4

The Road to Next Generation

No matter how successful IPv4 has been, in hindsight there's no denying that it could have been a better protocol. As its popularity increased, replacing or updating it has become increasingly more problematic. Lack of network address space is very likely the engine driving adoption of IPv6, but there are other reasons being proposed for moving to support IPv6. This chapter opens with a discussion of some of the improvements that have been proposed for the next generation of IP, followed by a brief history of the development effort for IP Next Generation (IPng).

4.1 Early Assumptions About the Internet Environment

Once it became clear that the Internet would soon grow beyond the capacity of IPv4, RFC 1287, "Towards the Future Internet Architecture," was published (December 1991). This document outlined the results of

a January 1991 meeting of the Internet Activities Board (IAB)[1] and the Internet Engineering Steering Group (IESG), including the basic assumptions that could (it was thought) be made about the future of the Internet and what were the most important areas for development of the Internet protocols.

The group's four broad assumptions were meant to characterize the best guess about what networking would be like during the next 5 to 10 years. Agreement on what the networking environment would be like led to appropriate planning for the future. The assumptions (and the eventual realities) were as follows.

- The TCP/IP protocol suite would coexist with its main rival, OSI, for some time. The International Organization for Standardization (ISO) developed the Open Systems Interconnection (OSI) architecture (source of the famous seven-layer OSI network protocol model). In fact, TCP/IP quickly gained the lion's share of the internetworking market. OSI continued to have influence only insofar as it had been chosen for use by government organizations.
- The Internet itself would become more complex, incorporating more diverse and a greater number of different types of networking technologies. In other words, instead of settling on one or a handful of network connectivity media, an increasing population of network connectivity media would become available and used over time. In fact, this is the case—sort of. Ethernet has come to dominate the LAN market, while a handful of other networking technologies (*ATM, Frame Relay, wireless Ethernet) have become dominant in other segments of the market.
- Access to the Internet would be provided by a variety of different carriers, including both public and private providers, for a wide variety of different networks. In other words, networks for many different types of organizations, including corporations, government agencies, educational institutions, and public services, will be connected through common carrier service providers as well as by privately maintained network connections. In fact, this assumption has also proven

[1]The IAB was later renamed the Internet Architecture Board, allowing the acronym to remain unchanged.

itself with some qualification. What might be called *ad hoc internetworking*—where one organization provided connectivity to one or more other organizations or individuals by routing those others' packets—was common in the early days of the Internet. However, by the mid-1990s backbone-oriented routing—where global Internet connectivity is offered to consumers, whether individuals, organizations, or government agencies—became the dominant model.

- The Internet must be able to interconnect as many as one billion (10^9) networks, although the consensus seemed to encompass a relatively broad range of anywhere from ten million to ten billion networks.

Even before NATs began masking untold numbers of hosts from any automated surveys, estimates of the global number of TCP/IP nodes were best-guesses. Organizations rarely advertise host IP numbers anymore, and with most new computers shipping with TCP/IP installed (whether Microsoft Windows, Apple, or *nix operating systems), the number of TCP/IP nodes can be assumed to approach the number of computers currently in use. By 2004, I doubt anyone would argue that there are fewer than 100 million or more than 1 billion IP nodes currently in operation. Thus, this assumption (the need to support as many as 1 billion networks or more) is clearly still within range.

The number of networks to be interconnected is still not entirely clear, although it has become clear that the IPv4 address space is insufficient. On the one hand, we could allot one network address to every computer in the world and still be well under the high-end estimate of networks needed. On the other hand, further rapid decreases in cost and size coupled with increases in the distribution of personal computers could create demand as high as (or higher than) one network for every human in the world, thus requiring on the order of at least 10 billion networks just to be assigned to individual people. Factoring in unforeseeable circumstances such as these led some to call for an address space that can handle at least a trillion globally unique networks.

4.2 Designated Areas for Internet Evolution

The January 1991 IAB/IESG meeting generated another list, this one of the areas that were deemed most important to further architectural growth.

The intention was to identify the areas on which development efforts should be focused. These included the following.

- Routing and addressing concerns
- Multiprotocol architecture
- Security architecture
- Traffic control and state
- Advanced applications

These areas, approaches to development, and other issues are discussed next.

4.2.1 Addressing and Routing

The address space was already clearly a problem, but the issue of ballooning routing tables was also of great concern. Another RFC published at about the same time cited routing tables with 5000 and 7000 entries as a looming impediment to performance on networks that were still growing rapidly. The authors of RFC 1287 suggest not only that the IPv4 address space will be depleted but also that at some point before then IPv4 routing algorithms will fail due to the large number of networks. They also suggest that multiple routes between sources and destinations will make possible type of service (ToS) variations and therefore require some mechanisms to control route selection.

Aggregation of network routes, through some mechanism to be determined, is suggested as one possible solution to the explosion of routes. Using some method of defining boundaries between large routing domains would help improve routing efficiency. Another suggestion solicits some efficient mechanism for the computation of network routes, as well as some mechanism for routers to maintain state associated with specific streams that are routed in some special way.

Potential addressing fixes include the use of the existing 32-bit address space as a nonglobally unique identifier. In other words, addresses might be reused in different parts of the network that don't interoperate directly. For example, dividing the world into different routing domains would allow a host address to be used once in each domain, with interoperation between the domains mediated by protocol gateways that rewrite the addresses as they pass over domain boundaries.

Another suggestion for addressing simply increased the size of the host address. A third suggestion expands the host address field and uses the entire field as a nonhierarchical address space, with a connection setup that gives routers the opportunity to map a host address to an administrative domain.

4.2.2 Multiprotocol Architecture

Support for interoperable transmission of OSI as well as TCP/IP traffic was thought to be an important criterion for further development. The perception at the time (up to 1991) was that Internet connectivity meant a host had an Internet address. If you didn't have an IP address and weren't running IP, you weren't connected. This viewpoint was already eroding by 1991, with the authors of RFC 1287 suggesting that connectivity could be based on access to the Internet through email gateways or, more simply, through some application. For example, users on NetWare networks at the time could run Internet applications like web browsers and email clients on their systems but use the Internetwork Packet eXchange (IPX) protocol to transport the data on their local Novell NetWare networks.

In practice, acceptance of TCP/IP as an internetworking protocol suite by most software and hardware vendors during the 1990s has largely driven out competing internetworking protocol suites. Even Novell finally deployed its NetWare network operating system as a native TCP/IP product by 1998.

More important, at least in hindsight, was the comment that TCP/IP could integrate or cross-pollinate with other application protocols. Interoperability, particularly between applications rather than at the lower layers of the protocol stack, was deemed to be a good thing.

4.2.3 Security Architecture

Department of Defense funding of significant research and development work that produced IP meant that the protocols were (at least according to the authors of RFC 1287) built with military security in mind. Although a set of vaguely military priority levels were defined for a first pass at quality of service at the IP layer (in RFC 791), there are no mechanisms for strong cryptographic authentication, access control, authorization,

or confidentiality evident at the IP layer until the early 1990s, when work on the IP Security Architecture (IPsec, see Chapter 6) began.

One specific suggestion for a desired security service is the use of *distinguished names* (an OSI construct used in X.500 directory specifications) that can be authenticated in order to implement access controls. Integrity enforcement was also suggested, with mechanisms to prevent modification of transmissions, spoofing of transmission origins, and defense against *replay attacks* (attacks in which an interceptor replays data stolen from an authorized stream). Other services include confidentiality (encrypted transmission), nonrepudiation (use of digital signature algorithms to prevent a sender from denying having sent a message), and protection from denial of service (DoS) attacks.

Other security issues raised in RFC 1287 include router/gateway protocol filtering (in other words, packet filtering firewalls) and encryption key management/storage.

4.2.4 Traffic Control and State

IPv4 is a connectionless protocol, but some applications—audio and video, for example—depend on some degree of traffic control to work properly. A video stream must arrive at its destination at a relatively dependable and predictable rate, not too fast (which might overwhelm the recipient node's buffers) and not too slow (which would degrade the quality of the transmission).

The authors of RFC 1287 suggest the need for some sort of packet queuing mechanims to provide traffic control; they also state that there should be some mechanism by which nodes can maintain status information for different *streams* of packets to more readily enable real-time applications to be carried over IP packets.

Noting that IPv4 implements a Type of Service (ToS) field, the authors also note that not only is ToS not generally implemented, it is not even clear how it could be implemented.

4.2.5 Advanced Applications

Rather than suggesting new applications, the authors of RFC 1287 suggest that improving and simplifying the processes involved in developing

new and advanced applications would be a more productive path. As a starting point, they suggest that the creation of common data formats for different types of data, particularly text, images and graphics, audio and video, workstation displays, and data objects. Also important to developing advanced applications are mechanisms for the exchange of these different types of data.

Suggested mechanisms include *store and forward* services, global file systems, interprocess communications, data broadcast, and a standardized method for accessing databases.

4.3 Room for Improvement

Other areas in which IPv4 could stand some improvement have been cited over the years as providing good reasons to upgrade the protocol. As it became more apparent that IPv4 could use some additional, or at least different, functionality, upgraders were faced with the opportunity to enhance IP in ways that go beyond adding network addressing capacity. This section highlights some of the areas where there is room for improvement, from network administration and automatic node configuration to rethinking ToS and IP options.

4.3.1 Network Administration and Configuration

IPv4 and most of the rest of the TCP/IP application protocol suite were never designed, by themselves, to be easy to use. For example, raw FTP (File Transfer Protocol) depends on what appear to be very arcane request and reply codes and uses a set of cryptic-seeming commands. Why do I mention this? Simply because these apparently complicated command and control mechanisms are actually designed to be standard across all platforms and to simplify access to software that understands the protocols. A system running IPv4 must be configured, correctly, with an apparently complicated set of parameters. These usually include a host name, IP address, subnet mask, default router, and some others (depending on the implementation). This is complicated—it means that the person who does the configuration must understand all these parameters or at least be given them by someone who does understand. What it means is that getting a system connected to an IPv4 network can be very complicated, time-consuming, and costly.

The Boot Protocol (BOOTP) took a first step toward simplifying the process of connecting a host to a network. This relatively simply protocol provided a mechanism for a host with minimal preconfiguration (often simply a terminal) to query a BOOTP server to get its IP configuration parameters. This approach failed to solve the entire problem because it only provided a mechanism for the BOOTP server to map IP address and other configuration information to a link layer address (for example, an Ethernet card interface address). To manage 100 hosts with BOOTP, you must assign each host its own IP address.

Address management and host configuration pose at least two big problems. First, if it is difficult to configure hosts, it costs money; second, if each host must tie up an IP address, whether or not it is connected, it costs address space. It would be nice if we could make host configuration a plug-and-play operation—in other words, so simple that you simply plug the system into the network and it is automatically configured. It would also be nice if we could figure out a way to share IP addresses among many hosts, so that if no more than half of our 100 hosts were connected at any given time, we could get away with sharing 50 IP addresses among them.

As it turns out, another protocol, called the Dynamic Host Configuration Protocol (DHCP), was built on top of the BOOTP framework in an attempt to address these issues. Still using a client/server model, clients can use DHCP to query a server for configuration information, just as with BOOTP. However, DHCP adds more flexibility in terms of what kind of configuration information can be provided as well as how IP addresses are allocated. There are three mechanisms for allocating addresses.

- Using *automatic allocation*, hosts request an IP address and are given a permanent one that they use each time they connect to the network.
- Using *manual allocation*, the server assigns specific IP addresses to individual hosts based on a list provided by a network administrator. These IP addresses are reserved, whether or not the hosts request them.
- Using *dynamic allocation*, the server doles out IP addresses on a first-come, first-served basis; hosts are allowed to use the addresses for a specific time period after which the address "lease" expires.

Both automatic and manual allocation will tend to inefficiently distribute IP addresses; using automatic allocation may tend to tie up IP addresses.

If an organization has more hosts than users, it could burn up as many IP addresses as it has hosts with this scheme. Manual allocation means network administrators must configure an IP address for each host, whether it connects once an hour or once a year to the network. Dynamic allocation, however, enables a relatively large population to share a relatively small number of IP addresses.

Unfortunately, DHCP falls short of enabling true plug-and-play configuration because it is stateful. That is, DHCP maintains the status of different IP addresses and the hosts using them. You have to explicitly set up a DHCP server that knows about your hosts, and the host to be configured with DHCP must know about the nearest DHCP server. True plug-and-play, which is a big part of the portability issue, doesn't happen with IPv4. As we'll see following, the inability of IPv4 to adequately support portability and network administration issues helps prompt the calls for upgrade to IPv6.

4.3.2 Type of Service (ToS)

IP uses a packet-switched network architecture. This means that a packet might take any of a number of different routes to reach its destination. Those routes differ: Some might cost more, some might allow greater throughput, some might have lower latency, and some might be more reliable than others. IPv4 provides a mechanism, the Type of Service field (ToS) that allows applications to tell IP how to handle their data streams. An application that needs lots of throughput—for example, FTP—might force the ToS to favor routes that have lots of bandwidth; an application that needs fast responses—for example, Telnet—might force the ToS to favor routes that have low delays.

This was a good idea that never really caught on that well with implementers. For one thing, it requires routing protocols to incorporate notions of preferential routes based on costs as well as the need to track values for latency, throughput, and reliability for available routes. For another thing, it requires that developers implement a function in their application that might request service that, ultimately, could affect performance. ToS is a choice of one, so if you decide that low latency is most important to your application, it might affect your ability to get higher bandwidth or more reliable routes for your application's packets.

4.3.3 IP Options

The IPv4 header includes a variable-length options field. IP options were meant to be the way to handle certain special functions. The original specifications left these options undefined, but eventually options for things like security as well as certain routing functions were added. Routing options include one (record route) to have each router handling the packet to record its address and another (timestamp) to have each router record its own address as well as the time it handles the packet. Source routing options are also available: Loose source routing specifies a list of routers that the packet must pass through on its way to the packet's destination, whereas strict source routing requires that the packet be routed only by the routers listed.

Options are an important part of IP, but the IPv4 implementation is not ideal. Although they are not often used, it is not because they are not useful so much as that the specification is suboptimal. Rather than throw options out, IPv6 improves the way they are used.

The problem with options is that they are special cases. IP datagrams without options are the vast majority and are the type of datagrams vendors optimize their routers to handle. The IP header without options is always five bytes long and is easy to process—especially when the router design optimizes for the processing of such headers. Performance is key to router sales, and because most traffic does not use IP options, the routers tend to handle those packets as exceptions, shunting them off to the side to be handled when it is convenient—and when it won't affect the router's overall performance.

Despite the benefits of using IPv4 options, the cost in terms of performance has been enough to keep them from being used very often.

4.4 IPng Candidates

Up to 1994, quite a few different proposals were made for the successor to IPv4. By 1992, the three dominant proposal families that would eventually be considered by the IETF in 1994 had already taken shape. RFC 1347, "TCP and UDP with Bigger Addresses (TUBA), A Simple Proposal for Internet Addressing and Routing," outlines one. *TUBA* can be characterized as

simply replacing IP with the OSI internetwork protocol, *Connection-Less Network Protocol* (*CLNP*). CLNP uses *Network Service Access* (*NSAP*) addresses that can be any length but that are often implemented in 20 bytes, providing more than enough address space. Furthermore, using CLNP would help IP and OSI to converge, while at the same time eliminating the need to build an entirely new protocol.

Another proposed IPng candidate was first known as IPv7 in 1992, and in 1993 was described in detail in RFC 1475 under the title "TP/IXTP/IX: The Next Internet." It is not clear what TP/IX stands for; according to Christian Huitema in *IPv6: The New Internet Protocol* (Prentice Hall PTR, 1998), the name expresses the desire of its proposer, Robert Ullman, to change not only IP but also TCP with the upgrade. TP/IX uses 64-bit addresses and adds an addressing layer to the hierarchy, above organizations, for administrations.

Under IPv7, eight-byte addresses are used to allocate three bytes to administrative domain, three to the organization's network, and two bytes for the host identifier. The IPv7 datagram header simplifies the IPv4 header, while adding a forward route identifier to be used by intermediate routers to determine how to handle datagrams. For example, the forward route identifier may be associated with a particular route based on certain values relating to the route itself (throughput or value) or to be associated with a particular datagram stream or even to be associated with data from a mobile host—that is, a host that moves from one network to another while maintaining open TCP connections. TP/IX not only modified TCP and UDP, but it also included a new routing protocol called *RAP*.

TP/IX later evolved into another proposal, described in RFC 1707, "CATNIP: Common Architecture for the Internet." CATNIP seems to have little in common with TP/IX, however, except that it retains the IPv7 designation. In its goal of providing a common architecture, the CATNIP specification makes allowances for the three most commonly used internetwork architectures: TCP/IP, OSI, and IPX, as well as discussion of how to integrate a competing proposed standard for the next generation of IP. The stated objective is to make it possible for all existing systems to continue to interoperate with **/no/** modifications, no changes in address, and no software upgrades for individual hosts. By making allowance for different network architectures, the CATNIP proposal meant to minimize impact on the actual infrastructure; however, it meant adding a layer of complexity in order to implement true interoperable internetworking.

The third proposal stream started out as something called *IP in IP*, or *IP Encaps* (for IP encapsulation). Under this proposal, there would be two layers of IP: One would be used for a global backbone, while the other would be used in more limited areas. The IP to be used in limited areas could continue to be IPv4, while the backbone would use a new layer with different addressing. Ultimately, this evolved and merged with other proposals to become the *Simple Internet Protocol Plus* (*SIPP*) proposal.

As explained in RFC 1710, "Simple Internet Protocol Plus White Paper," the SIPP working group grew from three different IETF working groups focused on developing an IPng. The first group was working on a version called *IP Address Encapsulation* (*IPAE*); the working group, chaired by Dave Crocker and Robert Hinden, proposed extensions to IPv4 that would carry larger addresses, and the group focused on developing transition mechanisms.

Somewhat later, Steve Deering proposed a new protocol evolved from IPv4 called the *Simple Internet Protocol* (*SIP*). A working group was formed to work on this proposal, which was chaired by Steve Deering and Christian Huitema. SIP used 64-bit addresses, a simplified header, and options in separate extension headers. After lengthy interaction between the two working groups and the realization that IPAE and SIP had a number of common elements and the transition mechanisms developed for IPAE would apply to SIP, the groups decided to merge and concentrate their efforts. The chairs of the new SIP working group were Steve Deering and Robert Hinden.

In parallel to SIP, Paul Francis (formerly Paul Tsuchiya) had founded a working group to develop the "P" Internet Protocol (Pip). Pip was a new Internet protocol based on a new architecture. The motivation behind Pip was that the opportunity for introducing a new Internet protocol does not come very often and given that opportunity important new features should be introduced. Pip supported variable-length addressing in 16-bit units, separation of addresses from identifiers, support for provider selection, mobility, and efficient forwarding. It included a transition scheme similar to IPAE.

After considerable discussion among the leaders of the Pip and SIP working groups, they came to realize that the advanced features in Pip could be accomplished in SIP without changing the base SIP protocol as well as keeping the IPAE transition mechanisms. In essence, it was possible to keep the best features of each protocol. Based on this, the groups decided

to merge their efforts. The new protocol was called Simple Internet Protocol Plus (SIPP). The chairs of the merged working group are Steve Deering, Paul Francis, and Robert Hinden.

Briefly, SIPP offers several changes from IPv4, including the following.

Routing and addressing expansion SIPP specifies 64-bit addresses, double the size of IPv4. The intention is to provide greater degrees of hierarchy within which routing can be accomplished. Another feature is the addition of *cluster addresses*, which identify regions of the network topology. SIPP address extensions, available in units of 64 bits, work with the cluster addresses to create the possibility of a much larger address space.

IP header simplification SIPP does away with some IPv4 header fields, while streamlining the structure to help improve routing efficiency.

Improvement in option implementation SIPP uses a more flexible approach to encoding and implementing IP options.

Quality of service SIPP makes it possible to label datagrams as belonging to specific data flows. Hosts can request special handling for the routing of these flows, especially useful for applications that depend on real-time delivery like that required by video or audio transmission.

Authentication and privacy SIPP adds extensions for authentication, data integrity, and confidentiality.

SIPP was the result of many people from several different groups working together. The finished specification includes many interesting new mechanisms, while still not straying too far from the goal of being an upgrade to IPv4 rather than an entirely new protocol built from the ground up. Notable is the use of routing similar to that in IPv4, still using CIDR to add flexibility and improve routing performance. Also important are new routing extensions that allow choice of routes from different providers based on various criteria (including performance, cost, provider policies for traffic, and so on). Other routing extensions include support for mobile hosts as well as automatic readdressing and extended addressing.

One other notable mechanism is the SIPP approach to IP options: Rather than including them as part of the basic IP header, SIPP segregates any

IP options from the main header. The options headers, if any, are simply inserted into the datagram after the header and before the transport layer protocol header. This way, routers can process datagrams without having to process the options headers unless it is necessary—thus improving performance overall for all datagrams.

RFC 1710 provides both a technical overview to the SIPP specification and a readable justification and narrative of the protocol. It is worth a look, if only to see how IPv6 as we know it came to be—because SIPP, with some modifications, was the specification recommended to and accepted by the IESG as the basis for IPng.

4.5 IPv6, The Next Generation

RFC 1752, "The Recommendation for the IP Next Generation Protocol," published in January 1995, is a fascinating document that outlines clearly what was needed and what was available, in terms of the candidate proposals for successors to IPv4. In its summary, the authors of RFC 1752 describe what IPng would look like.

This protocol recommendation includes a simplified header with a hierarchical address structure that permits rigorous route aggregation and is also large enough to meet the needs of the Internet for the foreseeable future. The protocol also includes packet-level authentication and encryption along with plug-and-play autoconfiguration. The design changes the way IP header options are encoded to increase the flexibility of introducing new options in the future while improving performance. It also includes the ability to label traffic flows.

The fifth item in a long list of specific recommendations is that IPng be based on SIPP with 128-bit addresses. The rest of the RFC provides an excellent resource for further historical background on how the Internet research community identified and approached the problems associated with IPv4, as well as detailed analysis of the three contenders, TUBA, CATNIP, and SIPP. The RFC examines each proposal and discusses how it meets (or fails to meet) the requirements and also presents the results of the proposal review process.

All three proposals are praised in some way, and all ultimately contributed something to the final recommendation. For example, SIPP did

not include a strong transition plan or a totally acceptable mechanism for autoconfiguration, so the recommendation draws on the TUBA proposal for those areas. And SIPP was not accepted in all its glory: The concept of address extensions was ultimately considered too experimental and potentially risky to incorporate into the IPng work, while the 64-bit address space was replaced with a 128-bit address space to cope with any future uncertainties.

The recommendations described in RFC 1752 include a variety of further tasks related to the actual design of the IPng and related protocols. SIPP and the others could be considered only as starting points, particularly if IPng were to be sufficiently robust to serve the Internet for years to come.

The first proposed standard RFCs (RFCs 1883 through 1887) to describe IPv6 and supporting protocols were published by early 1996, but they were not entirely complete and were soon followed by various additions and some slight modifications. By the end of the summer of 1998, new IPv6 RFCs were being approved for publication. In particular, RFC 2373, "IP Version 6 Addressing Architecture," replaced RFC 1883 and RFC 2374, "An IPv6 Aggregatable Global Unicast Address Format," replaced RFC 2073. Other newer RFCs approved for publication describe ICMPv6, neighbor discovery, and stateless autoconfiguration for IPv6.

Even as this book is going to press, the second round of IPv6 RFCs are being updated and in some cases replaced by a third wave of specifications. For example, RFC 2373 has been replaced with RFC 3513; other updates are still works-in-progress but can be expected to further hone IPv6 and related specifications over the coming years.

4.6 Summary

Few, if any, efforts in Internet engineering history have taken so long and involved so many different ideas, people, and groups as the project to upgrade the Internet Protocol. The process is instructive for students of networking history, network protocols, and the network protocol specification process. The result, IPv6, may ultimately be considered an improvement over IPv4—but as the product of many committees, there will invariably be those who feel that IPv6 could have been better than it is.

However, before IPv6 can be fully judged, it must be implemented. IPv6-related working groups have come up with a variety of approaches to the process of migrating from IPv4-only environments to networks capable of supporting IPv6. The next chapter discusses how IPv6 support may be deployed in existing networks.