An Empirical Investigation into Peer-to-Peer Network Address Translation (NAT) by using Internet Protocol Version 4 & 6

MOHAMMAD SADEGHPOUR NAZARI
Faculty of Arts, Computing, Engineering and Sciences,
Sheffield Hallam University, Sheffield, UK
Mohammad.Sadeghpournazari@student.shu.ac.uk

Abstract- **Although Internet Protocol version 6 (IPv6) rectified the valid IPv4 shortage, Network Address Translation (NAT) was deployed to decrease overwhelming demand for registered IPv4 addresses.**
By using NAT in both client / server and Peer-to-Peer networks, many problems which effect the rate of data downloading and uploading for users such as network overhead and packet discarding were encountered; however, the effecting rate of each problem relates to the use of TCP and UDP ports.
There are many different methods which are used to rectify P2P NAT problems. One of the best and most basic methods used in P2P NAT is Hole Punching. This technique, which has been discussed in this article, allows clients in private networks to connect together by using TCP and UDP ports.
During the next few years, although due to the explosion in use of IPv6 the request for NAT will decrease, address translation will still be needed by many clients, who use IPv4. Therefore, a new protocol, which is called Network Address Translation - Protocol Translation (NAT-PT), is necessary to translate the packet's header from IPv4 to IPv6. Replacing IPv4 by IPv6 has intensified these problems for the clients and ISPs; hence IPv6 has not been very popular so far.
The main objectives of this paper is aimed to analyze the NAT issues as throughput and overhead of the network via using different analyzers such as Network Observer by investigating into Peer - to - Peer NAT [Protocol Translation] when TCP/UDP ports and IPv4/6 are used. These later results are an important step for research in this field;
However, they also clearly highlight the existing lack of information in this domain.

Keywords*: *P2P NAT, TCP/UDP Hole Punching Algorithm, NAT-PT

I. Introduction

I.1) IPv4 Addressing

The IP address is one of the basic requirements for any device, which connects to any network. Traditionally, each IP address has 32 bits and is divided into 4 octets. [1] The Regional Internet Registry (RIR) has defined two different kinds of IPv4 address: public and private. Public IP addresses are routable, unique and known to all internet users; whilst private IP addresses are not permitted to be routed through the internet and organisations and private users can use them as much as they need.

Different scaling options such as CIDR (Classless Inter Domain Routing) and VLSM (Variable Length Subnet Mask) have been implemented to make IP addresses as usable as possible. However, by increasing the request for the public IP address, the number of available IP addresses is reduced.
To rectify this problem, two different techniques were deployed. The first method was a next-generation IP address definition, called "IP version 6" (IPv6). This version of IP address has 128 bits and 2^{128} addresses are available for all the users in the world. The second technique, which was defined to slow the depletion of IPv4 addressing, was address translation. [2]

I.2) Internet Protocol Version 6

The Internet Protocol Version 6 (IPv6) is defined by RFC 2460. According to Cisco Networking Academy (2010), by using IPv6, 3.4×10^{38} usable IP addresses can be allocated to the hosts. [3] There are many advantages for deploying IPv6. Global reach ability and flexibility; aggregation; auto configuration; end-to-end connection without address translation; renumbering; routing efficiency; no broadcasts; and no checksums are the most important benefits of using IPv6.

I.3) Peer-to-Peer Networks

Peer to Peer technology (P2P) is an emerging paradigm that is now viewed as a potential technology to redesign distributed architectures." [4]
One of the most important issues in Peer-to-Peer networks is connecting two or more PCs which are in different networks and have private IP addresses. For connecting these two nodes together and making a Peer-to-Peer network, address translation is needed. NAT in Peer-to-Peer networks causes well-known difficulties, because in P2P NAT the destination may not be reachable at any globally valid IP address.

I.4) Network Address Translation (NAT)

NAT is a process of addresses manipulation in the header of packets and is defined by RFC 1631. [4] It is used to translate private IP addresses to public IP addresses, which are routable through the internet and vice versa. It is usually used on the border of networks, especially "stub networks" (Figure 1).

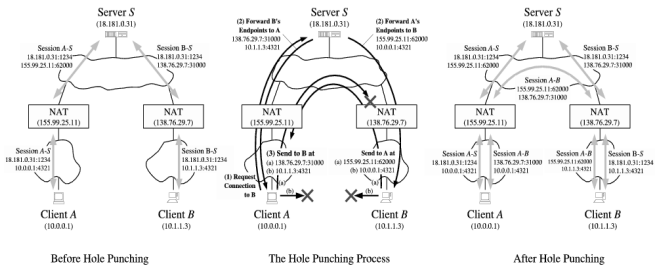


Figure 1: Schematic showing different NAT names.

There are three different types of address translation mapping on networks. They include:

1- Static NAT
2- Dynamic NAT
3- Overloaded NAT (PAT)

Many different techniques are used to implement P2P NAT and to rectify its problems. According to Hu (2005), UPnP (Universal Plug and Play), STUN (Simple Traversal UDP through Network Address Translation), ALG (Application Level Gateway) and UDP/TCP, Hole Punching is the basic technique used for Peer-to-Peer NAT. [6] The Hole Punching is widely used in both UDP- and TCP-based applications. This method is introduced by RFC 3027 and is one of the simplest methods to make an end-to-end TCP or UDP reliable session between two nodes. [7] (Figure 2)

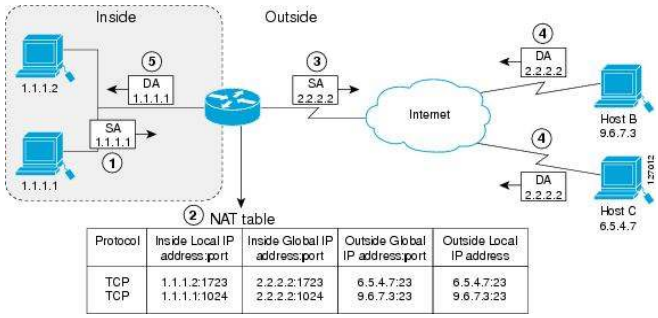


Figure 2: Hole Punching peer behind different NATs.

UDP and TCP Hole Punching are two different methods of Hole Punching algorithm. However, although TCP Hole Punching is more reliable, the UDP Hole Punching method is faster in sending and receiving data; easier to implement; and causes less congestion on the network. Therefore, most of the P2P networks that need a greater data transfer speed use UDP Hole Punching.

As mentioned earlier, in all kinds of P2P Address Translation the transport layer protocols have a very important role. Therefore, understanding the function of TCP and UDP ports and applications in P2P NAT is vital.

I.5) User Datagram Protocol (UDP)

UDP is the simplest protocol and is the only connectionless and unreliable protocol in this layer, which runs up to 40% faster than TCP. [8] It is defined by RFC 768 and has different characteristics such as: No reliability mechanism, No delivery guarantees and No buffering services.

I.6) Transmission Control Protocol (TCP)

TCP, which is defined by RFC 793, is the most important protocol in the transport layer and serves as the intermediary between application layer programs and network layer processes. [8] An overview of TCP shows that it is a connection-oriented service which has error recovery, sequencing, end-to-end reliability and flexibility. [9]

I.7) Network Address Translation - Protocol Translation (NAT-PT)

NAT-PT, which is defined by RFC 2765 & 2766, is used to establish a connection between IPv4 and IPv6 devices by translation of IPv4 packet's header to IPv6 and vice versa. [10]

I.8) Synchronisation of NAT with Transport Layer Protocols (Discussion)

Although NAT conserves registered IP addresses and increases the flexibility and reliability of connection to public networks, it loses end-to-end traceability. It, also, supports TCP and UDP traffic, which do not carry the IP addresses of sources and destinations. [4] In addition, that it increases the delay between sending and receiving packets, because the router CPU must check the header of each packet and decide whether it has to be translated or not. Therefore, the TTL (Time-to-Live) of some packets, which use TCP or UDP ports, might fall to zero, resulting in the packet being dropped. [11] Ruffi (2006) also believes that this might happen when the connection and translation slots are in idle time. According to him, "the idle times for translation slots in TCP connections are freed approximately every 60 seconds". After this time, the connection is closed and the packet will be dropped.

Blechschmidt (2005) believes that sometimes in P2P networks, the NAT traverse sends unrequested data to the NAT gateway through the UDP port. These packets are dropped by the firewall or NAT gateway. [12] However, this matter does not occur in the ports that use TCP protocol, because before using the TCP protocol a request must be sent to the data sender. Any requests sent through the TCP protocol, which uses three-way handshakes, should be received by another peer and then the connection is established. Therefore, the overhead of the network using TCP ports is more than that using UDP ports.

According to Ford et al. (2005), by using UDP port in Hole Punching Algorithm, the connection becomes unreliable, because after 20 seconds the UDP port reverts to idle time and the traffic is blocked. Therefore, if the application which uses UDP ports wants to be in an active state, it should send a periodic "keep-alive" message to guarantee that the relevant

translation state in the NATs does not disappear. Unfortunately this method provides an excessive amount of keep-alive traffic and is another cause of congestion in the network.

Guha and Francis (2005) believe that in TCP Hole Punching, both endpoints initiate a connection by sending SYN packets. If, when the SYN packets are delivered, both endpoints respond with SYN ACK packets, a connection is established. [13]

According to Hong et al. (2003), using IPv6 as the back bone of the network is more effective than using IPv4. [10] They believe that deploying IPv6 in the access layer, in which some nodes may still use IPv4, is one reason why packet process speed falls. In other words, NAT-PT is a cause of congestion in access layer networks.

I.9) Summary

It is clear that in both P2P and client-server NAT some packets are discarded. So, the rate of packets throughput is different in each network. On the client-server NAT most of the packets which are discarded use UDP protocol, and the main reasons for discarding them is the TTL of the packet; whilst in P2P NAT by using the simplest algorithm, which is the Hole Punching, the number of discarded packets, is related to various different factors, including:

1- The device which is used.
2- Different sizes of packets.
3- Time-to-Live (TTL) of the packet.
4- Use of TCP or UDP ports.
5- Idle and termination time of the TCP and UDP ports.
6- Termination of the connection by requester.
7- Different applications and services which are used.

In addition, network overhead and congestion (e.g. due to sending keep-alive messages) are other negative effects of using NAT in P2P networks by contributing IPv4 and IPv6.

II. Experimental Implementation and Data Collection

As mentioned earlier, the work of this article is based on experiments. In other words, two P2P NAT experiments, which are similar to each other, were implemented to survey objectives and draw a conclusion.

For this research, the TCP & UDP Hole Punching algorithms have been implemented. The topology of this research is presented below in Figure 3.

II.1) Experiment One

In the first scenario the DHCP service was enabled on R2 to send the IP addresses of peers behind each NAT to another peer. The IP addresses behind both R1 and R3 are private. Therefore, an address translation on R1 and another on R3 is needed to change the addresses and make them routable outside of the network. RIPv2, also, has been configured as dynamic routing protocol. In addition, Port Address Translation has been deployed on both R1 and R3, as border of each private network, to translate the private IP address to the Public IP address and vice versa.

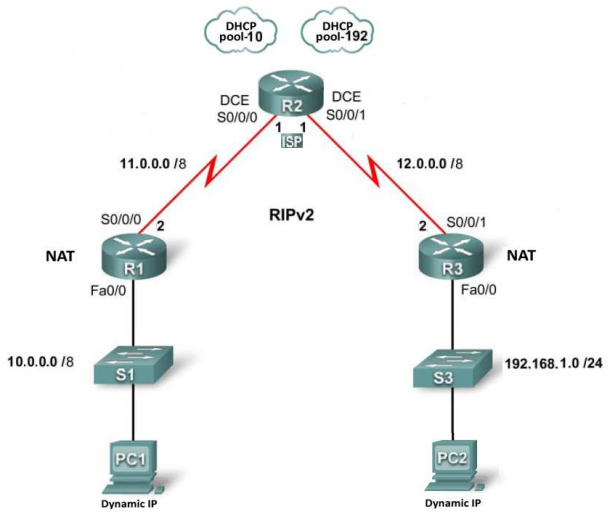


Figure 3: Schematic depicting the IPv4 scenario

II.2) Experiment Two

The second experiment of this research is similar to the first one. This experiment has been assigned based on IPv6. In other words, the serial connections between R1, R2 and R3 have been configured by IPv6. However, the Fast Ethernet connections of the R1 and R3 routers use IPv4. Therefore instead of NAT, the NAT-PT must be implemented on R1 and R3 to translate IPv4 to IPv6 and vice versa.

II.3) Software and Tools

For this research Network Observer has been used to to generate TCP and UDP packets and analyzes the raw data in this P2P network. Also, Network Traffic Generator and Monitor (NTGM) has been deployed to investigate network statistics is Network Traffic Generator and Monitor. This is one of the simplest tools used to present the inbound and outbound TCP/UDP/ICMP traffics of the clients. In addition, Q-Check is another used tool, which measures the throughput of the implemented network when TCP and UDP ports are in operation.

III. Outcomes and Results

III.1) Experiment One Outcomes

In the first part of this research, the Hole Punching algorithm has been simulated to check NAT functionality and to measure the rate of discarded packets in P2P NAT by using IPv4; however, before installing NAT, this experiment was conducted by sending 1514 byte packets to check the effect of P2P address translation in the rate of discarded packets on the TCP and UDP ports. In addition, network throughput and overhead, when using TCP and UDP ports, are other measurable factors to consider in this part of the research.

For investigation of the first objective, which involves measuring the rate of failed attempts in P2P NAT, different data packet sizes over different periods of time for both TCP and UDP ports were created and sent by packet generator.

The first data packet size is 1514 bytes, and this is sent through the TCP port. By using this data packet size, the rate of data transfer is automatically set to 10000 packets/second, because the maximum utilization of the router has been set to 100%.

The next factor for consideration, which is necessary to measure the rate of dropped packets, is a definition of the time period. The time period for data capturing is set between 20 seconds and 3600 seconds (1 hour). In other words, data monitoring is done in intervals at 20s, 60s, 300s, 900s, 1800s and 3600s. The results of this initial part are shown in Table 1 below:

Table 1: Results of the initial TCP experiment, pre-NAT configuration.

Time (Second)	*20*	*60*	*300*	*900*	*1800*	*3600*
Failed attempts	0	7	30	95	314	625
Total received Packets	2405	4001	7630	17774	31419	62238
Percent of discarded Packets (%)	0.00	0.17	0.39	0.53	1.00	1.00

Then, by increasing the size of packets to 8192 bytes, 16384 bytes the rate of discarded packet in different time periods for TCP Hole Punching algorithm was measured.

To investigate the second part of the first objective of Experiment One, as for the first part, different packet sizes were sent to the destination through the UDP port. Similarly, the utilization of the router was set to 100% to check its functionality in a critical state. Also, the same range of time and packet sizes (1514, 8192, 16384 bytes) were used to check the rate of failed attempts in P2P NAT by using a UDP port.

The second objective of Experiment One is investigation of the network throughput when TCP and UDP ports are used with IPv4. For this purpose, different data packet sizes using TCP and UDP ports were sent from PC1 to PC2 when P2P NAT was running. In addition, the same experiment was conducted without installing any address translation to survey the effect of NAT on the network throughput and network overhead.

The following table presents the results of an investigation into network throughput with NAT installation on TCP and UDP ports:

Table 2: Network Throughput with NAT

Data Packet Size (kb)	*TCP Throughput with NAT (kbps)*	*UDP Throughput (kbps)*
1	27.902	28.703
2	33.074	33.987
3	38.911	40.080
4	36.331	44.028
5	40.088	46.629
6	41.819	48.201
7	42.334	49.390
8	44.022	51.288
9	46.307	38.684
10	47.370	10.077
20	52.582	Overflow
50	56.538	Overflow
75	58.732	Overflow
100	58.686	Overflow

III.2) Experiment Two Outcomes

The second experiment was conducted based on IPv6. The same scenario as that used for Experiment One was implemented for this section; however, NAT was replaced by NAT-PT on R1 and R3.

NAT-PT functionality is completely different from NAT. This protocol translator gives a virtual IPv6 host in a v4 network to make it routable into the v6 network and send it to R2. It also gives a virtual IPv4 to the interface of R2, which is directly connected to the gateway of NAT-PT, to allow it to traverse into the v4 network.

By verification of this scenario, it is realised that links between routers have been connected and RIPng is working properly.

In addition, PC1 is able to ping R2 interface serial 0/0/0 and PC2 can ping interface serial 0/0/1 on R2, and vice versa. In other words, NAT-PT acts properly on both routers.

However, with both PC1 and PC2 able to ping R2 interfaces separately, they did not ping R2 interfaces together.

Also, PC1 and PC2 were not able to communicate directly with each other, although the connectivity of all network sections appeared to have been verified. In other words, this means that NAT-PT does not act in P2P networks. By further investigation into this field, it became apparent that for this experiment the functionality of tunnelling is better than NAT-PT.

So, this experiment concluded without the rate of discarded packets being measured, and without the network throughput and overhead being checked, since it was established that tunnelling between v4 and v6 networks would be better suited to this role, and this topic remains beyond the scope of this research.

IV. Conclusion and Discussion

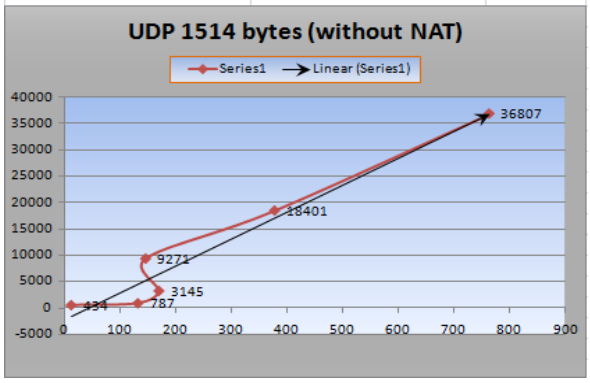
Certainly, the field of P2P Network Address Translation is very vast and researchers cannot cover all aspects of NAT problems in an article. Therefore, extensive investigation into different articles is needed to address different NAT issues. By comparison between the new issues raised in the recent scientific articles and the ones stated in this paper, it is clear that the issues around NAT which has been discussed in this article are clearly confirmed by the recent researches. Moreover, the issue of network throughput with and without installing NAT on network has not been addressed in the recent publications have been covered in this paper.

IV.1) Experiment One Conclusion without NAT

As already mentioned, in order to deduce a very specific result from experimentation, initial experiments were performed before configuring NAT to check the effect of NAT on the objectives. In other words, before NAT installation, data packets of 1514 bytes in both TCP and UDP ports were sent.

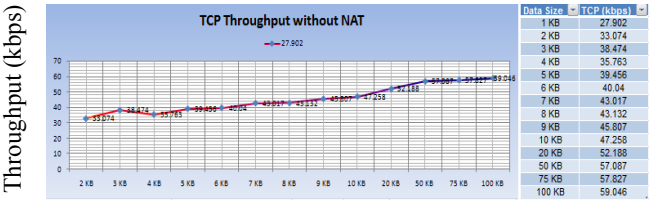
According to statistics, by using a TCP port and a data packet size of 1514 bytes, without using any address translation, a maximum of 1% of packets are dropped. This could be related to many different factors such as: the device which is used; the size of the packet; TTL; idle time; and so on. Using a UDP port for data traversal shows another result. According to the figures, the rate of discarded packets in a short period of time has a maximum of 16.77%, as expected. It can be seen that the rate of discarding packets in UDP ports is more than that in TCP ports, as was expected. As earlier discussed, UDP ports are not reliable and do not use a three-way hand-shake. Therefore, there is no guarantee as to the successful delivery of the packets when using UDP ports.

No of discarded packets	Total received packets	Percent
12	434	2.76%
132	787	16.77%
170	3145	5.41%
146	9271	1.57%
378	18401	2.05%
764	36807	2.08%

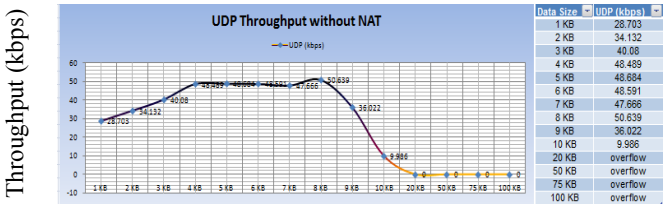


Graph 1: Graphical representation of the UDP experiment pre-NAT installation.

For the second step, the throughput of the network in both TCP and UDP ports was acquired in order to find the overhead of the network by comparing TCP and UDP protocols.



Graph 2: Graphical representation of TCP throughput before NAT installation.



Graph 3: Graphical representation of UDP throughput before NAT installation.

Comparison between Graphs 2 and 3 shows that the overhead associated with using TCP ports is more than that for UDP ports. This is because of ACK and SYN packets, which are sent only by TCP ports. In contrast to the previous result, when using a data packet size of more than 10 kb, the network overhead of sending data through the UDP port is more than that for TCP. The window size of the data packet could be another reason for the network overhead during this phase.

IV.2) Experiment One Conclusion, TCP Hole Punching

As mentioned earlier, after configuring NAT on both R1 and R3, different data packet sizes in different time periods were sent to check the rate of discarded packets. By using a TCP port, three different data packet sizes were sent.

Consideration of the outcome statistics reveals that the time increment for the number of discarded packets using TCP ports is increased when compared with that for UDP ports. Increasing the size of the packets, also, has the same effect on the rate of dropped packets.

Comparison between TCP Hole Punching and data transmission in P2P networks without any address translation shows that the number of transmitted packets without NAT is more than the number of sent packet with TCP Hole Punching. This means that NAT affects P2P networks and is a cause of delay and congestion in the network. Also, by TCP Hole Punching algorithm implementation, it was established that the number of discarded packets had increased. For example, the range of failed attempts in P2P networks without NAT by sending packets with a byte size of 1514 is 0 to 1%; while, in TCP Hole Punching using the same criteria it is between 0.04

and 3.15%. In addition, the maximum number of transmitted packets per hour declined from 62238 to 57118 packets.

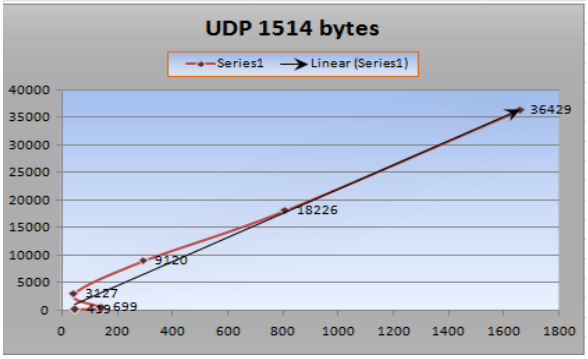
Checking the throughput of the P2P network whilst using the TCP Hole Punching algorithm was the next step of Experiment One. Comparison between outcome data, which shows the network throughput after address translation, and the TCP throughput received data before NAT installation shows that the TCP Hole Punching method has little influence on the throughput of network; however, this effect is better presented alongside the network response time.

IV.3) Experiment One Conclusion, UDP Hole Punching

After TCP Hole Punching, which is not very popular in P2P networks, UDP Hole Punching was tested. In this phase, the same sizes of data packet in the same time periods as used in TCP Hole Punching were checked.

By studying and analysing the figures, it can be understood that, in a similar way to TCP Hole Punching, increasing the time and data packet size reduces the total number of transmitted packets, but increases the rate of discarded packets. (Graph 4)

No of discarded packets	Total received packets	Percent
45	419	10.74%
141	699	20.17%
42	3127	1.34%
294	9120	3.22%
807	18226	4.43%
1657	36429	4.46%



Graph 4: Graphical representation of the UDP experiment after NAT installation.

Looking more closely at the analysed outcomes, more specific information can be gathered. For example, they show that the rate of failed attempts in UDP Hole Punching is mostly related to the short transmission times (maximum 300s), and occurs mostly due to the UDP port's idle time. In addition, UDP Hole Punching is used in real P2P networks for activities such as file sharing, video conferencing and so on, or when a large amount of data is going to be transmitted and needs a long time for transmission. So, the presented results justify expert opinion that UDP Hole Punching is more usable for big data size transmission in P2P networks than TCP Hole Punching.

Moreover, it was also justified that for sending data with a small size, TCP ports act better than UDP ports and are more reliable.

As in the last section of Experiment One, again for Experiment Two the throughput of the network once UDP Hole Punching had been installed was tested. As result of this section, the maximum packet size used to check the network throughput in UDP Hole Punching was 10 kb, because if packets of greater than 10 kb are used, the overflow message appears on the screen. It is illustrated that NAT has a clear effect on the throughput of the network.

On the other hand, comparison between TCP and UDP Hole Punching shows that when sending packets of less than 10 kb in size, the overhead of the network in TCP Hole Punching is more than that in UDP Hole Punching. This is because of the SYN and ACK packets.

Finally, according to Ford et al. (2005), the rate of success in UDP Hole Punching by using a Cisco router is 100%, whilst the results of this experiment say that the normal rate of NAT success for UDP ports is 94.25%. It is believed that this difference is related to the window size of the packets which have been sent, and to the transmitting time period. [6]

With regard to the rate of discarding packet on NAT, a new algorithm has been defined for video streaming on Peer to Peer networks by Wei and Pan (2011). [14] According to their result, the peer behind NAT cannot respond to all requests if the device receives a large number of packets and requests in a time slot. Therefore, the device randomly chooses packets to respond and discard the rest of requests. Network congestion is the second important NAT measured issue. According to Price & Tino (2010), the rate of network delay and congestion at any algorithms that sends keep alive message to make sure the network is active, is 20% more than the other mechanism that does not need to use keeps alive message. [15]

In addition, Zhang (2008) believes that, delay and congestion in P2P networks is related to three different ways of NAT traversal (STUN, TURN and MidCom) and is based on used application and services. [16] Last but not the least, the security of P2P networks is directly related to the NAT. This point has been clearly clarified by Pengfei & Yamei (2010) research, which is based on UDP Hole Punching. [17]

As it is concluded in this paper, by utilizing TCP/UDP Hole Punching, the rate of discarded packet, network overhead and congestion are increased, depending on applied device.

As it mentioned earlier, rate of discarded packet is related to different factors and can be investigated from different views. For example, number of neighbours behind of the NAT is one of the most important issues on rate of lost packet at video streaming. Wei and Pan (2011) have defined a new algorithm for video streaming; [14] however; they did not examine any

other factor in their research. With regards to the network congestion, this article has evidently emphasised on effect of SYN, ACK and keep alive message on TCP Hole Punching and this point has been confirmed by Price & Tino (2010) as well. [15] Moreover, regardless of the applied NAT methodology in network, used application and services are two important factors in the rate of network delay. By investigating into the TURN (Traversal Using Relay NAT) algorithm, which is the most typical ones in TCP based-NAT, Zhang (2008) has verified that the rate of congestion and delay in TCP-based NAT is more than UDP-based NAT and this might be a reason for discarding packets. Due to the security, reliability and NAT implementation into P2P networks, the UDP Hole Punching methodology is faster in sending and receiving data, easier to implement and causes less congestion on the network than TCP Hole Punching. [16] Therefore, most of the P2P networks that need a greater data transfer speed use UDP Hole Punching. This issue has been declared by Pengfei & Yamei (2010); [17] however, they did not state the effect of any other engaged issue in UDP Hole Punching, which have been utilised in this research, .including utilized device, different sizes of packets, Time-to-Live (TTL) of the packet, Idle and termination time of the UDP ports, or Different applications and services which are used.

IV.4) Experiment Two Conclusion

Experiment Two was based on IPv6. As explained, Network Address Translation – Protocol Translation (NAT-PT) was configured on both R1 and R3 to make a P2P NAT-PT connection between PC1 and PC2. Verification of this experiment showed that both translators work properly. In other words, IPv4 from network 10.0.0.0/8 was translated to 2001::/64 and vice versa. IPv4 in network 192.168.1.0/24, also, was translated to 2002::/64 and vice versa. However, despite using RIPng as a dynamic routing protocol, sending updates between routers, PC1 and PC2 were still not able to communicate with each other.

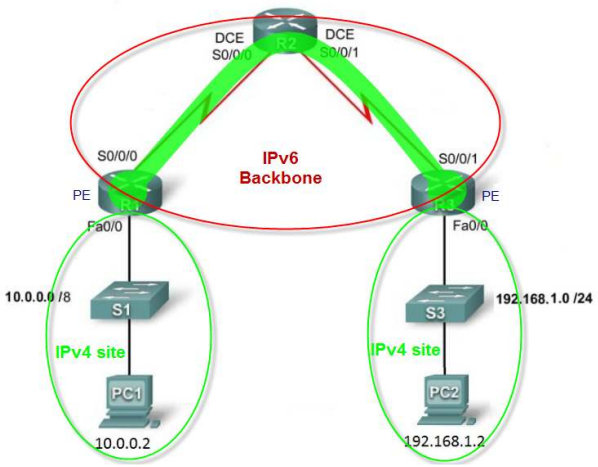


Figure 4: Schematic to show an IPv6 tunnel used between two IPv4 networks.

By investigation into this field, it was found that although NAT-PT is able to translate IPv4 to IPv6 and vice versa, it is not able to translate a converted IPv4 again. In other words, an IPv4 which has already been translated to an IPv6 is not able to be translated to another subnet of IPv4. This means that NAT-PT alone is used as the demarcation point between 2 domains.

Figure 4 shows that the best way for making a connection between two different IPv4 sites, which have different ranges of IPv4 and are connected by an IPv6 backbone, is to use IPv4 to IPv6 tunnelling.

So, keeping within the research field of this dissertation, which concerns NAT, the same objectives as used in Experiment One were not accessible to measure and survey the effect of IPv6 on P2P networks.

Whilst the implementation of NAT-PT in this experiment meant that the rate of discarded packets could not be measured, other results relating to IPv6 as a backbone were gathered.

Although IPv6 is the next generation of Internet Protocol and will become more popular over the next few years, it creates very significant congestion and delay in the backbone of networks. The most important reason for this is related to the header of IPv6, which has 128 bites.

This matter was justified by sending ICMP packets from PC1 and PC2 to R2. PC1 and PC2 were able to ping R2 serial interfaces separately; however, about 50% of the packet failed (see Figure 5).

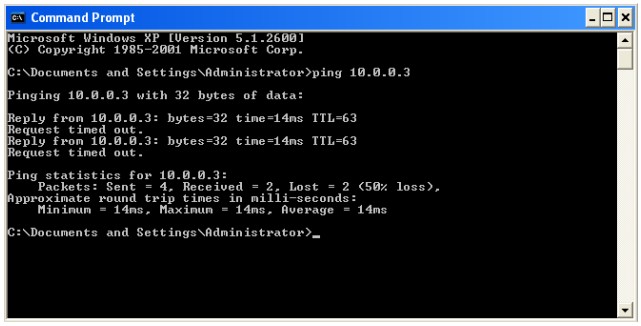


Figure 5: Commands showing the loss of 50% of packets after sending ICMP packets from PC1 and PC2 to R2 (pinged separately)

This occurrence is caused by the router not producing enough buffers to process all the packets, resulting in some of them failing. In addition, similar to when using NAT, NAT-PT itself could be a reason for this packet discarding.

So, it is recommended that during the next few years as IPv6 explodes onto the world market, the cache and buffer of the routers must be improved. In addition, according to the

popularity of P2P networks in the world, the reliability of UDP ports must be clarified, because most of these networks use UDP ports for data transmission.

V. Future Works

As was explained in the last few chapters, the first experiment of this research, which was based on IPv4, focused on the Hole Punching algorithm as the simplest technique on P2P NAT; however, there are other techniques that are used on some P2P networks for address translation, too. For instance, RNT, GSP, SPPS, SSP are some of the NAT traversal techniques that are used in application services. Therefore, investigation into the rate of discarding packets, network throughput, and network congestion to check the efficiency of each technique deserves further research in this field.

Regarding Experiment Two, it was mentioned that the rate of dropped data, network delay and congestion when using IPv6 are greater than their counterparts when IPv4 is used. Thus, a survey as to the causes of these problems in IPv6 networks, such as cache and buffer size of the routers, is another interesting area in the field of NAT-PT.

In addition, looking into the effects of increasing the number of hosts behind NATs in relation to the rate of discarding packets could prompt further study, especially because it was shown that when both PCs try to connect to the R2 router, which is on the v6 network, the 'request timeout' message appears for both PCs, a matter which only intensifies when the debug command is issued on the R2 router.

A final suggestion concerning the second experiment is related to tunnelling between IPv4 and IPv6 networks. It is clear that most of the implemented tunnelling involves 6-to-4 tunnels. In other words, two different IPv6 domains are connected together by using an IPv4 backbone. The DNS server has a vital role in this kind of tunnelling, but in this particular scenario (Experiment Two), the backbone of the network has been configured by IPv6. Hence, another kind of tunnelling must be deployed to rectify the problems of the second experiment. Implementation of this kind of tunnelling (using an IPv6 backbone) seems to be very difficult and would need serious study beforehand, especially since there is a lack of implementable knowledge in this field, with most researches explaining only the theory of this domain, choosing instead to implement v6-over-v4 tunnelling.

As a conclusion, it is strongly recommended that due to the explosion in popularity of IPv6 in the world, and so the necessity to accommodate this into existing networks, tunnelling between IPv4 hosts by deploying an IPv6 network as a backbone is employed, and the rate of discarding packets, delay and congestion of the networks are measured.

VI. Acknowledgment

Special thanks to ***Mr. Siavosh Haghighi Movahed*** *(S.Haghighi-Movahed@shu.ac.uk)* - Sheffield Hallam University - as technical supervisor and ***Dr. Babak Khazaei*** *(B.Khazaei@shu.ac.uk)* - Sheffield Hallam University - UK for providing guidance, review and feedback.

VII. References

[1] K. Cannon & C. Cannon & A. Chiarella, (2004), *CCNA Guide to Cisco Networking*, 4th Edition, Boston; USA, ISBN 978-0-84003119-8

[2] T. Lammle & S. Odom & K. Wallace, (2001), *CCNP routing study guide*, 1th Edition, San Francisco: Sybex, c2001, ISBN: 0-78212-712-6.

[3] Cisco Systems Inc., (2010), *"Cisco Networking Academy"*, [Online], available at: http://www.cisco.com/web/learning/netacad/index.html

[4] M. Lupu & B.Ooi & Q. Vu., (2010), *Peer-to-Peer Computing*, 1st Edition, Springer Heidelberg Dordrecht London New York, ISBN 978-3-642-03513-5

[5] Cisco Systems Inc. (Cisco Networking Academy Program), (2004), *Cisco Networking Academy Program: CCNA 3 and 4 companion guide / Cisco Systems Inc.*, Indianapolis, USA: Cisco Press. ISBN 1-58713-113-7.

[6] Z. Hu, Telecommunication Software and Multimedia Laboratory Helsinki University of Technology, (2005), *"NAT Traversal Techniques and Pee-to-Peer Applications"*, [Online], available at: http://www.tml.tkk.fi/Publications/C/18/hu.pdf

[7] B. Ford & D. Kegel & P. Serisuresh, Massachusetts Institute of Technology, (February 2005), *"Peer-to-Peer Communication across Network Address Translation",* [Online], available at: http://www.brynosaurus.com/pub/net/p2pnat/

[8] S. Convery, (2004), *Network Security Architectures*, Indianapolis, Ind.: Cisco Press, ISBN 1-58705-115-X

[9] B. Forouzan, (2009), *TCP/IP protocol suite*, 4th Edition, Boston; London: McGraw-Hill medical, ISBN 978-0-07-016678-3.

[10] Y. Hong & H. Kim & M. Shin, Protocol Engineering Center, and Electronics and Communication Research Institute Deaajeon, Korea, (2003), *"Application Translation for IPv6 at NAT-PT"*, IEEE 2003, [Online], available at: http://ieeexplore.ieee.org.lcproxy.shu.ac.uk/stamp/stamp.jsp?tp=&arnumber=1274343

[11] A. Rufi, (2006), *Network Security 1 and 2*: companion guide, Indianapolis, USA: Cisco Press, ISBN 1-58713-162-5.

[12] I. Blechschmidt, (2005), *"NAT-traverse - Use of UDP to traverse NAT gateways"*, [Online], available at: http://linide.sourceforge.net/nat-traverse/

[13] S. Guha, & P. Francis, Cornell University, (October 2005), *"Characterization and Measurement of TCP Traversal through NATs and Firewalls"*, [Online],

available at: http://nutss.gforge.cis.cornell.edu/pub/imc05-tcpnat.pdf
[14] Z. Wei & J. Pan, University of Victoria, Canada, (2011), *"Modeling BitTorrent-Based P2P Video Streaming Systems in the Presence of NAT Devices"*, IEEE (ICC) 2011, [Online], available at: http://ieeexplore.ieee.org.eresources.shef.ac.uk/stamp/stamp.jsp?tp=&arnumber=5963110
[15] R. Price & P. Tino, School of Computer Science, University of Birmingham, United Kingdom, (2010), *"Adapting to NAT timeout values in P2P Overlay Networks"*, IEEE 2010, [Online], available at: http://ieeexplore.ieee.org.eresources.shef.ac.uk/stamp/stamp.jsp?tp=&arnumber=5470785
[16] X. ZHANG, School of software, University of Electronic Science and Technology of China, (2008), *"DESIGN AND REALIZATION OF TCP-BASED NAT TRAVERSAL IN P2P"*, IEEE 2008, [Online], available at: http://ieeexplore.ieee.org.eresources.shef.ac.uk/stamp/stamp.jsp?tp=&arnumber=4770024
[17] C. Pengfei & Z. Yamei, Department of Computer Science, Henan Mechanical and Electrical Engineering College Xinxiang, China, (2010), *"Research on Using UDP to Traverse NAT under P2P Network Environmenton"*, IEEE (ICACTE) 2010, [Online], available at: http://ieeexplore.ieee.org.eresources.shef.ac.uk/stamp/stamp.jsp?tp=&arnumber=5579841

Biography

Mohammad Sadeghpour Nazari received his Master of Computer Science (MSc) degree with major of Networking Professional with Distinction grade from Sheffield Hallam University – UK by October 2010. He is also certified as a Cisco Certified Network Associate (CCNA) by the same time. In addition, He got his CCNA Security certificate by September 2010 and his MCSE certificate by May 2004.

He has done different researches in the field of Computer Networks such as "Implementation of 802.1X authentication by deploying the RADIUS Protocol for large scale networks" and "Advances in Wireless and Mobile broadband technology" for his MSc projects.

He has 8 years work experience in the field of wired and wireless networks and is currently working at industry.