Report 3: Network Address Translation

Department of Mathematical Sciences
Computer Science Division
University of Stellenbosch
7600 Stellenbosch
06 September 2022

INTRODUCTION

Network address translation is a fundamental component used on the internet today. It was implemented due to less space availability from ipv4, which led to the creation of ipv6 which solved the space availability constraint. NAT was initially proposed as the solution to the space constrain, which allows local nodes to form a network using unique IP address and communicate internally without packet modification. When sending from internal to external, the packet uses the NAT route and gets allocated a single IP address by some external authority such as ISP. The NAT route just modifies its header so that it appears as if it was sent from the NAT box using the allocated external address, will keeping an eye on ensuring that the requests and replies are matched up. The trade-off for using NAT it can influence the quality of the connection and it breaks the original envisioned model IP end-to-end connectivity on the internet and it is hard for protected intranet hosts to accept incoming connections.

GROUP MEMBERS:

[Takunda, Charles, Mudima, 23969156]
[Ben, Sibusiso, Mahlangu, 23629428]

OVERVIEW

This program aims to show how packets are sent through the NAT route which allows local computers to create a network of communication using unique IP addresses. It constitutes of five files namely device.java, deviceHandler.java, dhcp.java, natBox.java and tableContent.java. These files work concurrently to establish a connection of sending and receiving of packets between connected devices.

ADDITIONAL FEATURES IMPLEMENTED

In our implementation we added a GUI which allows the client select which connection (External or Internal) they would prefer to join the NAT box as. See picture below. The GUI also comes with an exit button that allows the client to exit if they choose too.

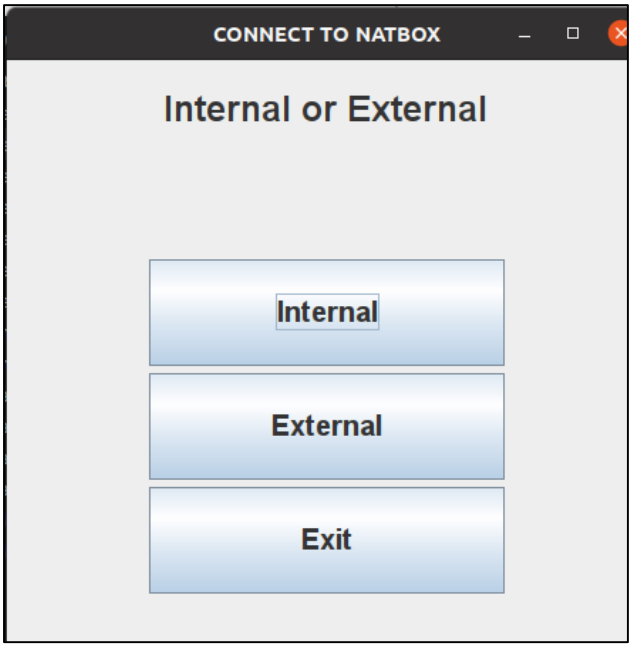

Figure 1: NAT box connection options

UNIMPLEMENTED FEATURES

External to internal communication before internal communication. Changing or addition of external IP address in the NAT table after communicating with the internal client. Instead, our program adds the external client to the NAT table immediately after connection. How we understood the project was that the external client is supposed to be added to the NAT table only after it has communicated with an internal client.

DESCRIPTION OF FILES

- **Device.java** consists of several functions, one be responsible for establishing connection(s) and connecting to the NAT box, the sending and receiving of packets and lastly, generate both IP and random Mac Address.
- **DCHP.java,** consists of the dchpclient() function, which dynamically allocates address to internal addresses, a function that sends an

acknowledgement and request to the NAT box and lastly, the ICMP which generate error messages to the source IP address when there is problem when delivering a packet

- **NatBox.java,** consists of a time that schedules an appointment for refreshing the natTable and creates a table of available private IP address
- **TableContent.java**, Object that creates a table structure and IP address entry.
- **DeviceHandler.java,** responsible for threading of messages between connected devices and print error messages in the NAT box

EXPERIMENTS AND TESTING

EXPERIMENT 1: ROUTING TESTS

TEST1:
Internal client to External client

HYPOTHESIS:
Packets are not altered and are expected to arrive successfully.

VARIABLES:
Two Clients connect to the NAT box one as an internal client (10.0.0.0) & the other as an external client (197.48.165.56)

METHOD:
In either of the clients use the following format to send a packet. See example below.
Format: SendCommand destinationIP Packet
$>> 197.48.165.56 hello [Sending packet from client 10.0.0.0]

RESULTS:
Packets were sent successfully.

TEST2:
Internal client to Internal client

HYPOTHESIS:
Paquet forwarding is forwarded without changing its data.

VARIABLES:
Two Clients connect to the NAT box as internal clients (10.0.0.1 & 10.0.0.2)

METHOD:
In either of the clients use the following format to send a packet. See example below.
Format: *SendCommand destinationIP Packet*
$>> 10.0.0.1 hello [Sending packet from client 10.0.0.0]

RESULTS:
Packets were supposed to have been sent successfully but due to our implementation model the packets are not being sent to another client. The NAT box receives the packets but does not translate them to the internal client.

TEST3:
External client to External client

HYPOTHESIS:
Paquet are dropped as they should be routed by external clients.

VARIABLES:
Two clients connected to the NAT box both as external clients (198.25.55.35 & 45.25.232.15)

METHOD:
Use the same command stated in Test1&2 replacing the destination an external client.

RESULTS:
Paquet were dropped as stated in the hypothesis.

TEST4:
External client to Internal Client

HYPOTHESIS:
The paquet header is routed according to the entry in the NAT table. If there is no corresponding entry of the external client in the NAT table, the paquet is dropped and an error paquet is supposed to be returned.

VARIABLES:
Two clients connected to the NAT box one as an internal (10.0.0.0) and the other as an external (142.25.24.123).

METHOD:
Use the same command stated in Test1&2 replacing the destination with the internal client IP address.

RESULTS:
The paquet were routed accordingly since provided that the External IP address was in the NAT table.

EXPERIMENT 2: REFRESHING NAT TABLE

TEST1:
Refreshing the NAT table at a specific interval.

HYPOTHESIS:
The NAT table is supposed to dynamically refresh the contents in the table and remove the connected IP addresses with the provided time interval.

VARIABLES:
Several connected clients (Internal or External. *Max number of IP addresses is 5.

METHOD:
Run the NAT box using *$java natBox [timeLimit].*
Next then connect several clients to the NAT box. Depending on the provided timeLimit, the NAT box with automatically remove the clients as expected and re-print out the table with all available IP addresses.

RESULTS:
The NAT table produced the desired results.

CONCLUSION

This project helped us understand and highlighted the problem at hand and how important the introduction of NAT box was to society and the internet. Although we did not fully implement the required implementations to the project those that were implemented provided us with a level of understanding of how NAT, IP, and DHCP work.

ISSUES ENCOUNTERED

During the implementation of project, we firstly struggled with the implementation of the overall project, but through discussions and consulting our peers we were able to understand the project framework to a great extent. In our project did not manage to solve the external to internal communication. As required by the project specifications external to internal is not supposed to occur unless the internal client communicated to external client first. In doing so the external client is then supposed to be added to NAT table, however our program adds the external client to the NAT table once connected to the NAT box.