National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member: Dr Hassan Khaliq **Dated: 5/3/2023**

Semester: 6th **Section: C**

EE-357 Computer and Communication Networks
Experiment - 11
NAT Network Address Translation

		CLO5-PLO9		
Name	**Reg. No**	**Individual and Teamwork** **5 Marks**	**Lab Report** **10 Marks**	**Quiz/viva** **5 Marks**
Muhammad Ahmed Mohsin	**333060**			
Imran Haider	**332569**			
Amina Bashir	**343489**			

NAT Network Address Translation:

The Internet is expanding at an exponential rate. As the amount of information and resources increases, it is becoming a requirement for even the smallest businesses and homes to connect to the Internet. Network Address Translation (NAT) is a method of connecting multiple computers to the Internet (or any other IP network) using one IP address. This allows home users and small businesses to connect their network to the Internet cheaply and efficiently.

In computer networking, **network address translation** (**NAT**) is the process of modifying IP address information in IP packet headers while in transit across a traffic routing device.

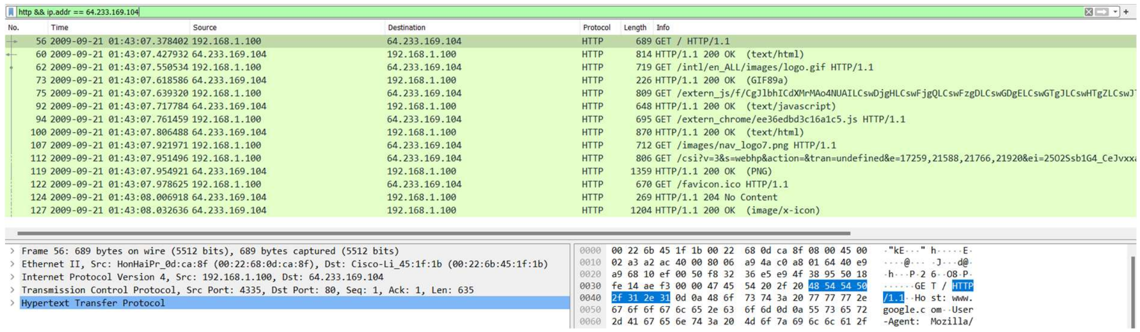
The simplest type of NAT provides a one to one translation of IP addresses. RFC 2663 refers to this type of NAT as **basic NAT**. It is often also referred to as **one-to-one NAT**. In this type of NAT only the IP addresses, IP header checksum and any higher level checksums that include the IP address need to be changed. The rest of the packet can be left untouched (at least for basic TCP/UDP functionality, some higher level protocols may need further translation). Basic NATs can be used when there is a requirement to interconnect two IP networks with incompatible addressing.

- The impetus towards increasing use of NAT comes from a number of factors:
- A world shortage of IP addresses
- Security needs
- Ease and flexibility of network administration

Questions:

1. What is the IP address of the client?

The IP address of the client is 192.168.1.100



http && ip.addr == 64.233.169.104

No.	Time	Source	Destination	Protocol	Length	Info
56	2009-09-21 01:43:07.378402	192.168.1.100	64.233.169.104	HTTP	689	GET / HTTP/1.1
60	2009-09-21 01:43:07.427932	64.233.169.104	192.168.1.100	HTTP	814	HTTP/1.1 200 OK (text/html)
62	2009-09-21 01:43:07.550534	192.168.1.100	64.233.169.104	HTTP	719	GET /intl/en_ALL/images/logo.gif HTTP/1.1
73	2009-09-21 01:43:07.618586	64.233.169.104	192.168.1.100	HTTP	226	HTTP/1.1 200 OK (GIF89a)
75	2009-09-21 01:43:07.639320	192.168.1.100	64.233.169.104	HTTP	809	GET /extern_js/f/CgJlbhICdXMrMAo4NUAILCswDjgHLCswFjgQLCswFzgDLCswGDgELCswGTgJLCswHTgZLCswJ
92	2009-09-21 01:43:07.717784	64.233.169.104	192.168.1.100	HTTP	648	HTTP/1.1 200 OK (text/javascript)
94	2009-09-21 01:43:07.761459	192.168.1.100	64.233.169.104	HTTP	695	GET /extern_chrome/ee36edbd3c16a1c5.js HTTP/1.1
100	2009-09-21 01:43:07.806488	64.233.169.104	192.168.1.100	HTTP	870	HTTP/1.1 200 OK (text/html)
107	2009-09-21 01:43:07.921971	192.168.1.100	64.233.169.104	HTTP	712	GET /images/nav_logo7.png HTTP/1.1
112	2009-09-21 01:43:07.951496	192.168.1.100	64.233.169.104	HTTP	806	GET /csi?v=3&s=webhp&action=&tran=undefined&e=17259,21588,21766,21920&ei=2502Ssb1G4_CeJvxx
119	2009-09-21 01:43:07.954921	64.233.169.104	192.168.1.100	HTTP	1359	HTTP/1.1 200 OK (PNG)
122	2009-09-21 01:43:07.978625	192.168.1.100	64.233.169.104	HTTP	670	GET /favicon.ico HTTP/1.1
124	2009-09-21 01:43:08.006918	64.233.169.104	192.168.1.100	HTTP	269	HTTP/1.1 204 No Content
127	2009-09-21 01:43:08.032636	64.233.169.104	192.168.1.100	HTTP	1204	HTTP/1.1 200 OK (image/x-icon)

```
> Frame 56: 689 bytes on wire (5512 bits), 689 bytes captured (5512 bits)
> Ethernet II, Src: HonHaiPr_0d:ca:8f (00:22:68:0d:ca:8f), Dst: Cisco-Li_45:1f:1b (00:22:6b:45:1f:1b)
> Internet Protocol Version 4, Src: 192.168.1.100, Dst: 64.233.169.104
> Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 1, Ack: 1, Len: 635
> Hypertext Transfer Protocol
```

```
0000  00 22 6b 45 1f 1b 00 22  68 0d ca 8f 08 00 45 00   ·"kE···" h·····E·
0010  02 a3 a2 ac 40 00 80 06  a9 4a c0 a8 01 64 40 e9   ····@··· ·J···d@·
0020  a9 68 10 ef 00 50 f8 32  36 e5 e9 4f 38 95 50 18   ·h···P·2 6··O8·P·
0030  fe 14 ae f3 00 00 47 45  54 20 2f 20 48 54 54 50   ······GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 77 77 77 2e   /1.1··Ho st: www.
0050  67 6f 6f 67 6c 65 2e 63  6f 6d 0d 0a 55 73 65 72   google.c om··User
0060  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f   -Agent:  Mozilla/
```

2. Consider now the HTTP GET sent from the client to the Google server (IP address 64.233.169.104) at time 02:43:07.378402. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET?

```
v Frame 56: 689 bytes on wire (5512 bits), 689 bytes captured (5512 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 21, 2009 01:43:07.378402000 Pakistan Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1253479387.378402000 seconds
    [Time delta from previous captured frame: 0.000214000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 7.109267000 seconds]
    Frame Number: 56
    Frame Length: 689 bytes (5512 bits)
    Capture Length: 689 bytes (5512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
```

Source IP: 192.168.1.100
Destination IP: 64.233.169.104
Source port: 4335
Destination Port: 80

3. At what time is the corresponding 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message?

01:43:07.427932

Source IP: 64.233.169.104

Destination IP: 192.168.1.100

Source port: 80

Destination Port: 4335

4. Recall that before a GET command can be sent to an HTTP server, TCP must first set up a connection using the three-way SYN/ACK handshake. At what time is the TCP connection ready? (Bilal)

Time: 01:43:07.378121

Source: 192.168.1.100, 4335
Destination: 64.233.169.104, 80

In the following we'll focus on the two HTTP messages (GET and 200 OK) identified above. Our goal below will be to locate these two HTTP messages in the trace file (NAT_ISP_side) captured on the link between the router and the ISP. Because these captured frames will have already been forwarded through the NAT router, some of the IP address and port numbers will have been changed as a result of NAT translation.

Open the NAT_ISP_side. Note that the time stamps in this file and in NAT_home_side are not synchronized since the packet captures at the two locations shown in Figure 1 were not started simultaneously.

5. In the NAT_ISP_side trace file, find the first HTTP GET message that was sent from the client to the Google server. At what time does this message appear in the NAT_ISP_side trace file? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET?

```
v Frame 85: 689 bytes on wire (5512 bits), 689 bytes captured (5512 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 21, 2009 01:43:07.800232000 Pakistan Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1253479387.800232000 seconds
    [Time delta from previous captured frame: 0.000414000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 6.069168000 seconds]
    Frame Number: 85
    Frame Length: 689 bytes (5512 bits)
    Capture Length: 689 bytes (5512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
```

01:43:07.800232

Source IP: 71.192.34.104

Destination IP: 64.233.169.104

Source port: 4335

Destination Port: 80

6. Compare these values with the corresponding values in the NAT_home_side file and comment whether NAT or NAPT is being used at the NAT router.

NAT modifies IP address in a header of an IP packet while it is travelling through an routing device. As in both cases the ip address of the source changed its header hence NAT or NAPT is being used at the NAT router.

7. In the NAT_ISP_side trace file, at what time is the 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message?

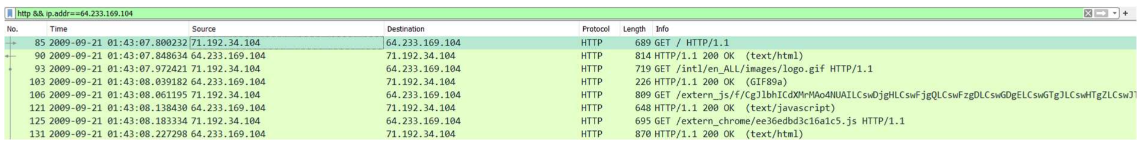
01:43:08.039182

Source IP: 64.233.169.104

Destination IP: 71.192.34.104

Source port: 80

Destination Port: 4335

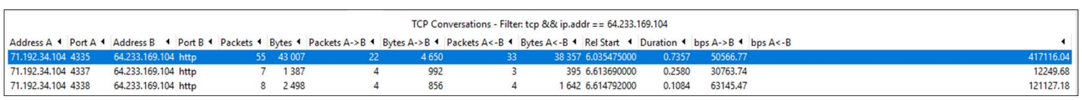


http && ip.addr==64.233.169.104

No.	Time	Source	Destination	Protocol	Length	Info
85	2009-09-21 01:43:07.800232	71.192.34.104	64.233.169.104	HTTP	689	GET / HTTP/1.1
90	2009-09-21 01:43:07.848634	64.233.169.104	71.192.34.104	HTTP	814	HTTP/1.1 200 OK (text/html)
93	2009-09-21 01:43:07.972421	71.192.34.104	64.233.169.104	HTTP	719	GET /intl/en_ALL/images/logo.gif HTTP/1.1
103	2009-09-21 01:43:08.039182	64.233.169.104	71.192.34.104	HTTP	226	HTTP/1.1 200 OK (GIF89a)
106	2009-09-21 01:43:08.061195	71.192.34.104	64.233.169.104	HTTP	809	GET /extern_js/f/CgJlbhICdXMrMAo4NUAILCswDjgHLCswFjgQLCswFzgDLCswGDgELCswGTgJLCswHTgZLCswJ
121	2009-09-21 01:43:08.138430	64.233.169.104	71.192.34.104	HTTP	648	HTTP/1.1 200 OK (text/javascript)
125	2009-09-21 01:43:08.183334	71.192.34.104	64.233.169.104	HTTP	695	GET /extern_chrome/ee36edbd3c16a1c5.js HTTP/1.1
131	2009-09-21 01:43:08.227298	64.233.169.104	71.192.34.104	HTTP	870	HTTP/1.1 200 OK (text/html)

8. Locate the TCP connection(s) made for this HTTP transaction. How many TCP connections have been made? Does the TCP connection addresses also get modified while passing through the NAT router?

Number of conversations: 3



TCP Conversations - Filter: tcp && ip.addr == 64.233.169.104

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start	Duration	bps A->B	bps A<-B
71.192.34.104	4335	64.233.169.104	http	55	43 007	22	4 650	33	38 357	6.035475000	0.7357	50566.77	417116.04
71.192.34.104	4337	64.233.169.104	http	7	1 387	4	992	3	395	6.613690000	0.2580	30763.74	12249.68
71.192.34.104	4338	64.233.169.104	http	8	2 498	4	856	4	1 642	6.614792000	0.1084	63145.47	121127.18

For the SYN, the source IP address has changed, For the ACK, the destination IP address has changed. The port numbers are unchanged.

Conclusion:

In this lab, we learned how to capture and analyze network packets using Wireshark. We first learned about the different types of network packets, including HTTP and TCP packets. We then learned about the internal structure of these packets. Finally, we performed packet capturing on a real network and analyzed the captured packets.We learned that HTTP packets are used to transfer hypertext documents over the network. They are composed of a header and a body. The header contains information about the request or response, such as the method, the path, and the version. The body contains the actual data being transferred.

We also learned that TCP packets are used to transfer data over the network in a reliable way. They are composed of a header and a payload. The header contains information about the packet, such as the source and destination addresses, the sequence number, and the acknowledgment number. The payload contains the actual data being transferred.By performing packet capturing and analyzing the captured packets, we gained a better understanding of how the internet works. We also learned how to use Wireshark to troubleshoot network problems.

In conclusion, this lab was a valuable learning experience. We learned about the different types of network packets, their internal structure, and how to use Wireshark to capture and analyze network packets. This knowledge will be useful in our future networking endeavors.