Robert M. Hinden

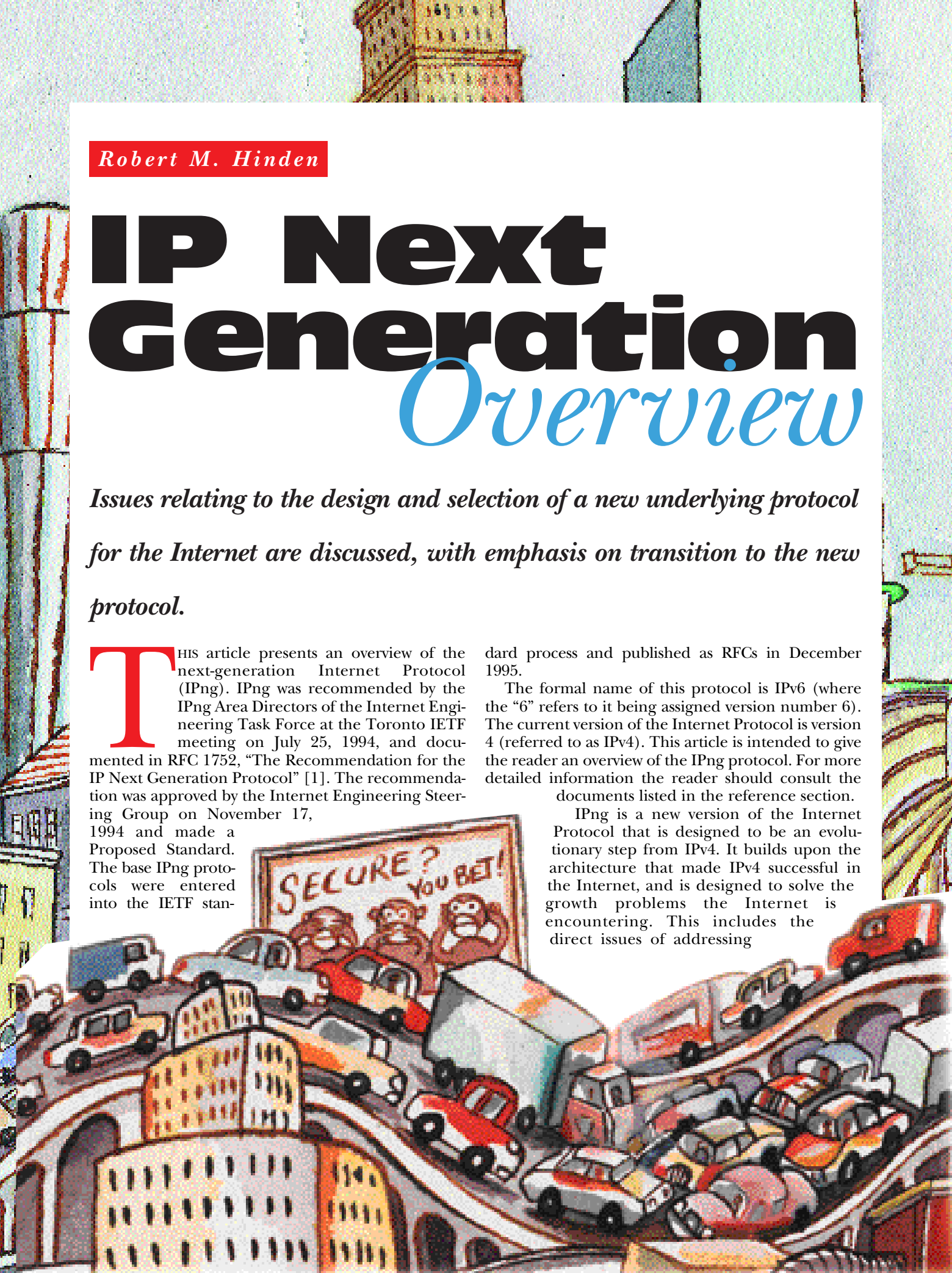
IP Next Generation Overview

Issues relating to the design and selection of a new underlying protocol for the Internet are discussed, with emphasis on transition to the new protocol.

THIS article presents an overview of the next-generation Internet Protocol (IPng). IPng was recommended by the IPng Area Directors of the Internet Engineering Task Force at the Toronto IETF meeting on July 25, 1994, and documented in RFC 1752, "The Recommendation for the IP Next Generation Protocol" [1]. The recommendation was approved by the Internet Engineering Steering Group on November 17, 1994 and made a Proposed Standard. The base IPng protocols were entered into the IETF standard process and published as RFCs in December 1995.

The formal name of this protocol is IPv6 (where the "6" refers to it being assigned version number 6). The current version of the Internet Protocol is version 4 (referred to as IPv4). This article is intended to give the reader an overview of the IPng protocol. For more detailed information the reader should consult the documents listed in the reference section.

IPng is a new version of the Internet Protocol that is designed to be an evolutionary step from IPv4. It builds upon the architecture that made IPv4 successful in the Internet, and is designed to solve the growth problems the Internet is encountering. This includes the direct issues of addressing



and routing, as well as dealing with long-term growth issues such as security, auto-configuration, real-time services, and transition.

IPng can be installed as a normal software upgrade in Internet devices and is interoperable with the current IPv4. Its deployment strategy was designed to not have any "flag" days. IPng is designed to run well on high-performance networks (e.g., ATM, Fast Ethernet, etc.) and at the same time is efficient for low-bandwidth networks (e.g., wireless). In addition, it provides a platform for new Internet functionality that will be required in the near future.

Key Issues

THERE are several key issues that should be considered when reviewing the design of the next-generation Internet protocol. Some are very straightforward. For example, the new protocol must be able to support large global internetworks. Others are less obvious. There must be a clear way to transition the current large installed base of IPv4 systems to the new protocol. It doesn't matter how good a new protocol is if there isn't a practical way to transition the current operational systems running IPv4 to the new protocol.

> The challenge for an IPng is for its transition to be complete before IPv4 routing and addressing break down.

Growth

Growth is the basic issue that created the need for a next-generation IP. If anything is to be learned from our experience with IPv4, it is that the addressing and routing must be capable of handling reasonable scenarios of future growth. It is important that we have an understanding of past growth and from where the future growth will come.

Currently IPv4 serves what could be called the computer market. The computer market has been the driver of the growth of the Internet. It comprises the current Internet and countless other smaller internets that are not connected to the Internet. Its focus is to connect computers together in the large business, government, and university education markets. This market has been growing at an exponential rate. One measure of this is that the number of networks in the current Internet (~80,000 as of January 1996) is doubling approximately every 12 months. The computers that are used at the endpoints of Internet communications range from PCs to supercomputers. Most are attached to local-area networks (LANs) and the vast majority are not mobile.

The next phase of growth will probably not be driven by the computer market. While the computer market will continue to grow at significant rates due to expansion into other areas such as schools (elementary through high school) and small businesses, it is doubtful it will continue to grow at an exponential rate. What is likely to happen is that other kinds of markets will develop. These markets will fall into several areas. They are extremely large and also bring with them a new set of requirements that were not as evident in the early stages of IPv4 deployment. The new markets are also likely to happen in parallel with one another. It may be that we will look back on the last 10 years of Internet growth as the time when the Internet was small and only doubling every year. The challenge for an IPng is to provide a solution that solves today's problems and is attractive in these emerging markets.

Nomadic personal computing devices seem certain to become ubiquitous as their prices drop and their capabilities increase. A key capability is that they will be networked. Unlike the majority of today's networked computers, they will support a variety of types of network attachments. When disconnected they will use radio-frequency wireless networks, when used in networked facilities they will use infrared attachment, and when docked they will use physical wires. This makes them ideal candidates for internetworking technology as they will need a common protocol that can work over a variety of physical networks. These types of devices will become consumer devices and will replace the current generation of cellular phones, pagers, and personal digital assistants. In addition to the obvious requirement of an Internet protocol that can support large-scale routing and addressing, they will require an Internet protocol that imposes a low overhead and supports autoconfiguration and mobility as a basic element. The nature of nomadic computing requires an Internet protocol to have built-in authentication and confidentiality. It goes without saying that these devices will also need to communicate with the current generation of computers. The requirement for low overhead comes from the wireless media. Unlike LANs, which will be very high speed, the wireless media will be several orders of magnitude slower due to constraints on available frequencies, spectrum allocation, error rates, and power consumption.

Another market is networked entertainment. The first signs of this emerging market are the proposals being discussed for 500 channels of television, video on demand, etc. This is clearly a consumer market. The possibility is that every television set will become an Internet host. As the world of digital high-definition television approaches, the differences between a computer and a television will diminish. As in the previous market, this market will require an Internet protocol that supports large-scale routing and addressing, and autoconfiguration. This market also requires a protocol suite that imposes the minimum overhead to get the job done. Cost will be the major

factor in the selection of an appropriate technology.

Yet another market that could use the next-generation IP is device control—the control of everyday devices such as lighting equipment, heating and cooling equipment, motors, and other types of equipment that are currently controlled via analog switches and in aggregate consume considerable amounts of electrical power. The size of this market is enormous and requires solutions that are simple, robust, easy to use, and very low cost. The potential payback is that networked control of devices will result in extremely large cost savings.

The challenge the IETF faced in the selection of an IPng was to pick a protocol that meets today's requirements and also matches the requirements of these emerging markets. These markets will appear with or without an IETF IPng. If the IETF IPng is a good match for these new markets, it is likely to be used. If not, these markets will develop something else. They will not wait for an IETF solution. If this should happen, it is probable that because of the size and scale of the new markets the IETF protocol would be supplanted. If the IETF IPng is not appropriate for use in these markets, it is also probable that each market will develop its own protocols, perhaps proprietary. These new protocols would not interoperate with each other. The opportunity for the IETF was to select an IPng that has a reasonable chance of being used in these emerging markets. This would have the very desirable outcome of creating an immense, interoperable, world-wide information infrastructure created with open protocols. The alternative is a world of disjointed networks with protocols controlled by individual vendors.

Transition

At some point in the next three to seven years the Internet will require a deployed new version of the Internet protocol. Two factors are driving this: routing and addressing. Global Internet routing based on the 32-bit addresses of IPv4 is becoming increasingly strained. IPv4 address do not provide enough flexibility to construct efficient hierarchies that can be aggregated. The deployment of Classless Inter-Domain Routing [8] is extending the lifetime of IPv4 routing by a number of years, but the effort expended to manage the routing will continue to increase. Even if the IPv4 routing can be scaled to support a full IPv4 Internet, the Internet will eventually run out of network numbers. There is no question that an IPng will be needed, only a question of when.

The challenge for an IPng is for its transition to be complete before IPv4 routing and addressing break down. The transition will be much easier if IPv4 addresses are still globally unique. The two transition requirements that are the most important are flexibility of deployment and the ability of IPv4 hosts to communicate with IPng hosts.

The deployment strategy for an IPng must be as flexible as possible. The Internet is too large for any kind of controlled rollout to be successful. The importance of flexibility in an IPng and the need for interoperability between IPv4 and IPng was well-stated in a message to the Simple Internet Protocol Plus (SIPP) mailing list by Bill Fink, who is responsible for a portion of NASA's operational Internet. In his message he said:

"Being a network manager and thereby representing the interests of a significant number of users, from my perspective it's safe to say that the transition and interoperation aspects of any IPng is *the* key first element, without which any other significant advantages won't be able to be integrated into the user's network environment. I also don't think it wise to think of the transition as just a painful phase we'll have to endure en route to a pure IPng environment, since the transition/coexistence period undoubtedly will last at least a decade and may very well continue for the entire lifetime of IPng, until it's replaced with IPngng and a new transition. I might wish it was otherwise but I fear they are facts of life given the immense installed base.

"Given this situation, and the reality that it won't be feasible to coordinate all the infrastructure changes even at the national and regional levels, it is imperative that the transition capabilities support the ability to deploy the IPng in the piecemeal fashion... with no requirement to need to coordinate local changes with other changes elsewhere in the Internet... .

"I realize that support for the transition and coexistence capabilities may be a major part of the IPng effort and may cause some headaches for the designers and developers, but I think it is a duty that can't be shirked and the necessary price that must be paid to provide as seamless an environment as possible to the end user and his basic network services such as email, ftp, gopher, X-Window clients, etc... .

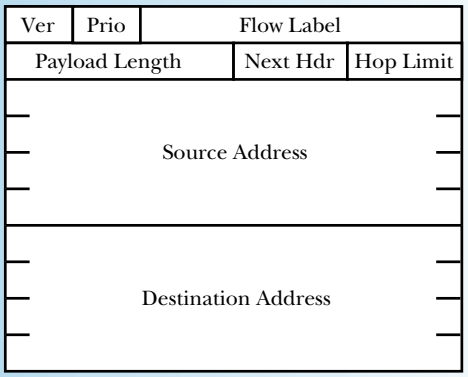
"The bottom line for me is that we must have interoperability during the extended transition period for the base IPv4 functionality... ."

ANOTHER way to think about the requirement for compatibility with IPv4 is to look at other product areas. In the product world, backward compatibility is very important. Vendors who do not provide backward compatibility for their customers usually find they do not have many customers left. For example, chip makers put considerable effort into making sure that new versions of their processor always run all of the software that ran on the previous model. It is unlikely that Intel would develop a new processor in the *x*86 family that did not run DOS and the tens of thousands of applications that run on the current versions of *x*86s.

Operating system vendors go to great lengths to make sure new versions of their operating systems are binary compatible with their old versions. For example, the labels on most PC or Macintosh software usually indicate that they require OS version *xx* or greater. It would be foolish for Microsoft to come out with a new version of Windows that did not run the

- Current address assignment policies are adequate.
- There is no current need to reclaim underutilized assigned network numbers.
- There is no current need to remember major portions of the Internet.
- CIDR-style assignments of parts of unassigned Class A address space should be considered.
- "Simple Internet Protocol Plus (SIPP) Spec. (128 bit ver)" [6] be adapted as the basis for IPng.
- The documents listed in Appendix C (of [4]) be the foundation of the IPng effort.
- An IPng Working Group be formed, chaired by Steve Deering and Ross Callon.
- Robert Hinden be the document editor for the IPng effort.
- An IPng Reviewer be appointed and that Dave Clark be the reviewer.
- An Address Autoconfiguration Working Group be formed, chaired by Dave Katz and Sue Thomson.
- An IPng Transition Working Group be formed, chaired by Bob Gilligan and TBA.
- The Transition and Coexistance Including Testing Working Group be chartered.
- Recommendations about the use of non-IPv6 addresses in IPv6 environments and IPv6 addresses in non-IPv6 environments be developed.
- The IESG commission a review of all IETF standards documents for IPng implications.
- The IESG task current IETF working groups to take IPng into account.
- The IESG charter new working groups where needed to revise old standards documents.
- Informational RFCs be solicited or developed describing a few specific IPng APIs.
- The IPng Area and Area Directorate continue until main documents are offered as Proposed Standards in late 1994.
- Support for the Authentication Header be required.
- Support for a specific authentication algorithm be required.
- Support for the Privacy Header be required.
- Support for a specific privacy algorithm be required.
- An "IPng framework for firewalls" be developed.

Figure 1. Elements of IPng Area Director's Recommendation of July 1994



Ver(sion)	4-bit Internet Protocol version number = 6.
Prio(rity)	4-bit Priority value. See IPv6 Priority section.
Flow Label	24-bit field. See IPv6 Quality of Service section.
Payload Length	16-bit unsigned integer. Length of payload, i.e., the rest of the packet following the IPv6 header, in octets.
Next Hdr	8-bit selector. Identifies the type of header immediately following the IPv6 header. Uses the same values as the IPv4 Protocol field [14].
Hop Limit	8-bit unsigned integer. Decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.
Source Address	128 bits. The address of the initial sender of the packet. See [10] for details.
Destination Address	128 bits. The address of the intended recipient of the packet (possibly not the ultimate recipient, if an optional Routing Header is present).

Figure 2. IPv6 Header format

applications that ran on the previous version. Microsoft even provides the ability for Windows applications to run on the NT operating system. This is an important feature. They understand that it was very important to make sure that the applications that run on Windows also run on NT.

The same requirement is also true for IPng. The Internet has a large installed base. Features need to be designed into an IPng to make the transition as easy as possible. As with processors and operating systems, it must be backward compatible with IPv4. Other protocols have tried to replace TCP/IP, for example, XTP and OSI. One element in their failure to reach widespread acceptance was that neither had any transition strategy other than running in parallel (sometimes called dual stack). New features alone are not adequate to motivate users to deploy new protocols. IPng must have a great transition strategy and new features.

History of the IPng Effort

The IPng protocol is the result of an evolution of many different IETF proposals and working groups focused on developing an IPng, and represents over three years of effort focused on this topic. A brief history follows:

By the winter of 1992 the Internet community had developed four separate proposals for IPng. These were "CNAT", "IP Encaps", "Nimrod", and "Simple CLNP". By December 1992 three more proposals followed: "The P Internet Protocol" (PIP), "The Simple Internet Protocol" (SIP) and "TP/IX". In the spring of 1992 the "Simple CLNP" evolved into "TCP and UDP with Bigger Addresses" (TUBA) and "IP Encaps" evolved into "IP Address Encapsulation" (IPAE).

By the fall of 1993, IPAE merged with SIP while still maintaining the name SIP. This group later merged with PIP and the resulting working group called themselves "Simple Internet Protocol Plus" (SIPP). At about the same time the TP/IX Working Group changed its name to "Common Architecture for the Internet" (CATNIP).

The IPng area directors made a recommendation for an IPng in July of 1994. This recommendation, from [4], includes the elements listed in Figure 1.

IPng Overview

As mentioned previously, IPng is a new version of the Internet Protocol, designed as a successor to IP version 4 [13]. IPng is assigned IP version number 6 and is formally called IPv6 [7].

IPv6 was designed to take an evolutionary—but not a radical—step from IPv4. Functions that work in IPv4 were kept in IPv6. Functions that didn't work were removed. The changes from IPv4 to IPv6 fall primarily into the following categories:

- Expanded Routing and Addressing Capabilities. IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable

nodes, and simpler autoconfiguration of addresses. The scalability of multicast routing is improved by adding a "scope" field to multicast addresses. A new type of address called a "anycast address" is defined, to identify sets of nodes where a packet sent to an anycast address is delivered to one of the nodes. The use of anycast addresses in the IPv6 source route allows nodes to control the path along which their traffic flows.

- Header Format Simplification. Some IPv4 header fields have been dropped or made optional, to reduce the common-case processing cost of packet handling and to keep the bandwidth cost of the IPv6 header as low as possible despite the increased size of the addresses. Even though the IPv6 addresses are four times longer than the IPv4 addresses, the IPv6 header is only twice the size of the IPv4 header.

- Improved Support for Options. Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

- Quality-of-Service Capabilities. A new capability is added to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling, such as non-default quality of service or "real-time" service.

Table 1. IPv6 Extension Header

Option	Function
Routing	Extended Routing (like IPv4 loose source route).
Fragmentation	Fragmentation and Reassembly.
Authentication	Integrity and Authentication.
Security Encapsulation	Confidentiality.
Hop-by-Hop Option	Special options which require hop-by-hop processing.
Destination Options	Optional information to be examined by the destination node.

- Authentication and Privacy Capabilities. IPv6 includes the definition of extensions that provide support for authentication, data integrity, and confidentiality. This is included as a basic element of IPv6 and will be included in all implementations.

The IPv6 protocol consists of two parts, the basic IPv6 header shown in Figure 2 and the IPv6 extension headers.

IPv6 Extensions

IPv6 includes an improved option mechanism over IPv4. IPv6 options are placed in separate extension headers that are located between the IPv6 header and the transport-layer header in a packet. Most IPv6 extension headers are not examined or processed by any router along a packet's delivery path until the packet arrives at its final destination. This facilitates a major improvement in router performance for packets containing options. In IPv4 the presence of any options requires the router to examine all options. The other improvement is that, unlike IPv4 options, IPv6 extension headers can be of arbitrary length and the total amount of options carried in a packet is not limited to 40 bytes. This feature, plus the manner in which they are processed, permits IPv6 options to be used for functions that were not practical in IPv4. A good example of this is the IPv6 Authentication and Security Encapsulation options.

In order to improve the performance when handling subsequent option headers and the transport protocol that follows, IPv6 options are always an integer multiple of 8 octets long, in order to retain this alignment for subsequent headers. The IPv6 extension headers that are currently defined are shown in Table 1.

Table 2. Format Prefix Allocation

Allocation	Prefix (binary)	Fraction of Address Space
Reserved	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP Allocation	0000 001	1/128
Reserved for IPX Allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Unassigned	001	1/8
Provider-Based Unicast Address	010	1/8
Unassigned	011	1/8
Reserved for Geographic-Based Unicast Addresses	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link Local Use Addresses	1111 1110 10	1/1024
Site Local Use Addresses	1111 1110 11	1/1024
Multicast Addresses	1111 1111	1/256

IPv6 Addressing

IPv6 addresses are 128-bits long and are identifiers for individual interfaces and sets of interfaces. IPv6 addresses of all types are assigned to interfaces, not nodes. Since each interface belongs to a single node, any of that node's interfaces' unicast addresses may be used as an identifier for the node. A single interface may be assigned multiple IPv6 addresses of any type.

IPv6 addresses are 128-bit identifiers for interfaces and sets of interfaces. There are three types of addresses:

- Unicast: An identifier for a single interface. A

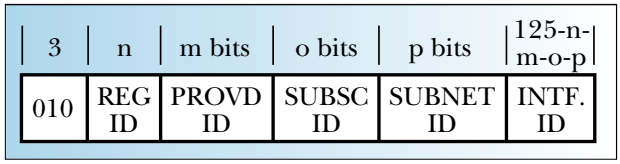


Figure 3. Assignment plan for unicast addresses

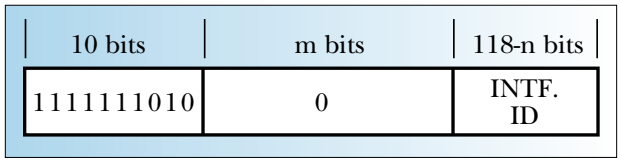


Figure 4. Link-Local address format

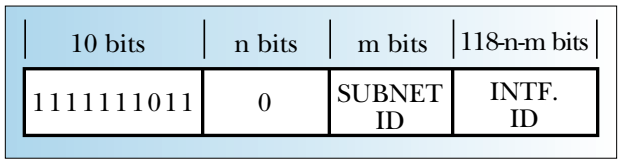


Figure 5. Site-Local address format

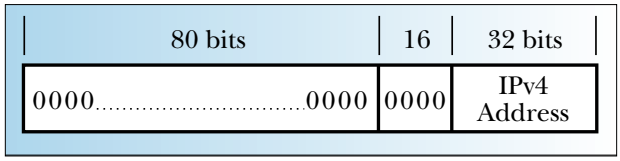


Figure 6. IPv4-compatible IPv6 address

packet sent to a unicast address is delivered to the interface identified by that address.

- Anycast: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance).
- Multicast: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address. There are no broadcast addresses in IPv6, their function being superseded by multicast addresses.

IPv6 supports addresses that are four times the number of bits as IPv4 addresses (128 vs. 32). This is 4 billion times 4 billion (2^{96}) times the size of the IPv4 address space (2^{32}). This works out to be 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses.

This is an extremely large address space. In a theoretical sense this is approximately 665,570,793,348,866,943,898,599 addresses per square meter of the surface of the planet Earth (assuming the Earth surface is 511,263,971,197,990 square meters).

In more practical terms the assignment and routing of addresses requires the creation of hierarchies that reduces the efficiency of the usage of the address space. Christian Huitema performed an analysis [11] that evaluated the efficiency of other addressing architectures (including the French telephone system, U.S. telephone systems, the current Internet using IPv4, and IEEE 802 nodes). He concluded that 128-bit IPv6 addresses could accommodate between $8x10^{17}$ to $2x10^{33}$ nodes assuming efficiency in the same ranges as the other addressing architectures. By even his most pessimistic estimate, this would provide 1,564 addresses for each square meter of the surface of the planet Earth. The optimistic estimate would allow for 3,911,873,538,269,506,102 addresses for each square meter of the surface of the planet Earth.

The specific type of IPv6 address is indicated by the leading bits in the address. The variable-length field comprising these leading bits is called the Format Prefix (FP). The initial allocation of these prefixes is shown in Table 2.

This allocation supports the direct allocation of provider addresses, local use addresses, and multicast addresses. Space is reserved for NSAP addresses, IPX addresses, and neutral-interconnect addresses. The remainder of the address space is unassigned for future use. This can be used for expansion of existing use (e.g., additional provider addresses, etc.) or new uses (e.g., separate locators and identifiers). Note that anycast addresses are not shown here because they are allocated out of the unicast address space. Approximately 15% of the address space is initially allocated. The remaining 85% is reserved for future use.

IPv6 nodes may have considerable or little knowledge of the internal structure of the IPv6 address, depending on the role the node plays (for instance, host versus router). At a minimum, a node may consider that unicast addresses (including its own) have no internal structure: a slightly sophisticated host (but still rather simple) may additionally be aware of subnet prefix(es) for the link(s) it is attached to, where different addresses may have different values for n. Still more sophisticated hosts may be aware of other hierarchical boundaries in the unicast address. Though a very simple router may have no knowledge of the internal structure of IPv6 unicast addresses, routers will more generally have knowledge of one or more of the hierarchical boundaries for the operation of routing protocols. The known boundaries will differ from router to router, depending on what positions the router holds in the routing hierarchy.

Unicast Addresses

There are several forms of unicast address assignment in IPv6. These are the global provider-based unicast address, the geographic-based unicast address, the NSAP address, the IPX hierarchical address, the site-local-use address, the link-local-use address, and the IPv4-capable host address. Additional address types can be defined in the future.

Provider-based Unicast Addresses

Provider-based unicast addresses are used for global

communication. They are similar in function to IPv4 addresses under CIDR. The assignment plan for unicast addresses is described in [15] and [16]. Their format is shown in Figure 3.

The first 3 bits identify the address as a provider-oriented unicast address. The next field (REGISTRY ID) identifies the Internet address registry that assigns provider identifiers (PROVIDER ID) to Internet service providers, which then assign portions of the address space to subscribers. This usage is similar to assignment of IP addresses under CIDR. The SUBSCRIBER ID distinguishes among multiple subscribers attached to the Internet service provider identified by the PROVIDER ID. The SUBNET ID identifies a specific physical link. There can be multiple subnets on the same physical link. A specific subnet can not span multiple physical links. The INTERFACE ID identifies a single interface among the group of interfaces identified by the subnet prefix.

Local-Use Addresses

A local-use address is a unicast address that has only local routability scope (within the subnet or within a subscriber network), and may have local or global uniqueness scope. They are intended for use inside of a site for "plug and play" local communication and for bootstrapping up to the use of global addresses [18].

There are two types of local-use unicast addresses defined. These are Link-Local and Site-Local. The Link-Local is for use on a single link and the Site-Local is for use in a single site. Link-Local addresses have the format shown in Figure 4. Link-Local addresses are designed to be used for addressing on a single link for purposes such as auto-address configuration, neighbor discovery, routing protocol neighbor adjacencies, or when no routers are present. Site-Local addresses have the format shown in Figure 5.

For both types of local use addresses the INTERFACE ID is an identifier that must be unique in the domain in which it is being used. In most cases these will use a node's IEEE-802 48-bit MAC address. The SUBNET ID identifies a specific subnet in a site. The combination of the SUBNET ID and the INTERFACE ID to form a local use address allows a large private internet to be constructed without any other address allocation.

Local-use addresses allow organizations that are not yet connected to the global Internet to operate without the need to request an address prefix from the global Internet address space. Local-use addresses can be used instead. If the organization later connects to the global Internet, it can use its SUBNET ID and INTERFACE ID in combination with a global prefix (e.g., REGISTRY ID + PROVIDER ID + SUBSCRIBER ID) to create a global address. This is a significant improvement over IPv4, which requires sites that use a private (non-global) IPv4 address to manually renumber when they connect to the Internet. IPv6 does the renumbering automatically [18].

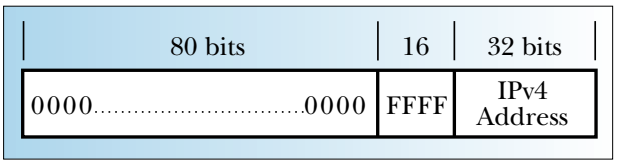


Figure 7. IPv4-mapped IPv6 address

IPv6 Addresses with Embedded IPV4 Addresses

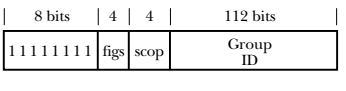
The IPv6 transition mechanisms include a technique for hosts and routers to dynamically tunnel IPv6 packets over IPv4 routing infrastructure. IPv6 nodes that utilize this technique are assigned special IPv6 unicast addresses that carry an IPv4 address in the low-order 32-bits. This type of address is termed an "IPv4-compatible IPv6 address" and has the format shown in Figure 6.

A second type of IPv6 address that holds an embedded IPv4 address is also defined. This address is used to represent the addresses of IPv4-only nodes (those that *do not* support IPv6) as IPv6 addresses. This type of address is termed an "IPv4-mapped IPv6 address" and has the format depicted in Figure 7.

Anycast Addresses

An IPv6 anycast address is an address that is assigned to more than one interface (typically belonging to different nodes), with the property that a packet sent to an anycast address is routed to the "nearest" interface having that address, according to the routing protocol's measure of distance.

Anycast addresses, when used as part of a route sequence, permit a node to select which of several Internet service providers it wants to carry its traffic. This capability is sometimes called "source selected policies". This would be implemented by configuring anycast addresses to identify the set of routers belonging to Internet service providers (e.g., one anycast address per Internet service provider). These anycast



11111111 at the start of the address identifies the address as being a multicast address.

FLGS is a set of 4 flags: | 0 | 0 | 0 | T |

The high-order 3 flags are reserved, and must be initialized to 0.

T=0 indicates a permanently assigned ("well-known") multicast address, assigned by the global Internet numbering authority.

T=1 indicates a non-permanently assigned ("transient") multicast address.

SCOP is a 4-bit multicast scope value used to limit the scope of the multicast group. The values are:

0	Reserved	8	Organization-local scope
1	Node-local scope	9	(unassigned)
2	Link-local scope	A	(unassigned)
3	(unassigned)	B	(unassigned)
4	(unassigned)	C	(unassigned)
5	Site-local scope	D	(unassigned)
6	(unassigned)	E	Global scope
7	(unassigned)	F	Reserved

GROUP ID identifies the multicast group, either permanent or transient, within the given scope.

Figure 8. Multicast address format

addresses can be used as intermediate addresses in an IPv6 routing header, to cause a packet to be delivered via a particular provider or sequence of providers. Other possible uses of anycast addresses are to identify the set of routers attached to a particular subnet, or the set of routers providing entry into a particular routing domain.

Anycast addresses are allocated from the unicast address space, using any of the defined unicast address formats. Thus, anycast addresses are syntactically indistinguishable from unicast addresses. When a unicast address is assigned to more than one interface, thus turning it into an anycast address, the nodes to which the address is assigned must be explicitly configured to know that it is an anycast address.

Multicast Addresses

A IPv6 multicast address is an identifier for a group of interfaces. An interface may belong to any number of multicast groups. Multicast addresses have the format illustrated and described in Figure 8.

IPv6 Routing

Routing in IPv6 is almost identical to IPv4 routing under CIDR except that the addresses are 128-bit IPv6 addresses instead of 32-bit IPv4 addresses. With very straightforward extensions, all of IPv4's routing algorithms (OSPF, RIP, IDRP, ISIS, etc.) can used to route IPv6.

IPv6 also includes simple routing extensions that support powerful new routing functionality. These capabilities include:

- Provider Selection (based on policy, performance, cost, etc.)
- Host Mobility (route to current location)
- Auto-Readdressing (route to new address)

The new routing functionality is obtained by creating sequences of IPv6 addresses using the IPv6 Routing option. The routing option is used by a IPv6 source to list one or more intermediate nodes (or topological group) to be "visited" on the way to a packet's destination. This function is very similar in function to IPv4's Loose Source and Record Route option.

In order to make address sequences a general function, IPv6 hosts are required in most cases to reverse routes in a packet it receives (if the packet was successfully authenticated using the IPv6 Authentication Header) containing address sequences in order to return the packet to its originator. This approach is taken to make IPv6 host implementations from the start support the handling and reversal of source routes. This is the key for allowing them to work with hosts that implement the new features such as provider selection or extended addresses. Figure 9 presents examples showing how the address sequences can be used.

IPv6 Quality-of-Service Capabilities

The Flow Label and the Priority fields in the IPv6 header may be used by a host to identify those packets for which it requests special handling by IPv6 routers, such as non-default quality of service or "real-time" service. This capability is important in order to support applications that require some degree of consistent throughput, delay, and/or jitter. These types of applications are commonly described as "multimedia" or "real-time" applications.

Flow Labels

The 24-bit Flow Label field in the IPv6 header may be used by a source to label those packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. This aspect of IPv6 is, at the time of writing, still experimental and subject to change as the requirements for flow support in the Internet become clearer. Hosts or routers that do not support the functions of the Flow Label field are required to set the field to zero when originating a packet, pass the field unchanged when forwarding a packet, and ignore the field when receiving a packet.

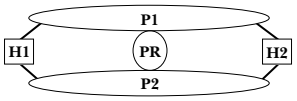
A flow is a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers. The nature of that special handling might be conveyed to the routers by a control protocol, such as a resource reservation protocol, or by information within the flow's packets themselves (e.g., in a hop-by-hop option).

Three examples show how the address sequences can be used. In these examples, address sequences are shown by a list of individual addresses separated by commas. For example:

SRC, 11, 12, 13, DST

Where the first address is the source address, the last address is the destination address, and the middle addresses are intermediate addresses.

For these examples assume that two hosts, H1 and H2 wish to communicate. Assume that H1 and H2's sites are both connected to providers P1 and P2. A third wireless provider, PR, is connected to both providers P1 and P2.



The simplest case (no use of address sequences) is when H1 wants to send a packet to H2 containing the addresses:

H1, H2

When H2 replied it would reverse the addresses and construct a packet containing the addresses:

H2, H1

In this example either provider could be used, and H1 and H2 would not be able to select which provider traffic would be sent to and received them.

If H1 decides that it wants to enforce a policy that all communication to/from H2 can only use provider P1, it would construct a packet containing the address sequence:

H1, P1, H2

This ensures that when H2 replies to H1, it will reverse the route and the reply it would also travel over P1. The addresses in H2's reply would look like:

H2, P1, H1

If H1 became mobile and moved to provider PR, it could maintain (not breaking any tranport connections) communication with H2 by sending packets that contain the address sequence:

H1, PR, P1, H2

This would ensure that when H2 replied it would enforce H1's policy of exclusive use of provider P1 and send the packet to H1 new location on provider PR. The reversed address sequence would be:

H2, P1, PR, H1

The address sequence facility of IPv6 can be used for provider selection, mobility, and readdressing. It is a simple but powerful capability.

Figure 9. Example use of address sequences

There may be multiple active flows from a source to a destination, as well as traffic that is not associated with any flow. A flow is uniquely identified by the combination of a source address and a non-zero flow label. Packets that do not belong to a flow carry a flow label of zero.

Table 3. Recommended Priority Values for Application Categories

Priority Value	Application Category
0	Uncharacterized traffic
1	"Filler" traffic (e.g., netnews)
2	Unattended data transfer (e.g., email)
3	(Reserved)
4	Attended bulk transfer (e.g., FTP, HTTP, NFS)
5	(Reserved)
6	Interactive traffic (e.g., telnet, X)
7	Internet control traffic (e.g., routing protocols, SNMP)

A flow label is assigned to a flow by the flow's source node. New flow labels must be chosen (pseudo-)randomly and uniformly from the range 1 to FFFFFF hex. The purpose of the random allocation is to make any set of bits within the Flow Label field suitable for use as a hash key by routers, for looking up the state associated with the flow.

All packets belonging to the same flow must be sent with the same source address, destination address, priority, and flow label. If any of those packets includes a Hop-by-Hop Options header, then they all must be originated with the same Hop-by-Hop Options header contents (excluding the Next Header field of the Hop-by-Hop Options header). If any of those packets includes a Routing header, then they all must be originated with the same contents in all extension headers up to and including the Routing header (excluding the Next Header field in the Routing header). The routers or destinations are permitted, but not required, to verify that these conditions are satisfied. If a violation is detected, it should be reported to the source by an ICMP Parameter Problem message, Code 0, pointing to the high-order octet of the Flow Label field (i.e., offset 1 within the IPv6 packet) [5].

Routers are free to "opportunistically" set up flow-handling state for any flow, even when no explicit flow establishment information has been provided to them via a control protocol, a hop-by-hop option, or other means. For example, upon receiving a packet from a particular source with an unknown, non-zero flow label, a router may process its IPv6 header and any necessary extension headers as if the flow label were zero. That processing would include determining the next-hop interface, and possibly other actions, such as updating a hop-by-hop option, advancing the pointer and addresses in a Routing header, or deciding on how to queue the packet based on its Priority field. The router may then choose to "remember" the results of those processing steps and cache that information, using the source address plus the flow label as the cache key. Subsequent packets with the same source address and flow label may then be handled by referring to the cached information rather than examining all those fields that, according to the requirements of the previous paragraph, can be assumed unchanged from the first packet seen in the flow.

Priority

The 4-bit Priority field in the IPv6 header enables a source to identify the desired delivery priority of its packets, relative to other packets from the same source. The Priority values are divided into two ranges: values 0 through 7 are used to specify the priority of traffic for which the source is providing congestion control (i.e., traffic that "backs off" in response to congestion, such as TCP traffic). Values 8 through 15 are used to specify the priority of traffic that does not back off in response to congestion, e.g., "real-time" packets being sent at a constant rate. For congestion-controlled traffic, the Priority values are recommended for particular application categories as shown in Table 3.

For non-congestion-controlled traffic, the lowest Priority value (8) should be used for those packets that the sender is most willing to have discarded under conditions of congestion (e.g., high-fidelity video traffic), and the highest value (15) should be used for those packets that the sender is least willing to have discarded (e.g., low-fidelity audio traffic). There is no relative ordering implied between the congestion-controlled priorities and the non-congestion-controlled priorities.

IPv6 Security

The current Internet has a number of security problems and lacks effective privacy and authentication mechanisms below the application layer. IPv6 remedies these shortcomings by having two integrated options that provide security services [1]. These two options may be used singly or together to provide differing levels of security to different users. This is very important because different user communities have different security needs.

The first mechanism, called the IPv6 Authentication Header, is an extension header that provides authentication and integrity (without confidentiality) to IPv6 datagrams [2]. While the extension is algorithm-independent and will support many different authentication techniques, the use of keyed MD5 is required to help ensure interoperability within the worldwide Internet. This can be used to eliminate a significant class of network attacks, including host masquerading attacks. The use of the IPv6 Authentication Header is particularly important when source routing is used with IPv6 because of the known risks in IP source routing. Its placement at the Internet layer can help provide host origin authentication to those upper layer protocols and services that currently lack meaningful protections. This mechanism

should be exportable by vendors in the U.S. and other countries with similar export restrictions because it only provides authentication and integrity, and specifically does not provide confidentiality. The exportability of the IPv6 Authentication Header encourages its widespread deployment and use.

The second security extension header provided with IPv6 is the IPv6 Encapsulating Security Header [3]. This mechanism provides integrity and confidentiality to IPv6 datagrams. It is simpler than some similar security protocols (e.g., SP3D, ISO NLSP) but remains flexible and algorithm-independent. To achieve interoperability within the global Internet, the use of DES CBC is being used as the standard algorithm for use with the IPv6 Encapsulating Security Header.

IPv6 Transition Mechanisms

The key transition objective is to allow IPv6 and IPv4 hosts to interoperate. A second objective is to allow IPv6 hosts and routers to be deployed in the Internet in a highly diffuse and incremental fashion, with few interdependencies. A third objective is that the transition should be as easy as possible for end users, system administrators, and network operators to understand and carry out.

The IPv6 transition mechanisms are a set of protocol mechanisms implemented in hosts and routers, along with some operational guidelines for addressing and deployment, designed to make transition from the Internet to IPv6 work with as little disruption as possible [9].

The IPv6 transition mechanisms provide a number of features, including:

- Incremental upgrade and deployment. Individual IPv4 hosts and routers may be upgraded to IPv6 one at a time without requiring any other hosts or routers to be upgraded at the same time. New IPv6 hosts and routers can be installed one by one.
- Minimal upgrade dependencies. The only prerequisite to upgrading hosts to IPv6 is that the DNS server must first be upgraded to handle IPv6 address records. There are no prerequisites to upgrading routers.
- Easy Addressing. When existing installed IPv4 hosts or routers are upgraded to IPv6, they may continue to use their existing address. They do not need to be assigned new addresses. Administrators do not need to draft new addressing plans.
- Low start-up costs. Little or no preparation work is needed in order to upgrade existing IPv4 systems to IPv6, or to deploy new IPv6 systems.

The mechanisms employed by the IPv6 transition mechanisms include:

- An IPv6 addressing structure that embeds IPv4 addresses within IPv6 addresses, and encodes other information used by the transition mechanisms.
- A model of deployment where all hosts and routers upgraded to IPv6 in the early transition phase are "dual" capable (i.e., implement complete IPv4 and IPv6 protocol stacks).
- The technique of encapsulating IPv6 packets within IPv4 headers to carry them over segments of the end-to-end path where the routers have not yet been upgraded to IPv6.
- The header translation technique to allow the eventual introduction of routing topologies that route only IPv6 traffic, and the deployment of hosts that support only IPv6. Use of this technique is optional, and would be used in the later phase of transition if it is used at all.

The IPv6 transition mechanisms ensures that IPv6 hosts can interoperate with IPv4 hosts anywhere in the Internet up until the time when IPv4 addresses run out, and allows IPv6 and IPv4 hosts within a limited scope to interoperate indefinitely after that. This feature protects the huge investment users have made in IPv4 and ensures that IPv6 does not render IPv4 obsolete. Hosts that need only a limited connectivity range (e.g., printers) need never be upgraded to IPv6. The incremental upgrade features of the IPv6 transition mechanisms allow the host and router vendors to integrate IPv6 into their product lines at their own pace, and allows the end-users and network operators to deploy IPv6 on their own schedules.

Current Status

The base IPv6 protocols have been approved by the Internet Engineering Steering Group of the IETF and are now IETF standards. These include:

- "Internet Protocol, Version 6 (IPv6) Specification" [7]
- "IP Version 6 Addressing Architecture" [10]
- "ICMP for the Internet Protocol Version 6 (IPv6)" [5]
- "IP Security Architecture" [1]
- "IP Authentication Header" [2]
- "IP Encapsulating Security Payload (ESP)" [3]
- "Transition Mechanisms for IPv6 Hosts and Routers" [9]
- "DNS Extensions to support IP version 6" [17]

Work is being completed on Neighbor Discovery, Address Auto-configuration, OSI NSAP Mappings, Routing Protocols, and specifications for how to run IPv6 over different types of subnetworks (e.g., Ethernet, FDDI, etc.).

A number of implementations of IPv6 are being developed by a variety of organizations. This includes both public domain and commercial implementations. As of April 1996 implementations include:

- Bay Networks
- Cisco Systems
- Digital Equipment Corporation
- FTP Software
- INRIA Rocquencourt
- Ipsilon Networks
- Mentat
- Naval Research Laboratories

- Pacific Softworks, Inc.
- Penril
- Siemens Nixdorf
- Sun Microsystems
- Swedish Institute for Computer Science (SICS)
- Telebit AS
- WIDE

Overall, IPv6 implementations are being done for all volume hardware and software platforms.

Why IPv6?

There are a number of reasons why IPv6 is appropriate for the next generation of the Internet Protocol. It solves the Internet scaling problem, provides a flexible transition mechanism for the current Internet, and was designed to meet the needs of new markets such as nomadic personal computing devices, networked entertainment, and device control. It does this in an evolutionary way that reduces the risk of architectural problems.

Ease of transition is a key point in the design of IPv6. It is not a capability that was added in at the end. IPv6 is designed to interoperate with IPv4. Specific mechanisms (embedded IPv4 addresses, pseudo-checksum rules etc.) were built into IPv6 to support transition and compatibility with IPv4. It was designed to permit a gradual and piecemeal deployment with a minimum of dependencies.

IPv6 supports large hierarchical addresses, which will allow the Internet to continue to grow and provide new routing capabilities not built into IPv4. It has anycast addresses that can be used for policy route selection and has scoped multicast addresses that provide improved scalability over IPv4 multicast. It also has local use address mechanisms that provide the ability for "plug and play" installation.

The address structure of IPv6 was also designed to support carrying the addresses of other Internet protocol suites. Space was allocated in the addressing plan for IPX and NSAP addresses. This was done to facilitate migration of these Internet protocols to IPv6.

IPv6 provides a platform for new Internet functionality. This includes support for real-time flows, provider selection, host mobility, end-to-end security, auto-configuration, and auto-reconfiguration. These new functions are also necessary to handle future growth.

In summary, IPng is a new version of IP designed as an evolutionary step from IPv4. It builds upon the architecture that made IPv4 successful in the Internet. It is designed to solve the growth problems the Internet is encountering including the direct issues of addressing and routing, as well as dealing with long-term growth issues such as security, auto-configuration, real-time services, and transition.

Where to Get Additional Information

A set of World-Wide Web pages is available describing IPng at http://playground.sun.com/ pub/ipng

These Web pages contain pointers to current specifications and implementations and will be updated on a regular basis. The documentation listed in the reference section of this article can be found in one of the IETF RFC and Internet Draft directories.

Acknowledgments

This article has described the work of IETF IPng, Addr Conf, and NG Trans working groups. One individual who deserves particular mention is Stephen E. Deering of Xerox PARC. Steve has provided the vision and direction that has made IPv6 possible. He is, simply stated, the architect of IPv6. A number of other people have also made important contributions to IPng. These include Ran Atkinson, Scott Bradner, Jim Bound, Ross Callon, Dave Crocker, Bill Fink, Paul Francis, Bob Gilligan, Ramesh Govindan, Christian Huitema, Allison Mankin, Thomas Nartin, Erik Nordmark, Dave Katz, Tony Li, Yakov Rekhter, Bill Simpson, and Sue Thompson.

References

1. Atkinson, R. *IP Security Architecture.* RFC-1825, August 1995.
2. Atkinson, R. *IP Authentication Header.* RFC-1826, August 1995.
3. Atkinson, R. *IP Encapsulating Security Payload (ESP).* RFC-1827, August 1995.
4. Bradner, B. and Mankin, A. *The Recommendation for the IP Next Generation Protocol.* RFC 1752, January 1995.
5. Conta, A., and Deering, S. *ICMP for the Internet Protocol Version 6 (IPv6).* RFC-1885, December 1995.
6. Deering, S. *Simple Internet Protocol Plus (SIPP) Specification (128-bit address version).* Internet Draft, July 1994.
7. Deering, S., and Hinden, R. *Internet Protocol, Version 6 (IPv6) Specification.* RFC-1883, December 1995.
8. Fuller, V., et al. *Supernetting: An Address Assignment and Aggregation Strategy.* RFC 1338, June 1992.
9. Gilligan, R., and Nordmark, E. *Transition Mechanisms for IPv6 Hosts and Routers.* RFC-1933, April 1996.
10. Hinden, R., and Deering, S., Eds., *IP Version 6 Addressing Architecture.* RFC-1884, December 1995.
11. Huitema, C. *The H Ratio for Address Assignment Efficiency.* RFC-1715, November 1994.
12. Nartin, T., Nordmark, E., and Simpson, W. *Neighbor Discovery for IP Version 6 (IPv6).* Internet Draft, February 1996.
13. Postel, J. *Internet Protocol.* RFC-791, September, 1981.
14. Postel, J. *Assigned Numbers.* RFC-1700, October 1994.
15. Rekhter, Y., and Li, T. *An Architecture for IPv6 Unicast Address Allocation.* RFC-1887, December 1995.
16. Rekhter, Y., Lothberg, P., Hinden, R., Deering, S., and Postel, J. *An IPv6 Global Unicast Address Format.* Internet Draft, February 1996.
17. Thompson, S., and Huitema, C. *DNS Extensions to support IP version 6.* RFC-1886, December 1995.
18. Thomson, S. *IPv6 Address Autoconfiguration.* Internet Draft, December 1995.

About the Author:

ROBERT M. HINDEN is Director of Software at Ipsilon Networks, Inc., and is currently the co-chair and document editor for the IPng working group and a member of the IPng Directorate. **Author's Present Address:** Ipsilon Networks, Inc., 2191 E. Bayshore Rd., Suite 100, Palo Alto, CA 94303. email: hinden@ipsilon.com