Understanding IPv6

The upsurge in use of the Internet has lead to an increased requirement for IP numbers, which are rapidly running out. A new standard for IP numbering is about to be introduced to help overcome some of the limitations of the old system and to provide enough addresses to see us all well into the next century.

By David Morton

Many of us have heard of the proposals to convert the Internet Protocol standard from the current version IPv4 to a new standard. Few of us are really aware of all the implications. The driving force of the new standard is the rapid growth of the Internet and IPv6 is being introduced to overcome the address space restrictions of the old one. At first glance the implications for the network administrator seem negligible, because many of us already have our allocated addresses. In some cases we've enough to last us for a considerable period of planned growth, so why should we be interested in IPv6?

IPv6 is a lot more than IPv4 with a couple of numbers bolted on the end to make the address space bigger. It's a ground-up re-think of what will be required from IP as a protocol in the future. IPv6 starts to take on board the modifications that will be needed to cope with the changing traffic that will start to appear on IP networks globally. It is anticipated that there will be much more emphasis on real-time transactions as the Internet and intranets metamorphose from old style data networks, into complex transmission systems carrying a vast wealth of data, entertainment and other services, some not yet even a twinkle in their innovator's eye.

At the same time IPv6 attempts to address one of the largest headaches of an IP network from the administrator's point of view; configuring the network in the first place. Despite time saving systems like **BOOTP** and **DHCP**, a huge number of networks are hand crafted, with IP numbers typed in at individual workstations by unfortunate support staff whose job is to do little else. Anything that automates this process will help, and IPv6 goes a long way in this respect.

Many network administrators dislike changes like IPv6 because they fear some hideous one-day change over that leaves life chaotic for days or even weeks afterwards. Thankfully this shouldn't be the case, since an IPv6 network can talk to an IPv4 network and vice-versa. If you have two sites with an IPv6 system, and your Internet connection between the two is only IPv4, then the two networks will still be able to talk – the IPv6 packets tunnelling through the IPv4 connection. So there are few reasons to fear IPv6, and every reason to start planning the changes now.

History

The Internet Architecture Board started studying the problem of the growth of the Internet, and the number of addresses that would be required, back in 1991, when it was still known as the Internet Activities Board. To some extent the growth had been anticipated – everyone knew there would be ever more computers connected to the Net. What was less obvious then was that the Net would extend beyond computers in the conventional sense, and that there would soon be a requirement to assign IP addresses to devices as diverse as mobile telephones, other communication devices and even motor cars. There is already a car on the market with a built-in modem for diagnostics, and no doubt more will follow.

These early studies lead to the appointment of a team of engineers and scientists, working under the Internet Engineering Steering Group, charged with defining the next generation Internet protocol. This group co-ordinated the efforts of a number of other teams studying the problem of address space size, enhancements to the Transport Control Protocol (TCP) and the problems of compatibility with other protocols – notably IPX.

After due deliberation, the first proposals were documented in the usual manner for Internet enhancements – a Request For Comments (RFC number 1752) entitled "The Recommendation for the IP Next Generation Protocol", issued in 1994. It took a year for these proposals to be finalised, which finally happened in July 1995.

January 1996 saw the publication of the detailed proposals, in the five further RFCs detailed in Figure 1. After a brief pause, perhaps the most vital RFC was issued in April 1996. RFC1933 covered the transition mechanism – how to switch over from one system to the other without the

RFC1883 – The IPv6 base protocol.
RFC1884 – The address specification.
RFC1885 – Description of the control protocol, known as ICMP.
RFC1886 – Addressing the problems of an enhanced Domain Name Service (DNS).
RFC1933 – The transition mechanism.

Figure 1 - RFCs covering the detailed proposals for TCP protocols.

whole Internet collapsing in a steaming heap.

Why Change?

The principle reason behind the requirement for a new Internet protocol is the sheer rate of growth. IPv4's 32-bit address space was generous when it was first introduced, but the addresses are running out fast. Every host on the Internet needs a unique address, of course, and while some techniques allow hosts to share addresses – some dial-up users have an address assigned to them temporarily for the period of the connection – these are only mild palliatives for a system that is rapidly running out of elbow room. 2^{32} ought to be able to support around 4.2 billion hosts. However, the need to assign addresses in strict hierarchies reduces the availability of addresses, and it's already becoming difficult for companies to get address allocations from InterNIC in the US or their equivalents in other jurisdictions.

This shortage has raised serious difficulties, particularly in those companies – frustrated at the lack of availability of legal IP addresses – that have gone ahead and allocated made-up addresses for their internal networks. This works, of course, until the company decides that it needs to be connected to the global Internet, at which point some unpleasant things can start happening. If those made-up addresses are unallocated, then the chances are that the Internet connection won't be problematic, until of course they are allocated.

There is at least one documented tale of small company with such random IP addresses causing serious problems for a large multi-national, merely by connecting to the Internet. The result was the exchange of strong words, along with the mention of substantial damages, when they were tracked down.

"The automatic configuration facilities, both for hosts and for routers, have been described by some as worth the cost of switching to IPv6 all on their own."

Perhaps this sounds unlikely, but at least one survey suggests that up to 60% of companies have non-conforming or illegal addresses somewhere on their networks. Clearly this situation can only get worse as addresses become scarcer, and there's more pressure on network administrators to add new machines to internal networks somehow, regardless of the niceties of InterNIC and the rules. Surprisingly, only a very small percentage of companies use any form of automation to allocate IP addresses, and so the change from an illegal address regime to a legal one is not necessarily a trivial exercise. By expanding the address space we make the allocation of legal addresses easier, and remove the need for network managers to fly by the seat of their pants to assign addresses to new hosts and workstations.

More Elbow Room

At first glance, the expansion of IP address space offered by IPv6 looks a little excessive. We can see that 32 bits was becoming restrictive, but the expansion to 128 bits means that there are a total of more than 3×10^{38} addresses – or to put it in terms that are just slightly easier to grasp, over 6×10^{23} addresses for every square metre on the Earth's surface. Even taking the most conservative analysis of how hierarchical allocation would reduce this, it still leaves us with many thousands of addresses per square metre. It's plausible that just about every consumer item with more than a whiff of electronics in it will have an IP address in years to come, but isn't this slightly overdoing it? This capacity doesn't just allow an IP address per toaster, or even an IP address per slice of bread in a multi-slice toaster, it is far greater than that.

"There is at least one documented tale of a small company with such random IP addresses causing serious problems for a large multi-national, merely by connecting to the Internet".

The point of this apparent overkill, of course, is to simplify the problem of routing. If we massively over-allocate the address space, it doesn't cost us much in resource terms, but means that we can create multi-level hierarchies of address allocation. This in turn means that routing algorithms, and the amount of space needed for routing tables, becomes hugely simplified. Router table explosion is a well-known phenomenon, and anything that makes the problem of directing packets to their correct destination easier is well worth the investment of effort.

What's still more exciting is that this hierarchical approach will make automatic router configuration a much more viable proposition than it is at the moment. As the Internet grows, and the number of addresses increases, so too the number of possible end-to-end paths increases as the square of the allocated addresses, while the number of possible intermediate routes expands by an even greater factor. Be-

IPv6

cause the number of allocated addresses increases, there becomes a point where automatic configuration is essential if you're not going to tie up everyone with more than a passing knowledge of computers configuring their routers, and their neighbour's routers.

Finally, the most obvious reason for the apparent overkill is that, while the IPv4 to IPv6 transition should be a moderately painless exercise for most people, given the amount of thought and effort which has been put in to the changeover and the backward compatibility of IPv6, we don't want to have to go through this sort of exercise any more often than we absolutely have to. To make a small change now, only to have to repeat the operation in 20 years' time, would be an appallingly shortsighted action.

"The proponents of IPv6 claim that to achieve the same level of security with IPv4 as is available with IPv6 would need more work, and would thus cost more money, than upgrading to the improved protocol."

A New Address

Most of us are familiar with the IPv4 address convention, where the address is written as four numbers between zero and 255 separated by dots. For example, one of my host addresses is 194.153.11.222. This is merely the accepted convention, as many users are rather more comfortable with decimal numbers than hexadecimal. However, when we extend the address from 32 bits to 128 things start to get rather unwieldy. No-one wants to have to type in large strings four times the length of the current IP address. Try remembering 194.153.11.222.128.17.135.44.240.36.97.66.205.221.52.4 2 for more than a few minutes and you'll see what I mean. Double that for the workstation IP address and the gateway address and we're into serious Post-it Note overload – not to mention mistyping errors.

IPv6 simplifies the problem in two ways: firstly it uses hex numbers (base 16, 0-F) instead of decimal, since people are rather more familiar with hex numbering schemes than once they were, and secondly it compresses the resulting address by allowing the removal of some zeros. So a typical address in its long form would look something like: DEAD:BEEF:0000:0000:0000:0073:FEED:F00D. Notice that the separators are now colons rather than full stops.

Now since a typical IPv6 address might have a number of zeros, this address can be shown in a shorthand version which is DEAD:BEEF::73:FEED:F00D. The convention here is that leading zeros within the four digit groups can be dropped, so 0073 becomes 73. A group of consecutive 16-bit numbers with the value of zero can be replaced with a double colon. It's only possible to replace one null string with the double colon, which can then be filled out to retrieve the original long form address. If there are two null strings, only one can be compressed like this because if both were compressed it wouldn't be possible to determine how long each one was, and so you'd have an ambiguous address.

Finally, there's a slightly modified form of IPv6 address for use when it's desirable to express an IPv4 address in IPv6 format. To save endless (and error prone) conversion between base 10 and base 16, this convention uses the old style dot notation for the last 32 bits of the address, so my IP address above appears as 0000:0000:0000:0000:0000:0000:194.153.11.222, which of course compresses into the short form address of ::194.153.11.222. So, despite the fact that we've quadrupled the address space, our old IP number can be expressed unambiguously in the new format with only two additional characters.

Headers

One of the deficiencies of IPv4 identified by the committees was the complexity of its headers. If these were allowed to grow by the same factor as the address space was to be enlarged, then things would start to get rather unwieldy. The IPv4 header has a total of 10 fields, the two 32-bit address fields (one for the source, one for the destination), and an options field which is padded to bring the whole header up to the correct length. Even with nothing in the options field, an IPv4 header is 20 bytes long, so clearly an 80 byte header for IPv6 was not a desirable thing.

So the IPv6 header is simplified by allowing headers to be chained together. There are now only six fields – the two 128 byte addresses for source and destination, and no options. Variations in the header that would have

"To make a small change now, only to have to repeat the operation in 20 years' time, would be an appallingly shortsighted action."

been contained within the IPv4 header, or its options field, are now identified using a new field, which specifies that another header is included after the current one but before the data itself. The first header defines the minimum needed for an IPv6 packet, including the version, priority, flow label, payload length and hop limit, and includes a field to say "and there's another header after this one". There is no limit to the number of headers that can be chained together in this way. As the next header field is an 8-bit number, there can be 255 different types of header. Only six different types are defined at present.

- Hop-by-hop options.
- Routing header.
- Fragment header.
- Authentication header.
- Encapsulating Security.
- Payload header.
- Destination options header.

The result of this simplification, and improved flexibility, is that the simplest IPv6 header is still only 40 bytes long – or double the size of the IPv4 header without options – despite the fact that the two addresses it incorporates are four times the size of the IPv4 header. Of course, if you decide to have all the trimmings, the header could get quite large, although this is not possible at present as only six header types have been defined. The new solution is much more elegant, in that straightforward tasks need only produce simple and lightweight headers, while allowing more complicated applications or systems to add whatever intricacy they need. The reduced complexity of the default IPv6 header clearly makes the task of the average router much easier than it otherwise might be.

"The simplest IPv6 header is still only 40 bytes long – or double the size of the IPv4 header without options – despite the fact that the two addresses it incorporates are four times the size of the IPv4 header."

Configuration

For most network administrators IPv6 seems, at first glance, to be something that solves someone else's problems. After all, most of us already have a nice big block of IP addresses allocated which will see us through the next few years. So, surely the only people who need it are those recent Internet users? In truth IPv6 is probably more significant to the administrator of an established IP network than it is to the newcomer. This is because IPv6 has significant features that enhance the ability of a host to configure itself. Most of us know that – despite all the aids like **BOOTP** and **DHCP** – many network administrators, or most likely their hard-pressed assistants, spend quite a bit of time typing IP numbers into address fields in one control panel or configuration utility after another. It's said that over half the IP networks in the world have manually defined addresses. Anything which will help this situation is to be welcomed. One research study has suggested that IPv6's configuration automation could pay for itself – compared with manually configured IPv4 – within 12 months.

The aim of the designers of this aspect of IPv6 was that a host should be able to discover automatically all the information it needs to connect to the Internet, without human intervention. This sounds like a tall order – especially to those of us used to IPv4 – but it's not as complex or as telepathic as it sounds. The minimum requirement for the host is that it should be able to generate one unique IP address, and discover at least one router address. It needs to be able to do this whether or not there's a server or a router on its local subnet.

In fully automatic mode (which might be the system chosen in a small office with no local IT skills available), the interface will assign an address to itself first by establishing a Link Local address – an address valid only on the local subnet. It will then use the IPv6 Link Local prefix as the beginning of the address, and add a unique number – perhaps the Ethernet card's physical address – as the suffix, with the intervening space padded out with zeros. Obviously, it's perfectly possible for two cheap clone ethernet cards to have the same address even though they're not supposed to – these things have been known. So the system then sends out a solicitation message which says "hello has anyone else out there got the same address as me?". If the answer is yes, the host with the duplicate address will reply, and the questioning host can add an offset – a random number in the simplest example – and try again. This system wouldn't make

"For most network administrators IPv6 seems, at first glance, to be something that solves someone else's problems. After all, most of us already have a nice big block of IP addresses."

IPv6

the optimum use of address space, but it would remove many of the problems associated with address configuration in small businesses.

In a more sophisticated environment, where there's a router between the host and the outside world, the host needs to determine the address of the router or routers. This is done with a different solicitation message that asks the routers to identify themselves. The response from the router will tell the host if it should continue to use automatic configuration, or if it should look for a DHCP server, and where to find it. If it's told to continue automatic configuration, the host will use the router's address information to create a routable address for itself.

In addition to the solicitation messages, IPv6 routers also send out advertisements, telling the hosts what IP addresses are available to that router. In this way it's possible to manage a switch-over between different Internet providers without having to manually reconfigure the hosts, even if DHCP or a similar scheme is not in use.

"The hierarchical approach to IPv6 will make automatic router configuration a much more viable proposition than it is at the moment."

Beyond Data

One of the significant shifts we're going to see in Internet traffic is a huge growth in the use of the Internet as a broadcast medium carrying video and audio in addition to the more customary Net traffic. Such things are in their early days at present, but there's little doubt that they will come to represent the bulk of data traffic on much of the Net eventually. While IPv4 is capable of dealing with such traffic, it lacks some of the features that would be designed in once the significance of such traffic was realised, including any future proofing. This re-engineering exercise for IPv6 allows the introduction of these features alongside other innovations.

The principle concern when you're mixing real-time data like video or audio broadcasts or conferencing with data, is that it's vital to be able to reserve network bandwidth for certain tasks. This is to protect network bandwidth from being swamped by the traffic from the more demanding applications. If the sales team can't quote prices to customers because their query requests to the server are being blocked by the CEO's MPEG home movies, then you've got a problem.

One dilemma with broadcast techniques across the Internet is that channels of various capacity may connect different recipients of the same data. It would be less than optimal for the system to deal with this by backing off the transmission to match the capacity of the slowest receiver. Imagine your TV losing colour information because someone in the next town turned on a monochrome set.

IPv6 copes with these situations by splitting data types into congestion-controlled traffic (mail, FTP, NFS) and non-congestion controlled traffic (real-time data) and assigning a value to a packet in the field labelled Drop Priority. In the CC group, control traffic is higher priority than interactive traffic like a telnet session, which in turn is a higher priority than mail. In the real-time category it's up to the application designer to define how the priorities would be assigned, but typically for a broadcast application, low bandwidth audio and video would have higher priority than those packets carrying the higher resolution parts of the audio and video data. In this way, if the channel gets congested, the low resolution signals get through, but the HDTV part gets dropped by the routers, so that at least the viewer on the end of a poor feed gets to see something sensible, rather than getting high quality video with no sound, or vice versa.

"One of the significant shifts we're going to see in Internet traffic is a huge growth in the use of the Internet as a broadcast medium carrying video and audio."

Making Things Secure

Perhaps the most stinging criticism that is aimed at the Internet in general, is the relatively poor level of security it offers at the core level. Even the popular press has carried articles containing dire warnings of the disaster that will result in sending your credit card details over the Internet. While much effort has been expended by many companies and organisations to address this, lingering doubts may still remain.

However, the IPv6 specification incorporates security right at the lowest level from day one. The security protocols are known as IP-Sec and are implemented using the optional headers to provide authentication and what's known as the Encapsulating Security Payload (ESP). Some of these facilities are available in IPv4. The proponents of IPv6 claim that to achieve the same level of security with IPv4 as is available with IPv6 would need more work, and would thus cost more money, than upgrading to the improved protocol.

The Authentication and ESP parts of the security specification can be im-

plemented independently, which is important because the present rather draconian US export regulations covering encryption techniques and algorithms may well forbid the export of the latter for general use. Nevertheless, the ESP header includes a field that defines the level of security and how it is implemented. While versions with keys greater than 40 bits may well be subject to export restrictions, those with weaker encryption schemes may be exported without controls.

The Big Switch

It is understood that a controlled roll-out of a new Internet protocol simply won't happen. The Internet is an anarchic place at best, without trying to persuade companies and individuals to upgrade their systems on a particular day. Even more restricting is that some OS vendors are not likely to be in a position to offer IPv6 as an option for some time.

If you're using Microsoft's Windows NT or Windows 95, for example, then Microsoft doesn't expect to ship an IPv6-compatible protocol stack for at least 18 months. There's always the option to purchase an IPv6 protocol stack from a third party specialist vendor. FTP Software is already shipping an IPv6 stack for both NT and 95, but this will be an additional cost item. Many companies and users are still happy to use the free IP stack bundled with their OS.

Clearly, the two different implementations of IP must be able to co-exist for a prolonged period, possibly many years. Not only must IPv4 hosts be able to communicate over IPv6 networks (which is the easier thing to do, obviously) but the reverse must be true. If you have two islands of IPv6, they must be able to communicate across an IPv4 connection. This latter task is clearly the more difficult, and is achieved by including enough information in the IPv6 headers to allow them to tunnel through the functionally narrower channel represented by the IPv4 connection.

The only trick that the Internet regulators will need to pull off to make this work, is to try to persuade the world to roll out IPv6 before the supply of IPv4 addresses runs out. If this happens, then any IPv4 addresses that have been applied for and correctly assigned will still be unique – although there will still be those rogue illegal addresses made up on the spur of the moment by desperate network administrators. If it doesn't happen in time, then there will be a small but significant number of duplicate IPv4 addresses – in which case, there may well be a requirement for header translating gateways to ensure proper communication between IPv4 hosts and IPv6 hosts.

"The driving force of the new standard is the rapid growth of the Internet, and IPv6 is being introduced to overcome the address space restrictions of the old one."

Conclusion

When I started to look into the prospect of implementing IPv6 I was both sceptical and nervous. It seemed to me that the roll-out would be far from problem free. The prospect of attempting to convert an entire company with many thousands of hosts spread over dozens of sites worldwide, while making sure that all their Internet providers understood the problems, and were ready (and willing) to undertake a synchronised changeover seemed to represent an insurmountable project management task. Even if it could be done, I found it hard to see the real benefit for an organisation that was not yet running short of addresses within its allocated space.

Further study has left me more than a little impressed. It's obvious with hindsight that the implementation of a new version of something as ubiquitous as IP would have to be thought out thoroughly, and would have to include forward and backward compatibility as part of its fundamental design. This has been achieved, and I'm much less concerned about the changeover than I once was.

What has impressed me much more, though, is the amount of effort that has been put into removing many of the bugbears of configuring a complex IP network. The automatic configuration facilities, both for hosts and for routers, have been described by some as worth the cost of switching to IPv6 all on their own.

It remains to be seen if these cost-benefit claims will turn out to be accurate, of course. I'm also sure that early adopters of IPv6 will experience their own teething troubles and specially refined version of chaos. Nevertheless, the switch will have to come, and it might be best to grasp the nettle sooner rather than later.

PCNA

The Author

David Morton is a network consultant and author. He can be contacted by email as dmorton@cix.co.uk.

Additional Resources

- **TCP/IP Tutorial**
- **The OSI 7 Layer Model Explained**
- **Understanding Frame Relay**
- **Understanding DHCP**
- **Virtual Private Networking Explained**

All these articles are available free online now at
www.pcnetworkadvisor.com

PCNA