

IP – The Future?

WJ Buchanan, SoC, Napier University

ABSTRACT

The Internet has been extremely successful, and is now used for applications which could never be conceived when it was initially developed. One of the major problems is that there is not enough IP addresses around to cover all the devices which might want to connect to the Internet. Thus mechanisms are being put in-place to migrate towards a future system which allows an increased number of IP addresses. There are also security weaknesses in the current version of IP addresses (IPv4), as the source of a host can be easily identified as the source address is added to the IP data packet. The main proposed standard is IPv6, which tries to overcome some of the current problems. It increases the addressing range with a 128-bit address, and thus will allow for an almost unlimited amount of addresses. This new standard will also allow autoconfiguration of network addresses, which should make it easier for devices to connect to the Internet. It will take a while before the Internet can be fully IPv6, thus this paper looks at other methods, including NAT/PAT, proxy servers and VPNs, which allow increased security, by hiding the source, and support private addressing structures.

Keywords: IPv4, IPv6, NAT, PAT, Proxy servers, VPNs

1. Introduction

The Internet has been more successful than could ever have been imagined. We have thus run out of precious IP addresses, and need to migrate towards an addressing system which makes the generation of IP addresses simpler; has more security, and will allow more addresses.

The IP header (IP Ver4) is added to higher-level data (as defined in RFC791). This header contains a 32-bit IP address of the destination node. Unfortunately, the standard 32-bit IP address is not large enough to support the growth in nodes connecting to the Internet. Thus a new standard, IP Version 6 (IP Ver6, aka, IP, The Next Generation, or IPng), has been developed to support a 128-bit address, as well as additional enhancements, such as authentication and data encryption. This paper discusses some of the new principles which are likely to be adopted to ease the transition to the future.

2. IPv6

The RFC1883/2373 specifications [1,2] outline the main changes as:

- **Expanded addressing capabilities.** The size of the IP address will be increased to 128 bits, rather than 32 bits. This will allow for more levels of addressing hierarchy, an increased number of addressable nodes and a simpler auto-configuration of addresses. With multicast routing, the scalability is improved by adding a scope field to the multicast addresses. As well as this, an anycast address has been added so that packets can be sent to any one of a group of nodes.
- **Improved IP header format.** This tidies the IPv4

header fields by dropping the least used options, or making them optional.

- **Improved support for extensions and options.** These allow for different encodings of the IP header options, and thus allow for variable lengths and increased flexibility for new options.
- **Flow labeling capability.** A new capability is added to enable the labeling of packet belonging to particular traffic *flows* for which the sender requests special handling, such as non-default quality of service or *real-time* service.
- **Authentication and privacy capabilities.** Extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv6.

2.1 Autoconfiguration and multiple IP addresses

IPv4 requires a significant amount of human intervention to set up the address of each of the nodes. IPv6 improves this by supplying autoconfiguration and renumbering facilities, which allows hosts to renumber without significant human intervention.

IPv4 has a stateful address structure, which either requires the user to manually set up the IP address of the computer or to use DHCP servers to provide IP addresses for a given MAC address. If a node moves from one subnet to another, the user must reconfigure the IP address, or request a new IP address from the DHCP. IPv6 supports a stateless autoconfiguration, where a host constructs its own IPv6. This occurs by adding its MAC address to a subnet prefix. The host automatically learns which subnet it is on by communicating from the router which is connected to the network that the host is connected to.

IPv6 supports multiple IP addresses for each host. These addresses can be either *valid*, *deprecated* or *invalid*. A *valid*

address would be used for new and existing communications. A *deprecated* address could be used only for the existing communications (as they perhaps migrated to the new address). An invalid address would not be used for any communications. When renumbering, a host would deprecate the existing IP address, and set the new IP address as valid. All new communications would use the new IP address, but connections to the previous address would still operate. This allows a node to gradually migrate from one IP address to another.

2.2 IPv6 header format

Figure 1 shows the basic format of the IPv6 header. The main fields are:

- Version number (4 bits) – contains the version number, such as 6 for IP Ver6. It is used to differentiate between IPv4 and IPv6.
- Priority (4 bits) – indicates the priority of the datagram, and gives 16 levels of priority (0 to 15). The first eight values (0 to 7) are used where the source is providing congestion control (which is traffic that backs-off when congestion occurs), these are:

- 0 defines no priority.
- 1 defines background traffic (such as netnews).
- 2 defines unattended transfer (such as e-mail), 3 (reserved).
- 4 defines attended bulk transfer (FTP, NFS), 5 (reserved).
- 6 defines interactive traffic (such as telnet, X-windows).
- 7 defines control traffic (such as routing protocols, SNMP).

The other values are used for traffic that will not back off in response to congestion (such as real-time traffic). The lowest priority for this is 8 (traffic which is the most willing to be discarded) and the highest is 15 (traffic which is the least willing to be discarded).

- **Flow label (24 bits)** – still experimental, but will be used to identify different data flow characteristics. It is assigned by the source and can be used to label data packets which require special handling by IPv6 routers, such as defined QoS (Quality of Service) or real-time services.
- **Payload length (16 bits)** – defines the total size of the IP datagram (and includes the IP header attached data).
- **Next header** – this field indicates which header follows the IP header (it uses the same IPv4). For example: 0 defines IP information; 1 defines ICMP information;

6 defines TCP information and 80 defines ISO-IP.

- **Hop limit** – defines the maximum number of hops that the datagram takes as it traverses the network. Each router decrements the hop limit by 1; when it reaches 0 it is deleted. This has been renamed from IPv4, where it was called time-to-live, as it better describes the parameter.
- **IP addresses (128 bits)** – defines IP address. There will be three main groups of IP addresses: unicast, multicast and anycast. A unicast address identifies a particular host, a multicast address enables the hosts within a particular group to receive the same packet, and the anycast address will be addressed to a number of interfaces on a single multicast address.

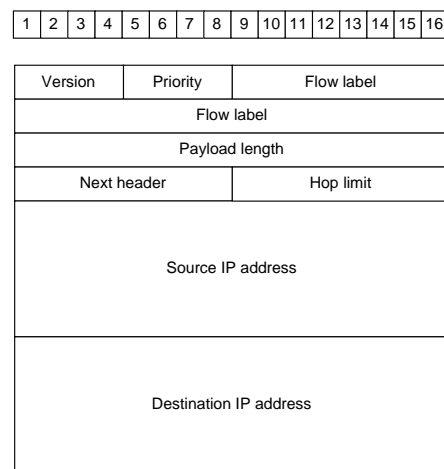


Figure 1 IP Ver6 header format

IPv6 has a simple header, which can be extended if required. These are:

- Routing header.
- Fragment header.
- Authentication header.
- Encrypted security payload.
- Destinations options header.

IPv6 addresses do not use the dotted notion and are written in a hexadecimal format, such as:

114F: 0000: 0000: 0000: 0006: 0600: 4411: CB1D

Often the leading zero's are omitted to give:

114F: 0: 0: 0: 6: 600: 4411: CB1D

This address can be shorted further by converting all zero values to a double colon, to give:

114F::6:600:4411:CB1D

The unicast address contains 128 bits, and has the following fields:

- **Field Prefix (FP) field (3 bits).** This identifies when the address is unicast, multicast, and so on). A value of 001 identifies aggregatable global unicasts.
- **Top-Level Aggregation Identifier (TLA ID field) (13 bits).** This is used to identify the authority responsible for the address at the highest level of the routing hierarchy.
- **Res field (8 bits).** This is reserved so that the TLA or NLA IDs can be expanded for future use.
- **NLA ID field (24 bits).** This is used to identify ISPs, and can be organized to reflect a hierarchy, or multi-tiered relationship, among providers.
- **SLA ID field (16 bits).** This is used by individual organizations in order to defined a local addressing hierarchy and to identify subnets.
- **Interface ID field (64 bits)** – This uses an IEEE EUI-64 format and is a unique ID for the network interface. In Ethernet-type networks, it uses the 16 bits from the MAC address of the network port.

The unicast address 0:0:0:0:0:0:0:1 is called the loop-back address, in compressed form it will be ::1.

2.3 IPv6 tunnelling over IPv4

It will take a long time before the Internet is fully defined by IPv6, thus devices which use IPv6 can still communicate over an IPv4 using IPv6 tunnelling over IPv4. This involves encapsulating an IPv6 packet in IPv4 packet.

3. Network Address Translation

Network address translation (NAT) is defined in RFC1631, and swaps one network address with another. This allows private networks (RFC1918) to be created, which are then translated to public address when they access the Internet. A router can operate at the border of a domain and translate addresses from private to public, and vice-versa. For example, a node could be given a private address of 192.168.10.12. The NAT could then translate this to a public address of 168.10.34.21. The NAT table would then have the mapping of:

Private	Public
192.168.10.12	168.10.34.21

If a host from outside the domain sends a data packet back to the domain, the NAT will translate the public address back into the private address. These translations can be statically assigned, such as where it is setup with a permanent mapping, or dynamically, where the tables can

change as the network requires. Figure 2 gives an example, where the destination address is 11.22.33.44. The address in this case is changed from 192.168.10.12 to 168.10.34.21, as the data packet goes out of the domain, and is changed back when it comes back into the domain.

NAT routers can use port address translation (PAT), which allows many internal address to be mapped to the same global address. This is also named as a *many-to-one* NAT, or address overloading. With PAT, the NAT router keeps a track of the connections, and the TCP/UDP ports that are being used. The NAT router then changes the global address back into a private address based on these.

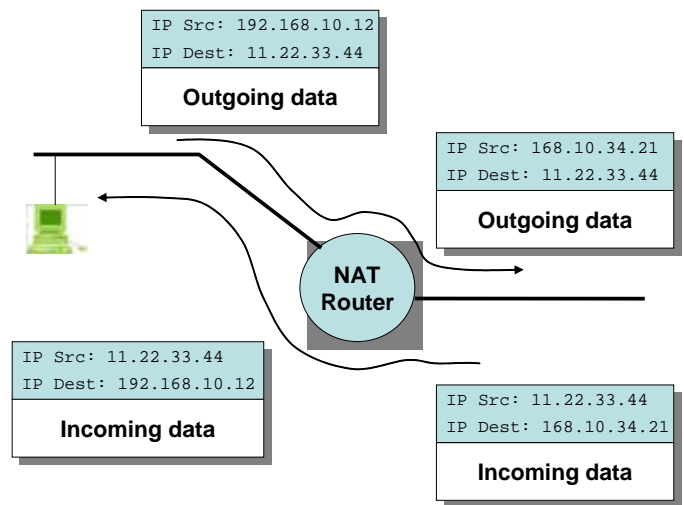


Figure 2 Example of NAT

It has the following advantages:

- NAT thus allows a virtual network to be created from within an organisation. It thus allows the easy creation of subnetworks and increases the range of addresses (Figure 3).
- NAT also enhances security as it limits external users in their connection to local network, as the translations of addresses will not be permanent (unless a static translation is implemented). NAT thus hides the topology of the network.
- NAT allows for increased mobility, as addresses can be allowed on an ad-hoc basis. This is especially important in a wireless communications, where the network address of a device can change is it connects to different wireless domains.

The Appendix gives an example of how NAT is programmed on a device.

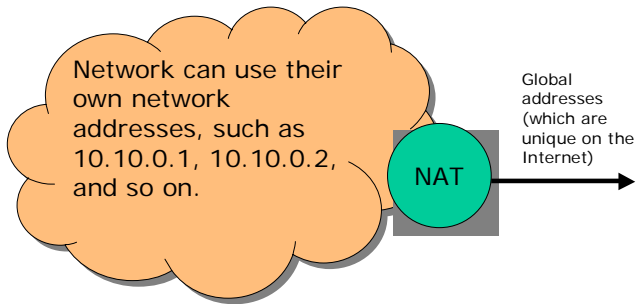


Figure 3 Hiding the network

4. Proxies

Proxies are intermediate devices which make requests to the Internet on behalf of a device. Thus, the internal network can have its own addressing structure, and the proxy is the one that must have a proper Internet address.

Proxies can thus give connectivity of nodes on a private network to a public one, and can allow for enhanced security. In Figure 4, Host1 makes a direct connection to the Internet through the Proxy. It does this by communicating the destination address and port to a specific port on the proxy server, which will then forward the request onto the destination. In this case, the proxy is setup to receive this traffic on port 1001. It then forwards this data onto the Internet.

In this case, 102.10.10.3 is a real Internet address, while the 192.168.0.x network is a private network, which cannot directly connect onto the Internet. The proxy thus offers Internet connectivity to any node on this network, through the proxy server.

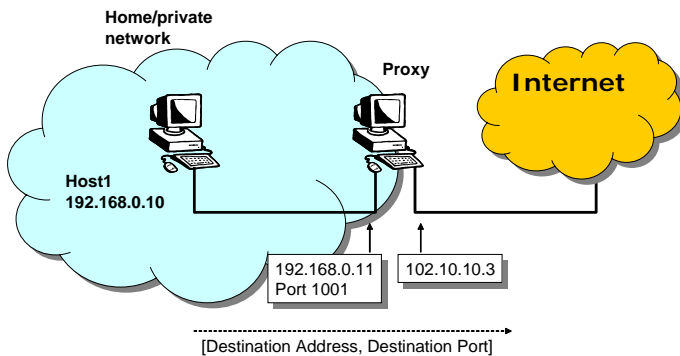


Figure 4 Proxy server

5. Virtual Private Networks

Virtual Private Networks (VPNs) use the Internet as a communications channel to link up two networks. The data on one network is then encapsulated in IP packets which are then sent over the Internet, and then are opened-up and put on the destination network. The two connected networks will thus act as a single virtual network. For security, the data can be encrypted. The two

networks can thus use private addresses, where only the network connects require Internet addresses. A common device which is used to connect users to a remote network is a Remote Access Server (RAS).

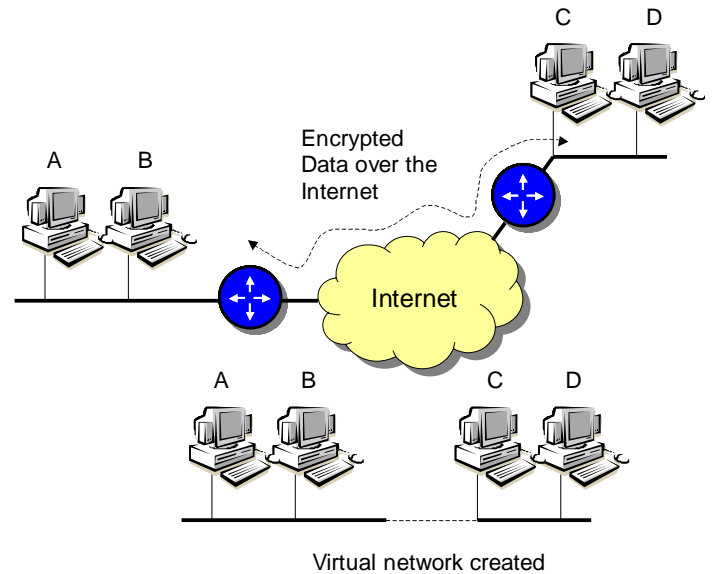


Figure 5 VPN

6. Conclusions

This paper has shown that the current version of IP has several weaknesses, including its lack of security, and its limited range. The future may be towards IPv6, but this will take some time to properly implement. At present there are several methods which can be used to overcome some of these weaknesses, these include NAT/PAT, Proxies and VPNs. These work well, and integrate with the current version of the Internet. A key element of these is that organisations can define private addresses, which then map to a public address. These private addresses are easier to administer, and also can be hidden from the rest of the Internet.

7. Author

Dr WJ Buchanan, SoC, Napier University

8. References

- [1] RFC 1884 - IP Version 6 Addressing Architecture Authors: R. Hinden, S. Deering. Date: December 1995.
- [2] RFC 2373 – Authors: R. Hinden , S. Deering.

9. Appendix

Network address translation allows private IP address to be translated to public address. This can either be achieved statically, where the translation is fixed by a translation table, or can be dynamic, where the translation

table is set-up as required by the network. Typically, a global address pool is used from which the public addresses are taken. The command for this has the format of:

```
RouterA# config t
RouterA(config)#ip nat pool name start-ip end-ip
    {netmask netmask | prefix-length prefix-length}
```

where the submask length is defined by the optional `netmask` argument (such as 255.255.255.0), or by a length using `prefix-length` (or 24 for the 255.255.255.0 subnet mask). After this, the types of packets which will be translated will be defined. This is achieved with the `access-list` command, and has the form:

```
RouterA# config t
RouterA(config)#access-list access-list-number permit source [source-wildcard]
```

A dynamic translation uses the `ip nat inside source list` command, such as:

```
Router(config)#ip nat inside source list access-list-number pool name
```

where the access list number is defined. This is then applied to one of the interfaces using the command (for s0):

```
RouterA# config t
RouterA (config) # int s0
RouterA (config-if)#ip nat inside
```

This will translate data packets which are coming into the port. To translate outgoing one, the `ip nat outside` command is used.

For example, to define a pool of addresses from 180.10.11.1 to 180.10.11.254:

```
RouterA(config)#ip nat pool org_pool 180.10.11.1
    180.10.11.254 netmask 255.255.255.0
```

which defines the global addresses as `org_pool`. This will be used to send translated data packets out in the Internet. An `access-list` command is then used to match the translation addresses:

```
RouterA(config)#access-list 2 permit 192.168.10.0
    0.0.0.255
RouterA(config)#ip nat inside source list 2 pool
    org_pool
```

which applies the access-list number 2 to the IP NAT pool of `org_pool`. This can then be applied to the interfaces with:

```
RouterA(config)#interface e0
RouterA(config-if)#ip nat inside
RouterA(config-if)#interface s0
```

```
RouterA(config0if)#ip nat outside
```

Thus if a host with an address of 192.168.10.10 sends a data packet out of the network, it will have one of the addresses from the pool, such as 180.10.11.1. All the hosts outside the network will use the address from the pool to communicate with the node. By default, these entries remain in the table for up to 24 hours (in order to allow communications to return). The time-out can be changed using the command:

```
RouterA(config)#ip nat translation timeout seconds
```

This is an important factor, especially when there is a large number of hosts which can only use a limited pool of addresses. A lower time-out will allow an address to be released, so that another node can use it.

NAT also enhances security as it limits external users in their connection to local network, as the translations of addresses will not be permanent (unless a static translation is implemented). NAT thus hides the topology of the network.

Static translation uses a fixed lookup table to translate the addresses, where each address which requires an Internet address has a corresponding public IP address. If it is used on its own, it cannot thus preserve IP address. Thus, typically the two methods are used, where important nodes, such as servers, will have a static entry, as this guarantees them an address, while other nodes, which are less important, will be granted a dynamic translation. This also aids security as the important devices can run enhanced security and monitoring software, which might not be possible on lower-level devices, which are typically administered on a daily basis by non-IT personnel.

Static addresses are also useful in translating network topologies from one network address structure to another, or even when individual nodes are moved from one subnet to another.

An example of configuring for static addresses of a node of 192.168.10.10 to the address of 180.10.11.1:

```
RouterA(config)#ip nat inside source static
    192.168.10.10 180.10.11.1
```

This can this be applied to the inside and outside interfaces with:

```
RouterA(config)#interface e0
RouterA(config-if)#ip nat inside
RouterA(config-if)#interface s0
RouterA(config-if)#ip nat outside
```

NAT allows organisations to quickly remap their addresses, as conditions require, such as changing Internet access provider, or to respond to a network breach.

One of the advanced features of NAT routers is their

ability to use Port Address Translation (PAT), which allows multiple inside addresses to map to the same global address. This is sometimes called a *many-to-one* NAT, or *address overloading*. With address overloading, many private addressed nodes can access the Internet using a single global address. The NAT router keeps track of the different conversations by mapping TCP and UDP port numbers in the translation table. A translation entry is one which maps one IP address and port pair to another, and is called an extended *table entry*. This table will match internal private IP addresses and ports, to the global address.

The NAT command is used to configure PAT with:

```
RouterA(config)#ip nat inside source list access-list-  
number pool name overload
```

For example, if a network has 20 IP global addresses from

180.10.11.1 to 180.10.11.20, then the router could be configured with:

```
RouterA(config)#ip nat pool org_pat_pool  
180.10.11.1 180.10.11.20 netmask 255.255.255.0  
RouterA(config)#access-list 2 permit 10.1.1.0  
0.0.0.255  
RouterA(config)#ip nat inside source list 2 pool  
org_pat_pool overload  
RouterA(config)#interface e0  
RouterA(config-if)#ip nat inside  
RouterA(config-if)#interface s0  
RouterA(config-if)#ip nat outside
```

This creates an access-list with a label of 2, which is applied using the overload method, to provide PAT. This method is obviously important in a home network, where users are granted an IP address for their router. The home network can then be setup with private addresses.