client. This is how the scripts work, but how do the server know that which script to run and how to GET the input data and where sent back the output. Well there are two ways the server script may start executing in response to either a GET http command. In fact there is a question mark in the command line or a POST http command. Now we shall see this in detail.

(Refer Slide Time: 09:52)

GET Command with a "?"

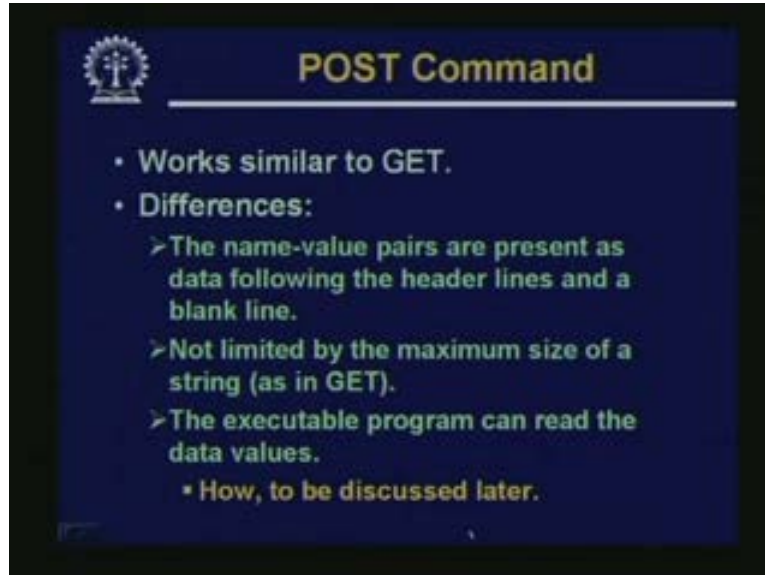
- Consider an example HTTP command:
`GET /cgi-bin/xyz.pl?roll=1234 & sex=M`
- What happens?
 - > Server identifies the "?" following the GET.
 - > Identifies xyz.pl as a program to be executed.
 - > Allows the xyz.com program to read the values present in the string following the "?".
 - How, to be discussed later
 - > The output generated by the xyz.com program is sent back to the client.

First let us look at the GET command with a question mark one example is shown here. So recall GET command with a question mark means, that whatever follows the question mark this will represent a so called query string. So there will be a number of name and value pairs separated by ampersand. For example roll is a name, 1234 is a value, sex is a name; m is the value. So there can be any number of such name value pairs you can put in separated by ampersand and whatever comes before this question mark, this does not refer to a web page. Rather this refers to an executable file or a script.

This is how the client can specify which program to run on the server side. So we explicitly specify the name of the script. So here for instance you are referring to a file called xyz.pl which is residing under the cgi bin directory. This file needs to be executed and the output has to be sent back. Now if you look at what happens internally, the server has to identify this question mark in the query in the total command line. If it is a GET command and if it finds a question mark then it identifies that whatever file name is specified before that is actually a program which is to be executed.

Now it starts executing the xyz., this is not com this is xyz.pl. It starts executing the xyz.pl program. Now the program is written in such a way that it can read in the values of this name value pairs which was supplied as part of the parameters. Now how this values can be read in this we shall be discussing later not now. And again the output generated by xyz.pl program that will be sent back to the client. And as I said typically the output is generated in the form of html page so that when it is sent back to the browser the browser can display it in a suitably formatted way on the screen.

(Refer Slide Time: 12:20)



Now if it is a POST command, it is in a way similar to GET, but the difference is that, the name value pairs which needs to be taken as the input to the program and not present on the command line. Rather they are present as data following one or more header lines and a blank line. So there will be a number of header lines after the header line there will be a blank line; after the blank line all the name value pairs will appear. This is the format in which POST sends the data to the web server. But in POST again the name of the file that you specify that will be the executable program.

And here there is one difference from GET is that you do not limit the number of such name value pairs or the total size of the string that you can send. And here again the executable program which runs can read in the data values. But how this again we shall be talking about later. Because we know that the program has to execute it we have explicitly named the program as part of the command. But when the program executes, how to get the values of those name value pairs. This we shall see later when we talk more about the cgi scripts and the way it works and how they are actually written.

(Refer Slide Time: 13:54)

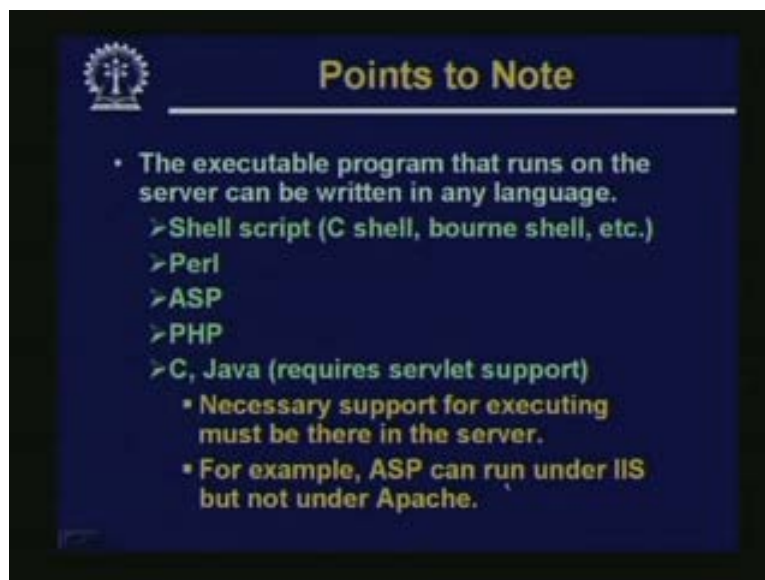


```
POST /cgi-bin/myscript.cgi HTTP/1.0
From: isg@hotmail.com
User-Agent: HTTPTool/1.0
Content-Type: application/x-www-form-
urlencoded
Content-Length: 32

Roll = 1234 & Sex = M & Age = 20
```

So an example of the POST command is shown here. So the first line is the actual POST command which specifies out here the name of the cgi script program. The name of the executable. Here the name is myscript.cgi. This is the name of the executable file then there are some header lines followed by a blank line out here, then the name value pairs. Now this name value pairs can appear in as many lines as you want. Not necessarily they will have to appear in a single line you can break it up in to a number of lines if you want.

(Refer Slide Time: 14:39)

- 
- The executable program that runs on the server can be written in any language.
 - > Shell script (C shell, bourne shell, etc.)
 - > Perl
 - > ASP
 - > PHP
 - > C, Java (requires servlet support)
 - Necessary support for executing must be there in the server.
 - For example, ASP can run under IIS but not under Apache.

Now there are a few points which you need to remember. First thing is that the executable program that we are talking about. The so called cgi script, cgi stands for

Common Gateway Interface. Now these executable programs can be potentially written in any language of your choice. It can be written in some shell script. This can be c shell, bourne shell, cone shell or any other shell. It can be written as a Perl. Perl is a popular choice for many. It can be written using some scripting languages asp or php. These are also quite popular. They may be written in conventional programming languages like C or Java. But one thing you remember in whatever language you write the program, you must ensure one thing.

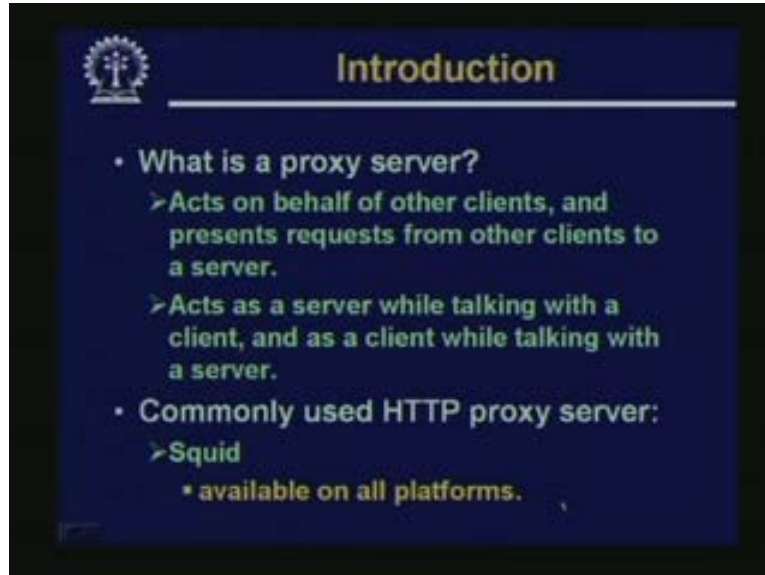
That the program when it is executing on the server side it must have some support for execution for example. If it is a c program then there has to be a c compiler to compile that first. If it is a java program, there has to be a java byte interpreter, it is a Perl program, there has to be a Perl interpreter. So in this way so to run a script written in a particular language, support for that particular language should exist on the side of the server. This is what you have to remember. For example asp a program written in asp it can be run under the internet information service, IIS which is available under the windows. But you cannot run it directly run it under the apache web server because IIS by default supports asp. It can interpret asp commands directly, that is why when you have a program written in asp you have to run it through IIS.

(Refer Slide Time: 16:40)



Now let us very quickly look at what is a proxy server? And why we use it for what are the facilities that a proxy server can provide?

(Refer Slide Time: 16:53)

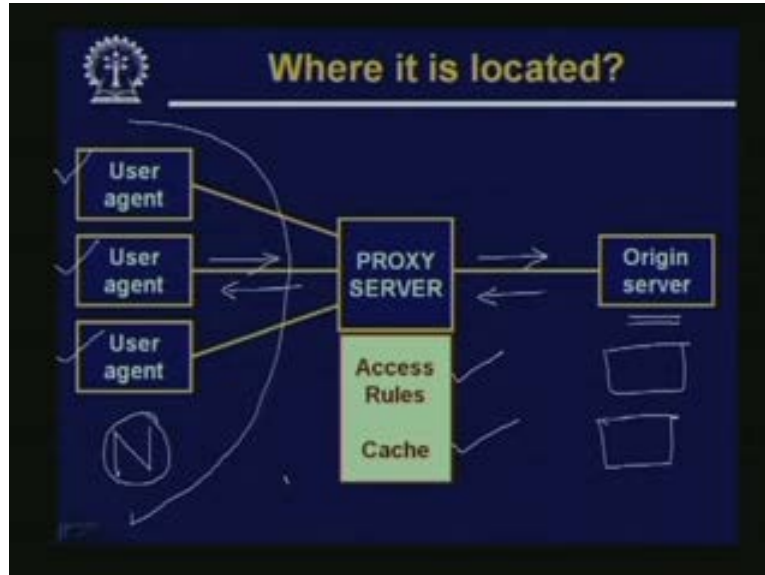


Now a proxy server we had mentioned before that it acts as a intermediating between a client and a server. Typically in the internet scenario the client will be a web browser like Netscape, like internet explorer. The web server can be some site from which you are trying to access the page. This web server is some times also called the origin server that is the origins of the document on the resource you are you are trying to access. Now in this scenario you are having the access through the proxy server which is intercepting all your requests. So whenever you want to send a request to the outside world you request will be first sent to the proxy server.

The proxy server will be sending the request on your behalf to the outside world. This is how the proxy server works. But let us see how? The first thing as I said, that it acts on behalf of other clients and presents requests from other clients to a server. The clients can be on the one side of it and the server is located on the other side of the proxy server. Now depending on which mode it is in, whether it is receiving a command from a client or sending a request to an external server, it can sometimes act as a web server, sometimes act as a web client. How? See the proxy server whenever it is receiving a command from a client then it acts as a web server to the client as if the client is sending a request to the web server.

So the proxy server accepts the request as if it is a web server. But it is not processing that request directly and locally rather it is forwarding the request to some origin server or web server which is outside the network. Now while making the second part of the request the proxy server acts as the client and the server located outside that acts as the http server. So a proxy server has a dual role. Sometimes it is a client, sometime it is a server. And the most commonly used proxy server is a program called squid which is freely available and it can be installed in almost all platforms.

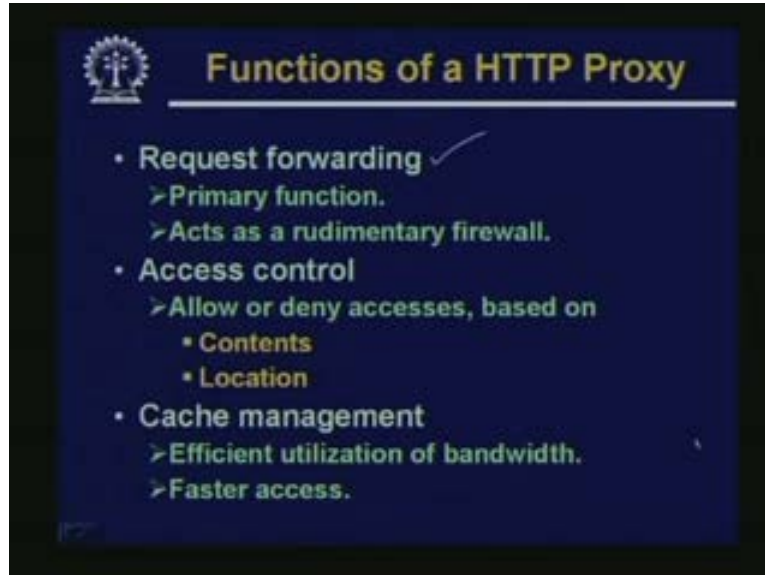
(Refer Slide Time: 19:33)



Diagrammatically a proxy server looks like this. Typically these user agents which are shown on this side, this part belong to a private network. This is a private network say N. These are three computers which are located inside the private network. These user agents may be nothing but may be these are simple browsers. There is some commands which have been typed on the browser and they are possibly referring to some origin server may be this is yahoo.com. There can be one origin server there can be several origin servers. But I am showing just one origin server in this diagram just to illustrate.

So the user agents will be sending the request to the proxy. So request will be coming like this. The proxy will be forwarding the request to the origin server. So the origin server after processing the request will be sending back the request to the proxy. The proxy in turn will be sending back the request to the original request user agent. This is how it works. Now in addition to this simple request forwarding and response receiving a proxy server typically also contains some access rules and cache. This we shall be talking about very shortly. These are some additional information which is maintained inside a proxy server.

(Refer Slide Time: 21:11)

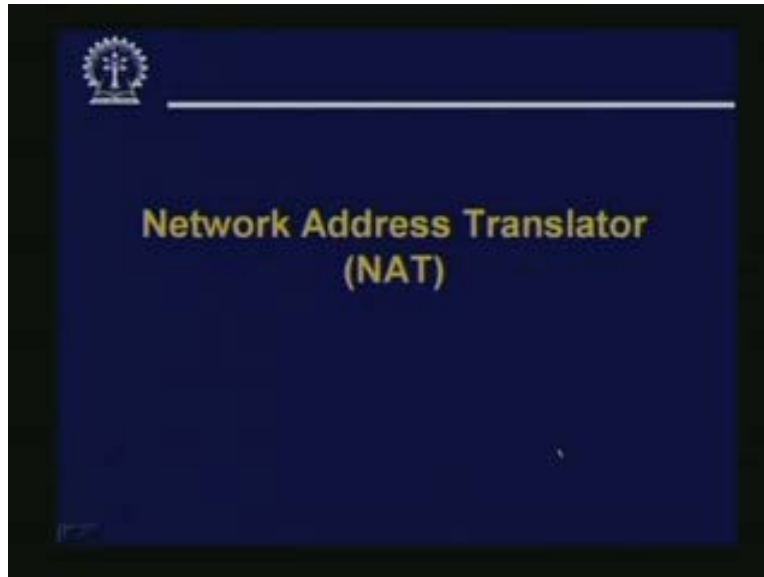


So talking about the functions of a proxy. First one we have already mentioned. It forwards request. See when it is forwarding a request it also acts as a simple and rudimentary fire wall. It can allow some requests, it can deny some requests. When someone from outside is trying to access an internal node your proxy can stop it. So these kinds of capabilities can be programmed in the proxy. So it can act as a simple firewall and you can have access control to a proxy. Access control means you can allow or deny certain access based on contents, based on location. Now when you say based on contents it may be based on some string which appear in the documents which we are trying to fetch. May be some string patterns in the web site name which you are trying to access.

Depending on the policy of the installation, you can set some rules that some body should not be able to access this kind of sites or these kinds of contents. Typically these are specified by mentioning a list of strings which are you can say, only access size which contains the strings or access the size which does not which does not contain the strings. So it can be either way you are allowing or denying based on some contents. Similarly you can also control access based on some location. You can specify that these are the websites that I do not want anybody to access from inside. So these are some websites which are blocked. This kind of things can be done very easily. The third thing is that a proxy maintains http cache.

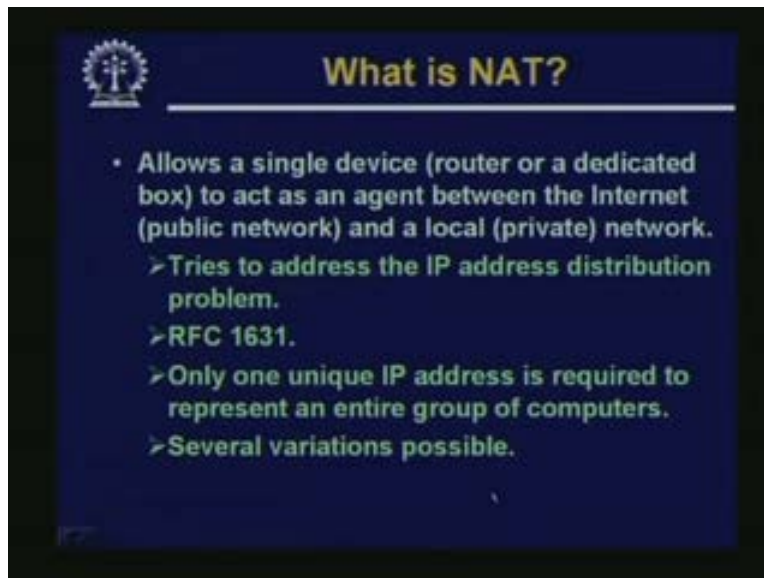
What the cache means is that, see http. Http requests are coming to the proxy. Proxy is forwarding the request and is getting back the requested information may be the web pages from the servers. These pages are forwarded back to the requesting client. But what the proxy does is that it also maintains or keeps a copy of these pages in its local disk. There is a disk area which called a cache where these are maintained are stored the idea is that if some client requests for a page which is already there in the cache. It need not send out a request outside and GET that page again. It can be directly forwarded from the local cache it of course saves bandwidth and allows for faster access.

(Refer Slide Time: 24:09)



Now let us talk about a very important device which is used in the internet. This is called a network address translator or a NAT.

(Refer Slide Time: 24:30)



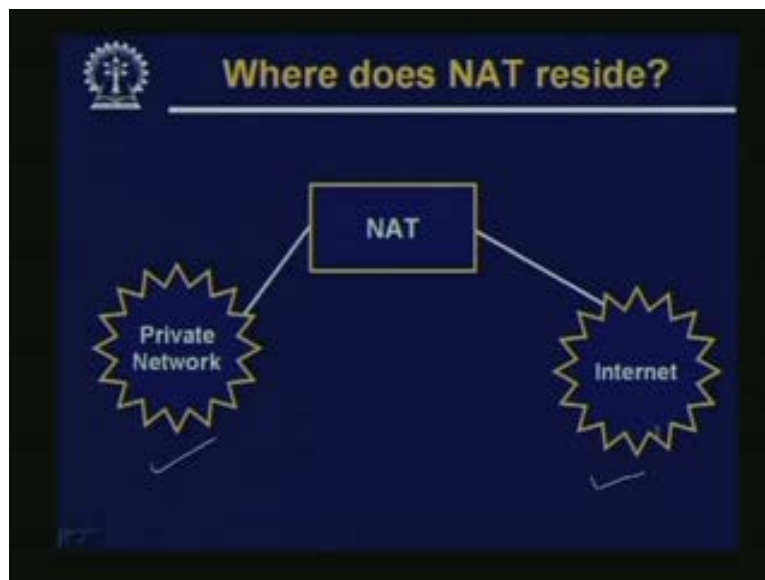
Now the reason why we use the NAT or manifold well in the simplest scenario a NAT allows a single device. This single device can be a router it can be a dedicated box. See this router can act as a NAT. This dedicated box can also act as a NAT. These are available commercially. So what we are saying is that, we are allowing this device to act as an intermediate agent between the internet and a local network. Now the local network

we are calling or we are referring to the private network and the internet or the external network we are referring to the public network. Now NAT sits between the public network and private network.

So you may argue that well it sounds very similar to a proxy server but well there are some differences. This will be clear as we go in to the details in the working of the NAT. See NAT does not only regulate access it also manages IP addresses see one bit problem many of them face today is that suppose if we have a organization which contains say 1000 computers, but we do not have 1000 valid IP addresses with us which we can assign to these computers rather the internet service provider through which we have obtained the internet connection, they give us only a set of few addresses. So how we can manage with this few addresses NAT is one such solution.

So NAT tries to address the IP address distribution problem the way NAT is specified in to works is specified in a RFC document. If you are interested you can have a look at 1631 is the number of the document. Now if you are using NAT then potentially one single unique IP address is sufficient to provide connectivity to an entire group of computers. Of course there are several variations which are possible. But in general even if you have a single IP address with you which is a registered IP address that may be sufficient for your organization requirements to have access to the outside world. But there is some restrictions and we shall see later that not all kinds of access are all. May be accesses from inside the network to outside will be allowed, but not the reverse. Someone from outside your network may not be able to directly connect to a computer inside.

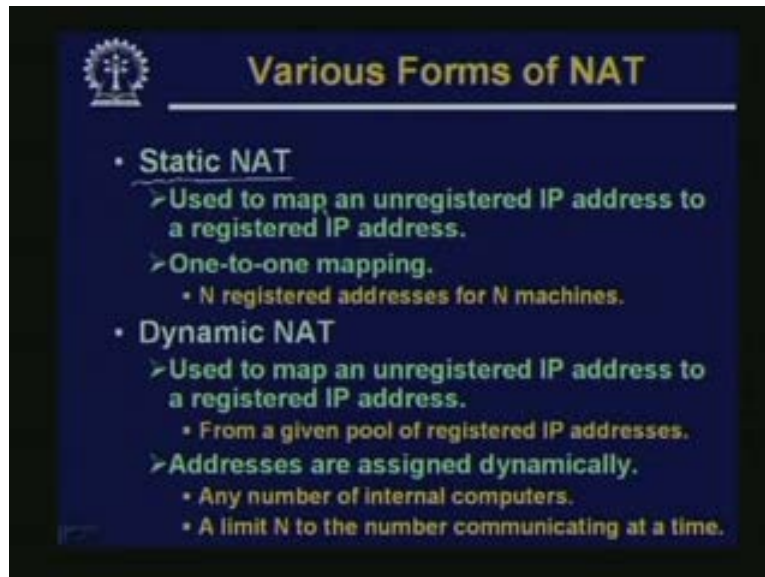
(Refer Slide Time: 27:33)



So as I mentioned NAT resides between the private network and the public network which is the internet. Now NAT is available as a separate box from the network vendors. Most of the routers which also connect a private network or the public network. They

also have the capability to act as a NAT. So NAT can be embedded inside a router or you can have a separate box that can act as a NAT.

(Refer Slide Time: 28:10)



Now let us see what are the various forms of NAT network address translation. Static NAT. Well, as the name implies we are trying to provide a static address translation. So what you are trying to do is that you are trying to provide a mapping of an unregistered IP address to a registered IP address. You remember this is a one to one mapping which means that if you have n number of unregistered IP addresses on n machines then to provide static NAT. You will be requiring N registered addresses. What this means is that say internally.

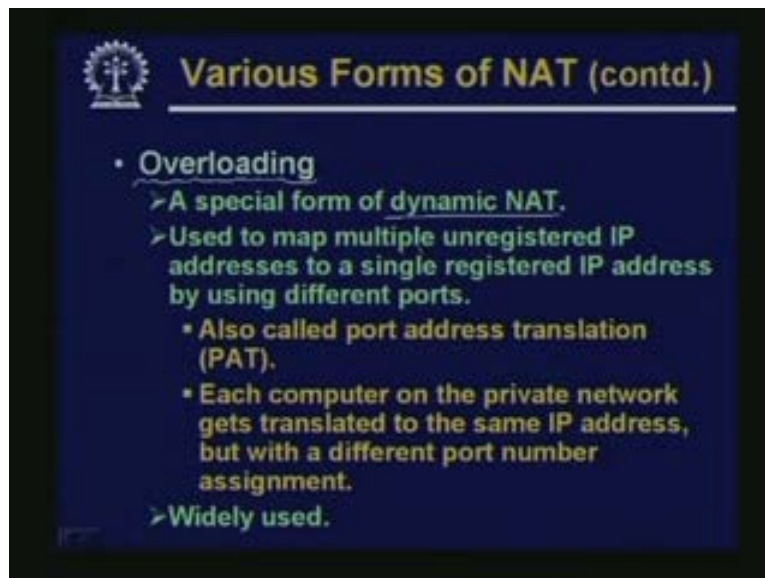
Suppose you have ten computers and you have 10 register's IP addresses with you. NAT will provide a correspondence between the internal computers and these IP addresses so that with this IP addresses they can access the outside world. Now static NAT is almost similar to providing these addresses directly to the computers. So it is not much useful other than some very specific needs. Why do want to assign some fixed IP address to some of the computers inside? Like for example inside your network, there is one computer which is the web server and you want the outside users. That means users are people who are residing outside your network they should also be able to access your web server.

So your web server must have a statically assigned IP address. But for others you may not need this. For others you have alternatives. One alternative is to go for dynamic network address translation dynamic NAT. Dynamic NAT also provides or maps an unregistered IP address to a registered IP address. But the difference from static NAT is that here you do not have one to one mapping. Rather here you are getting this registered IP address from a given pool of registered IP addresses. What I mean to say is here is something like this.

Suppose I have with me ten registered IP address suppose I am the NAT I have ten registered IP address with me and the on the other side in the public network there are 100 computers. So whenever I get a request from one of the computers in the private network I assign one of these 10 addresses to that computer and as long as the request is being processed that address will be statically assigned to that computer. But once the request processing is over that address will again be de allocated and we will be returning back to my pool.

So with this scheme I can have 10 simultaneous access connections at the same time. But we are not limiting the total number of computers that can possibly have connections with the outside world. In fact 100 or even 1000s if there are 1000 computers they can have access but not more than ten at a time. I have only ten valid addresses with me and I am allocating them based on demand. So these addresses are assigned dynamically you can have any number of computers. But you will be having a limit N depending on the number of addresses you have to the number that can communicate at a given time.

(Refer Slide Time: 32:11)

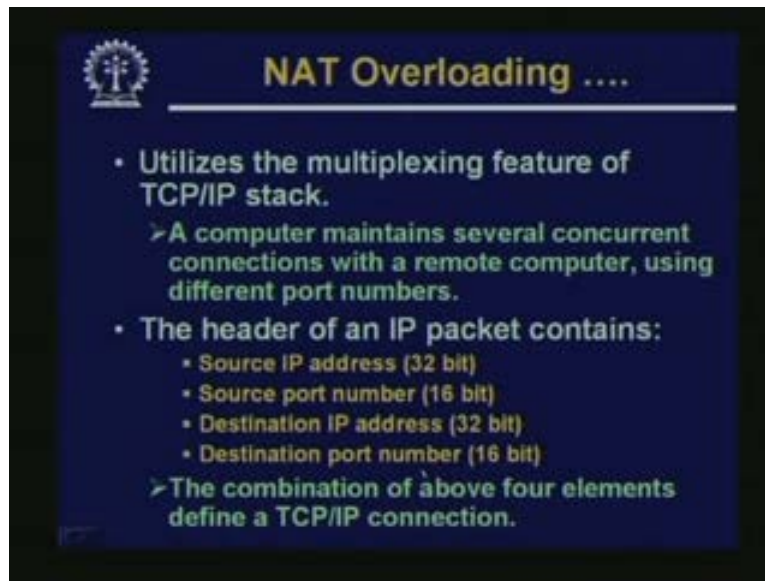


There is another form of NAT which is perhaps the most popular. This is called overloading. This in a sense is a special form of dynamic NAT because addresses are not assigned statically to the computers. They are somehow access permissions are generated dynamically. This is again is used to map multiple unregistered IP addresses to a single registered IP address. See here there is a difference you have a single registered address not multiple as discussing in the previous case. There is a single valid IP address available with me. But I can support 100 simultaneous requests from the private network.

How I can do this? I can do this by using the port numbers. So here I am using port numbers the different requests will be using different port numbers. Because we are using ports to distinguish the requesting computers. This method is also called port address

translation or PAT in short. So effectively in this scheme. Each computer in the private network will get translated to the same IP address. Only difference is that they will be having a different port number. So with this kind of a scheme you are no longer limited by the number of IP addresses you have. You can have as many simultaneous connections you desire using this scheme. These are said this is widely used.

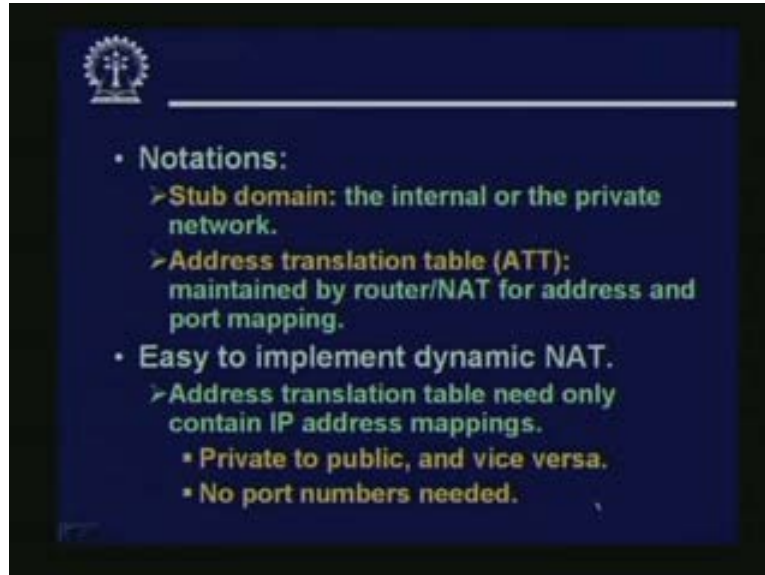
(Refer Slide Time: 33:55)



So let us look at NAT overloading in some detail. So as I said using multiple port numbers you can support several simultaneous connections. This is what is called multiplexing in the transport layer level of the TCP/IP protocol stack. Multiplexing means that a computer maintains several concurrent connections with a single remote computer. But several connections can be maintained by using different port numbers. Like for example I can have a connection with a machine x. But effectively I can have 100 connections with x. If I use 100 different port numbers, now each of this 100 connection may refer to some different server programs or may be the same server program.

There are multiple requests coming to it. I can use different port numbers to distinguish. Now this recalls that the IP packet header well including the extension header. This will contain the source and destination IP addresses and the port numbers. So the combination of these four elements will define a complete connection. So even if this source IP address and destination IP address are the same. If we can vary say for example the source port number, then these four tuples will still remain unique. May be the other three tuples have the same values. But at least one of the values of the four things must change between the different connections. This is how we can ensure that multiple connections can be established.

(Refer Slide Time: 35:48)



Now, some notations we define something called stub domain. Stub domain means the private network. So the domain which is behind the NAT which is a part of the private organization network, that is your stub domain. And for NAT for this address translation, we must maintain a table which is called address translation table or in short ATT. This ATT will have to be maintained by the router or the network address translator while it is carrying out the address mapping and also the port mapping. Well, let us look at the issue one by one. Suppose we want to implement simple dynamic NAT no overloading for the time being. Simple dynamic NAT means I have a pool of IP Addresses. If there are n addresses I can support n simultaneous connections.

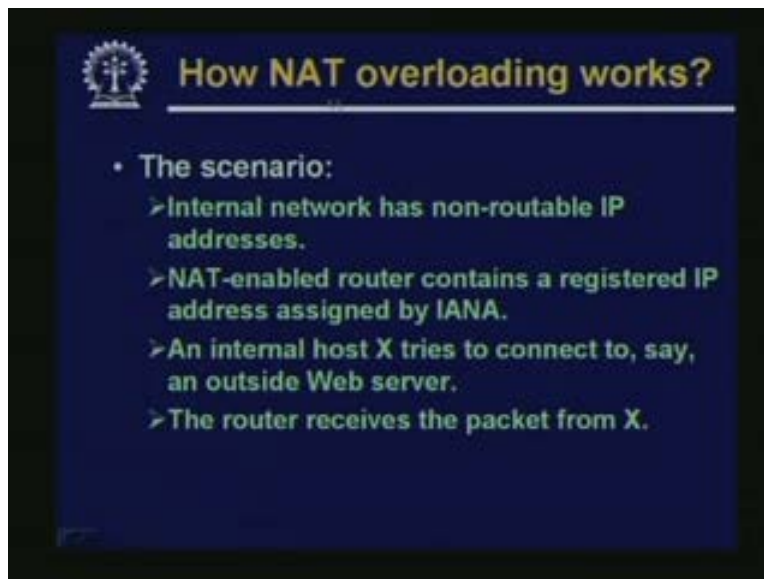
So far implementation of dynamic NAT, let us try to see what should the ATT contain? The ATT need only contain the IP addresses of the source and the IP address which has been allocated. Like you try to understand some computer had sent a request to the NAT. That computer, suppose had an address 10.5.6.7 say. The NAT assigns a valid address to it and in the table makes an entry that 10.5.6.7 has been assigned an address this. So for all the requests which are coming there will be an entry maintained in this table. Now what this table will allow it will allow outgoing connections. Of course, but it will also allow in coming connections. Why? Say as long as an entry like this, remains in the table you can have a connection from the outside world coming to you.

If you see that request is destined to an address, which is called 10.5.6.7, then you can possibly forward that request to the destination machine. Because you know that this particular address is there. But there is a problem if you are saying that the request is coming with 10. Address, which you may recall that refers to a private address. Maybe the external routers will be discarding them. So the packet may not reach the NAT at all. So what you can do instead of advertising this private address to the outside world. You tell the outside world, that well my NAT has given me this IP address and has assigned

me this. So you please connect me through this valid address. May be there is a valid IP address 2 200 3.10.5.17 or something like that.

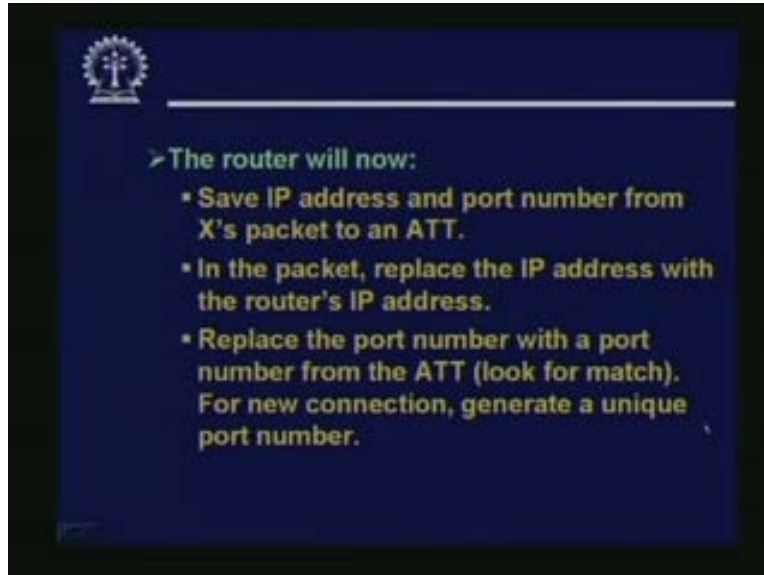
The outside agent will be using that IP address. The ATT will provide the translation and will be forwarding the request to the internal machine internal host. So using dynamic address mapping you can have some of the entries in the table fixed statically. Say for the web servers I give an example where you need a permanent address which is known to everybody. But the other addresses can be dynamically assigned and changed during the duration of a connection only. So in this way you can have the best of both worlds you can have some addresses which you need to be accessed from outside. And with the other IP addresses available you can provide dynamic access from the internal nodes. But you cannot have the other around a note from outside a host from outside cannot directly access an internal node. If it is does not have statically assigned IP address stored in the table. Now in this scheme port numbers are not needed.

(Refer Slide Time: 40:06)



But when you talk about NAT overloading you require port numbers. The scenario is like this. The internal as I said it possibly has non routable IP addresses or private IP addresses which if it appears as a destination address in a packet the router will ignore it. Router cannot forward a packet to an address which is a private address it will simply ignore it. NAT enabled routers or a NAT box which contains a registered IP address which will be assigned by the internet authorized access provider address provider IANA. So this one single IP address will be sufficient. Now suppose the scenario is like this. An internal host, say x tries to connect to an outside web server. So the request will first reach router. The router will receive this request packet from x.

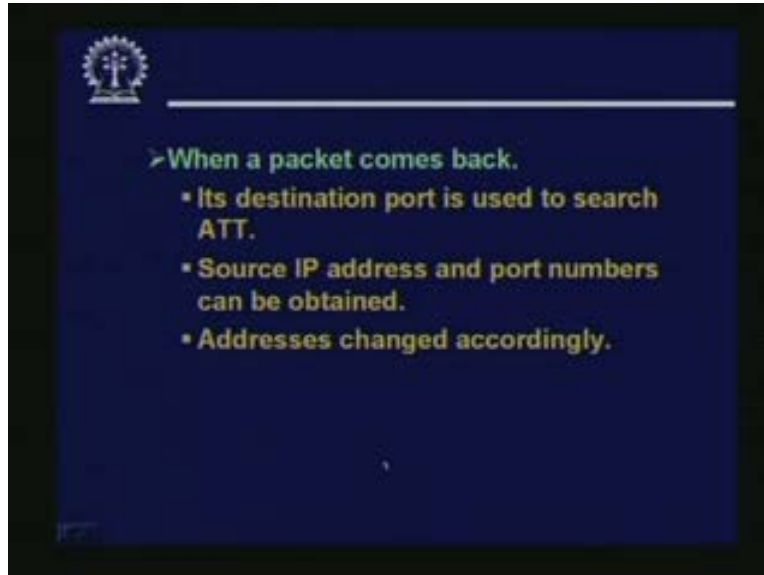
(Refer Slide Time: 41:11)



The router will now save the IP address of the requesting computer. This is a private IP address, please recall. And the port number which the computer was using for sending its request. This can be obtained from the packet that was received from x, it saves this two into the table address translation table. And then the packet it will replace the IP address which was the private address with the router's IP address. Replace the port number by the port number if the entry was already in the table or if it is a new connection you are trying to make to generate a unique port number.

So you are basically modifying a packet by changing the destination IP address and the destination port number. Some internal computer was sending the packet to the router of the NAT. The NAT changes the address of the package to some address of the outside world also the address of the source is changed because the request has to come back. Now if the source address still is the private address the request cannot come back. So it changes the source address by the valid IP address that the router contains and also port number is a unique number that it generates automatically.

(Refer Slide Time: 42:43)



So, on the other hand when the packet comes back the destination port number of the packet is used to search the ATT. So destination port number as I mentioned was assigned automatically and uniquely. So, that port number will uniquely identify an entry in the table that you can say that will acts as a primary keys of that ATT. So from that table you can obtain the source and port numbers source address and port numbers. You can accordingly change the source addresses of the IP packet and finally you can forward the packet to your to the host which was there in the internal network. So finally the packet will GET forwarded to the computer from which the request has originated.

(Refer Slide Time: 43:34)

A slide with a dark blue background and a white logo in the top left corner. The text is as follows:

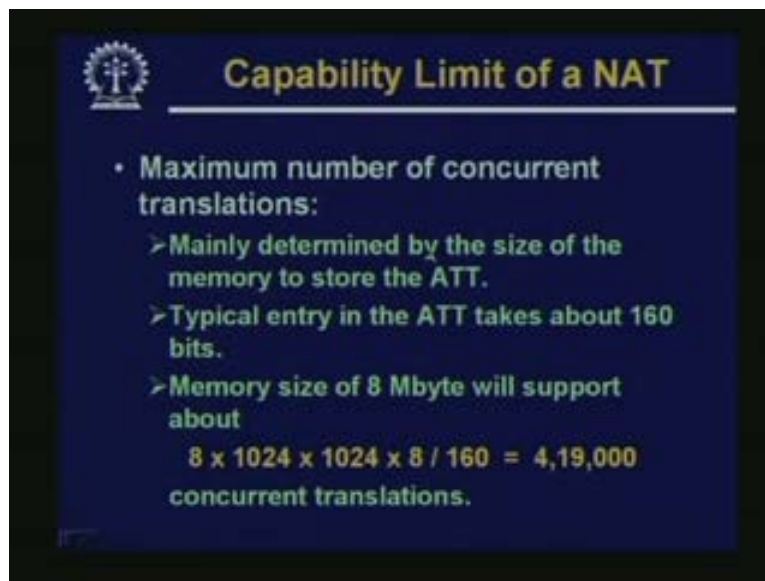
>The ATT looks like:

Source Computer	Source IP address	Source port number	NAT IP address	NAT port number
A	10.5.17.112	500	203.11.16.5	1
B	10.5.17.85	75	203.11.16.5	2
C	10.23.10.5	2480	203.11.16.5	3
D	10.22.5.118	1120	203.11.16.5	4

So a typical table this this address translation table may look like this. Well the first column is not really necessary but for documentation process it is sometimes kept. Source computer names are stored here. When a request comes the source computer addresses are stored and also the port number which the source computer was using. The NAT IP address there is only one address. So this will be the same and the NAT generates unique port numbers for every outgoing packet. So whenever an incoming packet comes it will come with this particular address as the destination address. But this port number will be the distinguishing factor. So whenever it is coming to port number 2, then through this table look up NAT will change this address to this particular address. And port number 2 to port number 75 and will be forwarding the packet to the internal network.

This is how the stable the NAT can provide address translation dynamically. But one thing you remember, this overloading although it appears to be a very powerful technique, but the only problem is that you cannot allow a permanent address through which the outside host can access. Accessing means one of your internal servers. Suppose you have one web server, you want to have a public IP address out level to it. Through NAT you cannot do that because all requests are processed dynamically by assigning a port number to it. So if you want to have this kind of facility also then you have to mix static NAT and this kind of overloading where some entries in the table will be permanently fixed. Some entries will be assigned port numbers and processed dynamically.

(Refer Slide Time: 45:38)



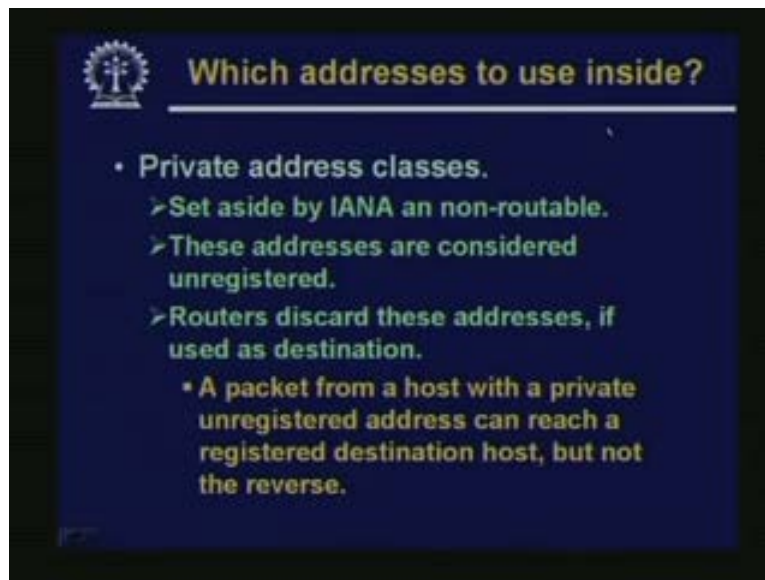
Capability Limit of a NAT

- Maximum number of concurrent translations:
 - > Mainly determined by the size of the memory to store the ATT.
 - > Typical entry in the ATT takes about 160 bits.
 - > Memory size of 8 Mbyte will support about
$$8 \times 1024 \times 1024 \times 8 / 160 = 4,19,000$$
concurrent translations.

Talking about maximum number of concurrent translations that a NAT will provide. This is primarily determined by the memory size. The size of the memory that we use to store this table. Just to show a simple calculation, a typical entry in the ATT may take around 160 bits. So if we have 8 megabytes of memory available with you, then for every request you will be needing our 20 bytes. So if you make a calculation these are the total number

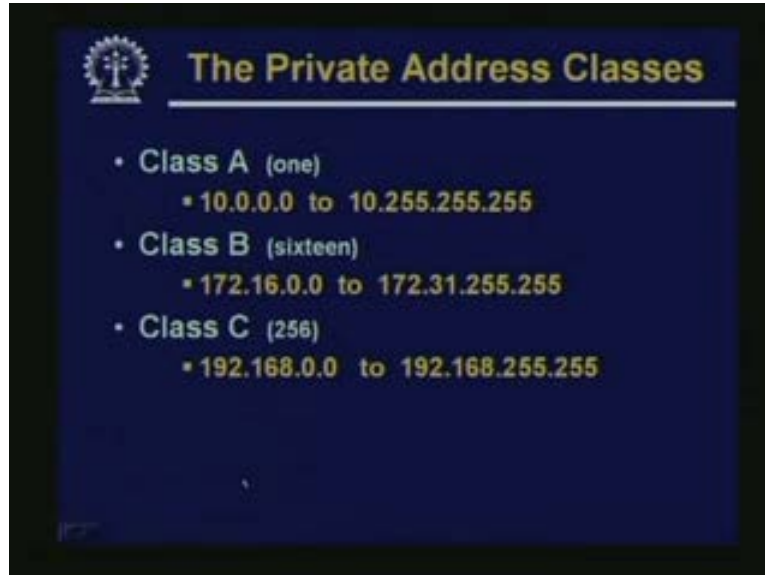
of bits divide by 160. This will come to more than 4 lakh concurrent translations. Which means you can have so many entries in the table potentially and port number will never be a constraint for you because port number is a 16 bit number. You can have up to 65000 distinct port numbers. So the number of distinct port numbers you can have that will ultimately provide the upper limit.

(Refer Slide Time: 46:44)



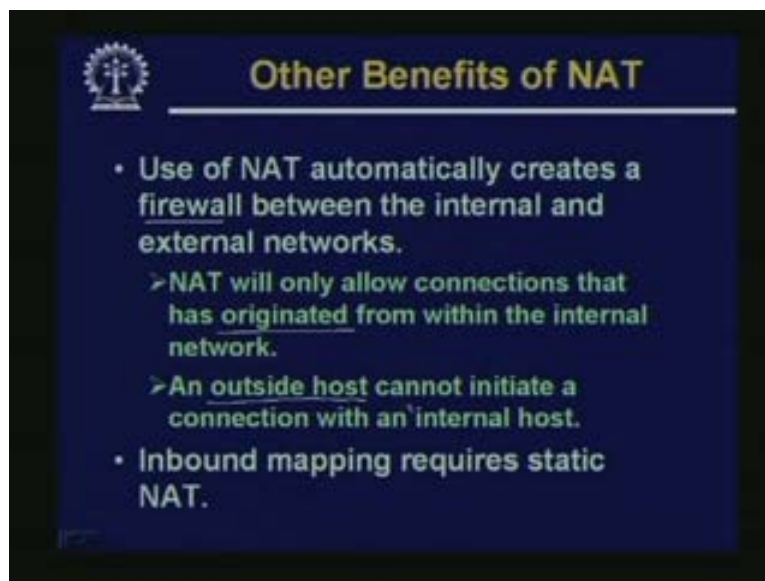
And talking about which addresses to use in the private network. There are some private address classes which also had mentioned before. Let us very quickly brush through it. This has been set aside by the address assignment agency as non-routable. Means these are unregistered routers will discard these addresses. If they are used as destination which means that a packet from a host within a private address can reach a registered host. But not the reverse because if you are trying to have the reverse connection, then the address of that private network has to be there in the source part which the routers will ignore. So from the private network you can reach an outside host but not the reverse. In order to achieve this you have to have a proxy server or a NAT sitting in between.

(Refer Slide Time: 47:39)



The private address class is just to brush up. There is one class A address 10 dot. There are 16 class B addresses 172.16, through 172.31, there are 256 class C addresses 192 68 0 up to 255. So many addresses are available to be used as private addresses depending on the size of your organization and requirements. You can possibly select one of these and use a proxy server or a NAT in conjunction with this to provide access to the outside world.

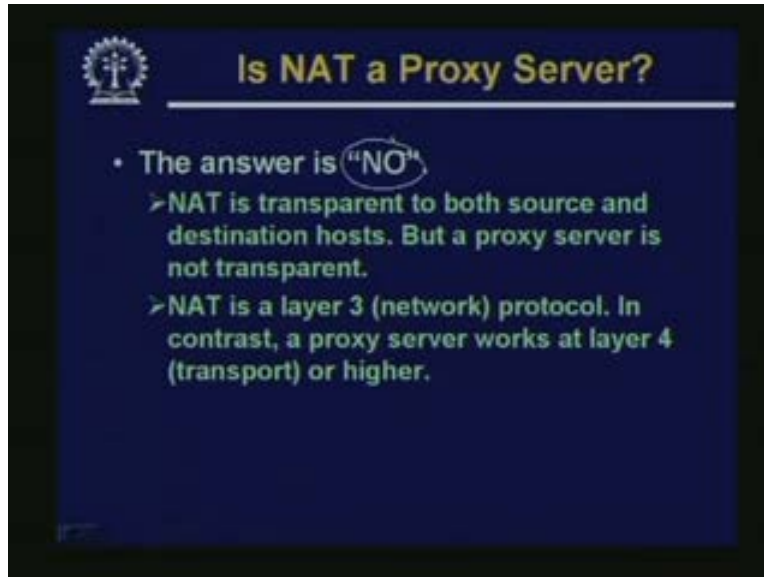
(Refer Slide Time: 48:16)



Now other benefits of NAT. Well NAT automatically creates some kind of a firewall between the internal external networks. Why? Because NAT only allows connections that

has originated from inside, an outside host just as I mentioned cannot directly establish a connection with an internal host because an internal host is having a private address and an outside host cannot directly specify that public, that particular private address to initiate a connection. So for cases where you need in bound mapping as I mentioned just for web server. For example you need static address assignment in those special cases. So in general you can have a combination of static and dynamic address mapping to have this flexibility. Some address static less signs some address dynamic.

(Refer Slide Time: 49:14)



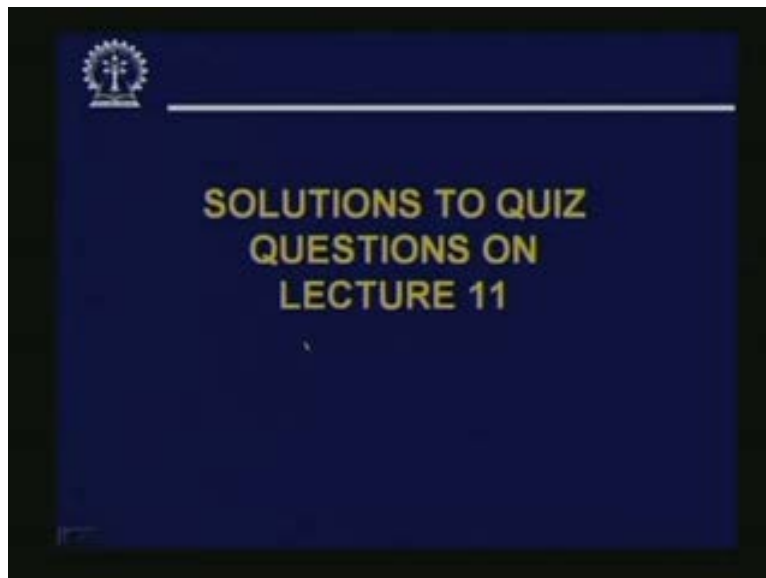
A question which sometimes arises that is NAT and proxy server are the same well technically speaking. Although they work in a similar way they are not the same. Why? The main reason is that NAT is transparent to both source and destination hosts. Neither the source nor the destination need to know that a NAT is present. But rather, in case of a proxy server. For example if you are accessing the internet through the proxy server you have to explicitly mention the name of the proxy server and the port number it is using in the configuration for your browser. That is something which is not transparent which you must explicitly specify that this is the proxy server I want to use. And the another difference is that NAT works in the network layer and a proxy server works at the transport layer or above.

(Refer Slide Time: 50:15)



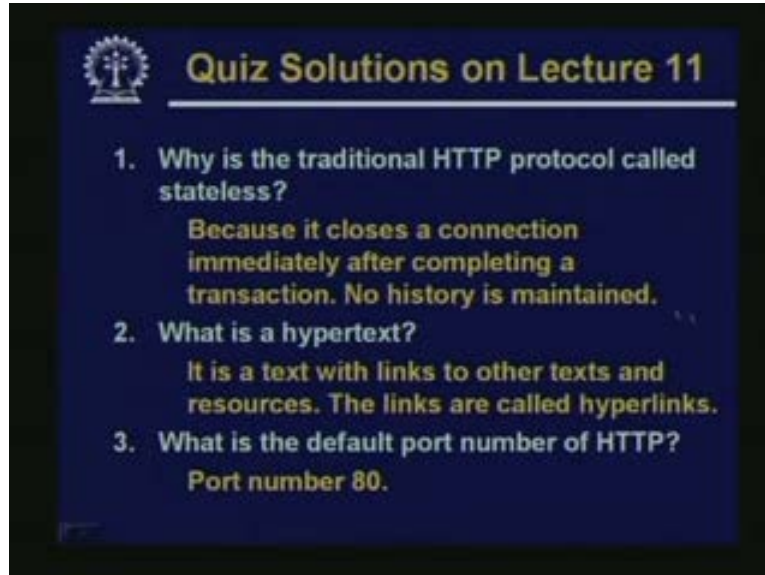
So with this we come to the end of today's lecture. We shall be now looking at the solutions that were POST for the problems of the last lecture and we shall also be presenting some quizzes for today.

(Refer Slide Time: 50:37)



So let us first look at the solutions to the quiz question on our previous lecture.

(Refer Slide Time: 50:42)



The first question was ,

Why is the traditional http protocol called stateless?

This however mentioned that it is stateless because the http server immediately closes the connection after completing a transaction and no history is maintained. But again you recall we mentioned that you can have a option either close or keep alive where you can specify whether you want these kind of stateless transaction or connection as default. Or you want to have a persistent connection or the connection will remain active over successive transactions or requests.

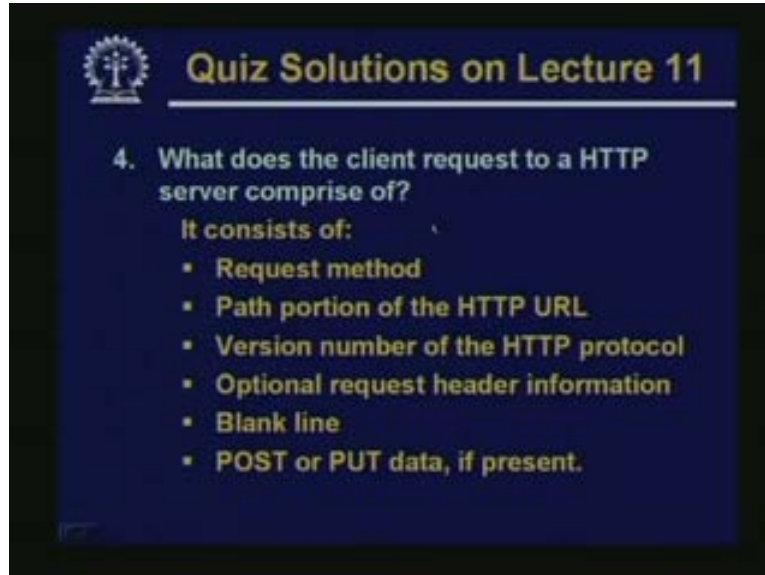
What is a hypertext?

Hypertext is a text which contains a some documents or text of course it also contains links to other texts. These links to other links are called hypertext. Now in the context of the World Wide Web the pages that we see on the browser they are mostly hypertext. They are usually typically written in html. There are other technologies also but html is most popular.

What is the default port number of http?

Which is port number 80.

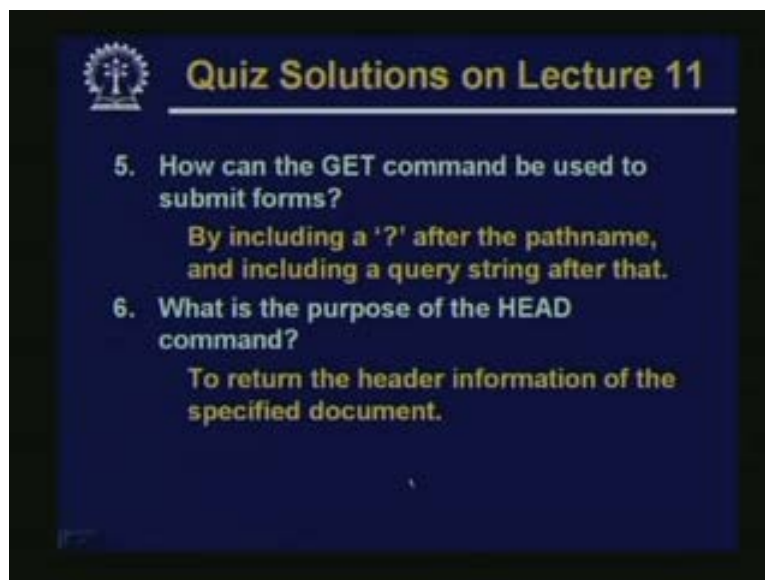
(Refer Slide Time: 52:08)



What does the client request to a http server comprise of?

It consists of several things: request method, path portion of the URL and the version number. They must appear on the same line. Depending on the command, depending on the request method you may have optional request headers. If you have an optional request header then there will be a blank line followed by some additional data if you want. For POST and put methods you need this additional data. So this all this taken to gather are the components of a client request.

(Refer Slide Time: 52:51)



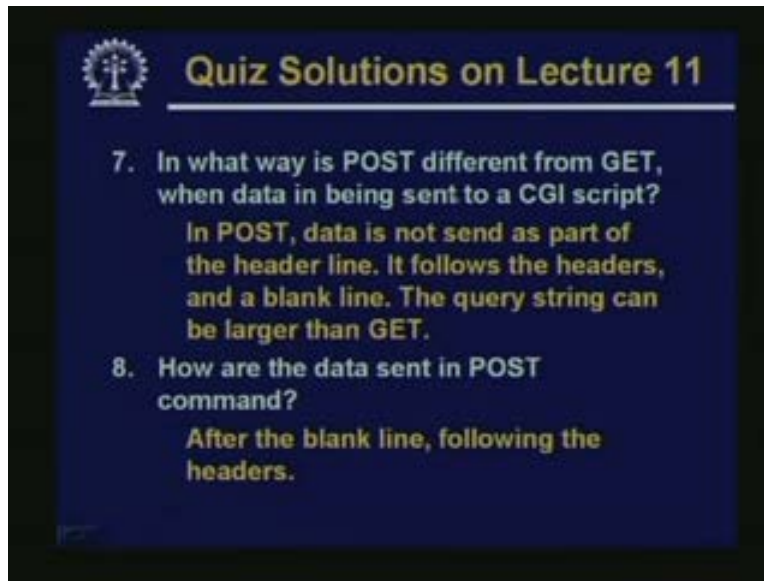
How can the GET command be used to submit forms?

Well this again we have mentioned today by including a question mark after the path name and including a query string after that question mark.

What is the purpose of head command?

To return the header information of the specified document because in many cases the client does need to have the full document, rather the header information. This may be used for a number of a I mean application cases. There is one application I can recite for instance in case of web search engines, when they try to maintain and update their database they simply look at the header rather than the complete document. So the header should contain some information which may be useful for the search engine to update its database.

(Refer Slide Time: 53:47)



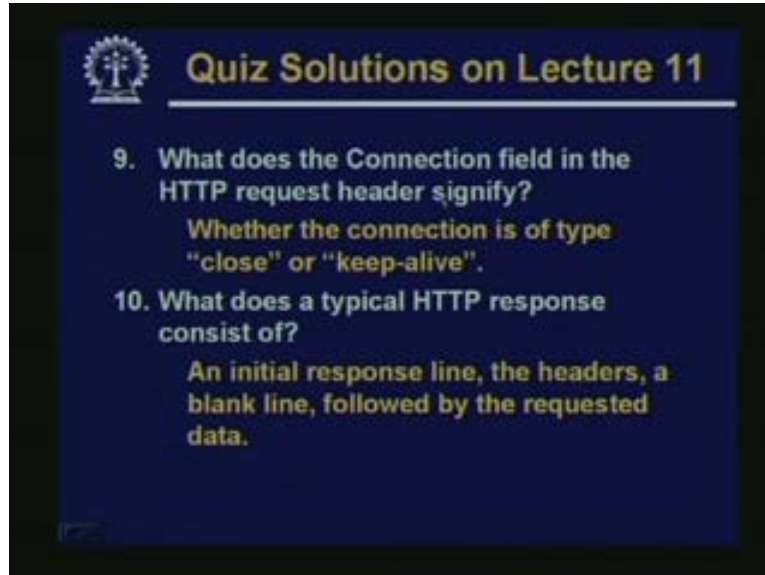
In what way is POST different from GET, when data is being sent to a common gateway interface script?

Well if you recall in POST we do not send data as part of the header line. Rather there will be the header lines, then there will be a blank line as the delimiter following the blank line we specify the query string. Now in case of POST the size of the query string can be larger than GET. For GET, it is limited to the maximum size of this string that the machine where the server is running supports. Typically it is 256 characters.

How are the data sent in POST command ?

Well after the blank line following the header lines.

(Refer Slide Time: 54:36)



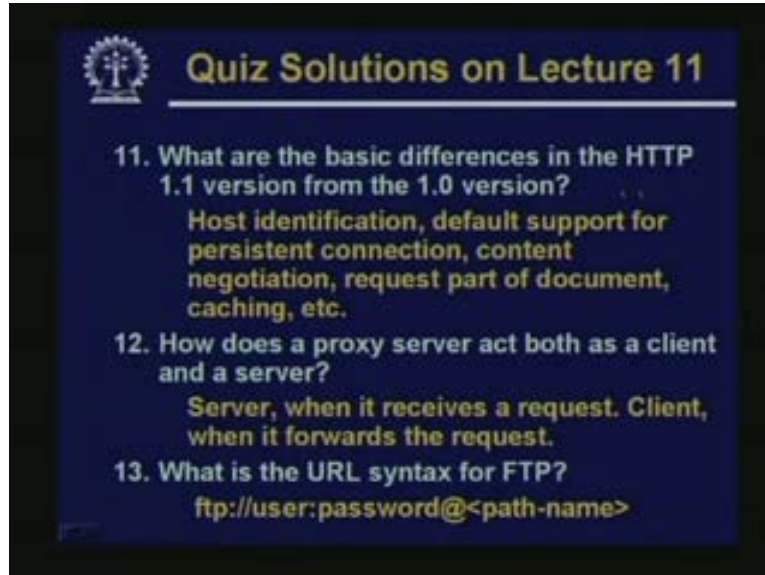
What does the connection field in the http request header signify?

Well it specifies whether you want to have a stateless kind of communication or so called stateful whether you want to close the connection immediately after transaction or whether you want to remain or retain the connection across several transactions.

What does a typical http response consists of?

Response is similar to request. Only difference is that the initial request header is not there. But in place of it there is an initial response line. Following the response line there will be some header lines. The blank lines followed by the requested data. So the remaining part is quite similar to a request the initial response line is only different.

(Refer Slide Time: 55:28)



What are the basic differences in the http “1.1” version from “1.0”?

Well the differences are host identification default support from persistent or keep alive connection, content negotiation, request a part of the document caching, these are the additional features which the version “1.1” supports.

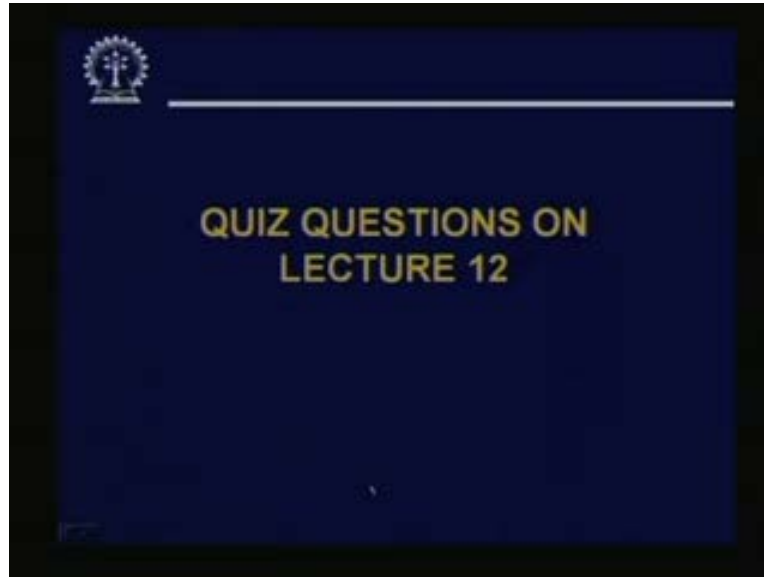
How does a proxy server act both as a client and a server?

Well it is a server when it is receiving a request from a internal client it is a client when it is forwarding the request to an external origin server.

What is the URL syntax for FTP?

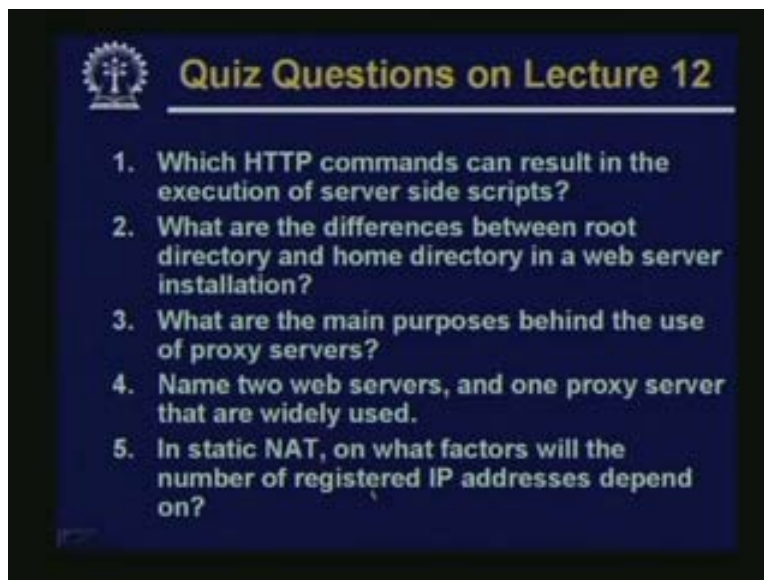
For ftp it will be starting with ftp, then it will be specifying the user name and password followed by colon. Then there is the at the rate sign symbol followed by the path name of the document you are going to transfer.

(Refer Slide Time: 56:21)



Now the questions from today's lecture.

(Refer Slide Time: 56:26)



Well which http commands can result in the execution of server side scripts?

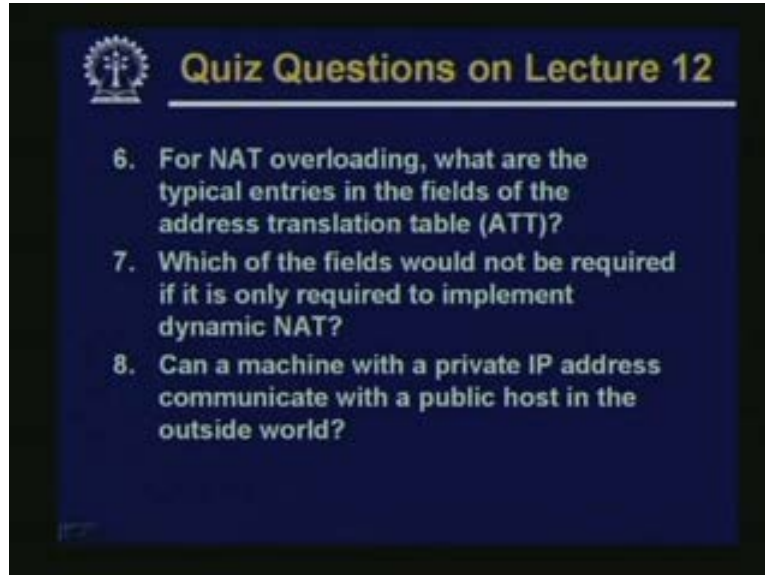
What are the differences between root directory and home directory in a web server installation?

What are the main purposes behind the use of proxy servers?

Name two web servers and one proxy server that are widely used.

In static NAT on what factors will the number of registered IP addresses depend on?

(Refer Slide Time: 56:54)



For NAT overloading what are the typical entries in the fields of the address translation table?

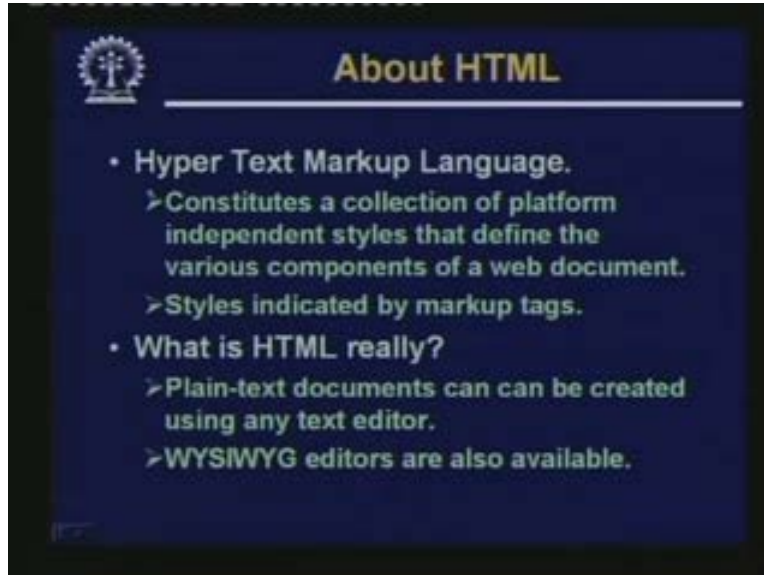
Which of the fields would not be required if it is only required to implement dynamic NAT?

Can a machine with a private IP address communicate with a public host in the outside world?

So these are questions from today's lecture. In our next lecture we shall be starting on our next module on web page design. We shall be starting with discussion on the hypertext mark up language html which is so popularly used for designing the web pages. That we shall see on our next class. Thank you.

Today we shall start our discussion on html which is the de facto language for designing web pages. Although today we have several other alternatives available with us but still html remains one of the most popular choices when it comes to the design of web pages. So today actually we shall be starting with the basic structure of an html document. What are the different things? What are the different so called tags and attributes that a typical html file contain? And we shall in our subsequent lecture what are the other features that you can support as part of html.

(Refer Slide Time: 58:27)



So the first thing is that html. The full form is Hyper Text Markup Language. So there are two components to this name one is Hyper Text; other is Markup. Well Hyper Text we had already talked about earlier. Hypertext is a kind of textual document where you can have links to other documents. In html this kind of links are allowed. So in that sense html is a hypertext document.