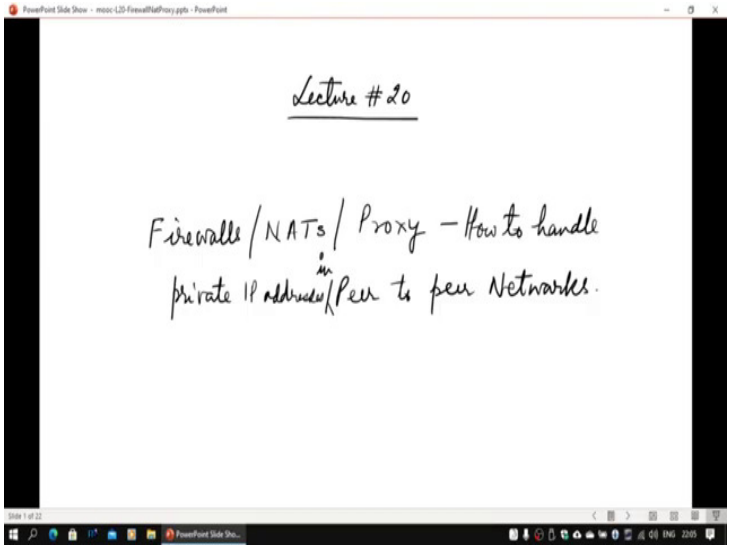
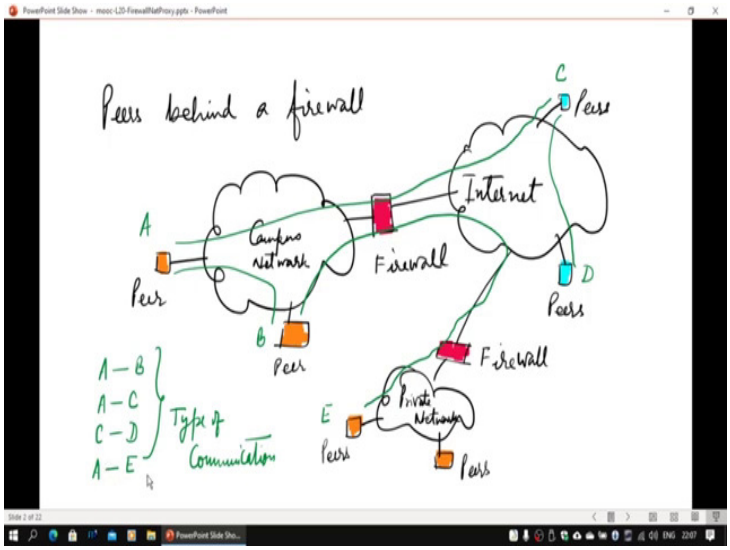
Peer to Peer Networks
Professor Y.N. Singh
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Lecture-20
P2P Nodes Communications Challenges in Heterogeneous Network Environments.

(Refer Slide Time: 0:15)



Welcome to the lecture number 24 Peer to Peer networks in our course. In this lecture, we will be talking about firewalls, NAT, which stands for network address translation devices, and proxy, the web proxy and HTTP proxy I am talking about. All of these are actually being handled and especially because these are used when you are going to use private IP addresses, and that becomes a problem with peer to peer network. Because, mostly, you can always communicate from inside the private network to outside Network through either NAT or through a proxy. How to build peer to peer clients? They will operate when they are inside those networks.

(Refer Slide Time: 1:03)

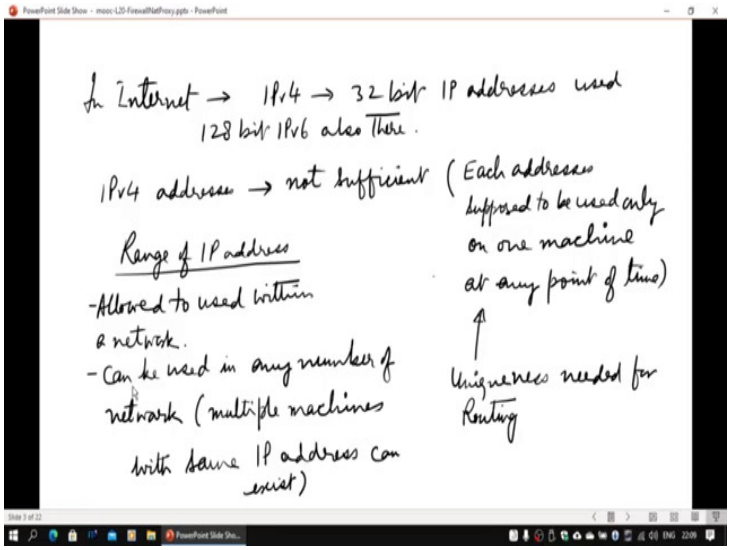


When a peer is behind a firewall, if for example, you can see there is a campus network here and then there is a firewall. This campus network is using a private IP block. A is a peer here, B is another peer, so when they talk to each other, there is no problem they can actually use their private IP and they can communicate with each other. But somehow they have to discover each other. And if for example peer A would like to communicate to peer C, which is directly on the internet is using a public IP address from inside A can actually communicate to C, but how C will communicate back to A is a problem.

A can set up either a TCP connection all the way to C if it knows C, then C will be able to communicate because he has created a path through this firewall, that is how the basically NAT will work. We will come back and we will talk about NAT after a few slides. Now there are peers who are directly on internet, they can directly talk to each other without any problem the way A and B were talking, C and D also can talk. And then there is a third scenario when peer B and peer E would like to talk, they both are actually on in different private networks and which are both actually being available through firewall.

This guy use it this guy's private IP address the internet will drop those packets. So these are basically various possible combinations of communication which could happen when the NATting firewalls or firewalls or proxies are used in the internet and which of course are there we cannot avoid them. So we have to have to build a peer to peer network which handles such scenarios. So in fact, I have listed all possible combinations A to B, which I mentioned earlier, A to C, C to D and A to E combination, or B to E combination. These are 4 particular categories which can happen; we should be able to handle this.

(Refer Slide Time: 3:10)

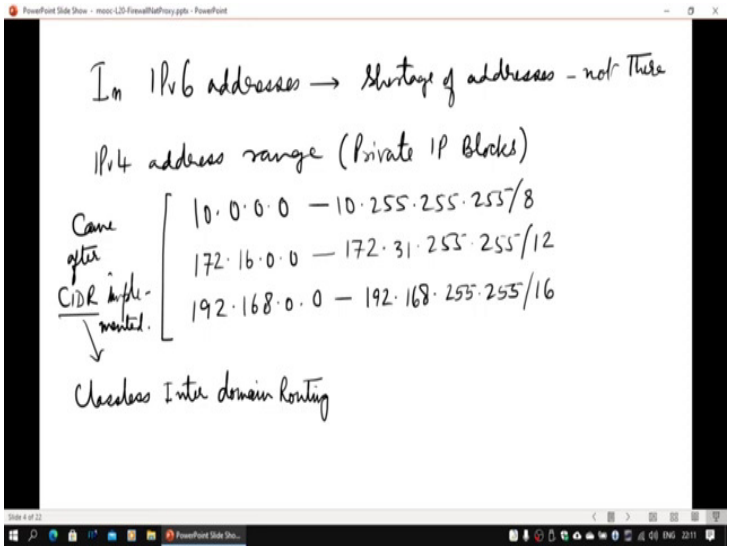


Now in the internet, we normally use IPv4, that is very common IPv6 also is in use now, which uses 128 bit, IPv6 addresses, IPv4 uses 32-bit IP addresses. IPv4 addresses from design people thought they will be sufficient, ultimate it was figured out they were actually not sufficient. And there were situations they were when they were in short supply. So people tried classless inter domain routing in between to sort out this particular issue. They also then also of course, you even after allocating all the class addresses was not possible.

People then start looking at alternative ways. So that is 1 thing they did was that they started with a private IP addresses. Now, when we actually look at public IP addresses, each address is supposed to be used only once anywhere in the world at any point of time. And that is where the problem actually came, so we need to reuse the addresses. So there are certain ranges which were defined which can be used in say IIT Kanpur it can be used, in some other Institute and some other organizations.

The only probably with the same IP address at any point of time exist in multiple places, how routing is going to happen. So since multiple machines with same IP addresses are exist. First of all the routing rules have to be updated in all routing, all internet routers.

(Refer Slide Time: 4:53)



One thing which was done was that these particular addresses, the private IP addresses were actually not are not allowed. And they need any packet which is destined for the specific range, private IP ranges have to be dropped. Of course, we do have IPv6 addresses, went there is no shortage of addresses they can be used, but still not everybody in the world has migrated to IPv6. But certainly, it is preferable to use IPv6 for peer to peer systems, because then we will not be facing this particular situation.

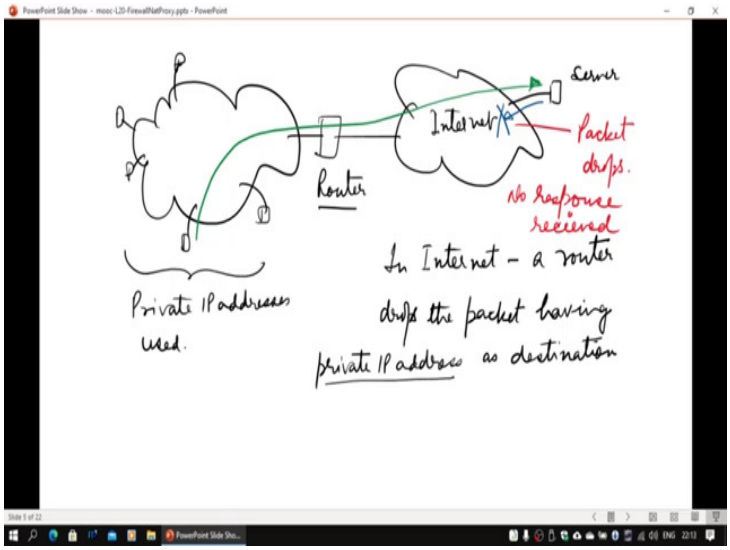
And you will be able to talk to anybody directly; you can set up a connection to him because that is a uniquely routable entity if he is using IPv6 address. Now, the three IPV4 address ranges are these 10.0.0.0 to 10. 255 .255.255 and I am telling 8 because the first 8 bit is what will tell you that network address is 10.0.0 .0. And so technically you are going to have 256^3, those many addresses are private IP addresses are possible in this and this can, these are generally used by big organizations.

They actually allocate this private IP block inside their organization. When they need to communicate to outside, then only actually they have to figure out a method so that they essentially technically do what we call network address swapping or translation they will do so that the communication, any machine from inside can go to outside world, they can also know even regulate because of this, that is where the firewalls come in.

And of course, these were actually dumb when even when your classless inter domain routing was also not able to handle the shortage of addresses. Basically, they CIDR allowed the reuse of class C addresses and they can be converted to bigger blocks. Earlier there were only 1

block size of a class C address can have at most of 256 posts inside. Actually there is only 255 hosts which are technically feasible.

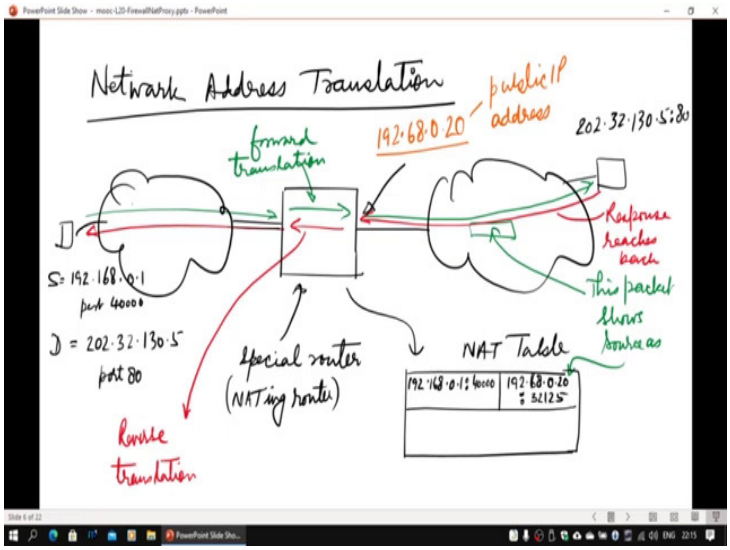
(Refer Slide Time: 7:02)



So how to access will be happening to the internet. So there will be a private IP address in the network which is being used here and there is actually various nodes, we have four nodes for example shown. They want to talk to internet, internet there is a server so they actually are being passed through router because these IP addresses are not permitted to go on internet. As I told that in internet router will drop the packet having private IP address as the destination.

For example, connection will have to be set up, so this node tries to set up a connection and then this address goes as it is. While returning this guy actually has put in his private IP address so that will become the destination, while in the form direction it was a source while the reverse it will become destination, internet router will drop the packet. And no response will be received that is what happens if you are in, this router simply is doing routing, not doing anything else.

(Refer Slide Time: 8:09)



So now we can actually implement some what we call network address translation for solving this particular issue. So, for example, now this source is having a private IP address. So, remember this is actually from the same block. So, this guy actually transmits the packet. So, this guy can this is going to be used as it has been told in the earlier slide this will be dropped at the Internet. So, what is done is we actually use a special router; we call it a NATing router.

So, this will replace this address by translating this forward when it is moving forward the source address and source port will be modified by a public address, maybe this public address will be 192.168.0.20. So once this is done, the packet will reach here. So, this server will see the source addresses this address is not 192 .168.0.1 but it will see 192.168.0.20.

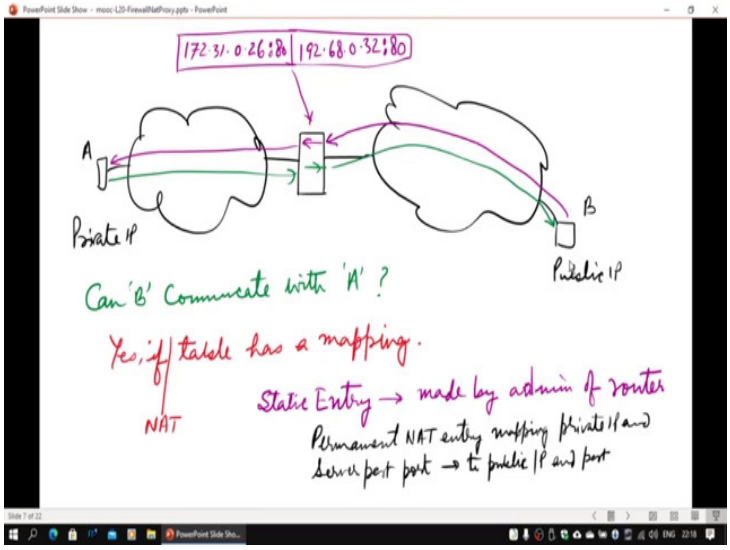
That will put that address as a destination so this internet will not drop it and this packet which is moving back you will be showing, in fact in forward direction in between this 192.168.20 as the source address, while in the reverse it will become a destination, the packet will reach here. Now this guy has to do a reverse translation. So, it has to remember that from the source IP address and source port, I translated to the public IP address and public port. So this has to be maintained in the table. There is going to be a NATing table because of that.

So, this Netting table will say that the packet which came from the source IP address port number was translated to this; in the reverse you do the same translation again. It is done injected into the private network which will then route this private IP and it will go to the destination. In fact, you are using the IP address but the outside world sees you as this

192.168.0.20. This is the mechanism which is used in such a scenario when this private IP block is used.

So a lot of peer to peer network will actually be seen as coming from this one particular IP address, but actually they are behind the NATing router and they may actually be working on a different IP address. But we need to just ensure that communication does happen uniquely without any disturbance.

(Refer Slide Time: 10:30)



Now, question is if A is there, A is not communicated to anybody and there is one another node which is outside and this would like to talk back to A, so this would like to make a setup a connection to A, can you do it. If it tries actually sending a packet back to her what is going to happen. So, it will not go because there is no entry in the table, A will need to have a mapping. So, in our case it was done because A tried to set up a condition that time entry got created.

And that entry will remain for some time, there is always a timer associated and within that time if the reverse packet will comes in, those packets will be allowed. So A's only job was at the time to keep on transmitting in the forward direction, so that this entry is maintained. Every time a packet passes through, timer is going to be reset in the mapping or Nat table. So static entry will be required and this is can mostly be done by the network admin of the organization. So, you can make any static entry so server actually is running inside.

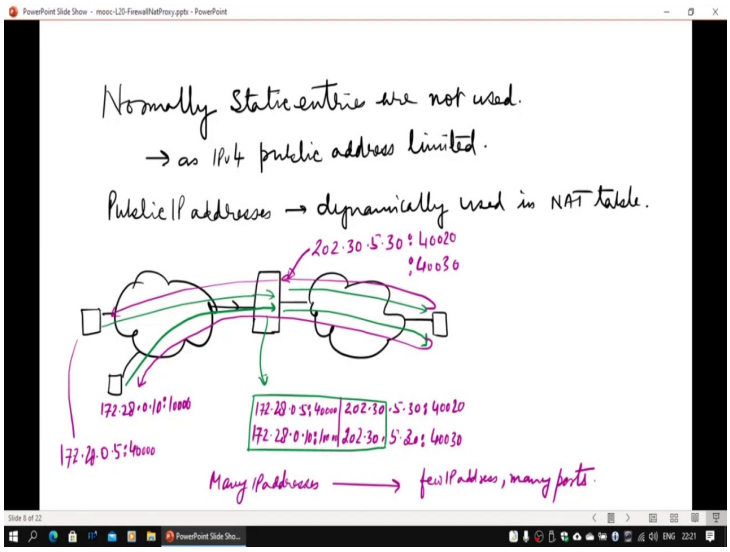
But you have to even say this is going to, if it is a web server, it is going to run NAT Port you will receive on NAT port so you will define public IP addressing public port is fixed and then

mapping to an internal IP address and port is going to be fixed. So that is how you can actually run servers even inside. So, you require a public permanent NAT entry mapping the private IP to the server port address, private IP and private port which is the server which is running the private the private IP, the public IP and port and this will be maintained in the NATing table.

So, this is how it is going to happen. In that case, if B communicates, A will be able to receive it and there has to be mapping which is going to be something like this, it will say that for this IP address,port number communicate to this. But in a web server request outside world will see this server is 192.168.0.32 but actually it is 172.31.0.26. And of course whenever B is going to make a communication there is a entry which is there a static entry stop. This IP address IP address or B is public IP address.

So it can always respond back which will just pass through, there is no NAT entry need to be created for this reverse thing. So this entry is going to be fixed and this is what is going to be used. So, while this packet is going it will be shown as if it is coming from here to this node B in this internet.

(Refer Slide Time: 13:41)



So normally we will never use a static entry these are being created dynamically. Because public addresses are very few. So, if within your organization, within your campus, I have say 10,000 machines, I do not have 10,000 IP addresses, I have very few. So these are mostly going, need to be allocated dynamically for various machines which are inside the campus so

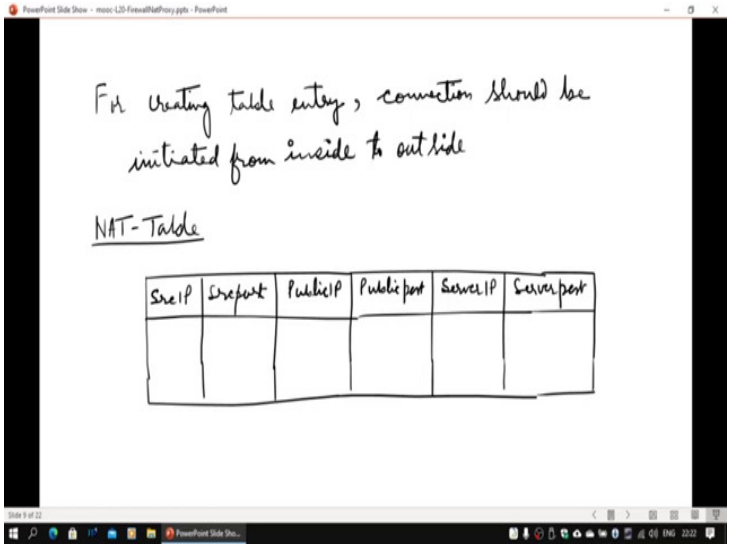
that they can communicate to the outside world. So, these are all dynamically allocation, static entries are done only for some servers, which are of important nature.

And it is kind of very precious resources; it is not easy to get it. So, here you can actually see a node actually is communicated, it is being communicating to the outside world. And there is a response which is going to come accordingly back to him, and then this response is coming to the first node. So, this was actually going with 172.28.0.5: 40,000, it is called 40,000, 40,000 is a port. There is a typical way that we actually write IP address and port number; we just separate them by a colon (:).

So, this entry is coming. Now another, actually machine talks to the same server what is going to happen, it will be given a different public IP address and port number pair. So, you can see that for 1 it was given as 202.3.5.30: 40,020 for other 1 it was given 40,030, only port numbers has been changed. Not there will be many machines with many public and many private IP addresses but there are very few public addresses.

So, normally these many private IP addresses will be mapped to few public IP addresses but many ports will be used in this case. So, the ports which are going to used will be much more in number then what are used here. So, the product of number of few IP addresses into ports. So, this and multiplied the many IP addresses into whatever the ports, directly will be equal. So, we are essentially now using multiplicity of ports where the IP addresses are few by using this dynamically configured NATing table.

(Refer Slide Time: 16:14)

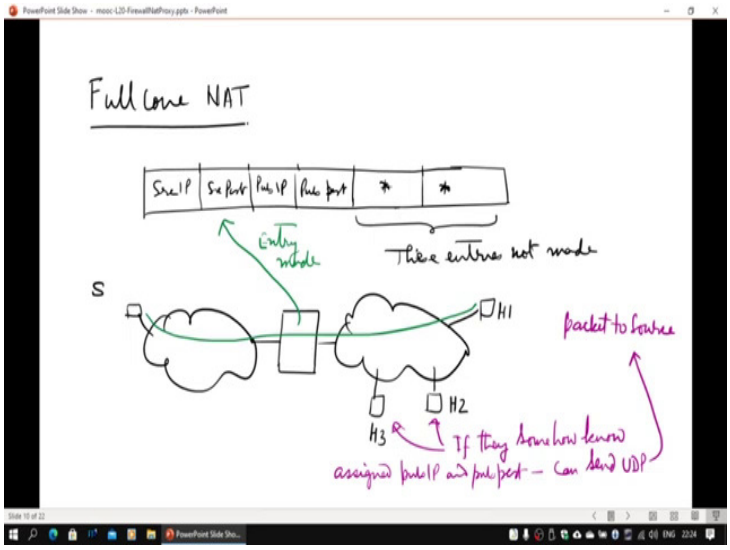


Now, one of the important thing is that in NATing table the entry can only be done when the connection is going from inside to outside. Because from outside when a connection comes, the outside guy will not be knowing what is the inside source IP address and port number, that is a private IP block it cannot even communicate to that, it can need to have some public IP address and port. So these are dynamically allocated whenever the first connection is done and normally there will be a timer, so I should actually keep a third field here which will be a timer here.

So, whenever an entry gets created, there is a timer, whenever there is a packet which goes into out, the timer is going to be reset. So far that entry is there, packet from outside to inside will be permitted as per this table. This table normally will have 6 fields source IP address, source port, public IP address which is being allocated by NAT and public port also located by NAT, there to whom you are communicating its IP address, server port and then there is a timer.

Now we need to fill up all the entries whenever the NATing has been done. So based on that how many entries you actually fill up, there are various kind of NATing mechanisms where they have they have been classified into various categories.

(Refer Slide Time: 17:45)



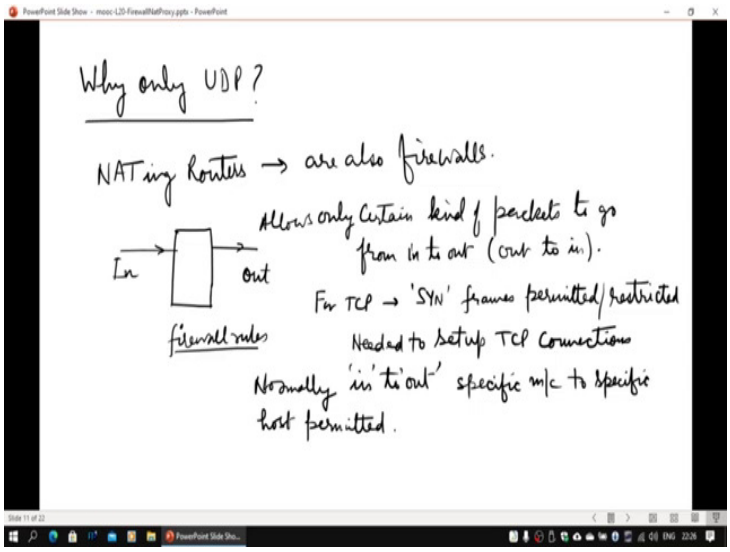
The first one is a full cone NAT. In this case, whenever a packet from inside to, goes from inside to outside. We actually make an entry for source IP, the source port, the correspondingly what is the public IP being allocated by NAT and allocated public port on the public interface side. But, these remaining two entries, which is a destination host and

which is the destination port that is not being maintained, that started. As a consequence, when the reverse side the packet is coming these cannot be verified.

You only just look at if it is coming from this public IP address and public port, replace them by this the source host IP address and port numbers you do not bother. So, this kind of NAT is known as full core NAT stop now the way it actually operates is very simple, when a request goes from source to host 1 (H1), the entry is going to be made. So, these four entries will be there this will be *, *.

So H1 now can respond, so when H1 comes back it will not be checking for H1's IP address and port, it will just look at what is the public IP address and port for which the packet is coming and it will do the address swapping and it will go into S. But because I am checking the host IP address and port even host 2 (H2) and host 3 (H3) can also communicate back to the same source. They somehow need to just figure out good for this source, this is the public IP address and public port which is there and they can then actually dump send the UDP packets back to the source. They cannot do TCP because TCP works in a slightly different way. TCP you have to set up the connection in the beginning.

(Refer Slide Time: 19:35)



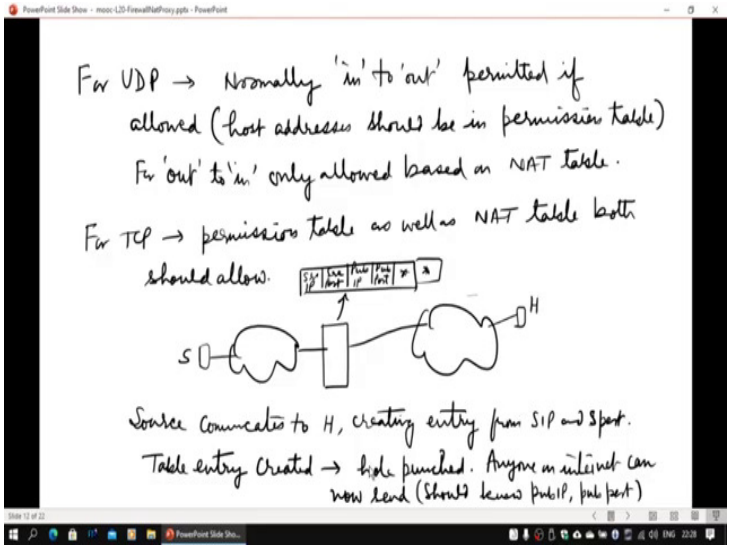
So why use only UDP? NATing routers normally are also firewalls and they allow only certain kind of packets to go through from in to out or from out to in. These are going to be firewall rules. So, we use IP tables, for example in any Linux machine for doing this job. For TCP frames normally there is a sync frame which is transmitted. So, when IP packet comes in, we actually look into the payload these firewalls look into the payload, whether it is a TCP

packet or UDP packet and they analyze the UDP or TCP header and based on that take the decision.

So, we can say the sync packet SYN can only go from in to out. That means connection can only be initialized when somebody from inside is trying to communicate to outside. So normally sync from outside to inside will not be permitted because of the attack issues. Or of course, it can be done but in that case you require static entry, otherwise this guy will not be having entry it cannot communicate back.

So, only for a static thing it will be done for in to out, normally will be the connection and we can actually have regulations like only part of the network can communicate to another part of the internet, all those kinds of rules can be done. So basically, default is you deny everybody then you start allowing rate specifically for to be on the safer side to be more secure.

(Refer Slide Time: 21:14)



For UDP normally in to out to be permitted because out to in when it comes, of course it does not know whom you have to pass it on unless there is a entry in the table. Out to in will only be permitted, will make sense if there is a NATing table entry. So, it will only be permitted if the NATing table entries there otherwise it will not be. For TCP permission table as well as the NAT table, both should actually allow the things to happen, then only it will be possible to set up the communication.

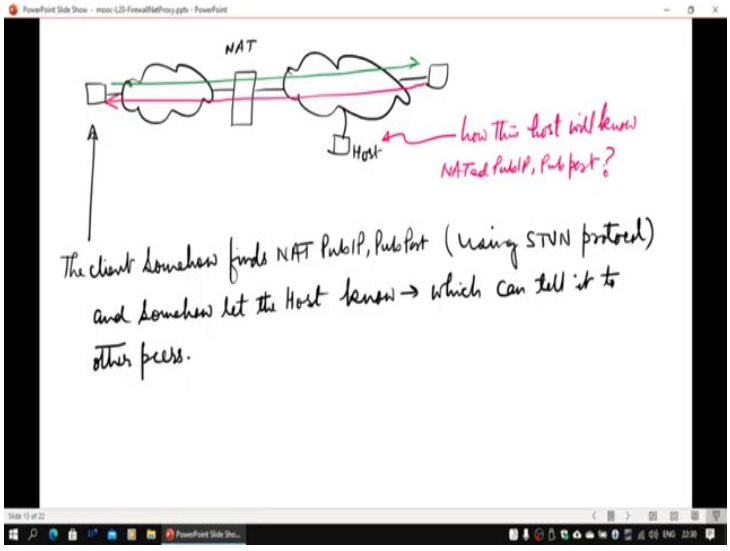
TCP also is actually permitted mostly NATs. It is not that we actually for example in IIT Kanpur using a NATing router from inside when I am setting up a TCP connection. So, sync

is permitted only from in to out but when sync is being forwarded, that time NAT entries also get created. In the reverse when the packet is coming in then those TCP frames or IP packets will be permitted after viewing into this and will be passed on to the source.

And if it is a full core NAT somebody else can also set up a, then they sync entry back but that sync will not be permitted because there is also a firewall rule, we say sync cannot come from out to in. But nobody from outside can set up a packet, but with UDP this will be feasible that is why in the previous slide, the previous to previous slide when there were 2 hosts they were also able to communicate back while from inside only for one particular host the UDP packet was sent.

And it was using the same table which was created for our connection was used by the other two hosts for out to in connection. Now whenever the UDP thing is done that actually means you are creating a hole and through this hole even other hosts can send the information back and in fact we call it a UDP hole punching because of this.

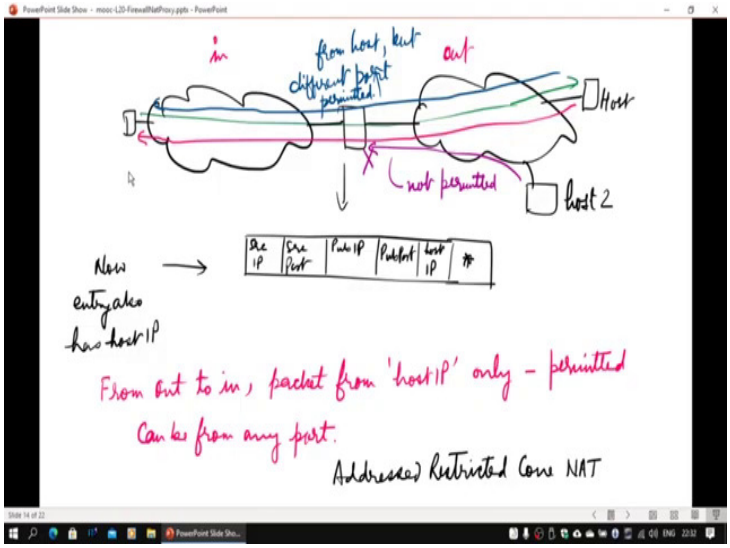
(Refer Slide Time: 23:16)



Now, the question is how this host, inside machine has client has communicated to some server or some other host will know that what is this NATed IP and public IP and NATed public port for this particular client. There has to be some mechanism by which this has to be done. One way is that we can actually have a protocol called STUN protocol. The client will now talk to server and server will figure out what is the corresponding public IP address and port from which the message has come.

It will actually put this in the payload and send the message back and when the message comes in the destination address will be only replaced by this address, but the payload will remain as it is and payload contains this address. So this guy can actually figure out what is the public IP address and public port has been allocated by that to me. And of course, this has to somehow tell this particular host that which particular NATed IP address and NATed port has been allocated to me, then this host can actually communicate back by this NATting router to this client. So this is actually slightly tricky the but that is the way it is.

(Refer Slide Time: 24:47)



Now, let us look at the second case. So previous one was full cone NAT, now we have in network we have an outside network, inside is using private IP block. Now that new entry, we will also be storing what is the host IP address to whom the connection was set up by the client from inside. I am going to make this entry but throughout I am not bothered. The moment you do it now there is a problem when this host come in comes, sends a packet back I am going to check from this host IP the packet is coming.

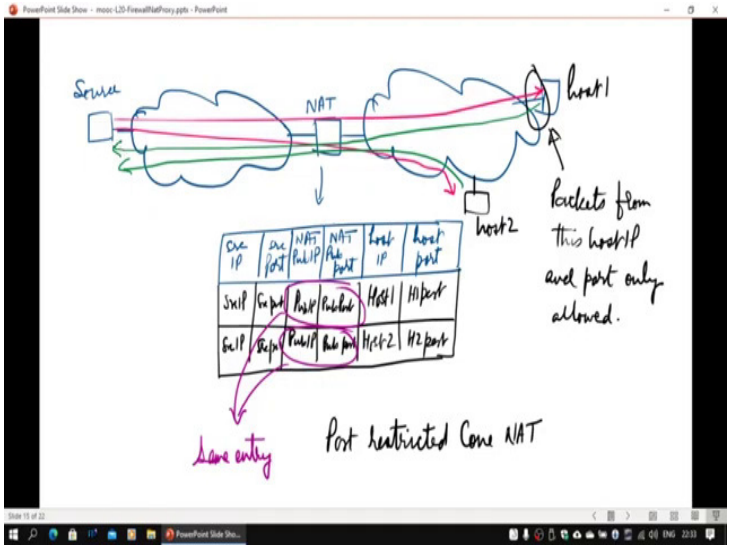
If it is not coming from this host, it can be coming from any port it does not matter. But it has to come from the same IP address if it comes from the same IP address I will permit, otherwise I will drop. One sequence this host to cannot send the message back to this client unless this client decides I am going to send a message to this host 2 for which may be a different public IP address and port can be allocated or maybe the same one.

And when it sends this information to this there is going to be another entry which is going to contain this host's IP address and corresponding public IP and public port. And then when it

is going to communicate back then communication will happen. So only those hosts can send you the information, those IP addresses can actually communicate back to whom you have communicated, nobody else can do it.

It is not a phone cone NAT, if there is a restriction, we call it an address restricted cone NAT because of this. Address has been restricted to the one to whom the inside client has communicated. This is the second kind of NAT so this is more restricted.

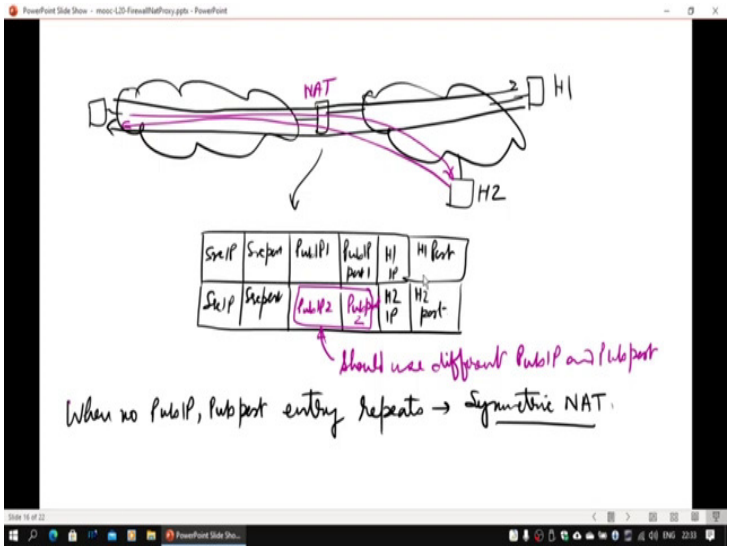
(Refer Slide Time: 26:40)



We can further extend this, we can say now not only I am going to actually put host IP but I will also put host port also in the table. Now it is very straight, if you need to communicate to host 1 to whichever port you have sent the signal sent the information, only that port can communicate back, no other port. Even other ports on this host cannot come. So that becomes a port restricted cone NAT. So other hosts also can do the same and interestingly when a source is communicating to other hosts, it can use same public IP address and same public port.

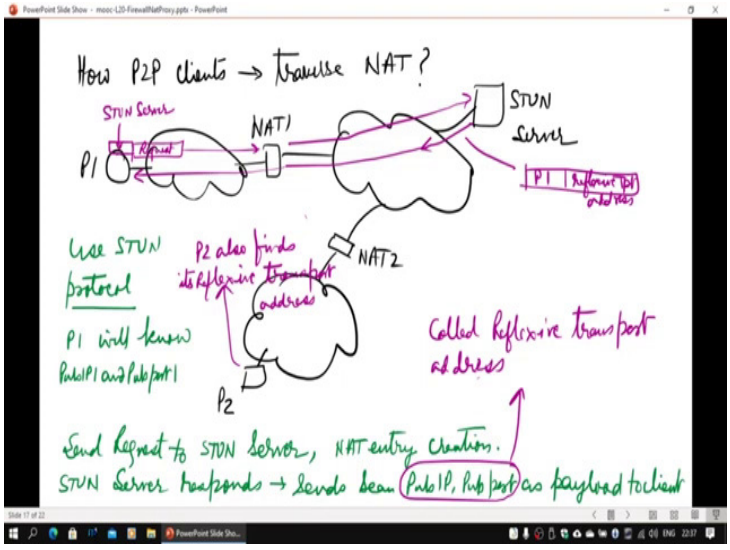
But this kind of a reuse which can happen, same entry can be there but these entries will be different. But only problem is this guy host 1 and host 2 both are actually now contending for the same public address, public IP and public port and inside if there is a common buffer there is a common buffer for the source. They may start competing for the buffer for this particular source. So, people actually even have built NATs which are even more restrictive.

(Refer Slide Time: 27:46)



So, these NATs are essentially only condition which you will do is that these public IP addresses and public ports cannot be reused. If it is used for one host H 1 something else has to some other pair has to be used for H2 at least one of them have to be different. So, such kind of are known as symmetric NATs these are very difficult to actually there is a UDP hole punching cannot be done, because other host cannot communicate back to you. So, symmetric NATs are much safer and these are also being used but these are also required larger table and larger decision making process whenever the packets are passing through.

(Refer Slide Time: 28:25)



Now, how the P2P clients will traverse through these NATing routers, so that remains a challenge. So, let us look at the ways and means by which this can be done. So, let us look at

host P1, now host P1 is now communicating to the internet it is having a NATing firewall, we can do it through a STUN server. So, STUN server is very simple that P1 will send a packet to the STUN server. STUN server will actually now look at, we will see this IP address and public port, public IP address and public port as the source of the packet.

It will just put that thing in the payload, we call this address as reflexive transport address. So, that is what is going to go in the payload and it will be sent back to P1. So P1 will be now having what is the address which it has. Same way P2 can also find out this reflexive address which is the address allocated here. So once this is there, somehow this P 2 if it knows what the reflexive address is for P1 and P1 knows, what is, the reflective address of P2, so any one of them actually should need to.

Only P1 knows about what is the P2's reflexive address, it can just communicate the message to this and this will get routed to this particular IP address. And since this has already created the entry, NATing entry via STUN server and this is a full cone NAT. So, the request this particular address will be seen as a host. So, this request will be permitted to go into the system. Even if it is not called restricted cone NAT, in that case the both of them need to know each others.

At least one of them, actually for example, this guy knows this address so this guy will directly communicate to this address but the entry will not be there, that is a problem so it will not be allowed. So, this guy needs to communicate to this NAT1, this guy need to communicate to this NAT2, whose public IP and port so and both of them will now the entry and after that if they start communicating that will be permitted, because now both of them have the entry in their NAT tables.

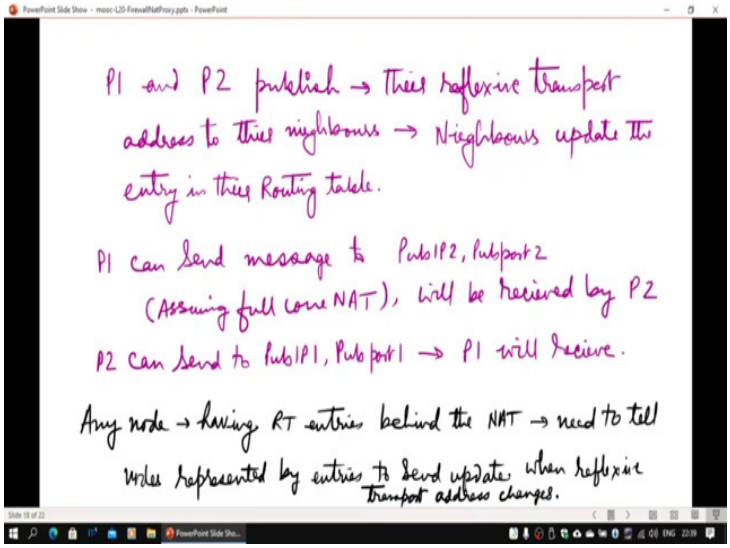
So, even with port cone NAT it is possible to set up this peer to peer connection between 2 peers who are behind NATs, both of them are in private IP, they do not need to actually have some third party router who route their messages to each other. So, this is one way to traverse the NAT, this actually works beautifully with full cone NAT, address assisted cone NAT as well as with port restricted cone NAT, all of them actually can be taken care of.

Both P 1 and P 2 need to find their reflexive transport addresses through a STUN server using that and then somehow they have to communicate to each other. Which can be done through messaging? Because, they can actually keep on pulling the information from because they can also set up a connection to somebody outside. So, somebody who is there on the public

IP and he is holding their addresses they can deposit their addresses and from there the action can happen, from their user ID to that particular public, node IP to public IP address they can always put in in a repository.

So, this guy also can do and then they can find out from user ID the node ID and from node ID what is the corresponding reflexive transport addresses and they can do a peer to peer communication. Only thing these 2 guys need to ensure that the UDP hole is punched in both the NATing firewalls because by periodically sending the packets out. The moment they stopped doing it they will be unreachable after that.

(Refer Slide Time: 32:08)

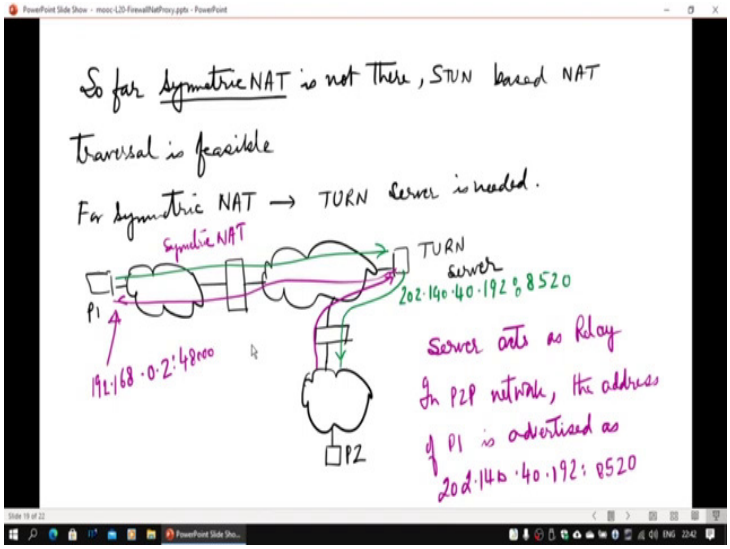


One of the ways is that both P1 and P2 will publish their reflexive transport addresses as their address to the neighbors and neighbors will update their entries in their routing tables. In fact, the guys who are holding P1 and P 2 in their routing tables, now they these P1 and P 2 need to keep on telling them what is the corresponding reflexive transport address they have. These guys who are actually holding P1 and P2 in their routing tables can keep on updating their entries and they can communicate that to them, so they can initiate the connection to P1 and P2.

So P1 will send message to the other P2's public IP address and port as I mentioned earlier and P2 will do the same. And any node having and every node needs to do something look into routing table and all the entries in the routing table which are behind NAT just keep on asking them to keep on updating their current reflexive transport address. These reflexive transport addresses can also change, they are not static.

So now there is something else, next time you send the by the time that entry, then router decided to allocate you something else they can be dynamically changed. So these need to be updated otherwise your network will not operate. So then you will be just going out of network for some point of time and they will ultimately come back in of course.

(Refer Slide Time: 33:47)



And so far, we have assumed that symmetric NAT is not there. How to take care of symmetric NAT? The STUN will not work actually in that case. STUN only allows the peer to figure out what is his reflexive transporters and some are communicated to other guy and they both can talk. But in symmetric NAT, the problem is that you have to essentially, the only way it can be done is when both of them as I mentioned earlier that P1 is going to talk to P2's reflexive transporters and P2 also does for the other guy simultaneously.

And they both get the entries done. And since they are in the reverse path, it is going to come through a different, the same transport, same basically public IP address and port numbers as they have sent the packet to, they can accept but the problem is here is that in a symmetric NAT, they have to identify what is the public IP address and port number, they cannot do this actually now.

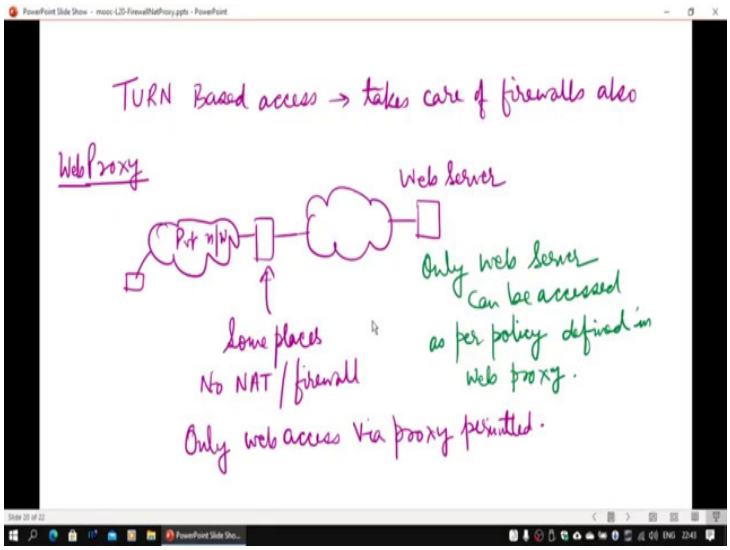
Because for that they have to go to some STUN server and then the STUN server for that particular IP address and public IP address and port number is located, this cannot be used for some other communication. So it cannot be able to communicate to some other hosts because it will be a little bit different public IP address and port number by the NATing router. It actually fails in case of symmetric NAT. In that case we need to go for TURN server.

Basically it is a relay node. So NATing traversal can be done through a relay node. So in this case the TURN server will be there and you will create a connection to the TURN server and TURN server on your behalf will allocate you a public IP address and port number which is on your behalf and this number will be told to you. So TURN will now keep a record of this particular public IP address and port number has been assigned to for P1. So whatever is the public IP address and port number coming from the symmetric NAT, so this is the mapped to this.

So when you are going to tell to somebody you will be telling somebody else as this is your IP address 202.140.140.192 : 8520. So when P2 wants to communicate, it will send on this particular IP address and TURN in turn will now reflect it back to the P1, so it is going to act like a reflector. Now, that is a pretty heavy load if there is a video or something is going to now be a lot of relaying on your behalf. So this is not that easy. So normally this is a paid service which is done, it is not a freely available thing.

Now you can find a lot of STUN servers which are freely available because they do not put much load, just go and query what your corresponding public is IP address and port number? This server acts as a relay on your behalf. So that is how the symmetric NAT is actually handled.

(Refer Slide Time: 36:52)

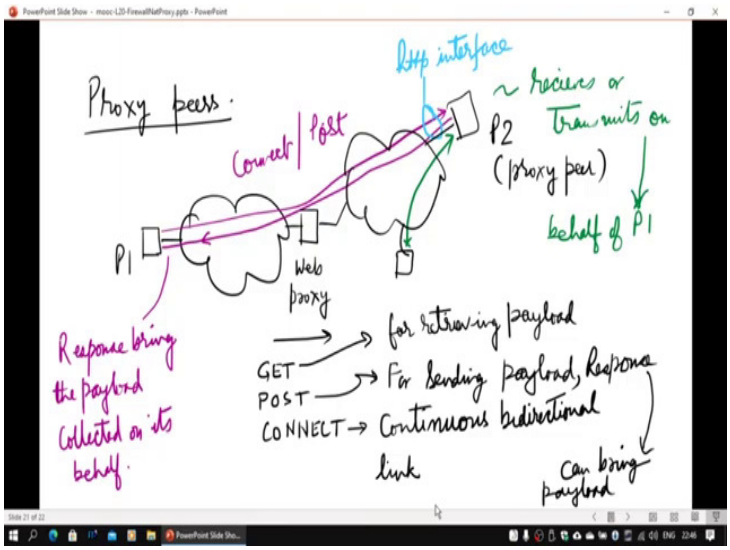


And of course, it also does take care of the firewalls automatically because connection is being initiated from inside and on that setup connection only the reverse path traffic is coming and forward is also going on that. So there is no need to worry about what is

permitted what is not permitted. So now other way things can be handled when for example, there is no NATing router if there is no firewall. So they say there is no traffic directly we know UDP or TCP packet can directly go from in to out.

So there is only going to be a web proxy. You can make a HTTP connection from your browser to the web proxy, it will do on your behalf a web proxy HTTP connection to the outside server and the return path information will come back. For GET, this is fine, GET only will send the URL and reverse they will get a payload. I cannot use GET actually for because I also need to send as a peon needs to send information to the peer who is outside. In that case, then there is a problem. Let us see how this can actually get solved.

(Refer Slide Time: 38:07)



In fact, on the outside is that shown in the previous slide, have shown in the web server. So I do not need a web server because I am not transacting information, which was going up normally in a UDP message or a message or a TCP thing. But now this, this is going through HTTP transport. So there is a web port which is available. So HTTP protocol has to be implemented by some peers. We call these peers as proxy peers, they are a special category.

So, I have to search for some proxy peer and essentially communicate to him. He will now make an entry that for me he is acting as a proxy. So what I will do is I will now use normally a connect. there are actually three modes of communication in HTTP, GET, POST or CONNECT. GET is not used because you can only retrieve the information, with POST you can display something and in response you can retrieve something.

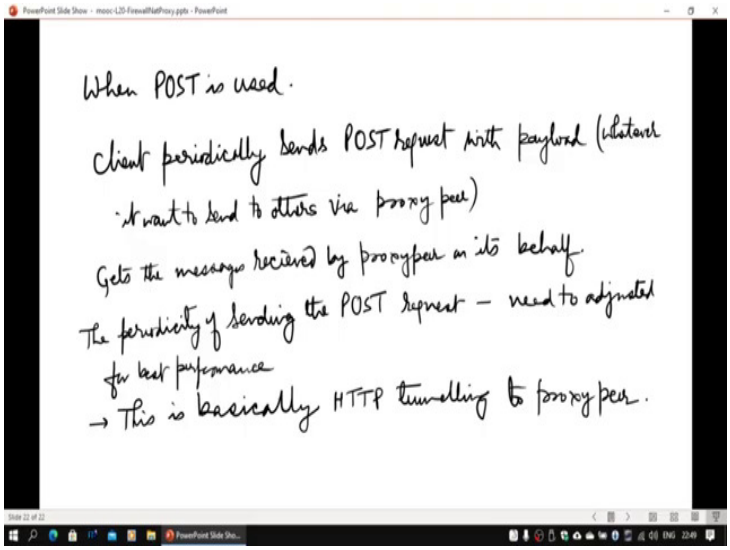
Only thing we have to have a timer. I do it now, then after 1 minute again I will do it. So by the time whatever is collected by him, collected by me I will send it to him whatever is collected by the proxy will be sent back to me as a payload. But, then this periodicity also needs to be I need to keep on adjusting so that there is not much backlog when the queue gets created.

So that proxy peer will have an HTTP interface through which I will be connecting. And I will be now publishing the IP address and port number of this proxy peer as my IP address and port number. We call it a, I will actually tell this is a proxy IP address. One of the guys who want to send something to me will send it to him. It will be encrypted, so he can decipher it, but he will route it back to me through HTTP. This is technically HTTP tunneling, which I am actually doing. There is somebody else who is working on my behalf, who is on the publicly on the internet.

So the worst case scenario, this is always going to work this how actually earlier even Skype used to work. And now we have also this CONNECT mode, CONNECT mode creates continuous pipe, so I can I need not do the timer wasting I can whatever things come I can keep on transmitting and this guy can keep on sending whatever is coming for me, there is a contrast pipe. This was basically been created for Web Sockets or for direct to peer to peer transaction on a continuous basis, streaming basis.

For that so now we can even use this, this far more efficient than doing POST and then adjusting timer again and again sending the request, setting up a new TCP connection. So that will no more be required actually.

(Refer Slide Time: 40:58)



And of course, as I told, whenever the POST method is used, we need to periodically send the POST request, so that I am going to send with that POST request the payload and whatever guy wants to send back to me will come as a payload that to me from the proxy peer. And then the message I received, the proxy peer will receive the messages on my behalf from other peers and that is what I will get from him in the return in the response. Only thing it requires now is the time out, after what time I have to send a POST request.

This needs to be continuously, dynamically adjusted, so, I do not have the buffer overflow at both the ends. So this will be one additional requirement which is going to be there. So in fact, as I told earlier, this is basically HTTP tunneling, being done to the proxy peer. So in today's lecture, we actually have looked at how the private IP problem can be resolved in a peer to peer systems, almost all peer to peer networks even like Skype or everybody is now actually operating either with a NAT, UDP hole punching.

And worst case, they then switch over to TURN or to HTTP tunneling, all 3 modes are normally supported. But normally most of our organization, nowadays do allow NAT or any kind of NAT but they HTTP tunneling only based organizations are very few, but if they are there, we need to provide HTTP tunneling for that. So, with that, I actually end this lecture and we will look at the next topic in the next video.