# 1

# IPv6 Rationale and Features

Back in the 1970s, the Internet Protocol (IP) was designed upon certain assumptions and key design decisions. After more than 25 years of deployment and usage, the resulting design has been surprisingly appropriate to sustain the growth of the Internet that we have seen and continue to see; not only the increase of the number of devices connected, but also of the kinds of applications and usage we are inventing everyday. This sustainability is a very impressive achievement of engineering excellence.

Despite the extraordinary sustainability of the current version (IPv4), however, it is suffering and the Internet Protocol needs an important revision. This chapter describes why we need a new version of the IP protocol (IPv6), by describing the Internet growth, the use of techniques to temper the consequences of that growth and the trouble experienced in deploying applications in current IPv4 networks. Some architecture considerations are then discussed and new features needed in current and future networks presented.

Next, the work towards IPv6 at the IETF is shown along with the key features of IPv6. Some milestones are also tabled. Finally, the IPv6 return on investment and drivers is discussed.

## 1.1 Internet Growth

The origin of IPv6 work lay in the imminent exhaustion of address space and global routing table growth; both could be summarized as Internet growth.

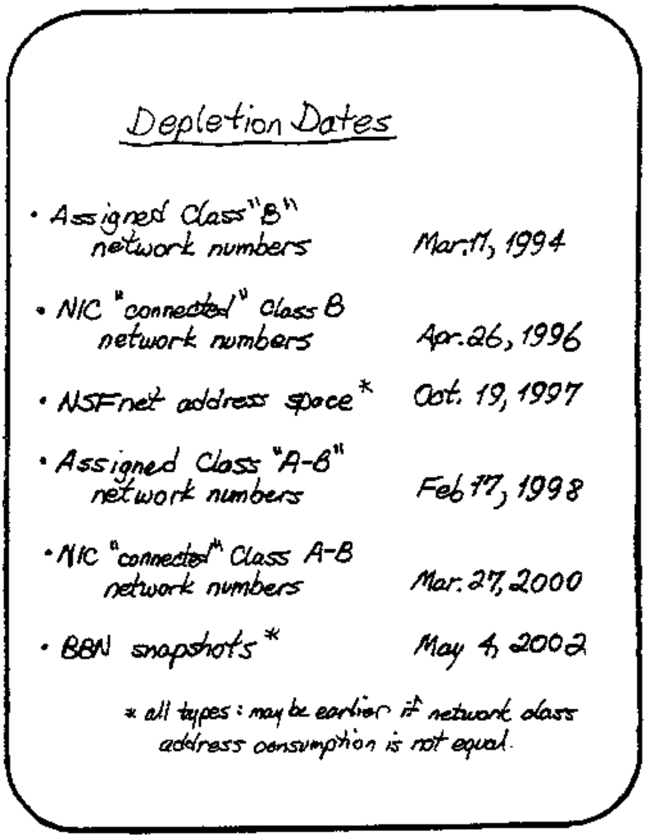
### *1.1.1 IPv4 Addressing*

The Internet is a victim of his own success. No one in the 1970s could have predicted this level of penetration into our lives.

*Migrating to IPv6: A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks* Marc Blanchet
© 2006 John Wiley & Sons, Ltd

In theory, 32 bits of IPv4 address space enables 4 billion hosts. Studies [RFC1715] have shown that the effectiveness of an address space is far less. For example, RFC1715 defines a H ratio as: H = log (number of objects using the network)/number of bits of the address space. Based on some empirical studies of phone numbers and other addressing schemes, the author concluded that this H ratio usually never reaches the value of 0.3, even with the most efficient addressing schemes. An optimistic H ratio is 0.26 and a pessimistic one (for not very efficient addressing schemes) is 0,14. At H = 0.26, with an addressing of 32 bits, the maximum number of objects, in the case of IPv4 the number of reachable hosts, is 200 000 000.[1] When IPv4 Internet reaches 200 million reachable nodes, the IPv4 addresses will be exhausted.

Moreover, the IPv4 address space was designed with three classes (A, B and C)[2] which makes the address space usage even less efficient than with the optimistic H ratio. In August 1990 at Vancouver IETF, a study [Solensky, 1990] demonstrated the exhaustion of class B address space by March 1994. Figure 1.1 shows the summary slide presented during that IETF. This was an important wakeup call for the whole Internet engineering community.



Depletion Dates

| | |
|---|---|
| • Assigned Class "B" network numbers | Mar. 11, 1994 |
| • NIC "connected" Class B network numbers | Apr. 26, 1996 |
| • NSFnet address space * | Oct. 19, 1997 |
| • Assigned Class "A-B" network numbers | Feb 17, 1998 |
| • NIC "connected" Class A-B network numbers | Mar. 27, 2000 |
| • BBN snapshots * | May 4, 2002 |

* all types: may be earlier if network class address consumption is not equal.

**Figure 1.1** Solensky slide on IPv4 address depletion dates

[1] RFC1715 was also used as input to define the IPv6 address length to 128 bits.

[2] D and E classes also exist but are not for unicast generic use.

At that time, most organizations requesting an address space pretty easily obtained a class B address block, since there was plenty of IPv4 address space. Assigning class C address blocks to organizations was the first cure; it decreased the initial address consumption problem but introduced more routes in the global routing table, therefore creating another problem.

### *1.1.2 IPv4 Address Space Utilization*

Let's talk about the current IPv4 address space utilization. The IPv4 address space is 32 bits wide. IANA allocates by 1/256th (0.4%) chunks to regional registries, which corresponds to a /8 prefix length or to the leftmost number in an IPv4 address. Since the 224.X.X.X to 239.X.X.X range is reserved for multicast addressing and the 240.X.X.X to 254.X.X.X range is the experimental class E addressing, the total unicast available address space is of 223 /8 prefixes.

Figure 1.2 shows the cumulative number of /8 prefixes allocated since the beginning of IPv4. At the end of 2004, there are 160 /8 prefixes allocated, representing 71% of the total unicast available address space.

In 2003, 5 /8 prefixes were allocated by IANA to the regional registries. In 2004, 9 /8 prefixes were allocated (80% annual increase). In January 2005 alone, 3 /8 prefixes were allocated. If every year after 2004, we are flattening the annual consumption to the 2004 number (9 /8 prefixes: i.e. 0% annual increase for the next 7 years), then Figure 1.3 shows the exhaustion of IPv4 address space (223 /8 prefixes) by 2011.

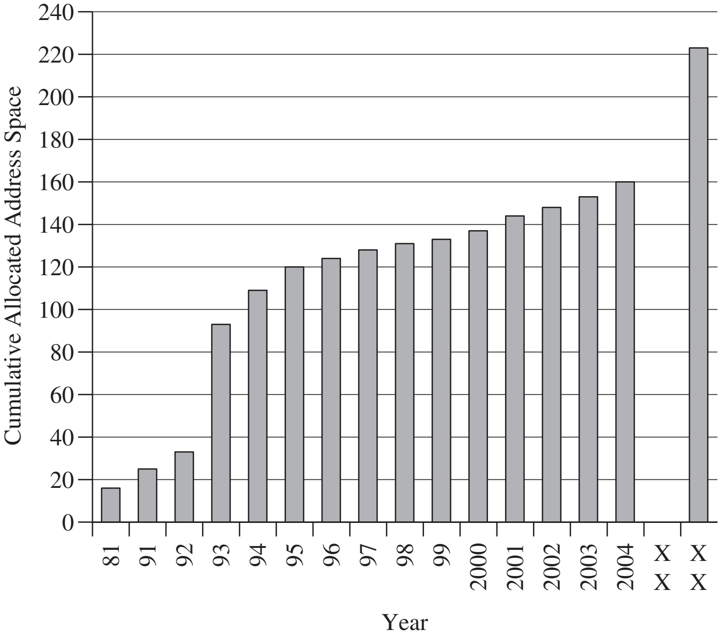


**Figure 1.2** IPv4 cumulative allocated address space as of 2004–12

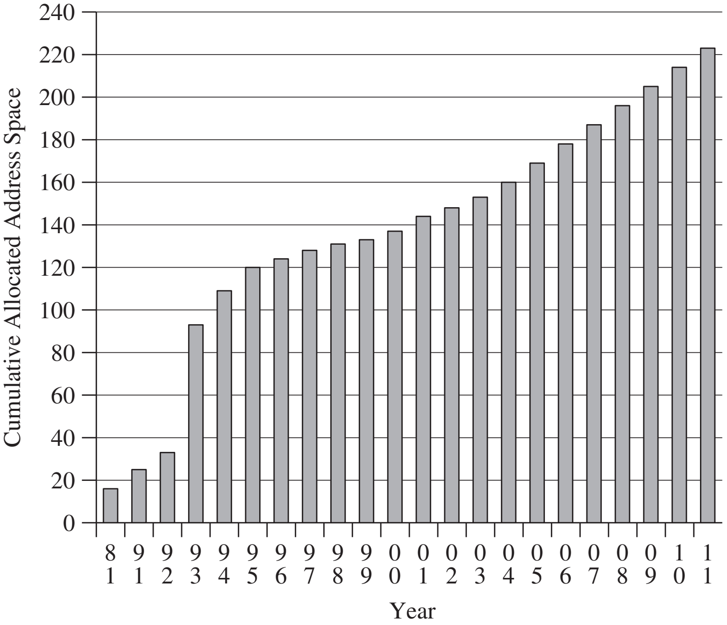


**Figure 1.3** Prediction of IPv4 allocated address space with flat annual consumption

If we are slightly more aggressive by increasing the annual consumption by 2 additional /8 prefixes every year after 2004, which results in an annual increase of 22%, then Figure 1.4 shows the exhaustion of IPv4 address space by 2009.

A 20% annual increase is pretty conservative, given that:

- large populations in China, India, Indonesia and Africa are not yet connected;
- world population net annual growth is 77 million people [Charnie, 2004];
- all kinds of electronic devices are increasingly being connected and always on;
- broadband connections incur permanent use of addresses instead of temporary addresses when dialing up;
- each 3G cell phone consumes at least one IP address.

On the other hand, mitigating factors may delay this exhaustion:

- some class A are assigned but not used and therefore could be reclaimed;
- as in economics, the rarer something is, the more difficult it is to get and more it costs, slowing the exhaustion but instead creating an address exchange market.

Despite this, the IPv4 address shortage is already happening, and severely, because

- organizations usually get just a few addresses (typically 4) for their whole network, limiting the possibilities of deploying servers and applications;
- some broadband providers are giving private address spaces to their subscribers, which means the subscriber computers cannot be reached from the Internet.

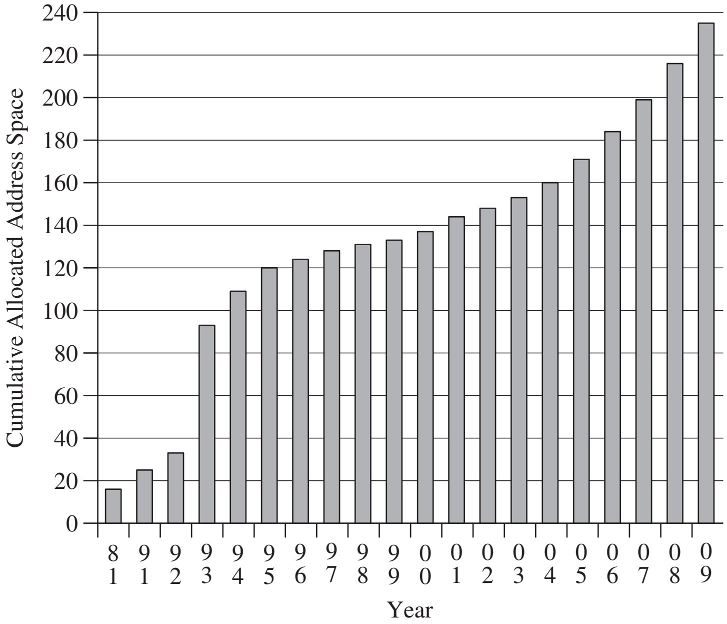


**Figure 1.4** Prediction of IPv4 allocated address space with incremented annual consumption

### *1.1.3 Network Address Translation*

The most important change regarding IP addressing is the massive use of Network Address Translation (NAT). The NAT functionality is usually implemented within the edge device of a network, combined with firewalling. For example, most organization networks have a firewall with NAT at the edge of their network and most home networks have a home router which implements firewalling and NAT.

NAT maps multiple internal private IP addresses to a single external IP address.[3] By allocating new external port numbers for each connection, essentially this NAT mapping process extends the address space by adding 16 bits of the port address space.

Figure 1.5 shows a basic network diagram of a private network with 2 computers (N1 and N2) and a public network, such as the Internet with one server (S). The private network uses private address space [RFC1918]. When internal nodes N1 and N2 connect to server S, the source addresses (10.0.0.3, 10.0.0.4) of the packets are translated to the NAT external IP address (192.0.2.2) when the packet is traversing the NAT. Server S receives connections coming from the same single source address (192.0.2.2), as if it comes from one single computer.

Table 1.1 shows how the detailed process works based on Figure 1.5. When the packet traverses the NAT, the source IP address and port are translated to the external IP address of the NAT and a new allocated port, respectively. For example, N1 source IP address 10.0.0.3 is translated to 192.0.2.2 and the source port 11111 is translated to the new allocated

[3] NAT can map multiple internal addresses to more than one external address, but for simplication we are discussing the most current used case: multiple internal to a single external address.

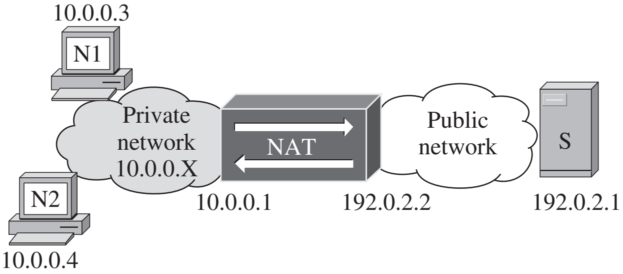


**Figure 1.5** NAT basic network diagram

**Table 1.1** NAT changing the source IP address and port number

| Flow | Packet header while in private network | | | | Packet header while in public network | | | |
|---|---|---|---|---|---|---|---|---|
| | Source IP address | Source port number | Destination IP address | Destination port | Source IP address | Source port number | Destination IP address | Destination port |
| N1 to S | 10.0.0.3 | 11111 | 192.0.2.1 | 80 | 192.0.2.2 | 32001 | 192.0.2.1 | 80 |
| N2 to S | 10.0.0.4 | 22222 | 192.0.2.1 | 80 | 192.0.2.2 | 32002 | 192.0.2.1 | 80 |

external port 32001 by the NAT. The mapping is kept inside the NAT for the lifetime of the connection. If the connection is to get a Web page from the HTTP server S, then the mapping will remain for the duration of the GET request. Any new connection, even to the same server, creates a new mapping. Also, Table 1.1 shows another connection from N2. From the S perspective, the two connections have the same source IP address, so they appear to come from the same source.

This translation technique effectively hides the nodes on the private network and conserves the public IP address space. The private address space [RFC1918] enables a very large network having millions of nodes hidden behind a single IP address, given that the total number of simultaneous connections is less than 65 K, since one connection takes one external port and port numbers are 16 bits wide. The proliferation of NATs enabled the whole Internet to continue growing at a much higher rate than the actual consumption rate of the IPv4 address space.

However, NAT does not come free. When internal nodes are using application protocols that negotiate the IP address and/or port numbers within the application protocol, then the application in server S will receive the private address of the node, not the public translated address by the NAT. The server application will then reply to the private address which is not reachable and routable from the public network. Therefore, the application does not work. For example, FTP [RFC959] with its separate control and data connections does not work if a NAT is in the path between the server and the client.

To overcome this limitation, the NAT must understand each application protocol that traverses it, inspect each application payload and modify the application payload to replace the private source address and port number by the external source IP address and port number.

This processing at the application layer is called an application level gateway (ALG). Every NAT implementation includes a FTP ALG to enable this widely used protocol to traverse NATs. An ALG does not work if the application payload is encrypted or integrity protected by the application protocol or by a layer below such as IPsec.

Moreover, when the IP header itself is integrity protected, for instance with the IPsec AH mode, then the translation of the source IP address and port number destroys the integrity protection.

NAT and its side effects are discussed more throughout this book.

### *1.1.4 HTTP Version 1.1 Virtual Hosting*

The simplicity of HTML and Web servers generated a lot of interest in the 1990s when everyone wanted to have their own Web server. This resulted in a very rapid growth of Web servers. Version 1.0 of the HyperText Transfert Protocol (HTTP) [RFC1945] required each Web site to have a specific public IP address. To aggregate resources, many Web sites are hosted on the same server, requiring the operating system to support multiple IPv4 address on the same interface, usually named secondary IP addresses.[4] This increased the consumption rate of IP addresses.

Version 1.1 [RFC2068] of HTTP supports virtual hosting, where multiple Web sites with different domain names (http://www.example1.com, http://www.example2.com) are served by the same IP address. A version 1.0 HTTP client sends only the path at the right of the domain name (for example: path=/a/b.html of the full URL: http://www.example1.com/a/b.html) to the HTTP server. A version 1.1 HTTP client sends the full hostname to the HTTP server (for example: http://www.example1.com/a/b.html), enabling the HTTP server to forward appropriately the request to the proper Web site handler. With version 1.1 of HTTP, the Web server now needs only one IP address to serve a virtually unlimited number of Web sites.

However, with this virtual hosting technique, IP filtering based on the address of the destination Web server is nearly impossible, since all Web sites share the same IP address. The filtering has to be done at the application level, requiring filtering devices to open the packet payload to inspect and parse the HTTP statements in order to identify the target Web server, which creates more burden on security gateways.

Compared to HTTP version 1.0, HTTP version 1.1 conserves the public IP address space by enabling virtual hosting.

In a typical enterprise scenario, the enterprise needs only two IP addresses: the external address of the NAT hiding its internal network and one address for all its Web sites. This created the defacto ISP practice to provide only four IPv4 addresses to organizations. The bad side effect is that the organizations now have to justify the need for more than four IPv4 addresses, moving the burden of allocation and usage of the IPv4 address space to the organization. Does your organization have to justify the need for more than four telephone numbers?

### *1.1.5 Variable Length Subnet Mask*

In organization networks, the original IPv4 design requires a single subnet mask throughout the network. An address plan identifies the subnet mask by finding the largest possible

[4] Secondary IP addresses were not supported on many OS at that time, which gave more headaches to network managers who hosted many Web sites.

number of hosts on a single subnet and using the according bit-level subnet mask. If the largest number of hosts on a single subnet in a network is 65, then the network will have a 7 bits subnet mask ($2^7 = 128 > 65$), which enables 128 hosts on each subnet. A subnet of 3 nodes in that network consumes 128 IPv4 addresses. A single subnet mask for the whole network decreases the efficiency of the IP address utilization.

To make address plans less sparse, resulting in conservation of address space, the variable length subnet mask technique (VLSM) [RFC1812] was introduced for routers and routing protocols. With VLSM, the routing infrastructure can handle a specific subnet mask on each subnet. Routing protocols such as the Routing Information Protocol (RIP) [RFC1058] had to be updated to support VLSM.

### *1.1.6 Classless IPv4*

Introduced to reduce the growth of the global routing table, the Classless Inter-Domain Routing (CIDR) [RFC1519] converts the classful IPv4 address space into a classless address space. In the classless model with CIDR, any network size is possible using any address number. The A class (from 0.0.0.0 to 127.255.255.255 with 24 bits each), the B class (from 128.0.0.0 to 191.255.255.255 with 16 bits each) and C classes (from 192.0.0.0 to 223.255.255.255 with 8 bits each) are no longer relevant. Therefore, the address consumption rate is decreased since allocation will be more effective.

Owing to the variable size of network prefixes, the /n notation (often named CIDR notation) after the prefix was introduced to make IP address writing notation shorter and more efficient. For example, '/25' in 192.0.2.0/25 identifies the number of significant leftmost bits in the address. Therefore, '/25' gives a $32 - 25 = 7$ bits prefix range, which gives $2^7 = 128$ addresses. '192.0.2.0/25' defines the range of addresses between 192.0.2.0 and 192.0.2.127. The CIDR notation is the only notation used to describe ranges or prefixes of IPv6 addresses and is the one used throughout this book for both IPv4 and IPv6.

### *1.1.7 Provider-based Assignment and Aggregation of IPv4 Network Prefixes*

Since the beginning of IPv4, organizations had been requesting IPv4 address space directly from IANA and IANA assigned a IPv4 address range of one class. This range was assigned to this organization permanently. These assignments were not related to the topology of the network, disabling any aggregation of the prefixes by a common provider. This is another cause of the growth of the global routing table.

In 1994, the policy of assignments changed. The new policy enforces provider-based assignments to the organizations. Now, IANA assigns blocks of addresses to regional registries (ARIN, RIPE, APNIC and others). Within their assigned blocks, regional registries assign smaller blocks of addresses to providers, which in turn assign smaller blocks to organizations. In this context, organization's prefixes are aggregated at the provider level, resulting in a more aggregated global routing table and decreased rate of growth of the table. This aggregation also has the benefit of more stability in the routing table, since organization's prefixes are not specifically announced in the global routing table. Since the leaf of the network (such as the link connecting the organization to the provider) is likely to be less stable, leaf announcements result in BGP updates in the routing table. On the other hand, the links connecting providers to exchange points are likely to be more stable.

An important side effect of the provider-based assignments is that the address space assigned to an organization is not owned by that organization but by the provider. If the organization changes provider, the organization receives a different address space from the new provider and will not be able to use the previously assigned address space anymore, since it is owned by the previous provider. This results in a renumbering of its entire network. Since IPv4 was not designed with mechanisms to facilitate renumbering, this situation results in huge trouble and inconvenience for organizations. Since they are locked by the address space of the provider and the cost of renumbering is high, the organization is locked to the provider. To overcome this big issue, organizations started to limit the use of public addresses to enable smoother change of providers. The limitation is accomplished by using a NAT at the edge of the network and limiting the number of servers using global address space.

### *1.1.8 Constrained Allocation Policy of IPv4 Addresses*

To further conserve IPv4 addresss space, successive versions of IPv4 address allocation policies and guidelines [RFC1466, RFC2050] were put in place by the IANA and the registries. These policies had the following goals [RFC2050]:

- conservation of address space to maximize the lifetime of the IPv4 address space;
- hierarchical routing for routing scalability on the public Internet;
- public registry.

Policy RFC2050 allocates IPv4 addresses space in smaller chunks to providers in a slow-start procedure. ISPs are asked to document the address assignments to the end organizations. The result is a slower rate of consumption of the IPv4 address space.

### *1.1.9 Global Routing*

Figure 1.6 [Huston, 2005] shows the size of the IPv4 BGP global routing table.

Despite the use of NAT, HTTP virtual hosting, VLSM, CIDR, provider-based aggregation and constrained address allocations since the mid 1990s, the growth of the global routing table has been mostly linear, while the slope is increasing in recent years. The growth, due mainly to the increase of small prefixes (/24), comes from the growth of the Internet itself, the use of multihoming and traffic engineering techniques using routing [Huston, 2001].

### *1.1.10 Summary of Internet Growth*

Whatever metric one take and despite all the invented solutions mentioned above, Internet growth is heading towards the exhaustion of IPv4 addresses in a few years and to ever increasing large global routing tables. A major fix to these issues must be deployed soon and IPv6 is the only solution currently worked out.

Let’s now take another perspective by discussing other current issues in IPv4 networks.

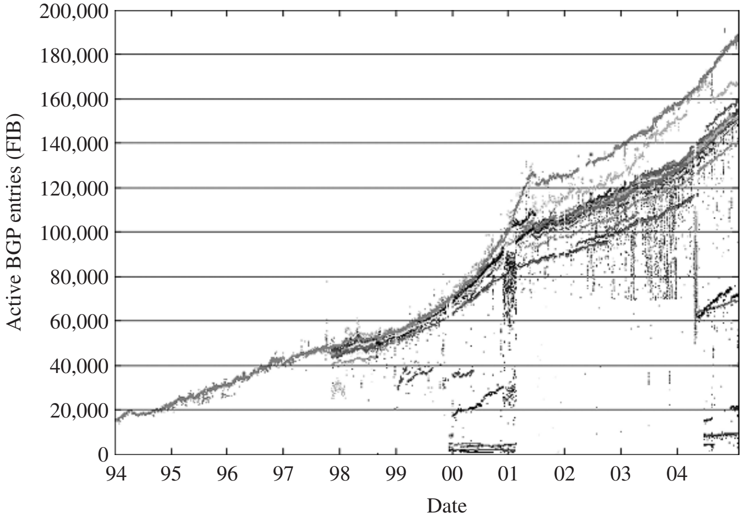


**Figure 1.6** IPv4 BGP Global Routing Table Size

## 1.2 Real Issues and Trouble with IPv4

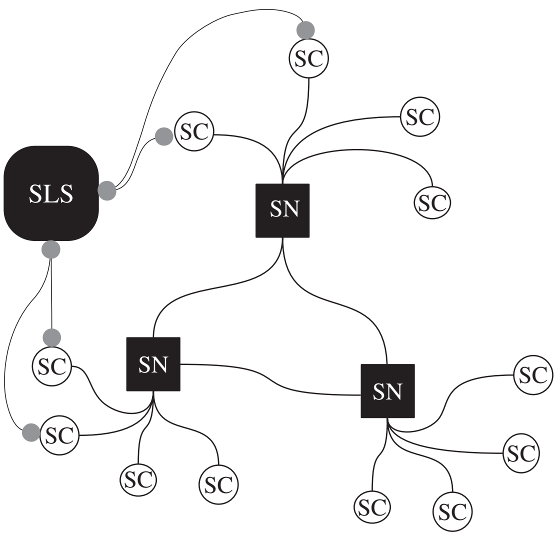
The shortage of IPv4 addresses is responsible for many issues and trouble in IP deployments today. Real world issues described in this section show the hidden costs of the lack of IPv4 addresses and the lack of functionalities in IPv4 for the current and future use of IP networking.

### *1.2.1 Deploying Voice over IP*

Skype [Skype] is a Peer-to-Peer (P2P) Voice over IP(VoIP) application and network. The Skype designers claim to traverse any NAT or firewalls to achieve P2P. Since the Skype protocol is not publicly disclosed, researchers have analyzed the protocol and described the process to traverse NAT and firewalls [Baset and Schulzrinne, 2004].

In a nutshell, a Skype client knows in advance some Skype gateways, named supernodes, and discovers others that help (the client) to find its external IPv4 address. The authors of the analysis think this technique is similar to STUN [RFC3489] and TURN [Rosenberg, 2004], discussed more in Section 1.3.1. The client tries to connect to the gateways using UDP; if unsuccessful, it tries TCP; if unsuccessful, it then tries TCP on the HTTP port (80); and if unsuccessful, it tries TCP on the HTTPS port (443). Since HTTP ports are usually opened for outgoing connections in most organization networks, Skype uses these ports as a last resort to traverse the firewall. When this last resort does not work, Skype loops again twice more and if still not successful, finally gives up.

In most cases, the voice traffic between the two VoIP peers goes through two other nodes, named supernodes, as shown in Figure 1.17. This figure shows the Skype network where small dots represents VoIP end-users. Supernodes are normal Skype nodes elected to be intermediary nodes, shown as bigger dots in Figure 1.7.



SN: Skype supernode
SC: Skype client
SLS: Skype login server

**Figure 1.7** Skype overlay network

The rationale behind electing supernodes in the Skype network is to enable fully automated NAT traversal without a specific network of servers. Any Skype node can be automatically elected as supernode if the node has a public address and some available CPU resources.

True peer-to-peer is almost impossible on IPv4 networks because of the presence of NAT. This Skype 4-nodes routing between the two peers through the supernodes obviously introduces delay, jitter and less performance. Also, since the election process to be a supernode is not managed by the user, a user bandwidth might be filled with the traffic of others, just because its computer was elected as supernode on the Skype overlay network. As a claimed peer-to-peer protocol, Skype is not a peer-to-peer protocol, not because of its authors failing to design a true peer-to-peer protocol, but because the deployed IPv4 networks disable peer-to-peer applications.

This demonstrates how unpredictable it is for an application to get the needed basic IP connections. It also demonstrates how the applications are involved in delivering basic connections, which should be handled by lower layers such as IP and transport.

Owing to the current issues with IPv4, applications have become much more complex just to reach other peers. The IP architecture was based on the end-to-end principle, where the network will be 'dumb' and end nodes can be reachable directly and easily. Current IPv4 networks have too much processing and break the end-to-end direct connectivity. Skype smarts or similar will need to be implemented in all applications requiring end-to-end connections, which makes applications complex, provokes latency in the application and network, makes application fragile to any network change and makes the overall connectivity unpredictable.

For the same reasons, an industry leader and well-known developer of open-source software stopped the development of a VoIP peer-to-peer application named SpeakFreely. Here is a excerpt from his announcement:

> The Internet of the near future will be something never contemplated when Speak Freely was designed, inherently hostile to such peer-to-peer applications. I am not using the phrase ‘peer to peer’ as a euphemism for ‘file sharing’ or other related activities, but in its original architectural sense, where all hosts on the Internet were fundamentally equal. Certainly, Internet connections differed in bandwidth, latency, and reliability, but apart from those physical properties any machine connected to the Internet could act as a client, server, or (in the case of datagram traffic such as Speak Freely audio) neither – simply a peer of those with which it communicated. Any Internet host could provide any service to any other and access services provided by them. New kinds of services could be invented as required, subject only to compatibility with the higher level transport protocols (such as TCP and UDP). Unfortunately, this era is coming to an end.
>
> [Walker, 2004]

It is terrible that the well designed IP protocol that offered so much innovation in its first 20 years is now stopping innovation, because of the introduction of NAT in the network.

SIP [RFC3261] is the IETF standard protocol for VoIP. SIP was designed for a pure IP network without NAT. It works fine only when no NAT is present between the peers. The pervasive presence of NAT means that SIP and its related protocols such as RTP are not deployable as in the current IPv4 networks with NAT dominance. These protocols have been augmented by various NAT traversal techniques. However, none of these techniques take care of all cases unless the audio path goes through a gateway, which disables the essence of VoIP performance which is to carry voice over IP on the direct path between the two peers.

I’ve been using a SIP softphone with a VoIP provider on my laptop. This SIP software implements most of the NAT traversal techniques. In many cases, VoIP calls just do not work. This software shows me the following error message: ‘Login timed out! Contact Network Admin.’ Very useful message! A few times, the error message was: ‘Cannot identify the Cone NAT correctly’. Another very useful message for an end-user.

As a user, I have no clue what kind of NAT, firewall or other network devices are in the path. So I called the VoIP provider technical support. As a technical person, I investigated and discovered the situation prior to the support call, but I took the ‘dumb’ user hat when calling technical support. After literally one and half hours of support over the phone (not the VoIP one obviously, but a plain old one), I had escalated two levels of technical support. It was suggested that I reinstall the SIP software and change the configuration of my operating system, which I did to the point where they asked me to reinstall the whole operating system, which I refused to do! 90 minutes of technical support, no good answer was given, the service was not restored and I was a frustrated customer. Why? There was a symmetric NAT in the path.[5] This kind of NAT is not supported by the NAT traversal techniques used by the SIP software. If most users of a VoIP deployment started calling the technical support and spent 90 minutes each, the VoIP company would go bankcrupt pretty fast!

These few examples show that VoIP is difficult to deploy over IPv4 networks because of NAT. True peer-to-peer is no longer possible. Innovation is hindered by current IP networks.

[5] Various kind of NATs are discussed in the book Web site: http://www.ipv6book.ca.

We need to restore this network to a good state in order to maintain innovation, user confidence and good experience. IPv6 provides the features needed to deploy applications seamlessly.

### *1.2.2 Deploying IP Security*

The IP protocol did not have any widespread security at the IP layer. Over time, security was added at the application layers, such as the secure socket layer (SSL) for the Web. Right now, we have similar and duplicated security functionality in several application protocols, creating a whole set of new problems, such as multiple, different and incompatible key management functions.

While discussing requirements of IPv6, the IETF decided to work on an IP security layer, named IPsec [RFC2401], to protect the whole IP packet for authentication, integrity protection and confidentiality. IPsec (see Chapter 13) is available for IPv4 as an option and mandatory for IPv6. By protecting the IP layer, the application layers over IP do not usually need additional protection.

However, the deployment of IPsec on current IPv4 networks have shown the difficulty of protecting IP packets when NATs are in the path. IPsec[6] protects the whole packet, so any modification of the packet between the source and the destination violates the security of the packet. NAT modifies addresses and port numbers of IP packets, therefore disabling the full protection of the IP packet, and disabling full security deployment.

Since IPv6 does not need NAT, full end-to-end IP security is deployable without those issues.

### *1.2.3 Deploying Application Security*

An enterprise has setup an e-commerce Web site with connection to its internal SQL database located in its private network. The Web site server is reachable from the internet. As shown in Figure 1.8, the connection from the Web server to the internal SQL database goes through the firewall which also implements NAT.

The SQL connection protocol between the Web server and the database backend negotiates IP addresses and port numbers within the protocol. So, by default, it does not traverse a NAT. The NAT-firewall product supports this protocol by inspecting the exchange and replacing the IP addresses and port numbers by the translated ones, within the application payload. This makes NAT and the SQL connection work. However, the organization then wants to encrypt

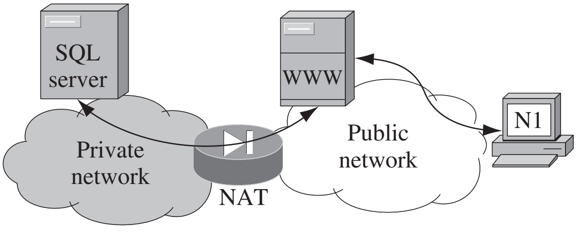


**Figure 1.8** Backend database connection to Web site

[6] More specifically the Authentication Header (AH) mode, see Chapter 13.

the data in the Web to SQL connection to prevent any snooping of the confidential data. By turning on encryption in the application protocol, the NAT-firewall is then unable to inspect and replace the IP addresses. The SQL encrypted connection does not work across NATs.

NAT disables the use of security in application protocols.

### *1.2.4 Videoconferencing*

A school board had to deliver a videoconferencing solution to help students in schools in remote communities to have access to professors in cities. Any professor from any school can give the course and any remote school class can attend the videocast. Remote students interact with the professor as if they were in the physical class.

Figure 1.9 shows the network where videoconferencing stations are located in remote networks. Multiple NATs are in the path between any combination of stations. The video feed station can be in any class and all sites are actively participating in the videoconference.

The videoconferencing software did not work by default in this configuration, since all the stations were hidden to the others by private address space. For any class, the teacher had to make a request to the IT department one week in advance, so that the IT department could configure a static mapping of addresses for all the NATs in the path and then configure the videoconferencing stations to use this mapping. This manual process of the IT department hindered the capability of the teachers to use the service as a commodity service. It is not the fault of the IT department, it is the trouble caused by the NATs: the inability to deploy and use applications.

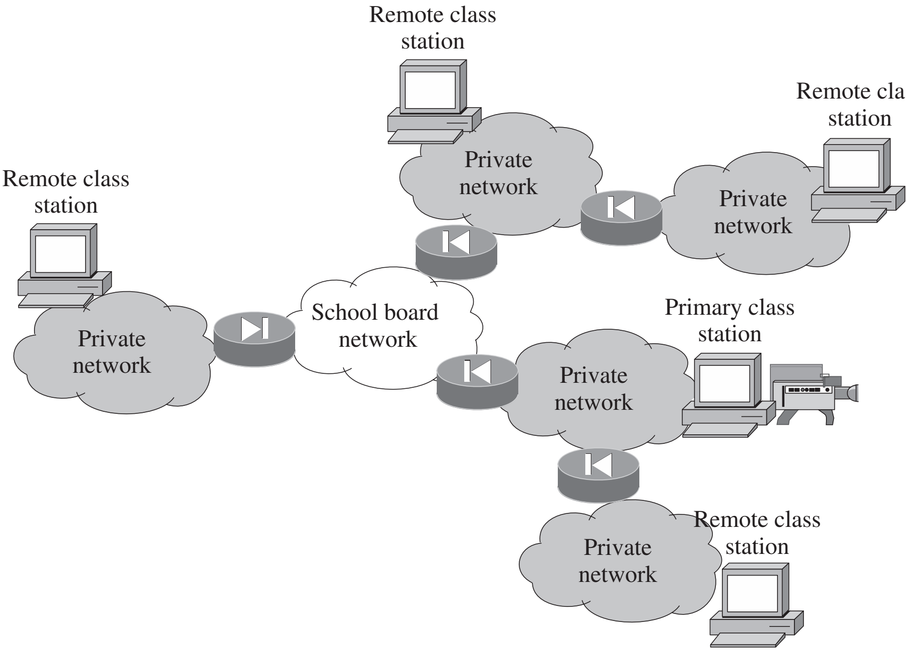


**Figure 1.9** Videoconferencing with multiple NAT in remote networks

Deploying IPv6 in this network solves the problem, since all nodes are reachable and videoconferencing is seamlessly working in all cases. A transition mechanism[7] can be used to enable IPv6 in the whole network before a full upgrade becomes possible.

### *1.2.5 A Simple Web Server at Home*

With digital cameras and powerful computers, people have a library of digital pictures on home computers connected to a home network with broadband connectivity. They would like to share these pictures with friends and family that could access them remotely from the Internet. Available broadband bandwidth makes this possible.

Figure 1.10 shows a home network with the computer that has the picture library and runs a Web server.

This Web server is not reachable from the Internet because it has a private address hidden by the NAT function implemented in the home gateway. Therefore, friends cannot access the pictures. To solve the issue, the home gateway is configured with a static mapping of the external address to the internal address of the Web server for the HTTP port. If the home network has multiple computers to be Web servers (such as parents and teenagers Web servers), this mapping works only for a single computer. Moreover, it requires some IP networking knowledge beyond most end-users. There is a good chance that you, the reader, who has good IP networking knowledge, is being requested by the non-techie friends and family to set up these devices! Right? Point taken?

As we can see, the bandwidth, the computer, the data and the software are all available to make this simple application possible. What disables the application from actually working is the NAT.

Home gateways have evolved into very complex devices these days. It is common to receive a 200 page user's manual, discussing a lot of complex IP configurations. A majority of these pages and the overall complexity come from the NAT presence and its related issues.

By restoring reachability, IPv6 in this home network setup will make the application work seamlessly.

### *1.2.6 Using Remote Procedure Calls*

Remote procedure calls (RPC) are used for distributed computing, where applications developers have access to an application programming interface (API) to access services on remote computers by a simple function call. RPC processes in the distributed network talk to each other by exchanging their IP addresses and by dynamically allocating ports.

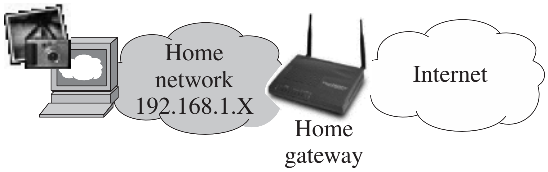


**Figure 1.10** Home network

[7] In this case the TSP tunnel broker, see Section 16.2.9.

When a NAT is in the path between the computers, RPC no longer works. Very complex configurations prone to any changes in the network can overcome some of the simple setups, but then do not scale well and maintain states in the network.

Distributed computing networks are difficult to deploy over IPv4: IPv6 deployment solves this RPC problem right away.

### *1.2.7 Remote Management of Applications and Servers*

Many organizations are outsourcing their IT services to a third party. This third party organization usually sets up a network operations center (NOC) to manage remotely the servers, networks and applications of its multiple customers. As shown in Figure 1.11, the NOC is connected to the customer's networks through private networks or the Internet.

However, many organizations have one or many NATs in their internal network. The remote management station in the NOC cannot reach the servers behind NAT. One has to define static translations on all the NATs to make this work, when possible. Even if they make it happen, a lot of static configuration is introduced in the NAT network, where any fault NAT will make the network unreachable by the NOC. The NOC is responsible for troubleshooting and keeping the network running, while it has no tools to manage the network! The support organization can then not deliver its service level agreement conditions such as 99.99% uptime, since the NOC cannot manage the network. This has nothing to do with security and firewalling, but rather lack of address space and the related presence of NAT.

Using IPv6 in this scenario solves the reachability issue. Moreover, it will enhance security since in all paths, end-to-end security and end-to-border security can be established.

### *1.2.8 VPN Between Same Address Space*

Many organizations, subsidiaries within an organization, divisions within an organization or recently merged or acquired organizations have separate IT departements. For that matter,

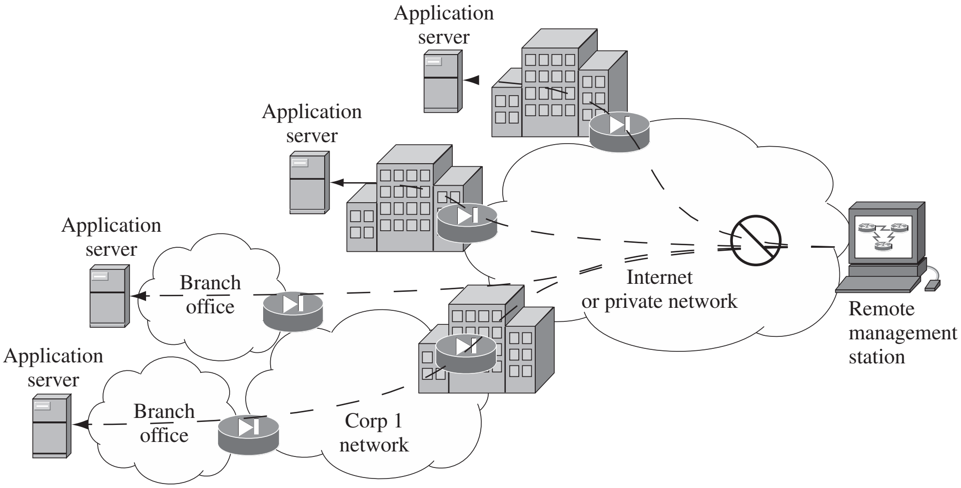


**Figure 1.11** Remote management of servers in private networks

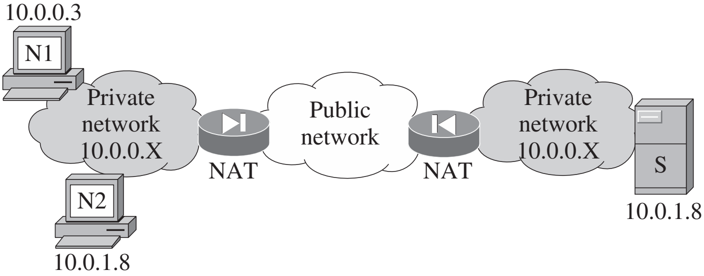


**Figure 1.12** VPN between same address spaces

they manage their own address space. In nearly all cases, each network uses the 10.X.X.X private address space. When two or more of these networks are connected together, an address collision happens.

Figure 1.12 shows a simple case of this situation. A VPN is created between the two NATs, at the borders of each network. When N1 needs to reach the server S in the remote network, it sends the packet to S address: 10.0.1.8. However, the packet never reaches S but instead reaches N2 in the same network as N1.

To overcome this situation, one defines address views for each host available to the other network, creating a large static map on both NATs (often called double NAT), not dynamically managed as DNS and routing are good for. When a user calls the IT tech support for a problem reaching the other side, it is very hard to troubleshoot because of this double NAT process. This situation creates network management and support costs and does not scale well. The alternative is to renumber one of the networks, entailing important work and causing downtime on the network.

With its huge address space, IPv6 does not have these address collision issues.

### *1.2.9 Deploying Services in the Home Network*

Figure 1.13 shows an example of a remote monitoring service in the home network, where network cameras are placed in the home. The owner, while out of his home, wants to see what is happening in the home using its PC or its graphical cell phone.

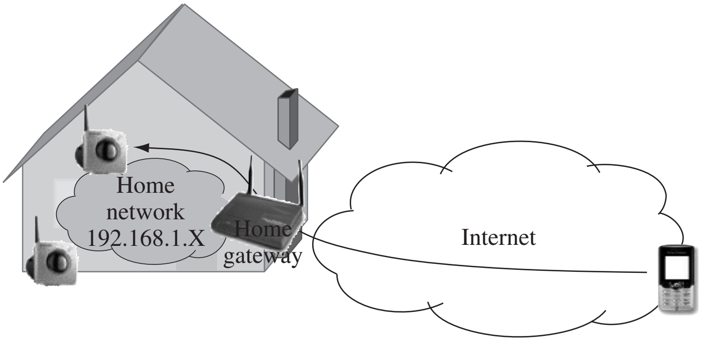


**Figure 1.13** Video monitoring in a home network

With multiple cameras in the home, there is no simple way to reach all of them from a device on the Internet. One would have to configure the home gateway with multiple static port translations, or to create a specific VPN to access the home network. The more services one has at home, the more these tricks become painful. Many applications are related to sensors and appliances in the home that are accessed or controlled from outside the home network.

With IPv6, these home services are straightforward to enable, manage and use, because reachability is restored by the addressing.

### *1.2.10 Merging or Connecting Two Networks Together*

When two organizations merge or connect their networks, their address spaces collide because both usually use the same address space 10.X.X.X. Figure 1.14 shows such a situation.

This creates similar problems to the one discussed in Section 1.2.8. With a large address space, IPv6 addressing in networks will not collide when networks connect or merge.

### *1.2.11 Large Networks*

For some large corporate networks, given the non-optimized allocation of address space with subnet masks, the private address space 10.X.X.X is just not sufficient for their numbering [Hain, 2004]. As the networks expand, they need to have more address space. Many organizations are using non-allocated address space such as 1.X.X.X for their additional address space. As we can see, large networks need more private address space than is available in IPv4.

On the other hand, IPv6 has sufficient public and private address space to support these scenarios.

### *1.2.12 Address Plans and Secondary Addresses*

An enterprise address plan identifies a subnet mask for each link, which establishes the maximum number of nodes on that link. When more hosts than the maximum are put on one link, a second prefix is used on that link. Figure 1.15 shows such a situation where the initial prefix is 192.168.1.X/24. Among the computers on that link, N1 has 192.168.1.2 and the router has 192.168.1.1.

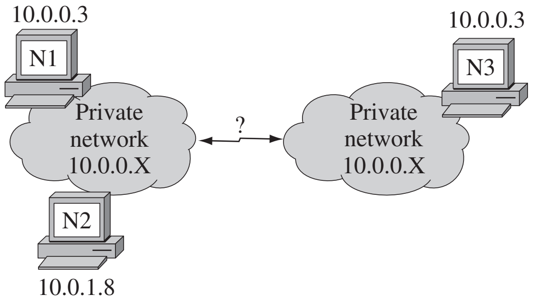


**Figure 1.14** Merging two networks with the same address space

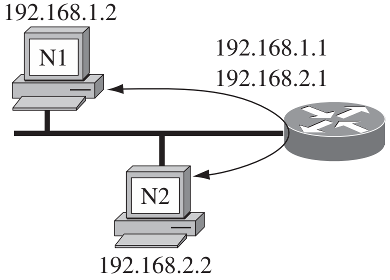


**Figure 1.15** Secondary address traffic

When the maximum number of nodes is reached, then another prefix (192.168.2.X/24) is added on the link. The router now has an additional address, 192.168.2.1, and N2 is part of the second prefix (192.168.2.2). For simpler network management, nodes do not participate in routing. To send a packet to N2, N1 finds that N2 is not on the same link, given that it does not have the same prefix. N1 sends the packet to its default router and the router resends the packet on the same link to N2. So all communications between the two nodes are duplicated on the same link, adding delay and decreasing the available bandwidth by half.

With virtually unlimited numbers of addresses for nodes on a link, IPv6 does not suffer from this behavior.

### *1.2.13 Provider VPN Address Collisions*

Nowadays, most organizations are using 10.0.0.0/8 address spaces inside their corporate network. When a provider offers the VPN service to its enterprise customers, the routing inside the VPN core carries many 10.0.0.0/8 routes originating from different networks, as shown in Figure 1.16.

This address collision in the routing table makes the routing incoherent and exposes the organization’s networks to others within the provider network. To overcome this problem, a route distinguisher is added to the routing protocols to identify uniquely each organization

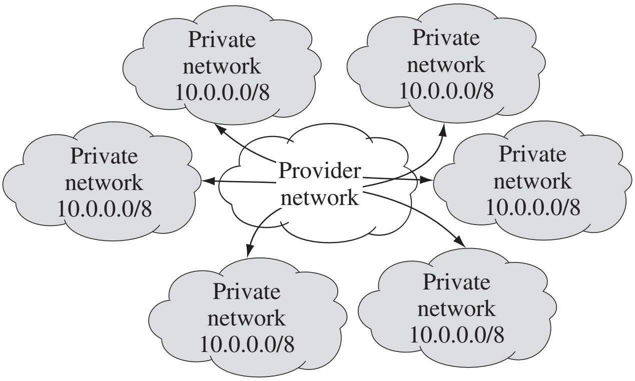

Private network 10.0.0.0/8
Private network 10.0.0.0/8
Private network 10.0.0.0/8
Provider network
Private network 10.0.0.0/8
Private network 10.0.0.0/8
Private network 10.0.0.0/8

**Figure 1.16** Provider VPN address collision

network inside the provider network. This is an example of extending the IPv4 address space for a very specific need, which only solves that need, and does not help with other issues. IPv6 does not have that problem because each network has a unique large address space.

### *1.2.14 Should IP Addresses be Free?*

In many markets nowadays, providers are billing IPv4 addresses. For example, some broadband providers are asking a premium for more than one public IP address to the home networks. This cost is a result of the lack of IPv4 address space and also the lack of functionality in the IP framework to deliver a range of addresses to a large number of networks, such as home networks. By restoring address space and providing prefix delegation methods, IPv6 solves these issues and IPv6 addresses should be free. [8]

### *1.2.15 Summary*

Each example above indicates the costs and issues, often combined together, related to IPv4 networks today. The mitigation techniques used to keep IPv4 up and running were also discussed. This set is just a sample and many other instances exist. For each example, a possible 'workaround' might exist, but these workarounds combined together create an important network management problem.

Moreover, the end-to-end reachability is now lost, disabling innovative applications and security to be deployed.

Many issues are related to the existence of NAT. Either we exacerbate the problem by continuing to procrastinate, incurring more and more costs, or we solve the problem by deploying IPv6. With the right transition tool, the deployment of IPv6 costs less than the current visible and hidden costs of NAT.

## 1.3 Architectural Considerations

IP architecture considerations are at the core of the issues facing IPv4 today.[9] In architecture terminology, IPv4 has lost transparency, defined as:

> the original Internet concept of a single universal logical addressing scheme, and the mechanisms by which packets may flow from source to destination essentially unaltered.
> [RFC2775]

Transparency is related to the existence of the end-to-end principle, at the core of the design of IPv4 and the Internet. This end-to-end principle may be summarized as [RFC2775]:

- Certain functions can only be accomplished by the end nodes. For example, failures in transmission and end-to-end security can only be managed by the end nodes. As such, state of the end-to-end communication must only be kept by end nodes and not by the

[8] Apart, that is, from some fees paid by the providers to the registries for the registry own operations. However, these fees are near to zero when shared over all the provider's customers.

[9] This section is based on RFC2775 and RFC2993, very good documents to read for a more exhaustive description.

network. The network is enabled to re-route packets transparently and efficiently, since no state is kept in the network.

- Transport protocols are designed to provide the required functions over a non-guaranteed IP network. Enhancements [RFC2581] were also integrated in end-nodes to better manage congestion.
- Packets can flow unaltered throughout the network and IP addresses are used as unique labels for end systems.

Implications of NAT in the network are illustrated by the following issues [RFC2993]:

- NAT is a single point of failure. Since a NAT keeps state, any failure of the NAT requires that all the current connections of all nodes behind the NAT be re-established.
- Application-level gateways(ALG) are complex. ALG are used in NAT devices to inspect application protocol packets to modify them on the fly. Any application requires a synchronization of all ALG in the field to support the deployment of the application.
- NAT violates TCP states. TCP states are defined for end nodes to manage the connections. A device in the network that is assigning transient addresses and ports without managing TCP states will collide with non terminated TCP connections.
- NAT requires symmetric state management. In the event of link flappings, multiple NATs must be fully synchronized in real time in order to keep the state of connections and address and port assignments.
- NAT disables the use of a global name for advertising services. NAT hides devices such that services behind cannot be advertised in the DNS to be accessed from anywhere.
- Private address space used for VPNs are colliding. L2TP tunnels and other VPN technologies enable networks to be connected together. However, the address spaces usually collide since private networks use the same 10.X.X.X address space.
- Correlation in network events is difficult. Since source addresses are changed on the fly, correlation of network events based on IP address becomes a huge problem since it requires the dynamic state of translation of all the NATs in the path being saved and then correlated by some qualitative heuristic.

During an IETF plenary session, Steve Deering, primary author of multicast [RFC1112] and IPv6 [RFC1883], described the initial IP architecture model as an hourglass. The following figures are from his presentation [Deering, 2001]. The initial and true model of the Internet Protocol is shown in Figure 1.17.

The hourglass architecture was based on the following design criteria:

- An internet layer to
  - make a bigger network (than a layer 2 layer such as ATM);
  - provide global addressing (instead of local addressing which makes connecting networks very difficult);
  - virtualize the network to isolate end-to-end protocols from network details/changes.
- A single internet protocol to
  - maximize interoperability;
  - minimize the number of service interfaces.
- A narrow internet protocol that
  - assumes least common network functionality to maximize the number of usable networks.

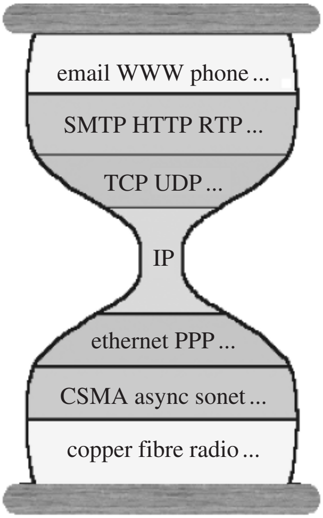


**Figure 1.17** IP hourglass model

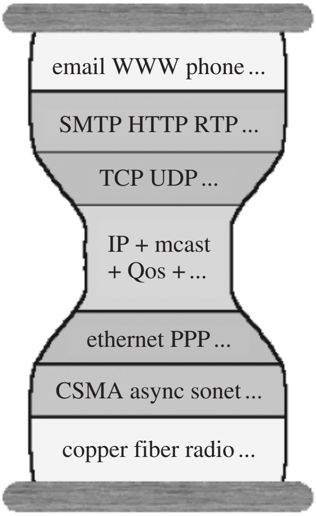


**Figure 1.18** IP hourglass architecture fattening

Over time, this model got 'fatter on the waist', as shown in Figure 1.18, by adding additional services to the IP protocol itself, such as multicast, QoS, security, MPLS, L2TP, and others.

This fattening requires additional functionality from the underlying layers, which makes these new functionalities more difficult to deploy. Moreover, the introduction of network address translation (NAT) and application level gateways (ALG) broke the IP model as shown in Figure 1.19. With these middle boxes, state management is introduced in the network and behavior is unpredictable.

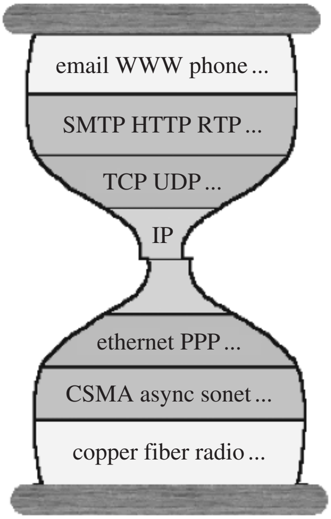


**Figure 1.19** NAT and ALG breaking the IP hourglass architecture

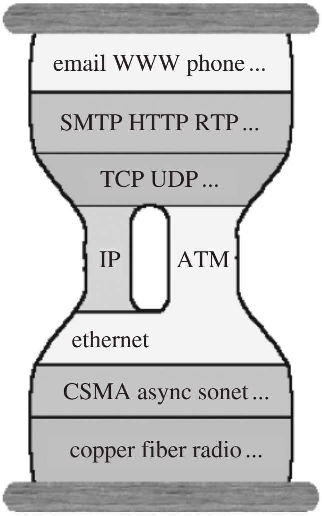


**Figure 1.20** ATM replacing IP in the hourglass architecture

Asynchroneous Transfer Mode (ATM) tried to become a layer 3 protocol, as shown in Figure 1.20, but was unsuccessful.

Eventually, the IP layer might become overloaded making the architecture too fat, as shown in Figure 1.21.

On the positive side, for the architecture, we have used IP tunneling to overlay networks, as shown in Figure 1.22.

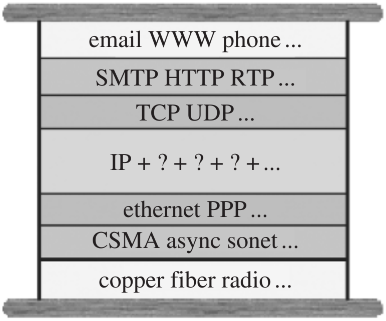


**Figure 1.21** IP overloaded and IP hourglass architecture too fat

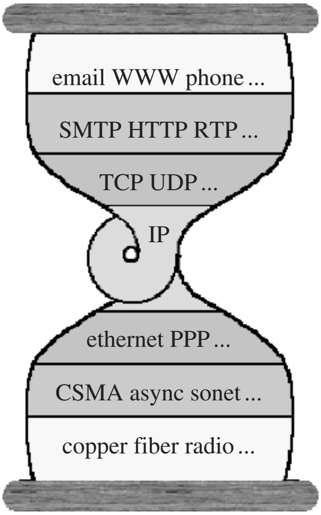


**Figure 1.22** IP tunneling still in the IP hourglass architecture

The IP tunneling techniques do not change the IP hourglass model, since the layering is preserved. Such a technique is used to encapsulate IPv6 in IPv4 packets, enabling fast IPv6 deployment over current IPv4 networks, as discussed in Chapter 16.

Figure 1.23 shows the dual-stack integration of IPv4 with IPv6 where the two are used together. This is the main path we are taking towards deploying IPv6, as discussed throughout this book. However, it introduces two service interfaces, requires changes on both upper and lower layers and is not interoperable.

Figure 1.24 shows the target architecture where IPv4 is replaced by IPv6 to restore a thin layer 3 architecture, leaving the lower layer to handle the wires and the upper layers to handle the application requirements.

This final target would fully restore the initial IP architecture, leaving the maximum of flexibility for transport and application layers. It is hoped that, at some point in time, we will drink the wine!

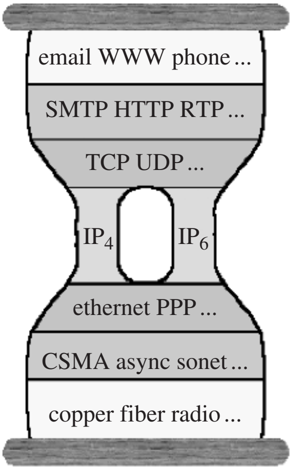


**Figure 1.23** Dual stack in the IP hourglass architecture

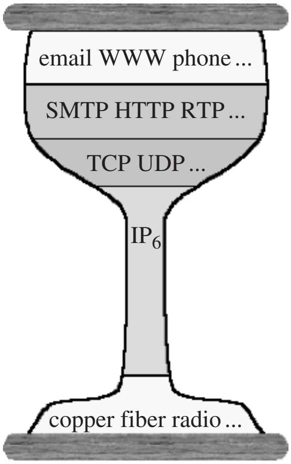


**Figure 1.24** End target with IPv6 for the IP hourglass architecture

### *1.3.1 Network Address Translator Variations*

Worst of all, the engineering community have found that NATs have a wide variety of behaviors[RFC3022, RFC3027, RFC3235]. This situation became apparent while application protocol designers were defining techniques to overcome NAT limitations in their respective application protocol. These techniques were based on the early identification of NAT laziness to make the translation of IP addresses and ports as precise as possible. Therefore, various techniques of NAT-traversal [RFC3489, Rosenberg, 2004] have been designed, none of which works in all cases and seamlessly enables the application. This single issue might by itself create sufficient network management costs to justify a full and rapid deployment of IPv6. The book Web site contains a section describing the variety of NATs.

## 1.4 Paradigm Shift

The networking world has changed since the 1970s. The concept of hosts and networks has changed. Table 1.2 lists the changes in networking that demonstrates the paradigm shift. It also lists the requirements for this new networking.

The listed requirements are all part of the design features of IPv6.

**Table 1.2** Changes in networking

| Past role | New Role | Description | Requirements |
|---|---|---|---|
| Host | Node or device | A host used to be a big fixed computer. Now, an IP host may be a tiny sensor, a control on an airplane, an identification tag on a cow or a videocamera. | IP efficiency.<br>Autoconfiguration. |
| Host | Server | With the advent of personal computers and client-server models, the hosts became clients. Nowadays, with VoIP, peer-to-peer and multimedia services, the host is offering services to the network, as an edge device. | IP reachability.<br>Large address space. |
| Host | Router | Where before a host was present, now it is a router. With personal area networks, one brings a PDA, a cell phone and a laptop: one of them is the router for the others. With broadband access to the home, where once a single host was attached, now it is a router with a network. | Automated routing and router configuration.<br>Network prefix delegation. |
| Static | Mobile | With Wifi, 3G and other wireless technologies, and the pretty small devices existing today, the devices are mobile and society becomes accustomed to using mobility services. | IP mobility. |
| Network | Unmanaged small network | Networks used to be large, managed by IP routing experts. Nowadays, networks in the home, personal area networks and sensor networks are all examples of small networks not managed by IP routing experts, but mostly unmanaged. | Automated network deployments. |
| Friendly | Not friendly | Back in the early Internet, there was a good level of trust between the organizations, mainly universities and research centers, connected to the Internet. The requirements for security were basic. Now, the trust that a user should have when connecting to the Internet is probably near zero. | Security |

*Note*: Well, we (the engineering community) try hard to make the home networks unmanaged, but when one looks at the configuration needed on a typical home gateway, we have still a long way to go to make it easy!

## 1.5 IETF Work Towards IPv6

As discussed in Section 1.1.1, the Solensky study [Solensky, 1990] that demonstrated the exhaustion of class B address space by March 1994 was the first wakeup call in the IETF. Table 1.3 shows the steps taken by the IETF [RFC1752] towards what was first named IPng and then renamed IPv6, when the new IP version number was assigned to 6.

From 1991 to 1995 and parallel with the IETF requirements work thread, many IPng protocol candidates were designed and discussed. Table 1.4 lists the candidates.

Figure 1.25 shows the generation tree of the proposals. Initially, there were six different protocols, and over time, some merged and some were not considered in the final evaluation step for IPng.

**Table 1.3** IETF major steps towards IPv6

| Date | Step | Description | Reference |
|---|---|---|---|
| August 1990 | Predicted exhaustion of class B addresses by 1994 | First wakeup call within the IETF. The class B address space could be exhausted in 4 years! | [Solensky, 1990] |
| November 1991–November 1992 | Routing and addressing(ROAD) working group formed | ROAD wg was formed to address the routing and addressing issues.<br>Recommendations were: CIDR and new protocol through a request for proposal process. | [RFC1380] |
| June 1992 | Internet Architecture Board (IAB) recommendation on IP version 7 | IAB recommended to use CLNP instead of a new IP protocol. The proposal was named IP version 7. | [IAB, 1992] |
| July 1992 | IETF rejects IAB proposal | During the IETF meeting, the IETF rejects the IAB proposal for IP version 7 based on CLNP. | [IABREJECT] |
| September 1993 | Recommendation to use Classless Inter-Domain Routing (CIDR) | CIDR removes the class structure, enabling more efficient address assignments and aggregation. | [RFC1519] |
| December 1993 | Solicitation for IPng requirements and selection criteria | The IPng area directors sollicit contributions on requirements and selection criteria for the new IPng protocol. | [RFC1550] |
| July 1994 | IPv4 address exhaustion estimated between 2005 and 2011 | The Address Lifetime Expectations (ALE) working group was chartered to estimate the remaining lifetime of IPv4 address space. They concluded that the IPv4 address space end of life is between 2005 and 2011. | [RFC1752] |

(*continued overleaf*)

**Table 1.3** (*continued*)

| Date | Step | Description | Reference |
|---|---|---|---|
| August 1994 | 20 White Papers on IPng requirements and selection criteria | 20 white papers were contributed responding to the solicitation [RFC1550], from the following subjects or industries: cable TV, cellular, electric power, military, ATM, mobility, accounting, routing, security, large corporate networking, transition, market acceptance, host implementations and others. | [RFC1667], [RFC1668], [RFC1669], [RFC1670], [RFC1671], [RFC1672], [RFC1673], [RFC1674], [RFC1675], [RFC1676], [RFC1677], [RFC1678], [RFC1679], [RFC1680], [RFC1681], [RFC1682], [RFC1683], [RFC1686], [RFC1687], [RFC1688], [RFC1753] |
| December 1994 | IPng Area Formed | A new area within IETF is formed to manage the IPng effort. The framework of efforts is also defined. | [RFC1719] |
| December 1994 | Technical criteria to choose IPng | A list of criteria is defined and going to be used against all the IPng protocol proposals. | [RFC1726] |
| January 1995 | Recommendation for IPng | The IPng area directors main recommendation is to use the SIPP 128bits version as the basis of the new IPng protocol. Many working groups are formed. | [RFC1752] |
| December 1995 | IPv6 specification | The first version of the IPv6 specification is published. | [RFC1883] |
| December 1996 | Ngtrans working group first meeting | The Next generation transition (ngtrans) working group is formed to handle the transition to IPv6. | [IETF37ngtrans] |
| December 1998 | New version of IPv6 specification | Based on implementations and additional work, a new version of the IPv6 specification is published. It slightly changes the header format, clarifies many items such as Path MTU, traffic class, flow label and jumbograms. | [RFC2460] |

*Note*: When an RFC document is the reference, the date is the publication date of the RFC. In most cases, the actual step happened many months before the publication date.

SIPP, TUBA and CATNIP were the protocol candidates reviewed more carefully by the IPng directorate [RFC1752]. SIPP was chosen with some additional modifications, such as increasing the address size from 64 bits to 128 bits, after a long debate on the appropriate size of the address space.

**Table 1.4** IPng protocol candidates

| Protocol Name | Full name | Reference |
|---|---|---|
| IP encaps | Internet Protocol Encapsulation | [RFC1955] |
| SIP | Simple Internet Protocol | |
| PIP | P Internet Protocol | [RFC1621], [RFC1622] |
| Simple CLNP | Simple Connectionless-mode Network Layer Protocol | |
| Nimrod | New IP Routing and Addressing Architecture | [RFC1753], [NIMROD], [RFC1992] |
| TP/IX | | [RFC1475] |
| IPAE | IP Address Encapsulation | |
| TUBA | TCP and UDP with Bigger Addresses,<br>TCP/UDP Over CLNP-Addressed Networks | [RFC1347], [RFC1561] |
| CATNIP | Common Architecture for Next-Generation IP | [RFC1707] |
| SIPP | Simple Internet Protocol Plus | [RFC1710] |
| SIPP 128bits ver | Simple Internet Protocol Plus, 128 bits address version | |
| IPng | Internet Protocol Next Generation | [RFC1883] |
| IPv6 | Internet Protocol version 6 | [RFC1883] |

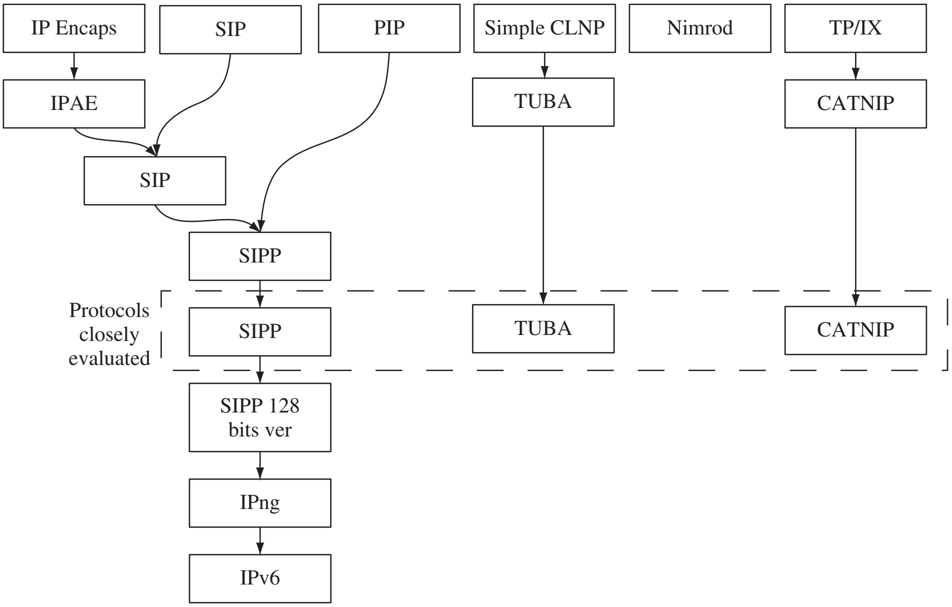


**Figure 1.25** IPng protocol proposals generation tree

## 1.6 IPv6 Main Features

> "IPv6 is a new version of the Internet Protocol. It has been designed as an evolutionary, rather than revolutionary, step from IPv4. Functions which are generally seen as working in IPv4 were kept in IPv6. Functions which don't work or are infrequently used were removed or made optional. A few new features were added where the functionality was felt to be necessary." [RFC1752]

Table 1.5 lists the main features of IPv6, which introduces many concepts, discussed in the following chapters.

**Table 1.5** IPv6 features

| Feature | Implementation | Benefit | Book chapter or section |
|---|---|---|---|
| Larger addresses | 128 bit addresses | From 32 bit address space in IPv4 to 128 bit address space. It enables all nodes to be addressable and reachable, removing the need for network address translation and restoring the end-to-end model for end-to-end capabilities such as security. | 4 |
| More levels of addressing hierarchy | Address architecture | Multiple levels in the addressing hierarchy provide better aggregation of routes, easier allocation of addresses to downstreams and scalability of the global routing table. | 4 |
| Scoping in the address | Specific bits in the address | Address scoping enables easy filtering at boundaries, such as link or site and better security against remote attack on link layer protocols. | 4 |
| Simple and fixed address architecture | /48 for sites, /64 for a link | Simplified address architecture enables easier addressing plans, which decreases the network management costs. Now, subnet masks are fixed and provide virtually unlimited numbers of nodes on a link. | 4 |
| Privacy addresses | Specific bits in the address | Provides privacy for the end-user where the IP address cannot be used for tracking traffic usage. | 13.4 |
| Multiple addresses on an interface | IPv6 stack | Multiple addresses on interfaces enables multiple use, virtual hosting, easier renumbering and a method for multihoming. | 5 |
| Autoconfiguration of nodes | IPv6 stack, router advertisements | Auto-configuration is based on advertisements about the link addressing sent by the routers. Nodes insert their MAC address into the host part of the IPv6 address. It enables fast and reliable configuration of nodes, as well as easy renumbering. | 5 |

| | | | |
|---|---|---|---|
| No address conflicts on links | IPv6 stack | Embedding the unique link address (MAC) into the host part of the IPv6 address and a duplicate address detection method guarantee uniqueness of the address on the link. | 5 |
| Better reliability in auto configuration | Router advertisements | Each router on a link sends auto-configuration information to nodes, so if one router is dead, others are still sending. The router infrastructure is always nearer to the host and more fault tolerant than DHCP servers. | 5 |
| Multicast address scoping | Specific bits in the address | A multicast address now contains a scope. IPv4 multicast had to rely on TTL to manage the reachability of a multicast channel, which makes multicast management complex. IPv6 multicast is easy to manage since the scope of the channel is within the IPv6 multicast address. | 4 |
| Simpler and more efficient IP header | Less number of fields, no checksum, 64 bit aligned fields | Routers process the packets faster and more efficiently, which improves the forwarding performance. | 3 |
| Extension headers | Options are placed after the base IPv6 header | Options for IPv6 packets are implemented as extension headers and are tagged with processing options. Routers do not have to look at most extension headers which increases their forwarding performance. New headers can be added incrementally without any impact on implementations. | 3 |
| Mandatory IP security | IPsec | IPsec is mandatory in IPv6, which makes all nodes in a position to secure their traffic, if they have the necessary underlying key infrastructure. | 13 |
| Source routing | Extension header | Source routing is implemented in a way so that routers not directly involved in the source routing can still make policy decisions based on the destination address. This feature makes source routing more deployable. | 9.2 |
| Simple and flexible transition | Transition protocols | In the foundation and requirements of IPv6, there was a clear need to make a smooth transition. The requirements were: incremental upgrade, incremental deployment, easy addressing and low start-up costs. | 16, 17, 18 |
| Labeling flows for QoS | Flow label header field | A flow label is defined in a specific field in the basic header, enabling the labeling and policing of traffic by the routers, without the need to inspect the application payload by the routers, resulting in more efficient QoS processing. | 14 |

(*continued overleaf*)

**Table 1.5** (*continued*)

| Feature | Implementation | Benefit | Book chapter or section |
|---|---|---|---|
| Multihoming capabilities | Multiple prefix on the same link and on interfaces | Multiple prefixes can be announced in router advertisements, which creates multiple addresses on interfaces. Lifetimes of prefixes are managed by the nodes which provides an easy way to multihome nodes. | 9.12 |
| More efficient use of links | Neighbor discovery | Link scope interactions between nodes and between nodes and routers are optimized. | 5.2.2, 6.1, 6.2 |
| Use of Multicast for discovery and link-local interaction | Neighbor discovery | No broadcasts are used in IPv6. In most cases, only relevant nodes receive the requests. | 5.2.2, 6.1, 6.2, 6.3 |
| Mobility | MobileIPv6 | Mobility is integrated in IPv6 headers, stacks and implementations, making mobility a seamless and deployable feature. | 11 |
| Private but unique address space | Unique local address space | Private addresses are used for unconnected networks to the Internet. Different than RFC1918 private IPv4 address space, private IPv6 address space remains unique to the site, which makes it easy to connect private networks together. | 4.3.2.3 |

## 1.7 IPv6 Milestones

Table 1.6 lists some major milestones of IPv6.

## 1.8 IPv6 Return on Investment

A study [Pau, 2002a] has established a return on investment framework for IPv6:

- adopters of IPv6 run smaller risks than waiting;
- targeted ratio of approx. 16% of IPv6 creates positive ROI on incremental deployment;
- migration costs can hardly be a deciding factor in deploying IPv6.

Another study [Pau, 2002b] from the same author uses an analytical model to reveal that for the ISP operator, net revenue with IPv6 is intrinsically and systematically higher than for IPv4.

**Table 1.6** Some IPv6 milestones

| Date | Step | Description | Reference |
|---|---|---|---|
| August 1990 | Predicted exhaustion of class B addresses by 1994 | First wakeup call within the IETF. The class B address space could be exhausted in 4 years! | [Solensky, 1990] |
| January 1995 | Recommendation for IPng | The IPng area directors main recommendation is to use the SIPP 128 bits version as the basis of the new IPng protocol. Many working groups are formed. | [RFC1752] |
| December 1995 | IPv6 specification | The first version of the IPv6 specification is published. | [RFC1883] |
| July 1996 | First IPv6 test network over Internet (6bone) | The 6bone IPv6 test backbone is started. | [6bonehistory] |
| February 1999 | Freenet6 service started | During an IPng working group interim meeting, the Freenet6 tunnel broker service is announced, providing the world community with easy access to the IPv6 Internet using automated tunnels. | http://www.freenet6.net |
| July 1999 | Registry-based IPv6 address space allocation is started | The regional registries, RIPE, ARIN and APNIC, start allocating IPv6 address space to providers. | [RIPE-196] |
| July 1999 | IPv6Forum | The IPv6Forum body is formed. | http://www.ipv6forum.com |
| February 2000 | Solaris8 | The first commercial OS to include IPv6 in the product as standard feature is Sun Solaris 8. | http://www.sun.com/ipv6 |
| March 2000 | FreeBSD 4.0 | FreeBSD open source operating system now includes IPv6 in its standard distribution. | http://www.freebsd.org |
| May 2001 | Freenet6 second generation | The freenet6 service second generation uses the TSP tunnel broker protocol. | http://www.freenet6.net |

## 1.9 What Happened to IPv5?

IPv5 is the IP protocol number of the Stream Protocol (ST) [RFC1190], an experimental protocol for streaming traffic. Figure 1.26 shows where ST fits in the IP architecture, including some specific streaming transport protocols named PVP and NVP.

To differentiate IPv4 packets from ST IP packets at the link layer, ST requires a specific IP version number. At the time of ST, the next version number available for IP was ‘5’. So IANA [IANA, 2001] allocated 5 to ST, so ST is also known as IPv5. When IPng was designed, the next version number available for IP was ‘6’, so IPng is IPv6.

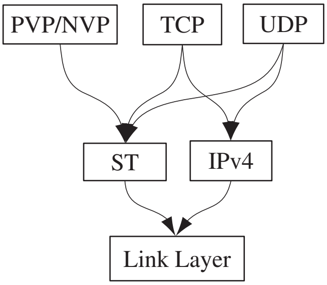


**Figure 1.26** Streaming Protocol and IPv4 Architecture

ST is an experimental protocol and is not deployed. Streaming is handled with multicast, RTP and other protocols.

## 1.10 Summary

IPv4 was deployed mainly in the university networks before the Internet became so pervasive. At that time, the Internet was for information sharing and electronic communications. For information sharing, ftp sites were used, then came Archie to index ftp sites, then Gopher which structures the information, then Veronica to index the gopher sites and then came the Web and the first client Mosaic. Before the Web and Mosaic, there was ‘no killer application’. However, there was a playground fertile for future innovations, and now it is part of our daily life.

As we have seen in the first sections of this chapter, the current IPv4 protocol is no longer fertile for innovations and is now a pretty constrained network.

IPv6 restores the fully-fledged network needed to deploy new applications, most of them probably still unknown. In the near future, there will be very little or zero IPv4 address space remaining.

The drivers for IPv6 are multiple, such as mobility, reachability, network management, multimedia, and others described in this chapter. More trouble and higher costs with IPv4, combined with new applications requiring the new IPv6 functionalities, is driving IPv6.

## 1.11 References

[6bonehistory] Ngtrans historic milestones, http://www.6bone.net/ngtrans/ngtrans_charter.html

Baset S. and Schulzrinne H., ‘An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol’, Columbia University, September 15, 2004, http://www.cs.columbia.edu/~library/TR-repository/reports/reports-2004/cucs-039-04.pdf

Charnie, J., ‘Statement to the commission on population and development’, United Nations, March 2004, http://www.un.org/esa/population/cpd/37OPEN.pdf

Deering, S., ‘Watching the Waist of the Protocol Hourglass’, Proceedings of the 51st Internet Engineering Task Force, August 2001, http://www.ietf.org/proceedings/01aug/slides/plenary-1/index.html

Hain, T., ‘Expanded Address Allocation for Private Internets’, Internet-Draft draft-hain-1918bis-00, April 2004.

Huston, G., ‘Analyzing the Internet’s BGP Routing Table’, January 2001, http://www.potaroo.net/papers/ipj/4-1-bgp.pdf

Huston, G., ‘Growth of the BGP Table’, January 2005, http://bgp.potaroo.net/

IAB, Internet Activities Board Meeting Minutes, June 1992, http://www.iab.org/documents/IABmins/IABmins.1992-06-18.html

[IABREJECT] Proceeding of the twenty-fourth Internet Engineering Task Force, http://www.ietf.org/proceedings/prior29/IETF24.pdf

IANA, 'IP protocol numbers', ftp://ftp.iana.org/assignments/version-numbers, November 2001.

[IETF37ngtrans] Proceedings of the 37th Internet Engineering Task Force, CNRI, December 1996, http://www.ietf.org/proceedings/96dec/toc.html

Nimrod: 'A Scalable Routing Architecture for Large, Heterogeneous, and Dynamic Internetworks', http://www.ir.bbn.com/projects/nimrod/nimrod-index.html

Pau, L.-F., (2002a) 'IPv6 Return on Investment (R.O.I) Analysis Framework at a Generic Level, and First Conclusions', Erasmus Research Institute of Management (ERIM), ERIM Report Series Research in Management, Rotterdam, The Netherlands, ERS-2002-78-LIS, September 2002.

Pau, L.-F., (2002b) 'A Business Evaluation of The Next Generation IPv6 Protocol in Fixed And Mobile Communication Services: An Analytical Study and Calculation', Erasmus Research Institute of Management (ERIM), ERIM Report Series Research in Management, Rotterdam, The Netherlands, ERS-2002-78-LIS, September 2002.

[RFC959] Postel, J. and Reynolds, J., 'File Transfer Protocol', STD 9, RFC 959, October 1985.

[RFC1058] Hedrick, C., 'Routing Information Protocol', RFC 1058, June 1988.

[RFC1112] Deering, S., 'Host Extensions for IP multicasting', RFC 1112, August 1989.

[RFC1190] Casner, S., Lynn, C., Park, P., Schroder, K. and Topolcic, C., 'Experimental Internet Stream Protocol: Version 2 (ST-II)', RFC 1190, October 1990.

[RFC1347] Callon, R., 'TCP and UDP with Bigger Addresses (TUBA), A Simple Proposal for Internet Addressing and Routing', RFC 1347, June 1992.

[RFC1380] Gross, P. and Almquist, P., 'IESG Deliberations on Routing and Addressing', RFC 1380, November 1992.

[RFC1466] Gerich, E., 'Guidelines for Management of IP Address Space', RFC 1466, May 1993.

[RFC1475] Ullmann, R. 'TP/IX: The Next Internet', RFC 1475, June 1993.

[RFC1519] Fuller, V., Li, T., Yu, J. and Varadhan K., 'Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy', RFC 1519, September 1993.

[RFC1550] Bradner, S. and Mankin, A., 'IP: Next Generation (IPng) White Paper Solicitation', RFC 1550, December 1993.

[RFC1561] Piscitello, D., 'Use of ISO CLNP in TUBA Environments', RFC 1561, December 1993.

[RFC1621] Francis, P., 'Pip Near-term Architecture', RFC 1621, May 1994.

[RFC1622] Francis, P., 'Pip Header Processing', RFC 1622, May 1994.

[RFC1667] Symington, S., Wood, D. and Pullen, J., 'Modeling and Simulation Requirements for IPng', RFC 1667, August 1994.

[RFC1668] Estrin, D., Li, T. and Rekhter Y., 'Unified Routing Requirements for IPng', RFC 1668, August 1994.

[RFC1669] Curran, J., 'Market Viability as a IPng Criteria', RFC 1669, August 1994.

[RFC1670] Heagerty, D., 'Input to IPng Engineering Considerations', RFC 1670, August 1994.

[RFC1671] Carpenter, B., 'IPng White Paper on Transition and Other Considerations', RFC 1671, August 1994.

[RFC1672] Brownlee, N., 'Accounting Requirements for IPng', RFC 1672, August 1994.

[RFC1673] Skelton, R., 'Electric Power Research Institute Comments on IPng', RFC 1673, August 1994.

[RFC1674] Taylor, M., 'A Cellular Industry View of IPng', RFC 1674, August 1994.

[RFC1675] Bellovin, S., 'Security Concerns for IPng', RFC 1675, August 1994.

[RFC1676] Ghiselli, A., Salomoni, D. and Vistoli, C., 'INFN Requirements for an IPng', RFC 1676, August 1994.

[RFC1677] Adamson, R., 'Tactical Radio Frequency Communication Requirements for IPng', RFC 1677, August 1994.

[RFC1678] Britton, E. and Tavs, J., 'IPng Requirements of Large Corporate Networks', RFC 1678, August 1994.

[RFC1679] Green, D., Irey, P., Marlow, D. and O'Donoghue, K., 'HPN Working Group Input to the IPng Requirements Solicitation', RFC 1679, August 1994.

[RFC1680] Brazdziunas, C., 'IPng Support for ATM Services', RFC 1680, August 1994.

[RFC1681] Bellovin, S., 'On Many Addresses per Host', RFC 1681, August 1994.

[RFC1682] Bound, J., 'IPng BSD Host Implementation Analysis', RFC 1682, August 1994.

[RFC1683] Clark, R., Ammar, M. and Calvert, K., 'Multiprotocol Interoperability In IPng', RFC 1683, August 1994.

[RFC1686] Vecchi, M., 'IPng Requirements: A Cable Television Industry Viewpoint', RFC 1686, August 1994.

[RFC1687] Fleischman, E., 'A Large Corporate User's View of IPng', RFC 1687, August 1994.

[RFC1688] Simpson, W., 'IPng Mobility Considerations', RFC 1688, August 1994.

[RFC1707] McGovern, M. and Ullmann, R., ‘CATNIP: Common Architecture for the Internet’, RFC 1707, October 1994.

[RFC1710] Hinden, R., ‘Simple Internet Protocol Plus White Paper’, RFC 1710, October 1994.

[RFC1715] Huitema, C., ‘The H Ratio for Address Assignment Efficiency’, RFC 1715, November 1994.

[RFC1719] Gross, P., ‘A Direction for IPng’, RFC 1719, December 1994.

[RFC1726] Partridge, C. and Kastenholz, F., ‘Technical Criteria for Choosing IP The Next Generation (IPng)’, RFC 1726, December 1994.

[RFC1752] Bradner, S. and Mankin, A., ‘The Recommendation for the IP Next Generation Protocol’, RFC 1752, January 1995.

[RFC1753] Chiappa, J., ‘IPng Technical Requirements of the Nimrod Routing and Addressing Architecture’, RFC 1753, December 1994.

[RFC1812] Baker, F., ‘Requirements for IP Version 4 Routers’, RFC 1812, June 1995.

[RFC1883] Deering, S. and Hinden, R., ‘Internet Protocol, Version 6 (IPv6) Specification’, RFC 1883, December 1995.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G. and Lear, E., ‘Address Allocation for Private Internets’, BCP 5, RFC 1918, February 1996.

[RFC1945] Berners-Lee, T., Fielding, R. and Nielsen, H., ‘Hypertext Transfer Protocol – HTTP/1.0’, RFC 1945, May 1996.

[RFC1955] Hinden, R., ‘New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG’, RFC 1955, June 1996.

[RFC1992] Castineyra, I., Chiappa, N. and Steenstrup, M., ‘The Nimrod Routing Architecture’, RFC 1992, August 1996.

[RFC2050] Hubbard, K., Kosters, M., Conrad, D., Karrenberg, D. and Postel, J., ‘Internet Registry IP Allocation Guidelines’, BCP 12, RFC 2050, November 1996.

[RFC2068] Fielding, R., Gettys, J., Mogul, J., Nielsen, H. and Berners-Lee T., ‘Hypertext Transfer Protocol – HTTP/1.1’, RFC 2068, January 1997.

[RFC2401] Kent, S. and Atkinson, R., ‘Security Architecture for the Internet Protocol’, IETF RFC 2401, November 1998.

[RFC2460] Deering, S. and Hinden, R., ‘Internet Protocol, Version 6 (IPv6) Specification’, RFC 2460, December 1998.

[RFC2581] Allman, M., Paxson, V. and Storens, W., ‘TCP Congestion Control’ RFC 2581, April 1999.

[RFC2775] Carpenter, B., ‘Internet Transparency’, RFC 2775, February 2000.

[RFC2993] Hain, T., ‘Architectural Implications of NAT’, RFC 2993, November 2000.

[RFC3022] Srisuresh, P. and Egevang, K., ‘Traditional IP Network Address Translator (Traditional NAT)’, RFC 3022, January 2001.

[RFC3027] Holdrege, M. and Srisuresh, P., ‘Protocol Complications with the IP Network Address Translator’, RFC 3027, January 2001.

[RFC3235] Senie, D., ‘Network Address Translator (NAT)-Friendly Application Design Guidelines’, RFC 3235, January 2002.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler E., ‘SIP: Session Initiation Protocol’, RFC 3261, June 2002.

[RFC3489] Rosenberg, J., Weinberger, J., Huitema, C. and Mahy, R., ‘STUN – Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)’, RFC 3489, March 2003.

[RIPE-196] RIPE NCC, ‘Provisional IPv6 Assignment and Allocation Policy Document’, RIPE-196, July 1999.

Rosenberg, J., ‘Traversal Using Relay NAT (TURN)’, Internet-Draft draft-rosenberg-midcom-turn-04, February 2004.

Solensky F., ‘Continued Internet Growth’, Proceedings of the 18th Internet Engineering Task Force, August 1990, http://www.ietf.org/proceedings/prior29/IETF18.pdf

[Skype] http://www.skype.com

Walker, J., ‘Speak Freely: End of Life Announcement’, January 2004, http://www.fourmilab.ch/speakfree/unix

## 1.12 Further Reading

Walker, J., ‘The Digital Imprimatur’, November 2003, http://www.fourmilab.ch/documents/digital-imprimatur/