Methods

There are two parts of the translator.

1. Network Address Port Translation and Protocol Translation.
2. DNS Application level gateway.
3. **Network Address Port Translation – Protocol Translation**

Since IPv6 has 128 bit address space and IPv4 has 32 bit address network address translation is required to make communication between IPv4 and IPv6 networks.

IPv64 Node

NAPT - PT

IPv6 Node

IPv6

IPv6 Packet IPv4 Packet

IPv4

DNS - Gateway

(Linux system as Router/Translator)

Fig. 1. Architecture of Translator

When IPv4 address is translating into IPv6 address a 96 bit prefix is added to 32 bit IPv4 address to make it 128 bit IPv6 address.

When IPv6 address is translating into IPv4 address lower 32 bits of IPv6 are taken to make it 32 bit IPv4 address.

Linux system is used as a translator. Two NIC cards are installed, one is connected to IPv4 network and other is connected to IPv6 network. The translation is dynamic as well as static based on requirements.

A module is installed in kernel which captures the packets from both the NIC card and translate the IPv4/IPv6 addresses and forward the new generated packet forward to desired network. If the packet is not recognized by the module, packet is dropped or rejected. An available IPv4 address and port number from pool is assigned to the new connection.

IPv4 - IPv6 Header Translation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Version | IHL | | Type of service | Total Length | | |
| Identification | | | | Flags | | Fragment offset |
| Time to live | | Protocol | | Header Checksum | | |
| Source Address | | | | | | |
| Destination Address | | | | | | |
| Options | | | | | Padding | |

Fig 2.1 IPv4 Header Format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Traffic class | | Flow label | |
| Payload length | | Next Header | | Hop limit |
| Source Address | | | | |
| Destination Address | | | | |

Fig 2.2 IPv6 Header Format

**Fields to be translated :-**

IPv4 Header

* Version :- 4
* Total length :- Payload length of IPv6 Header plus header length of IPv4 header.
* Time To Live:- Hop Limit value from IPv6 header decreased by one
* Protocol:- Next Header copied from IPv6 header.
* Header Checksum:- Computed when the IPv4 header has been created
* Source Address :- In 32 bit format.
* Destination Address :- In 32 bit format.

IPv6 Header

* Version :- 6
* Payload length:- Total length of IPv4 header minus IPv4 header length.
* Next Header:- Protocol field copied from IPv4 header.
* Hop Limit:- Time To Live value from IPv4 header decreased by one
* Source Address :- 128 bit format
* Destination Address:- 128 bit format

Other fields are directly copied or discarded.

Since Main purpose of translator is to make communication between IPv4 and IPv6 network without dual-stack system. Our purpose will fail if we use one-to-one mapping of address i.e. IPv4 address are assigned to each communicating devices, so Port translation is required.

**Port Translation**

In IPv6-IPv4 translation the purpose of port translation is same as in IPv4 port translation. Only one public IPv4 address is assigned to the group of IPv6 machines and they will communicate with IPv4 network by this IP only.

Port from available pool is assigned to every IPv6 machine which tries to communicate through translator. An entry is made somewhere for further communication.

**Protocol Translation**

Protocol translation is of Translation of UDP, TCP and ICMP. In TCP and UDP we need to update checksum of the packet and forward to the desired network.

*new\_cksum = orig\_cksum + (cksum(new\_src\_addr, new\_dst\_addr) - cksum(orig\_src\_addr, orig\_dst\_addr))*

ICMP protocols are different in both networks. For IPv4 it is ICMPv4 and for IPv6 it is ICMPv6. So two kinds of packet are translated into each other.

**Implementation**

System is implemented on Linux machine i.e. Linux machine will act as a IPv6/IPv4 translator gateway.

kernel modules are created and installed in the system and translator machine is connected to the networks. It act as a Translator cum Router which receives the packets, translate them and forward to desired networks.

Following steps taken by translator:-

1. From IPv6 to IPv4
   * Start making connection by adding a 96 bit prefix to the address of destination machine to make it 128 bit IPv6 address.
   * Packet is received by translator and added 96 bit is removed by translator and an available IPv4 address and port number from pool is assigned to the machine.
   * Packet is forwarded to IPv4 network.
2. From IPv4 to IPv6
   * IPv4 packet is received by translator and the table is searched for original destination of packet i.e. IPv6 address and port number.
   * 96 bit prefix is added to source IPv4 address and destination address is fetched from the allotted addresses.
   * Packet is forwarded to IPv6 network.

This system is tested on IIT-Kanpur Network lab for following applications.

* Http/Https
* SSH
* Telnet
* Ftp (Passive mode)
* nslookup

**Translator is tested on following configuration**

Translator IPv6 address :- 4001:4490::c0a8:103

Translator IPv4 address :- 172.31.210.225

IPv6 machine address :- 4001:4490::c0a8:105

Now IPv6 machine tries to communicate with multiple IPv4 network over LAN and Internet.

**Packet from IPv6 Machine:-**

Source:- 4001:4490::C0Aa:103 Source port :- 15432

Destination:- 4001:4490::172.31.1.212 Destination port :- 80

**After Translation :-**

Source:- 172.29.0.1 Source port :- 5000 **(From Pool)**

Destination:- 172.31.1.212 Destination port :- 80

**Packet from IPv4 Machine:-**

Source:- 172.31.1.212 Source port :- 80

Destination:-172.29.0.1 Destination port :- 5000

**After Translation :-**

Source:- 4001:4490::172.31.1.212 Source port :- 80 **(From Table)**

Destination:- 4001:4490::c0a8:103 Destination port :- 15432

Prefix :- 4001:4490::/96

**Multithreaded DNS – Gateway**

DNS Gateway is a mechanism for synthesizing AAAA records from A records. DNS Gateway is used with an IPv6/IPv4 translator to enable client-server communication between an IPv6-only client and an IPv4-only server, without requiring any changes to either the IPv6 or the IPv4 node, for the class of applications that work through NATs.

DNS gateway add 96 bit prefix to A records to make it AAAA records. When IPv6 machine tries communicate with IPv4 address machine ask DNS for AAAA records but there is no entry for such records then DNS gateway returns a AAAA record to IPv6 client.

**Steps Involved**

* IPv6 client send AAAA query to DNS gateway.
* DNS gateway change AAAA query into A query and send it to DNS.
* DNS gateway receive the response from DNS.
* DNS gateway add 96 bit address in 32 bit address response from the server and make the query AAAA type.
* Now send it back to IPv6 client.
* Client receives the AAAA response and it will think it is communicating with IPv6 machine.

Example :-

IPv6 machine with IP address 4001:4490::c0a8:105 want to know the AAAA address if [www.iitk.ac.in](http://www.iitk.ac.in).

* Gateway changes the query into A and send it to DNS.
* Response from DNS is 172.31.1.212 type A.
* The address is changed to 4001:4490::172.31.1.212 type AAAA

Now IPv6 client understands the address and start communicate with [www.iitk.ac.in](http://www.iitk.ac.in) with the help of IPv6/IPv4 translator gateway.

Since the response of DNS is large and sometimes it contains many answers and it takes some time to search the CNAME, IP address from the response. So gateway will be busy until it gives back response to IPv6 client. If there are multiple queries then it decreases the efficiency of DNS gateway.

To overcome this the concept of Multithreading is used. A new thread is created for every query and client. By multithreaded DNS gateway multiple queries will be resolved at same time.