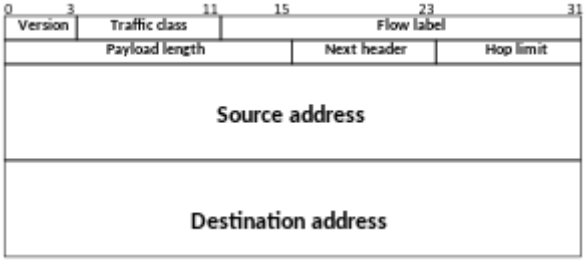
IPv6

Internet Protocol version 6 (**IPv6**) is the most recent version of the Internet Protocol (IP), the communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet. IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion. IPv6 is intended to replace IPv4.[1] In December 1998, IPv6 became a Draft Standard for the IETF,[2] who subsequently ratified it as an Internet Standard on 14 July 2017.[3][4]

Internet Protocol Version 6

Communication protocol



IPv6 header

Purpose	Internetworking protocol
Developer(s)	Internet Engineering Task Force
Introduced	December 1995
Based on	IPv4
OSI layer	Network layer
RFC(s)	RFC 2460, RFC 8200

Devices on the Internet are assigned a unique IP address for identification and location definition. With the rapid growth of the Internet after commercialization in the 1990s, it became evident that far more addresses would be needed to connect devices than the IPv4 address space had available. By 1998, the IETF had formalized the successor protocol. IPv6 uses a 128-bit address, theoretically allowing 2^{128}, or approximately 3.4×10^{38} addresses. The actual number is slightly smaller, as multiple ranges are reserved for special use or completely excluded from use. The two protocols are not designed to be interoperable, and thus direct communication between them is impossible, complicating the move to IPv6. However, several transition mechanisms have been devised to rectify this.

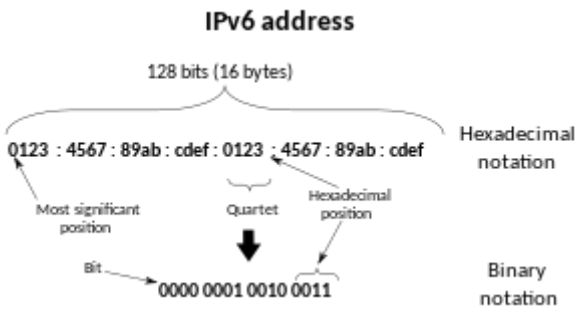
IPv6 provides other technical benefits in addition to a larger addressing space. In particular, it permits hierarchical address allocation methods that facilitate route aggregation across the Internet, and thus limit the expansion of routing tables. The use of multicast addressing is expanded and simplified, and provides additional optimization for the delivery of services. Device mobility, security, and configuration aspects have been considered in the design of the protocol.

IPv6 addresses are represented as eight groups, separated by colons, of four hexadecimal digits. The full representation may be shortened; for example, *2001:0db8:0000:0000:0000:8a2e:0370:7334* becomes *2001:db8::8a2e:370:7334*.

Contents

Main features

IPv6 is an Internet Layer protocol for packet-switched internetworking and provides end-to-end datagram transmission across multiple IP networks, closely adhering to the design principles developed in the previous version of the protocol, Internet Protocol Version 4 (IPv4).



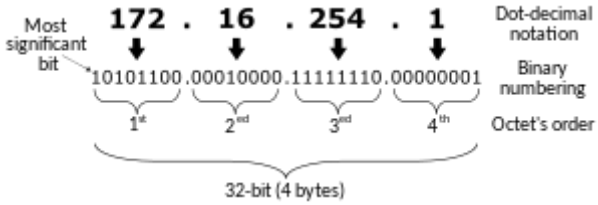
Glossary of terms used for IPv6 addresses

In addition to offering more addresses, IPv6 also implements features not present in IPv4. It simplifies aspects of address configuration, network renumbering, and router announcements when changing network connectivity providers. It simplifies processing of packets in routers by placing the responsibility for packet fragmentation into the end points. The IPv6 subnet size is standardized by fixing the size of the host identifier portion of an address to 64 bits.

The addressing architecture of IPv6 is defined in RFC 4291 (https://datatracker.ietf.org/doc/html/rfc4291) and allows three different types of transmission: unicast, anycast and multicast.[5]:210

Motivation and origin

IPv4 address exhaustion



Decomposition of the dot-decimal IPv4 address representation to its binary value

Internet Protocol Version 4 (IPv4) was the first publicly used version of the Internet Protocol. IPv4 was developed as a research project by the Defense Advanced Research Projects Agency (DARPA), a United States Department of Defense agency, before becoming the foundation for the Internet and the World Wide Web. IPv4 includes an addressing system that uses numerical identifiers consisting of 32 bits. These addresses are typically displayed in dot-decimal notation as decimal values of four octets, each in the range 0 to 255, or 8 bits per number. Thus, IPv4 provides an addressing capability of 2^{32} or approximately 4.3 billion addresses. Address exhaustion was not initially a concern in IPv4 as this version was originally presumed to be a test of DARPA's networking concepts.[6] During the first decade of operation of the Internet, it became apparent that methods had to be developed to conserve address space. In the early 1990s, even after the redesign of the addressing system using a classless network model, it became clear that this would not suffice to prevent IPv4 address exhaustion, and that further changes to the Internet infrastructure were needed.[7]

The last unassigned top-level address blocks of 16 million IPv4 addresses were allocated in February 2011 by the Internet Assigned Numbers Authority (IANA) to the five regional Internet registries (RIRs). However, each RIR still has available address pools and is expected to continue with standard address allocation policies until one /8 Classless Inter-Domain Routing (CIDR) block remains. After that, only blocks of 1,024 addresses (/22) will be provided from the RIRs to a local Internet registry (LIR). As of September 2015, all of Asia-Pacific Network Information Centre (APNIC), the Réseaux IP Européens Network Coordination Centre (RIPE_NCC), Latin America and Caribbean Network Information Centre (LACNIC), and American Registry for Internet Numbers (ARIN) have reached this stage.[8][9][10] This leaves African Network Information Center (AFRINIC) as the sole regional internet registry that is still using the normal protocol for distributing IPv4 addresses. As of November 2018, AFRINIC's minimum allocation is /22 or 1024 IPv4 addresses. A LIR may receive additional allocation when about 80% of all the address space has been utilized.[11]

RIPE NCC announced that it had fully run out of IPv4 addresses on 25 November 2019,[12] and called for greater progress on the adoption of IPv6.

It is widely expected that the Internet will use IPv4 alongside IPv6 for the foreseeable future.

Comparison with IPv4

On the Internet, data is transmitted in the form of network packets. IPv6 specifies a new packet format, designed to minimize packet header processing by routers.[2][13] Because the headers of IPv4 packets and IPv6 packets are significantly different, the two protocols are not interoperable. However, most transport and application-layer protocols need little or no change to operate over IPv6; exceptions are application protocols that embed Internet-layer addresses, such as File Transfer Protocol (FTP) and Network Time Protocol (NTP), where the new address format may cause conflicts with existing protocol syntax.

Larger address space

The main advantage of IPv6 over IPv4 is its larger address space. The size of an IPv6 address is 128 bits, compared to 32 bits in IPv4.[2] The address space therefore has 2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses (approximately 3.4×10^{38}). Some blocks of this space and some specific addresses are reserved for special uses.

While this address space is very large, it was not the intent of the designers of IPv6 to assure geographical saturation with usable addresses. Rather, the longer addresses simplify allocation of addresses, enable efficient route aggregation, and allow implementation of special addressing features. In IPv4, complex Classless Inter-Domain Routing (CIDR) methods were developed to make the best use of the small address space. The standard size of a subnet in IPv6 is 2^{64} addresses, about four billion times the size of the entire IPv4 address space. Thus, actual address space utilization will be small in IPv6, but network management and routing efficiency are improved by the large subnet space and hierarchical route aggregation.

Multicasting

Multicast structure in IPv6

Multicasting, the transmission of a packet to multiple destinations in a single send operation, is part of the base specification in IPv6. In IPv4 this is an optional (although commonly implemented) feature.[14] IPv6 multicast addressing has features and protocols in common with IPv4 multicast, but also provides changes and improvements by eliminating the need for certain protocols. IPv6 does not implement traditional IP broadcast, i.e. the transmission of a packet to all hosts on the attached link using a special *broadcast address*, and therefore does not define broadcast addresses. In IPv6, the same result is achieved by sending a packet to the link-local *all nodes* multicast group at address ff02::1, which is analogous to IPv4 multicasting to address 224.0.0.1. IPv6 also provides for new multicast implementations, including embedding rendezvous point addresses in an IPv6 multicast group address, which simplifies the deployment of inter-domain solutions.[15]

In IPv4 it is very difficult for an organization to get even one globally routable multicast group assignment, and the implementation of inter-domain solutions is arcane.[16] Unicast address assignments by a local Internet registry for IPv6 have at least a 64-bit routing prefix, yielding the smallest subnet size available in IPv6 (also 64 bits). With such an assignment it is possible to embed the unicast address prefix into the IPv6 multicast address format, while still providing a 32-bit block, the least significant bits of the address, or approximately 4.2 billion multicast group identifiers. Thus each user of an IPv6 subnet automatically has available a set of globally routable source-specific multicast groups for multicast applications.[17]

Stateless address autoconfiguration (SLAAC)

IPv6 hosts configure themselves automatically. Every interface has a self-generated link-local address and, when connected to a network, conflict resolution is performed and routers provide network prefixes via router advertisements.[18] Stateless configuration of routers can be achieved with a special router renumbering protocol.[19] When necessary, hosts may configure additional stateful addresses via Dynamic Host Configuration Protocol version 6 (DHCPv6) or static addresses manually.

Like IPv4, IPv6 supports globally unique IP addresses. The design of IPv6 intended to re-emphasize the end-to-end principle of network design that was originally conceived during the establishment of the early Internet by rendering network address translation obsolete. Therefore, every device on the network is globally addressable directly from any other device.

A stable, unique, globally addressable IP address would facilitate tracking a device across networks. Therefore, such addresses are a particular privacy concern for mobile devices, such as laptops and cell phones.[20] To address these privacy concerns, the SLAAC protocol includes what are typically called "privacy addresses" or, more correctly, "temporary addresses", codified in RFC 4941, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6".[21] Temporary addresses are random and unstable. A typical consumer device generates a new temporary address daily and will ignore traffic addressed to an old address after one week. Temporary addresses are used by default by Windows since XP SP1,[22] macOS since (Mac OS X) 10.7, Android since 4.0, and iOS since version 4.3. Use of temporary addresses by Linux distributions varies.[23]

Renumbering an existing network for a new connectivity provider with different routing prefixes is a major effort with IPv4.[24][25] With IPv6, however, changing the prefix announced by a few routers can in principle renumber an entire network, since the host identifiers (the least-significant 64 bits of an address) can be independently self-configured by a host.[18]

The SLAAC address generation method is implementation-dependent. IETF recommends that addresses are deterministic but semantically opaque.[26]

IPsec

Internet Protocol Security (IPsec) was originally developed for IPv6, but found widespread deployment first in IPv4, for which it was re-engineered. IPsec was a mandatory part of all IPv6 protocol implementations,[2] and Internet Key Exchange (IKE) was recommended, but with RFC 6434 the inclusion of IPsec in IPv6 implementations was downgraded to a recommendation because it was considered impractical to require full IPsec implementation for all types of devices that may use IPv6. However, as of RFC 4301 IPv6 protocol implementations that do implement IPsec need to implement IKEv2 and need to support a minimum set of cryptographic algorithms. This requirement will help to make IPsec implementations more interoperable between devices from different vendors. The IPsec Authentication Header (AH) and the Encapsulating Security Payload header (ESP) are implemented as IPv6 extension headers.[27]

Simplified processing by routers

The packet header in IPv6 is simpler than the IPv4 header. Many rarely used fields have been moved to optional header extensions.[28] With the simplified IPv6 packet header the process of packet forwarding by routers has been simplified. Although IPv6 packet headers are at least twice the size of IPv4 packet headers, processing of packets that only contain the base IPv6 header by routers may, in some cases, be more efficient, because less processing is required in routers due to the headers being aligned to match common word sizes.[2][13] However, many devices implement IPv6 support in software (as opposed to hardware), thus resulting in very bad packet processing performance.[29] Additionally, for many implementations, the use of Extension Headers causes packets to be processed by a router's CPU, leading to poor performance or even security issues.[30]

Moreover, an IPv6 header does not include a checksum. The IPv4 header checksum is calculated for the IPv4 header, and has to be recalculated by routers every time the time to live (called hop limit in the IPv6 protocol) is reduced by one. The absence of a checksum in the IPv6 header furthers the end-to-end principle of Internet design, which envisioned that most processing in the network occurs in the leaf nodes. Integrity protection for the data that is encapsulated in the IPv6 packet is assumed to be assured by both the link layer or error detection in higher-layer protocols, namely the Transmission Control Protocol (TCP) and

the User Datagram Protocol (UDP) on the transport layer. Thus, while IPv4 allowed UDP datagram headers to have no checksum (indicated by 0 in the header field), IPv6 requires a checksum in UDP headers.

IPv6 routers do not perform IP fragmentation. IPv6 hosts are required either to perform path MTU discovery, perform end-to-end fragmentation, or send packets no larger than the default maximum transmission unit (MTU), which is 1280 octets.

Mobility

Unlike mobile IPv4, mobile IPv6 avoids triangular routing and is therefore as efficient as native IPv6. IPv6 routers may also allow entire subnets to move to a new router connection point without renumbering.[31]

Extension headers

The IPv6 packet header has a minimum size of 40 octets (320 bits). Options are implemented as extensions. This provides the opportunity to extend the protocol in the future without affecting the core packet structure.[2] However, RFC 7872 notes that some network operators drop IPv6 packets with extension headers when they traverse transit autonomous systems.

Several examples of IPv6 extension headers.

Jumbograms

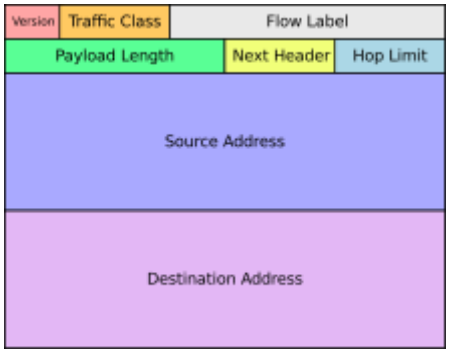
IPv4 limits packets to 65,535 ($2^{16}-1$) octets of payload. An IPv6 node can optionally handle packets over this limit, referred to as jumbograms, which can be as large as 4,294,967,295 ($2^{32}-1$) octets. The use of jumbograms may improve performance over high-MTU links. The use of jumbograms is indicated by the Jumbo Payload Option extension header.[32]

IPv6 packets

An IPv6 packet has two parts: a header and payload.

The header consists of a fixed portion with minimal functionality required for all packets and may be followed by optional extensions to implement special features.

The fixed header occupies the first 40 octets (320 bits) of the IPv6 packet. It contains the source and destination addresses, traffic class, hop count, and the type of the optional extension or payload which follows the header. This *Next Header* field tells the receiver how to interpret the data which follows the header. If the packet contains options, this field contains the option type of the next option. The "Next Header" field of the last option points to the upper-layer protocol that is carried in the packet's payload.



IPv6 packet header

The current use of the IPv6 Traffic Class field divides this between a 6 bit Differentiated Services Code Point[33] and a 2-bit Explicit Congestion Notification field.[34]

Extension headers carry options that are used for special treatment of a packet in the network, e.g., for routing, fragmentation, and for security using the IPsec framework.

Without special options, a payload must be less than 64 kB. With a Jumbo Payload option (in a *Hop-By-Hop Options* extension header), the payload must be less than 4 GB.

Unlike with IPv4, routers never fragment a packet. Hosts are expected to use Path MTU Discovery to make their packets small enough to reach the destination without needing to be fragmented. See IPv6 packet fragmentation.

Addressing

IPv6 addresses have 128 bits. The design of the IPv6 address space implements a different design philosophy than in IPv4, in which subnetting was used to improve the efficiency of utilization of the small address space. In IPv6, the address space is deemed large enough for the foreseeable future, and a local area subnet always uses 64 bits for the host portion of the address, designated as the interface identifier, while the most-significant 64 bits are used as the routing prefix.[35] While the myth has existed regarding IPv6 subnets being impossible to scan, RFC 7707 notes that patterns resulting from some IPv6 address configuration techniques and algorithms allow address scanning in many real-world scenarios.

A general structure for an IPv6 unicast address

Address representation

The 128 bits of an IPv6 address are represented in 8 groups of 16 bits each. Each group is written as four hexadecimal digits (sometimes called hextets[36][37] or more formally hexadectets[38] and informally a quibble or quad-nibble[38]) and the groups are separated by colons (:). An example of this representation is *2001:0db8:0000:0000:0000:ff00:0042:8329*.

For convenience and clarity, the representation of an IPv6 address may be shortened with the following rules.

- One or more leading zeros from any group of hexadecimal digits are removed, which is usually done to all of the leading zeros. For example, the group *0042* is converted to *42*.
- Consecutive sections of zeros are replaced with two colons (::). This may only be used once in an address, as multiple use would render the address indeterminate. RFC 5952 (https://datatracker.ietf.org/doc/html/rfc5952) requires that a double colon not be used to denote an omitted single section of zeros.[39]

An example of application of these rules:

> Initial address: *2001:0db8:0000:0000:0000:ff00:0042:8329*.
> After removing all leading zeros in each group: *2001:db8:0:0:0:ff00:42:8329*.
> After omitting consecutive sections of zeros: *2001:db8::ff00:42:8329*.

The loopback address *0000:0000:0000:0000:0000:0000:0000:0001* is defined in RFC 5156 (https://datatracker.ietf.org/doc/html/rfc5156) and is abbreviated to *::1* by using both rules.

As an IPv6 address may have more than one representation, the IETF has issued a proposed standard for representing them in text.[40]

Because IPv6 addresses contain colons, and URLs use colons to separate the host from the port number, RFC2732[41] specifies that an IPv6 address used as the host-part of a URL should be enclosed in square brackets, e.g. http://[2001:db8:4006:812::200e] or http://[2001:db8:4006:812::200e]:8080/path/page.html.

Link-local address

All interfaces of IPv6 hosts require a link-local address. IPv6 link-local addresses have the prefix *fe80::/10*. This prefix is combined with a 64-bit suffix, which the host can compute and/or assign by itself—without configuration and without the presence or cooperation of an external network component like a DHCP server.

The Link-Local Unicast Address structure in IPv6

The lower 64 bits of the link-local address (the suffix) were originally derived from the MAC address of the underlying network interface card. As this method of assigning addresses would cause undesirable address changes when faulty network cards were replaced, and as it also suffered from a number of security and privacy issues, RFC 8064 has replaced the original MAC-based method with the hash-based method specified in RFC 7217.

Address uniqueness and router solicitation

IPv6 uses a new mechanism for mapping IP addresses to link-layer addresses (MAC addresses), because it does not support the broadcast addressing method, on which the functionality of the Address Resolution Protocol (ARP) in IPv4 is based. IPv6 implements the Neighbor Discovery Protocol (NDP, ND) in the link layer, which relies on ICMPv6 and multicast transmission.[5]:210 IPv6 hosts verify the uniqueness of their IPv6 addresses in a local area network (LAN) by sending a neighbor solicitation message asking for the link-layer address of the IP address. If any other host in the LAN is using that address, it responds.[42]

A host bringing up a new IPv6 interface first generates a unique link-local address using one of several mechanisms designed to generate a unique address. Should a non-unique address be detected, the host can try again with a newly generated address. Once a unique link-local address is established, the IPv6 host determines whether the LAN is connected on this link to any router interface that supports IPv6. It does so by sending out an ICMPv6 router solicitation message to the all-routers[43] multicast group with its link-local address as source. If there is no answer after a predetermined number of attempts, the host concludes that no routers are connected. If it does get a response, known as a router advertisement, from a router, the response includes the network configuration information to allow establishment of a globally unique address with an appropriate unicast network prefix.[44] There are also two flag bits that tell the host whether it should use DHCP to get further information and addresses:

- The Manage bit, which indicates whether or not the host should use DHCP to obtain additional addresses rather than rely on an auto-configured address from the router advertisement.
- The Other bit, which indicates whether or not the host should obtain other information through DHCP. The other information consists of one or more prefix information options for the subnets that the host is attached to, a lifetime for the prefix, and two flags:[42]
  - On-link: If this flag is set, the host will treat all addresses on the specific subnet as being on-link and send packets directly to them instead of sending them to a router for the duration of the given lifetime.
  - Address: This flag tells the host to actually create a global address.

Global addressing

The assignment procedure for global addresses is similar to local-address construction. The prefix is supplied from router advertisements on the network. Multiple prefix announcements cause multiple addresses to be configured.[42]

The global unicast address structure in IPv6

Stateless address autoconfiguration (SLAAC) requires a */64* address block, as defined in RFC 4291 (https://datatracker.ietf.org/doc/html/rfc4291). Local Internet registries are assigned at least */32* blocks, which they divide among subordinate networks.[45] The initial recommendation stated assignment of a */48* subnet to end-consumer sites (RFC 3177 (https://datatracker.ietf.org/doc/html/rfc3177)). This was replaced by RFC 6177 (https://datatracker.ietf.org/doc/html/rfc6177), which "recommends giving home sites significantly more than a single */64*, but does not recommend that every home site be given a */48* either". */56*s are specifically considered. It remains to be seen whether ISPs will honor this recommendation. For example, during initial trials, Comcast customers were given a single */64* network.[46]

IPv6 in the Domain Name System

In the Domain Name System (DNS), hostnames are mapped to IPv6 addresses by AAAA ("quad-A") resource records. For reverse resolution, the IETF reserved the domain ip6.arpa, where the name space is hierarchically divided by the 1-digit hexadecimal representation of nibble units (4 bits) of the IPv6 address. This scheme is defined in RFC 3596 (https://datatracker.ietf.org/doc/html/rfc3596).

When a dual-stack host queries a DNS server to resolve a fully qualified domain name (FQDN), the DNS client of the host sends two DNS requests, one querying A records and the other querying AAAA records. The host operating system may be configured with a preference for address selection rules RFC 6724 (https://datatracker.ietf.org/doc/html/rfc6724).[47]

An alternate record type was used in early DNS implementations for IPv6, designed to facilitate network renumbering, the *A6* records for the forward lookup and a number of other innovations such as *bit-string labels* and *DNAME* records. It is defined in RFC 2874 (https://datatracker.ietf.org/doc/html/rfc2874) and its references (with further discussion of the pros and cons of both schemes in RFC 3364 (https://datatracker.ietf.org/doc/html/rfc3364)), but has been deprecated to experimental status (RFC 3363 (https://datatracker.ietf.org/doc/html/rfc3363)).

Transition mechanisms

IPv6 is not foreseen to supplant IPv4 instantaneously. Both protocols will continue to operate simultaneously for some time. Therefore, IPv6 transition mechanisms are needed to enable IPv6 hosts to reach IPv4 services and to allow isolated IPv6 hosts and networks to reach each other over IPv4 infrastructure.[48]

According to Silvia Hagen, a dual-stack implementation of the IPv4 and IPv6 on devices is the easiest way to migrate to IPv6.[49] Many other transition mechanisms use tunneling to encapsulate IPv6 traffic within IPv4 networks and vice versa. This is an imperfect solution, which reduces the maximum transmission unit (MTU) of a link and therefore complicates Path MTU Discovery, and may increase latency.[50][51]

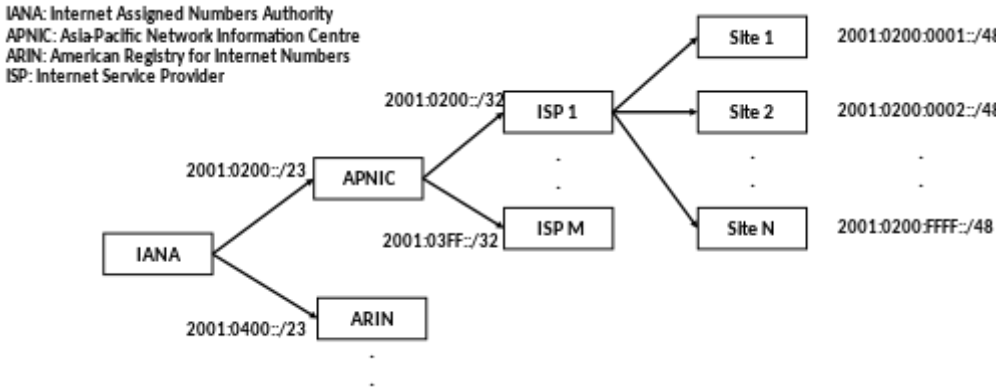
Dual-stack IP implementation

Dual-stack IP implementations provide complete IPv4 and IPv6 protocol stacks in the operating system of a computer or network device on top of the common physical layer implementation, such as Ethernet. This permits dual-stack hosts to participate in IPv6 and IPv4 networks simultaneously. The method is defined in RFC 4213 (https://datatracker.ietf.org/doc/html/rfc4213).[52]

A device with dual-stack implementation in the operating system has an IPv4 and IPv6 address, and can communicate with other nodes in the LAN or the Internet using either IPv4 or IPv6. The Domain Name System (DNS) protocol is used by both IP protocols to resolve fully qualified domain names (FQDN) and IP addresses, but dual stack requires that the resolving DNS server can resolve both types of addresses. Such a dual stack DNS server would hold IPv4 addresses in the A records, and IPv6 addresses in the AAAA records. Depending on the destination that is to be resolved, a DNS name server may return an IPv4 or IPv6 IP address, or both. A default address selection mechanism, or preferred protocol, needs to be configured either on hosts or the DNS server. The IETF has published Happy Eyeballs to assist dual stack applications, so that they can connect using both IPv4 and IPv6, but prefer an IPv6 connection if it is available. However, dual-stack also needs to be implemented on all routers between the host and the service for which the DNS server has returned an IPv6 address. Dual-stack clients should only be configured to prefer IPv6, if the network is able to forward IPv6 packets using the IPv6 versions of routing protocols. When dual stack networks protocols are in place the application layer can be migrated to IPv6.[53]

While dual-stack is supported by major operating system and network device vendors, legacy networking hardware and servers don't support IPv6.

ISP customers with public-facing IPv6



IPv6 Prefix Assignment mechanism with IANA, RIRs, and ISPs

Internet service providers (ISPs) are increasingly providing their business and private customers with public-facing IPv6 global unicast addresses. However, if in the local area network (LAN) IPv4 is still used, and the ISP can only provide a public facing IPv6, the IPv4 LAN addresses are translated into the public facing IPv6 address using NAT64, a network address translation (NAT) mechanism. Some ISPs cannot provide their customers with public-facing IPv4 and IPv6 addresses, thus supporting dual stack networking, because some ISPs have exhausted their globally routable IPv4 address pool. Meanwhile, ISP customers are still trying to reach IPv4 web servers and other destinations.[54]

A significant percentage of ISPs in all regional Internet registry (RIR) zones have obtained IPv6 address space. This includes many of the world's major ISPs and mobile network operators, such as Verizon Wireless, StarHub Cable, Chubu Telecommunications, Kabel Deutschland, Swisscom, T-Mobile, Internode and Telefonica.[55]

While some ISPs still allocate customers only IPv4 addresses, many ISPs allocate their customers only an IPv6 or dual stack IPv4 and IPv6. ISPs report the share of IPv6 traffic from customers over their network to be anything between 20% and 40%, but by mid-2017 IPv6 traffic still only accounted for a fraction of total traffic at several large Internet exchange points (IXPs). AMS-IX reported it to be 2% and SeattleIX reported 7%. A 2017 survey found that many DSL customers that were served by a dual stack ISP did not request DNS servers to resolve fully qualified domain names into IPv6 addresses. The survey also found that the majority of traffic from IPv6-ready webserver resources were still requested and served over IPv4, mostly due to ISP customers that did not use the dual stack facility provided by their ISP and to a lesser extent due to customers of IPv4-only ISPs.[56]

Tunneling

The technical basis for tunneling, or encapsulating IPv6 packets in IPv4 packets, is outlined in RFC 4213. When the Internet backbone was IPv4-only, one of the frequently used tunneling protocols was 6to4.[57] Teredo tunneling was also frequently used for integrating IPv6 LANs with the IPv4 Internet backbone. Teredo is outlined in RFC 4380 and allows IPv6 local area networks to tunnel over IPv4 networks, by encapsulating IPv6 packets within UDP. The Teredo relay is an IPv6 router that mediates between a Teredo server and the native IPv6 network. It was expected that 6to4 and Teredo would be widely deployed until ISP networks would switch to native IPv6, but by 2014 Google Statistics showed that the use of both mechanisms had dropped to almost 0.[58]

IPv4-mapped IPv6 addresses

IPv4-compatible IPv6 unicast address

IPv4-mapped IPv6 unicast address

Hybrid dual-stack IPv6/IPv4 implementations recognize a special class of addresses, the IPv4-mapped IPv6 addresses. These addresses are typically written with a 96-bit prefix in the standard IPv6 format, and the remaining 32 bits written in the customary dot-decimal notation of IPv4. IPv4-mapped addresses are specified in RFC 6890 (https://datatracker.ietf.org/doc/html/rfc6890)[59] section 2.2.3 Table 20 and are defined in RFC 4291.

Addresses in this group consist of an 80-bit prefix of zeros, the next 16 bits are ones, and the remaining, least-significant 32 bits contain the IPv4 address. For example, ::ffff:192.0.2.128 represents the IPv4 address 192.0.2.128. Another format, called "IPv4-compatible IPv6 address", is ::192.0.2.128; however, this method is deprecated.[60]

Because of the significant internal differences between IPv4 and IPv6 protocol stacks, some of the lower-level functionality available to programmers in the IPv6 stack does not work the same when used with IPv4-mapped addresses. Some common IPv6 stacks do not implement the IPv4-mapped address feature, either because the IPv6 and IPv4 stacks are separate implementations (e.g., Microsoft Windows 2000, XP, and Server 2003), or because of security concerns (OpenBSD).[61] On these operating systems, a program must open a separate socket for each IP protocol it uses. On some systems, e.g., the Linux kernel, NetBSD, and FreeBSD, this feature is controlled by the socket option IPV6_V6ONLY, as specified in RFC 3493 (https://datatracker.ietf.org/doc/html/rfc3493).[62]

RFC 6052 (https://datatracker.ietf.org/doc/html/rfc6052) defines a class of IPv4-embedded IPv6 addresses with the address prefix 64:ff9b::/96 for use in NAT64 transition methods. For example, 64:ff9b::192.0.2.128 represents the IPv4 address 192.0.2.128.

Security

A number of security implications may arise from the use of IPv6. Some of them may be related with the IPv6 protocols themselves, while others may be related with implementation flaws.[63][64]

Shadow networks

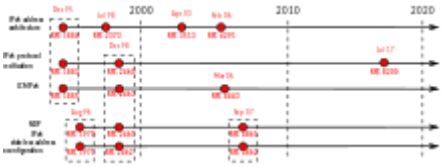
The addition of nodes having IPv6 enabled by default by the software manufacturer, may result in the inadvertent creation of *shadow networks*, causing IPv6 traffic flowing into networks having only IPv4 security management in place. This may also occur with operating system upgrades, when the newer operating system enables IPv6 by default, while the older one did not. Failing to update the security infrastructure to accommodate IPv6 can lead to IPv6 traffic bypassing it.[65] Shadow networks have occurred on business networks in which enterprises are replacing Windows XP systems that do not have an IPv6 stack enabled by default, with Windows 7 systems, that do.[66] Some IPv6 stack implementors have therefore recommended disabling IPv4 mapped addresses and instead using a dual-stack network where supporting both IPv4 and IPv6 is necessary.[67]

IPv6 packet fragmentation

Research has shown that the use of fragmentation can be leveraged to evade network security controls, similar to IPv4. As a result, RFC 7112 (https://datatracker.ietf.org/doc/html/rfc7112) requires that the first fragment of an IPv6 packet contains the entire IPv6 header chain, such that some very pathological fragmentation cases are forbidden. Additionally, as a result of research on the evasion of RA-Guard in RFC 7113 (https://datatracker.ietf.org/doc/html/rfc7113), RFC 6980 (https://datatracker.ietf.org/doc/html/rfc6980) has deprecated the use of fragmentation with Neighbor Discovery, and discouraged the use of fragmentation with Secure Neighbor Discovery (SEND).

Standardization through RFCs

Working-group proposals



A timeline for the standards governing IPv6

Due to the anticipated global growth of the Internet, the Internet Engineering Task Force (IETF) in the early 1990s started an effort to develop a next generation IP protocol.[5]:209 By the beginning of 1992, several proposals appeared for an expanded Internet addressing system and by the end of 1992 the IETF announced a call for white papers.[68] In September 1993, the IETF created a temporary, ad hoc *IP Next Generation* (IPng) area to deal specifically with such issues. The new area was led by Allison Mankin and Scott Bradner, and had a directorate with 15 engineers from diverse backgrounds for direction-setting and preliminary document review:[7][69] The working-group members were J. Allard (Microsoft), Steve Bellovin (AT&T), Jim Bound (Digital Equipment Corporation), Ross Callon (Wellfleet), Brian Carpenter (CERN), Dave Clark (MIT), John Curran (NEARNET), Steve Deering (Xerox), Dino Farinacci (Cisco), Paul Francis (NTT), Eric Fleischmann (Boeing), Mark Knopper (Ameritech), Greg Minshall (Novell), Rob Ullmann (Lotus), and Lixia Zhang (Xerox).[70]

The Internet Engineering Task Force adopted the IPng model on 25 July 1994, with the formation of several IPng working groups.[7] By 1996, a series of RFCs was released defining Internet Protocol version 6 (IPv6), starting with RFC 1883 (https://datatracker.ietf.org/doc/html/rfc1883). (Version 5 was used by the experimental Internet Stream Protocol.)

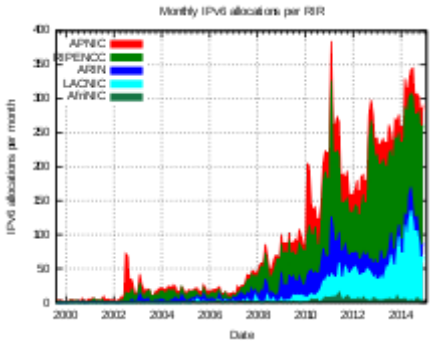
RFC standardization

The first RFC to standardize IPv6 was the RFC 1883 (https://datatracker.ietf.org/doc/html/rfc1883) in 1995, which became obsoleted by RFC 2460 (https://datatracker.ietf.org/doc/html/rfc2460) in 1998.[5]:209 In July 2017 this RFC was obsoleted by RFC 8200 (https://datatracker.ietf.org/doc/html/rfc8200), which elevated IPv6 to "Internet Standard" (the highest maturity level for IETF protocols).[3]

Deployment

The 1993 introduction of Classless Inter-Domain Routing (CIDR) in the routing and IP address allocation for the Internet, and the extensive use of network address translation (NAT), delayed IPv4 address exhaustion to allow for IPv6 deployment, which began in the mid-2000s.

Universities were among the early adopters of IPv6. Virginia Tech deployed IPv6 at a trial location in 2004 and later expanded IPv6 deployment across the campus network. By 2016, 82% of the traffic on their network used IPv6. Imperial College London began experimental IPv6 deployment in 2003 and by 2016 the IPv6 traffic on their networks averaged between 20% and 40%. A significant portion of this IPv6 traffic was generated through their high energy physics collaboration with CERN, which relies entirely on IPv6.[71]



Monthly IPv6 allocations per regional Internet registry (RIR)

The Domain Name System (DNS) has supported IPv6 since 2008. In the same year, IPv6 was first used in a major world event during the Beijing 2008 Summer Olympics.[72][73]

By 2011, all major operating systems in use on personal computers and server systems had production-quality IPv6 implementations. Cellular telephone systems presented a large deployment field for Internet Protocol devices as mobile telephone service made the transition from 3G to 4G technologies, in which voice is provisioned as a voice over IP (VoIP) service that would leverage IPv6 enhancements. In 2009, the US cellular operator Verizon released technical specifications for devices to operate on its "next-generation" networks.[74] The specification mandated IPv6 operation according to the *3GPP Release 8 Specifications (March 2009)*, and deprecated IPv4 as an optional capability.[74]

The deployment of IPv6 in the Internet backbone continued. In 2018 only 25.3% of the about 54,000 autonomous systems advertised both IPv4 and IPv6 prefixes in the global Border Gateway Protocol (BGP) routing database. A further 243 networks advertised only an IPv6 prefix. Internet backbone transit networks offering IPv6 support existed in every country globally, except in parts of Africa, the Middle East and China.[75] By mid-2018 some major European broadband ISPs had deployed IPv6 for the majority of their customers. British Sky Broadcasting provided over 86% of its customers with IPv6, Deutsche Telekom had 56% deployment of IPv6, XS4ALL in the Netherlands had 73% deployment and in Belgium the broadband ISPs VOO and Telenet had 73% and 63% IPv6 deployment respectively.[76] In the United States the broadband ISP Comcast had an IPv6 deployment of about 66%. In 2018 Comcast reported an estimated 36.1 million IPv6 users, while AT&T reported 22.3 million IPv6 users.[77]

See also

- China Next Generation Internet
- Comparison of IPv6 support in operating systems
- Comparison of IPv6 support in common applications
- DoD IPv6 product certification
- Happy Eyeballs
- List of IPv6 tunnel brokers
- University of New Hampshire InterOperability Laboratory

References

1. New Zealand IPv6 Task Force. "FAQs" (https://www.ipv6.org.nz/ipv6-faqs/). Retrieved 26 October 2015.
2. S. Deering; R. Hinden (December 1998), *Internet Protocol, Version 6 (IPv6) Specification*, Internet Engineering Task Force (IETF), RFC 2460 (https://tools.ietf.org/html/rfc2460) Obsoletes RFC 1883.
3. S. Deering; R. Hinden (July 2017), "Internet Protocol, Version 6 (IPv6) Specification", *Ietf Request for Comments (RFC) Pages - Test*, Internet Engineering Task Force (IETF), ISSN 2070-1721 (https://www.worldcat.org/issn/2070-1721), RFC 8200 (https://tools.ietf.org/html/rfc8200) Obsoletes RFC 2460.
4. Siddiqui, Aftab (17 July 2017). "RFC 8200 – IPv6 has been standardized" (https://www.internetsociety.org/blog/2017/07/rfc-8200-ipv6-has-been-standardized/). Internet Society. Retrieved 25 February 2018.
5. Rami Rosen (2014). *Linux Kernel Networking: Implementation and Theory*. New York: Apress. ISBN 9781430261971. OCLC 869747983 (https://www.worldcat.org/oclc/869747983).
6. *Google IPv6 Conference 2008: What will the IPv6 Internet look like?* (https://www.youtube.com/watch?v=mZo69JQoLb8). Event occurs at 13:35.
7. Bradner, S.; Mankin, A. (January 1995). *The Recommendation for the IP Next Generation Protocol* (https://tools.ietf.org/html/rfc1752). IETF. doi:10.17487/RFC1752 (https://doi.org/10.17487%2FRFC1752). RFC 1752 (https://tools.ietf.org/html/rfc1752).
8. Rashid, Fahmida. "IPv4 Address Exhaustion Not Instant Cause for Concern with IPv6 in Wings" (https://www.eweek.com/c/a/IT-Infrastructure/IPv4-Address-Exhaustion-Not-Instant-Cause-for-Concern-with-IPv6-in-Wings-287643/). eWeek. Retrieved 23 June 2012.
9. Ward, Mark (14 September 2012). "Europe hits old internet address limits" (https://www.bbc.co.uk/news/technology-19600718). *BBC News*. BBC. Retrieved 15 September 2012.
10. Huston, Geoff. "IPV4 Address Report" (https://www.potaroo.net/tools/ipv4/).
11. "African Network Information Center : -" (https://my.afrinic.net/help/policies/afpol-v4200407-000.htm). *my.afrinic.net*. Retrieved 28 November 2018.
12. news, Publication date: 25 Nov 2019-; ipv4; Depletion, Ipv4; ipv6; Release, Press. "The RIPE NCC has run out of IPv4 Addresses" (https://www.ripe.net/publications/news/about-ripe-ncc-and-ripe/the-ripe-ncc-has-run-out-of-ipv4-addresses). *RIPE Network Coordination Centre*. Retrieved 26 November 2019.
13. Partridge, C.; Kastenholz, F. (December 1994). "Technical Criteria for Choosing IP The Next Generation (IPng)" (https://www.ietf.org/rfc/rfc1726.txt). RFC 1726 (https://tools.ietf.org/html/rfc1726).
14. RFC 1112 (https://datatracker.ietf.org/doc/html/rfc1112), *Host extensions for IP multicasting*, S. Deering (August 1989)

15. RFC 3956 (https://datatracker.ietf.org/doc/html/rfc3956), *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*, P. Savola, B. Haberman (November 2004)
16. RFC 2908 (https://datatracker.ietf.org/doc/html/rfc2908), *The Internet Multicast Address Allocation Architecture*, D. Thaler, M. Handley, D. Estrin (September 2000)
17. RFC 3306 (https://datatracker.ietf.org/doc/html/rfc3306), *Unicast-Prefix-based IPv6 Multicast Addresses*, B. Haberman, D. Thaler (August 2002)
18. Thomson, S.; Narten, T.; Jinmei, T. (September 2007). "IPv6 Stateless Address Autoconfiguration" (https://tools.ietf.org/html/rfc4862). RFC 4862 (https://tools.ietf.org/html/rfc4862).
19. RFC 2894 (https://datatracker.ietf.org/doc/html/rfc2894), *Router Renumbering for IPv6*, M. Crawford, August 2000.
20. T. Narten; R. Draves; S. Krishnan (September 2007). "Privacy Extensions for Stateless Address Autoconfiguration in IPv6" (https://www.internetsociety.org/resources/deploy360/2014/privacy-extensions-for-ipv6-slaac/). *www.ietf.org*. Retrieved 13 March 2017.
21. Narten, Thomas; Draves, Richard; Krishnan, Suresh. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6* (https://tools.ietf.org/html/rfc4941). doi:10.17487/RFC4941 (https://doi.org/10.17487%2FRFC4941). RFC 4941 (https://tools.ietf.org/html/rfc4941).
22. "Overview of the Advanced Networking Pack for Windows XP" (https://web.archive.org/web/20170907013704/https://support.microsoft.com/en-us/help/817778/overview-of-the-advanced-networking-pack-for-windows-xp). Archived from the original (http://support.microsoft.com/kb/817778) on 7 September 2017. Retrieved 15 April 2019.
23. "Privacy Extensions for IPv6 SLAAC" (https://www.internetsociety.org/resources/deploy360/2014/privacy-extensions-for-ipv6-slaac/). *Internet Society*. 8 August 2014. Retrieved 17 January 2020.
24. Ferguson, P.; Berkowitz, H. (January 1997). "Network Renumbering Overview: Why would I want it and what is it anyway?" (https://tools.ietf.org/search/rfc2071). RFC 2071 (https://tools.ietf.org/html/rfc2071).
25. Berkowitz, H. (January 1997). "Router Renumbering Guide" (https://tools.ietf.org/html/rfc2072). RFC 2072 (https://tools.ietf.org/html/rfc2072).
26. Cooper, Alissa; Gont, Fernando; Thaler, Dave. *Recommendation on Stable IPv6 Interface Identifiers* (https://tools.ietf.org/html/rfc8064). doi:10.17487/RFC8064 (https://doi.org/10.17487%2FRFC8064). RFC 8064 (https://tools.ietf.org/html/rfc8064).
27. Silvia Hagen (2014). *IPv6 Essentials: Integrating IPv6 into Your IPv4 Network* (3rd ed.). Sebastopol, CA: O'Reilly Media. p. 196. ISBN 978-1-4493-3526-7. OCLC 881832733 (https://www.worldcat.org/oclc/881832733).
28. "The History of Domain Names | IPv6" (https://web.archive.org/web/20180612211153/http://www.historyofdomainnames.com/ipv6/). *www.historyofdomainnames.com*. Archived from the original (http://www.historyofdomainnames.com/ipv6/) on 12 June 2018. Retrieved 12 June 2018.
29. Zack, E. (July 2013). "IPv6 Security Assessment and Benchmarking" (http://www.ipv6hackers.org/meetings/ipv6-hackers-1).
30. Gont, F. (March 2016). "Operational Implications of IPv6 Packets with Extension Headers" (https://tools.ietf.org/html/draft-gont-v6ops-ipv6-ehs-packet-drops-03). *draft-gont-v6ops-ipv6-ehs-packet-drops-03*.
31. RFC 3963 (https://datatracker.ietf.org/doc/html/rfc3963), *Network Mobility (NEMO) Basic Protocol Support*, V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert (January 2005)
32. RFC 2675 (https://datatracker.ietf.org/doc/html/rfc2675), *IPv6 Jumbograms*, D. Borman, S. Deering, R. Hinden (August 1999)
33. RFC 2474 (https://datatracker.ietf.org/doc/html/rfc2474)
34. RFC 3168 (https://datatracker.ietf.org/doc/html/rfc3168)

35. RFC 4291 (https://datatracker.ietf.org/doc/html/rfc4291), p. 9.
36. Graziani, Rick (2012). *IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6* (https://books.google.com/books?id=FbYjJjZNA5gC&pg=PA55). Cisco Press. p. 55. ISBN 978-0-13-303347-2.
37. Coffeen, Tom (2014). *IPv6 Address Planning: Designing an Address Plan for the Future* (https://books.google.com/books?id=dZU8BQAAQBAJ&pg=PT170). O'Reilly Media. p. 170. ISBN 978-1-4919-0326-1.
38. Horley, Edward (2013). *Practical IPv6 for Windows Administrators* (https://books.google.com/books?id=u50QAwAAQBAJ&q=17&pg=PA17). Apress. p. 17. ISBN 978-1-4302-6371-5.
39. S. Kawamura (August 2010). "A Recommendation for IPv6 Address Text Representation" (https://tools.ietf.org/html/rfc5952#section-4.2.2). section 4.2.2. RFC 5952 (https://tools.ietf.org/html/rfc5952).
40. S. Kawamura (August 2010). "A Recommendation for IPv6 Address Text Representation" (https://tools.ietf.org/html/rfc5952). RFC 5952 (https://tools.ietf.org/html/rfc5952).
41. "Format for Literal IPv6 Addresses in URL's" (https://tools.ietf.org/html/rfc2732). August 2010. RFC 2732 (https://tools.ietf.org/html/rfc2732).
42. Narten, T. (August 1999). "Neighbor discovery and stateless autoconfiguration in IPv6". *IEEE Internet Computing*. **3** (4): 54–62. doi:10.1109/4236.780961 (https://doi.org/10.1109%2F4236.780961).
43. T. Narten (September 2007). "Neighbor Discovery for IP version 6 (IPv6)" (https://tools.ietf.org/html/rfc4861#section-6.3.7). section 6.3.7. RFC 4861 (https://tools.ietf.org/html/rfc4861).
44. S. Thomson (September 2007). "IPv6 Stateless Address Autoconfiguration" (https://tools.ietf.org/html/rfc4862#section-5.5.1). section 5.5.1. RFC 4862 (https://tools.ietf.org/html/rfc4862).
45. "IPv6 Address Allocation and Assignment Policy" (https://www.ripe.net/ripe/docs/ripe-512). RIPE NCC. 8 February 2011. Retrieved 27 March 2011.
46. Brzozowski, John (31 January 2011). "Comcast Activates First Users With IPv6 Native Dual Stack Over DOCSIS" (https://corporate.comcast.com/comcast-voices/comcast-activates-first-users-with-ipv6-native-dual-stack-over-docsis). *corporate.comcast.com*. Comcast. Retrieved 15 April 2019.
47. Silvia Hagen (2014). *IPv6 Essentials: Integrating IPv6 into Your IPv4 Network*. O'Reilly Media, Inc. p. 176. ISBN 9781449335267.
48. "IPv6 Transition Mechanism / Tunneling Comparison" (https://www.sixxs.net/faq/connectivity/?faq=comparison). Sixxs.net. Retrieved 20 January 2012.
49. Silvia Hagen (2014). *IPv6 Essentials: Integrating IPv6 into Your IPv4 Network*. O'Reilly Media, Inc. pp. 222–223. ISBN 9781449335267.
50. "Advisory Guidelines for 6to4 Deployment" (https://tools.ietf.org/html/rfc6343). IETF. RFC 6343 (https://tools.ietf.org/html/rfc6343). Retrieved 20 August 2012.
51. "IPv6: Dual stack where you can; tunnel where you must" (https://web.archive.org/web/20080511161659/http://www.networkworld.com/news/tech/2007/090507-tech-uodate.html). networkworld.com. 5 September 2007. Archived from the original (http://www.networkworld.com/news/tech/2007/090507-tech-uodate.html) on 11 May 2008. Retrieved 27 November 2012.
52. "Basic Transition Mechanisms for IPv6 Hosts and Routers" (https://tools.ietf.org/html/rfc4213). IETF. RFC 4213 (https://tools.ietf.org/html/rfc4213). Retrieved 20 August 2012.
53. Silvia Hagen (2014). *IPv6 Essentials: Integrating IPv6 into Your IPv4 Network*. O'Reilly Media, Inc. p. 222. ISBN 9781449335267.
54. Juniper TechLibrary (31 August 2017). "Understanding Dual Stacking of IPv4 and IPv6 Unicast Addresses" (https://www.juniper.net/documentation/en_US/junos/topics/concept/ipv6-dual-stack-understanding.html). *www.juniper.net*. Retrieved 13 March 2017.
55. "IPv6" (https://www.nro.net/ipv6/). *www.nro.net*. Retrieved 13 March 2017.

56. Enric Pujol (12 June 2017). "What stops IPv6 traffic in a dual-stack ISP?" (https://blog.apnic.net/2017/06/13/stops-ipv6-traffic-dual-stack-isp/). *www.apnic.net*. Retrieved 13 June 2017.
57. Steven J. Vaughan-Nichols (14 October 2010). "Five ways for IPv6 and IPv4 to peacefully co-exist" (https://www.zdnet.com/article/five-ways-for-ipv6-and-ipv4-to-peacefully-co-exist/). *www.zdnet.com*. Retrieved 13 March 2017.
58. Silvia Hagen (2014). *IPv6 Essentials: Integrating IPv6 into Your IPv4 Network*. O'Reilly Media, Inc. p. 33. ISBN 9781449335267.
59. "Special-Purpose IP Address Registries" (https://tools.ietf.org/html/rfc6890#section-2.2.3). IETF. RFC 6890 (https://tools.ietf.org/html/rfc6890).
60. Hinden, Robert M.; Deering, Stephen E. "RFC 4291 - IP Version 6 Addressing Architecture, section 2.5.5.1. IPv4-Compatible IPv6 Address" (https://tools.ietf.org/html/rfc4291#section-2.5.5.1). *tools.ietf.org*. Retrieved 23 September 2019.
61. `inet6(4) (https://man.openbsd.org/inet6.4)` – OpenBSD Kernel Interfaces Manual
62. "Basic Socket Interface Extensions for IPv6" (https://tools.ietf.org/html/rfc3493#page-22). IETF. February 2003. p. 22. RFC 3493 (https://tools.ietf.org/html/rfc3493). Retrieved 28 November 2017.
63. Gont, Fernando (10 March 2019), *IPv6 Security for IPv4 Engineers* (https://www.internetsociety.org/wp-content/uploads/2019/03/deploy360-ipv6-security-v1.0.pdf) (PDF), retrieved 30 August 2019
64. Gont, Fernando (10 January 2019), *IPv6 Security Frequently Asked Questions (FAQ)* (https://www.internetsociety.org/wp-content/uploads/2019/02/Deploy360-IPv6-Security-FAQ.pdf) (PDF), retrieved 30 August 2019
65. Mullins, Robert (5 April 2012), *Shadow Networks: an Unintended IPv6 Side Effect* (https://web.archive.org/web/20130411113334/http://www.networkcomputing.com/ipv6-tech-center/shadow-networks-an-unintended-ipv6-side/232800326), archived from the original (http://www.networkcomputing.com/ipv6-tech-center/shadow-networks-an-unintended-ipv6-side/232800326) on 11 April 2013, retrieved 2 March 2013
66. Cicileo, Guillermo; Gagliano, Roque; O'Flaherty, Christian; et al. (October 2009). *IPv6 For All: A Guide for IPv6 Usage and Application in Different Environments* (https://www.ipv6forum.com/dl/books/ipv6forall.pdf) (PDF). p. 5. Retrieved 2 March 2013.
67. Jun-ichiro itojun Hagino (October 2003). "IPv4-Mapped Addresses on the Wire Considered Harmful" (https://tools.ietf.org/html/draft-itojun-v6ops-v4mapped-harmful-02).
68. Bradner, S.; Mankin, A. (December 1993). "IP: Next Generation (IPng) White Paper Solicitation" (https://tools.ietf.org/html/rfc1550). RFC 1550 (https://tools.ietf.org/html/rfc1550).
69. "History of the IPng Effort" (https://web.archive.org/web/20140523072903/http://grnlight.net/index.php/programming-articles/103-history-of-the-ipng-effort). *The Sun*. Archived from the original (http://grnlight.net/index.php/programming-articles/103-history-of-the-ipng-effort) on 23 May 2014.
70. "The Recommendation for the IP Next Generation Protocol – Appendix B" (https://tools.ietf.org/html/rfc1752#appendix-B). RFC 1752 (https://tools.ietf.org/html/rfc1752).
71. *State of IPv6 Deployment 2018* (https://www.internetsociety.org/resources/2018/state-of-ipv6-deployment-2018/), Internet Society, 2018, p. 3
72. "Beijing2008.cn leaps to next-generation Net" (https://web.archive.org/web/20090204051327/http://en.beijing2008.cn/news/official/preparation/n214384681.shtml) (Press release). The Beijing Organizing Committee for the Games of the XXIX Olympiad. 30 May 2008. Archived from the original (http://en.beijing2008.cn/news/official/preparation/n214384681.shtml) on 4 February 2009.
73. Das, Kaushik (2008). "IPv6 and the 2008 Beijing Olympics" (http://ipv6.com/articles/general/IPv6-Olympics-2008.htm). *IPv6.com*. Retrieved 15 August 2008.

74. Derek Morr (9 June 2009). "Verizon Mandates IPv6 Support for Next-Gen Cell Phones" (http://www.circleid.com/posts/20090609_verizon_mandates_ipv6_support_for_next_gen_cell_phones/). CircleID.
75. *State of IPv6 Deployment 2018* (https://www.internetsociety.org/resources/2018/state-of-ipv6-deployment-2018/), Internet Society, 2018, p. 6
76. *State of IPv6 Deployment 2018* (https://www.internetsociety.org/resources/2018/state-of-ipv6-deployment-2018/), Internet Society, 2018, p. 7
77. *State of IPv6 Deployment 2018* (https://www.internetsociety.org/resources/2018/state-of-ipv6-deployment-2018/), Internet Society, 2018, pp. 7–8

External links

- IPv6 in the Linux Kernel (https://www.haifux.org/lectures/187) by Rami Rosen.
- Free Pool of IPv4 Address Space Depleted (https://www.nro.net/news/ipv4-free-pool-depleted)
- An Introduction and Statistics about IPV6 (https://www.google.com/intl/en/ipv6/)