[Template:Use dmy dates](/wiki/Template:Use_dmy_dates" \o "Template:Use dmy dates) [Template:IPstack](/wiki/Template:IPstack) **Internet Protocol version 6** (**IPv6**) is the most recent version of the [Internet Protocol](/wiki/Internet_Protocol) (IP), the [communications protocol](/wiki/Communications_protocol) that provides an identification and location system for computers on networks and routes traffic across the [Internet](/wiki/Internet). IPv6 was developed by the [Internet Engineering Task Force](/wiki/Internet_Engineering_Task_Force) (IETF) to deal with the long-anticipated problem of [IPv4 address exhaustion](/wiki/IPv4_address_exhaustion). IPv6 is intended to replace [IPv4](/wiki/IPv4).[[1]](#cite_note-1) Every device on the Internet is assigned an [IP address](/wiki/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 than the IPv4 address space has available were necessary to connect new devices in the future. By 1998, the [Internet Engineering Task Force](/wiki/Internet_Engineering_Task_Force) (IETF) had formalized the successor protocol. IPv6 uses a 128-bit address, theoretically allowing 2128, or approximately [Template:Val](/wiki/Template:Val) addresses. The actual number is slightly smaller, as multiple ranges are reserved for special use or completely excluded from use. The total number of possible IPv6 address is more than [Template:Val](/wiki/Template:Val) times as many as IPv4, which uses 32-bit addresses and provides approximately 4.3 billion addresses. The two protocols are not designed to be [interoperable](/wiki/Interoperable), complicating the transition to IPv6. However, several [IPv6 transition mechanisms](/wiki/IPv6_transition_mechanisms) have been devised to permit communication between IPv4 and IPv6 hosts.

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](/wiki/Routing_table). 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 of four [hexadecimal](/wiki/Hexadecimal) digits with the groups being separated by colons, for example 2001:0db8:0000:0042:0000:8a2e:0370:7334, but methods to abbreviate this full notation exist.

[Template:Toc level](/wiki/Template:Toc_level)

## Contents

* 1 Main features[[edit](/index.php?title=(none)&action=edit&section=1)]
* 2 Motivation and origin[[edit](/index.php?title=(none)&action=edit&section=2)]
  + 2.1 IPv4[[edit](/index.php?title=(none)&action=edit&section=3)]
  + 2.2 Working-group proposals[[edit](/index.php?title=(none)&action=edit&section=4)]
* 3 Comparison with IPv4[[edit](/index.php?title=(none)&action=edit&section=5)]
  + 3.1 Larger address space[[edit](/index.php?title=(none)&action=edit&section=6)]
  + 3.2 Multicasting[[edit](/index.php?title=(none)&action=edit&section=7)]
  + 3.3 Stateless address autoconfiguration (SLAAC)[[edit](/index.php?title=(none)&action=edit&section=8)]
  + 3.4 Network-layer security[[edit](/index.php?title=(none)&action=edit&section=9)]
  + 3.5 Simplified processing by routers[[edit](/index.php?title=(none)&action=edit&section=10)]
  + 3.6 Mobility[[edit](/index.php?title=(none)&action=edit&section=11)]
  + 3.7 Options extensibility[[edit](/index.php?title=(none)&action=edit&section=12)]
  + 3.8 Jumbograms[[edit](/index.php?title=(none)&action=edit&section=13)]
  + 3.9 Privacy[[edit](/index.php?title=(none)&action=edit&section=14)]
* 4 Packet format[[edit](/index.php?title=(none)&action=edit&section=15)]
* 5 Addressing[[edit](/index.php?title=(none)&action=edit&section=16)]
  + 5.1 Address representation[[edit](/index.php?title=(none)&action=edit&section=17)]
  + 5.2 Address uniqueness[[edit](/index.php?title=(none)&action=edit&section=18)]
  + 5.3 Link local address[[edit](/index.php?title=(none)&action=edit&section=19)]
  + 5.4 Dual IP stack implementation[[edit](/index.php?title=(none)&action=edit&section=23)]
  + 5.5 Tunneling[[edit](/index.php?title=(none)&action=edit&section=24)]
    - 5.5.1 Automatic tunneling[[edit](/index.php?title=(none)&action=edit&section=25)]
    - 5.5.2 Configured and automated tunneling (6in4)[[edit](/index.php?title=(none)&action=edit&section=26)]
  + 5.6 Proxying and translation for IPv6-only hosts[[edit](/index.php?title=(none)&action=edit&section=27)]
* 6 IPv6 readiness[[edit](/index.php?title=(none)&action=edit&section=28)]
  + 6.1 Software[[edit](/index.php?title=(none)&action=edit&section=29)]
    - 6.1.1 IPv4-mapped IPv6 addresses[[edit](/index.php?title=(none)&action=edit&section=30)]
  + 6.2 Hardware and embedded systems[[edit](/index.php?title=(none)&action=edit&section=31)]
  + 6.3 Shadow networks[[edit](/index.php?title=(none)&action=edit&section=32)]
* 7 Deployment[[edit](/index.php?title=(none)&action=edit&section=33)]
* 8 See also[[edit](/index.php?title=(none)&action=edit&section=34)]
* 9 References[[edit](/index.php?title=(none)&action=edit&section=35)]
* 10 External links[[edit](/index.php?title=(none)&action=edit&section=36)]

## Main features[[edit](/index.php?title=(none)&action=edit&section=1)]

[thumb|300px|Decomposition of the](/wiki/File:Ipv6_address_leading_zeros.svg) [IPv6 address](/wiki/IPv6_address) representation into its binary form IPv6 is an [Internet Layer](/wiki/Internet_Layer) protocol for [packet-switched](/wiki/Packet-switched) [internetworking](/wiki/Internetworking) and provides end-to-end [datagram](/wiki/Datagram) transmission across multiple IP networks, closely adhering to the design principles developed in the previous version of the protocol, [Internet Protocol Version 4](/wiki/IPv4) (IPv4). IPv6 was first formally described in [Internet standard](/wiki/Internet_standard) document RFC 2460, published in December 1998.<ref name=rfc2460>RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*, [S. Deering](/wiki/Steve_Deering), R. Hinden (December 1998)</ref> In addition to offering more addresses, IPv6 also implements features not present in IPv4. It simplifies aspects of address assignment ([stateless address autoconfiguration](/wiki/IPv6_address#Stateless_address_autoconfiguration)), 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](/wiki/Subnetwork) size is standardized by fixing the size of the host identifier portion of an address to 64 bits to facilitate an automatic mechanism for forming the host identifier from [link layer](/wiki/Link_layer) addressing information ([MAC address](/wiki/MAC_address)). [Network security](/wiki/Network_security) was a design requirement of the IPv6 architecture, and included the original specification of [IPsec](/wiki/IPsec).

IPv6 does not specify interoperability features with IPv4, but essentially creates a parallel, independent network. Exchanging traffic between the two networks requires translator gateways employing one of several transition mechanisms, such as [NAT64](/wiki/NAT64), or a [tunneling protocol](/wiki/Tunneling_protocol) like [6to4](/wiki/6to4), [6in4](/wiki/6in4), or [Teredo](/wiki/Teredo_tunneling).

## Motivation and origin[[edit](/index.php?title=(none)&action=edit&section=2)]

### IPv4[[edit](/index.php?title=(none)&action=edit&section=3)]

[right|300px|thumb|Decomposition of the quad-dotted](/wiki/Image:Ipv4_address.svg) [IPv4 address](/wiki/IPv4_address) representation to its binary value [Internet Protocol Version 4](/wiki/IPv4) (IPv4) was the first publicly used version of the [Internet Protocol](/wiki/Internet_Protocol). IPv4 was developed as a research project by the [Defense Advanced Research Projects Agency](/wiki/DARPA) (DARPA), a [United States Department of Defense](/wiki/United_States_Department_of_Defense) [agency](/wiki/Government_agency), before becoming the foundation for the [Internet](/wiki/Internet) and the [World Wide Web](/wiki/World_Wide_Web). It is currently described by [IETF](/wiki/IETF) publication RFC 791 (September 1981), which replaced an earlier definition (RFC 760, January 1980). IPv4 included an addressing system that used numerical identifiers consisting of 32 bits. These addresses are typically displayed in [quad-dotted notation](/wiki/Quad-dotted_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 232 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.[[2]](#cite_note-2) 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](/wiki/Classless_Inter-Domain_Routing) model, it became clear that this would not suffice to prevent [IPv4 address exhaustion](/wiki/IPv4_address_exhaustion), and that further changes to the Internet infrastructure were needed.<ref name=rfc1752>[Template:Cite web](/wiki/Template:Cite_web)</ref>

The last unassigned top-level address blocks of 16 million IPv4 addresses were allocated in February 2011 by the [Internet Assigned Numbers Authority](/wiki/Internet_Assigned_Numbers_Authority) (IANA) to the five [regional Internet registries](/wiki/Regional_Internet_registry) (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](/wiki/Classless_Inter-Domain_Routing) (CIDR) block remains. After that, only blocks of 1024 addresses (/22) will be provided from the RIRs to a [local Internet registry](/wiki/Local_Internet_registry) (LIR). As of September 2015, all of [Asia-Pacific Network Information Centre](/wiki/Asia-Pacific_Network_Information_Centre) (APNIC), the [Réseaux IP Européens Network Coordination Centre](/wiki/RIPE_NCC) (RIPE\_NCC), [Latin America and Caribbean Network Information Centre](/wiki/Latin_America_and_Caribbean_Network_Information_Centre) (LACNIC), and [American Registry for Internet Numbers](/wiki/American_Registry_for_Internet_Numbers) (ARIN) have reached this stage.[[3]](#cite_note-3)[[4]](#cite_note-4)[[5]](#cite_note-5) This leaves [African Network Information Center](/wiki/AFRINIC) (AFRINIC) as the sole regional internet registry that is still using the normal protocol for distributing IPv4 addresses.

### Working-group proposals[[edit](/index.php?title=(none)&action=edit&section=4)]

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.[[6]](#cite_note-6) 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](/wiki/Scott_Bradner), and had a directorate with 15 engineers from diverse backgrounds for direction-setting and preliminary document review:<ref name=rfc1752/>[[7]](#cite_note-7) The working-group members were [J. Allard](/wiki/J._Allard) (Microsoft), [Steve Bellovin](/wiki/Steven_M._Bellovin) (AT&T), Jim Bound (Digital Equipment Corporation), Ross Callon (Wellfleet), [Brian Carpenter](/wiki/Brian_Carpenter_(Internet_engineer)) (CERN), [Dave Clark](/wiki/David_D._Clark) (MIT), [John Curran](/wiki/John_Curran_(businessman)) (NEARNET), [Steve Deering](/wiki/Steve_Deering) (Xerox), Dino Farinacci (Cisco), Paul Francis (NTT), Eric Fleischmann (Boeing), Mark Knopper (Ameritech), Greg Minshall (Novell), Rob Ullmann (Lotus), and [Lixia Zhang](/wiki/Lixia_Zhang) (Xerox).[[8]](#cite_note-8) The Internet Engineering Task Force adopted the IPng model on 25 July 1994, with the formation of several IPng working groups.<ref name=rfc1752/> By 1996, a series of [RFCs](/wiki/Request_for_comments) was released defining Internet Protocol version 6 (IPv6), starting with RFC 1883. (Version 5 was used by the experimental [Internet Stream Protocol](/wiki/Internet_Stream_Protocol).)

It is widely expected that the Internet will use IPv4 alongside IPv6 for the foreseeable future. Direct communication between the IPv4 and IPv6 network protocols is not possible; therefore, intermediary trans-protocol systems are needed as a communication conduit between IPv4 and IPv6 whether on a single device or among network nodes.

## Comparison with IPv4[[edit](/index.php?title=(none)&action=edit&section=5)]

On the Internet, data is transmitted in the form of [network packets](/wiki/Network_packet). IPv6 specifies a new [packet format](/wiki/IPv6_packet), designed to minimize packet header processing by routers.<ref name=rfc2460/><ref name=rfc1726>[Template:Cite web](/wiki/Template:Cite_web)</ref> Because the headers of IPv4 packets and IPv6 packets are significantly different, the two protocols are not interoperable. However, in most respects, IPv6 is an extension of IPv4. 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 [FTP](/wiki/File_Transfer_Protocol) and [NTP](/wiki/Network_Time_Protocol), where the new address format may cause conflicts with existing protocol syntax.

### Larger address space[[edit](/index.php?title=(none)&action=edit&section=6)]

The main advantage of IPv6 over IPv4 is its larger address space. The length of an IPv6 address is 128 bits, compared with 32 bits in IPv4.<ref name=rfc2460/> The address space therefore has 2128 or approximately [Template:Val](/wiki/Template:Val) addresses.

In addition, the IPv4 address space is poorly allocated, with approximately 14%, in 2011, of all available addresses utilized.[[9]](#cite_note-9) While these numbers are large, it was not the intent of the designers of the IPv6 address space to assure geographical saturation with usable addresses. Rather, the longer addresses simplify allocation of addresses, enable efficient [route aggregation](/wiki/Route_aggregation), and allow implementation of special addressing features. In IPv4, complex [Classless Inter-Domain Routing](/wiki/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 264 addresses, the square of the size of the entire IPv4 address space. Thus, actual address space utilization rates will be small in IPv6, but network management and routing efficiency are improved by the large subnet space and hierarchical route aggregation.

Renumbering an existing network for a new connectivity provider with different routing prefixes is a major effort with IPv4.[[10]](#cite_note-10)[[11]](#cite_note-11) 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.<ref name=rfc4862>[Template:Cite web](/wiki/Template:Cite_web)</ref>

### Multicasting[[edit](/index.php?title=(none)&action=edit&section=7)]

[Multicasting](/wiki/Multicast), 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.<ref name=rfc1112>RFC 1112, *Host extensions for IP multicasting*, S. Deering (August 1989)</ref> IPv6 multicast addressing shares common features and protocols with IPv4 multicast, but also provides changes and improvements by eliminating the need for certain protocols. IPv6 does not implement traditional [IP broadcast](/wiki/Broadcast_IP_address), 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 can be 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.<ref name=rfc3956>RFC 3956, *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*, P. Savola, B. Haberman (November 2004)</ref>

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.[[12]](#cite_note-12) Unicast address assignments by a [local Internet registry](/wiki/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.[[13]](#cite_note-13)

### Stateless address autoconfiguration (SLAAC)[[edit](/index.php?title=(none)&action=edit&section=8)]

[Template:See also](/wiki/Template:See_also)

IPv6 hosts can configure themselves automatically when connected to an IPv6 network using the [Neighbor Discovery Protocol](/wiki/Neighbor_Discovery_Protocol) via [Internet Control Message Protocol version 6](/wiki/Internet_Control_Message_Protocol_version_6) (ICMPv6) router discovery messages. When first connected to a network, a host sends a [link-local](/wiki/Link-local_address) router solicitation multicast request for its configuration parameters; routers respond to such a request with a router advertisement packet that contains Internet Layer configuration parameters.<ref name=rfc4862/>

If IPv6 stateless address auto-configuration is unsuitable for an application, a network may use stateful configuration with the [Dynamic Host Configuration Protocol version 6](/wiki/DHCPv6) (DHCPv6) or hosts may be configured manually using static methods.

Routers present a special case of requirements for address configuration, as they often are sources of autoconfiguration information, such as router and prefix advertisements. Stateless configuration of routers can be achieved with a special router renumbering protocol.<ref name=rfc2894>RFC 2894, *Router Renumbering for IPv6*, M. Crawford, August 2000.</ref>

### Network-layer security[[edit](/index.php?title=(none)&action=edit&section=9)]

[Internet Protocol Security (IPsec)](/wiki/IPsec) was originally developed for IPv6, but found widespread deployment first in IPv4, for which it was re-engineered. IPsec was a mandatory specification of the base IPv6 protocol suite,<ref name=rfc2460/><ref name=rfc4301>RFC 4301, *IPv6 Node Requirements", J. Loughney (April 2006)</ref> but has since been made optional.<ref name=rfc6434>RFC 6434,* IPv6 Node Requirements*, E. Jankiewicz, J. Loughney, T. Narten (December 2011)</ref>*

### Simplified processing by routers[[edit](/index.php?title=(none)&action=edit&section=10)]

In IPv6, the packet header and the process of packet forwarding have been simplified. Although IPv6 packet headers are at least twice the size of IPv4 packet headers, packet processing by routers is generally more efficient,<ref name=rfc2460/><ref name=rfc1726/> because less processing is required in routers. This furthers the [end-to-end principle](/wiki/End-to-end_principle) of Internet design, which envisioned that most processing in the network occurs in the leaf nodes.

The packet header in IPv6 is simpler than the IPv4 header. Many rarely used fields have been moved to optional header extensions.

IPv6 routers do not perform [IP fragmentation](/wiki/IP_fragmentation). IPv6 hosts are required to either perform [path MTU discovery](/wiki/Path_MTU_discovery), perform end-to-end fragmentation, or to send packets no larger than the default [Maximum transmission unit](/wiki/Maximum_transmission_unit) (MTU), which is 1280 [octets](/wiki/Octet_(computing)).

The IPv6 header is not protected by a [checksum](/wiki/Checksum). Integrity protection is assumed to be assured by both the link layer or error detection and correction methods in higher-layer protocols, such as TCP and UDP. In IPv4, UDP may actually have a checksum of 0, indicating no checksum; IPv6 requires a checksum in UDP. Therefore, IPv6 routers do not need to recompute a checksum when header fields change, such as the [time to live](/wiki/Time_to_live) (TTL) or [hop count](/wiki/Hop_count).

The *TTL* field of IPv4 has been renamed to *Hop Limit* in IPv6, reflecting the fact that routers are no longer expected to compute the time a packet has spent in a queue.

### Mobility[[edit](/index.php?title=(none)&action=edit&section=11)]

Unlike mobile IPv4, [mobile IPv6](/wiki/Mobile_IPv6) avoids [triangular routing](/wiki/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.<ref name=rfc3963>RFC 3963, *Network Mobility (NEMO) Basic Protocol Support*, V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert (January 2005)</ref>

### Options extensibility[[edit](/index.php?title=(none)&action=edit&section=12)]

The IPv6 packet header has a minimum size of 40 octets. Options are implemented as extensions. This provides the opportunity to extend the protocol in the future without affecting the core packet structure.<ref name=rfc2460/> However, recent studies indicate that there is still widespread dropping of IPv6 packets that contain extension headers.<ref name=draft-ietf-v6ops-ipv6-ehs-in-real-world-01>[Template:Cite web](/wiki/Template:Cite_web)</ref>

### Jumbograms[[edit](/index.php?title=(none)&action=edit&section=13)]

IPv4 limits packets to 65,535 (216−1) octets of payload. An IPv6 node can optionally handle packets over this limit, referred to as [jumbograms](/wiki/Jumbogram), which can be as large as 4,294,967,295 (232−1) octets. The use of jumbograms may improve performance over high-[MTU](/wiki/Maximum_transmission_unit) links. The use of jumbograms is indicated by the Jumbo Payload Option header.<ref name=rfc2675>RFC 2675, *IPv6 Jumbograms*, D. Borman, [S. Deering](/wiki/Steve_Deering), R. Hinden (August 1999)</ref>

### Privacy[[edit](/index.php?title=(none)&action=edit&section=14)]

Like IPv4, IPv6 supports globally unique [IP addresses](/wiki/IP_address) by which the network activity of each device can potentially be tracked. 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. In this approach each device on the network has a unique address globally reachable directly from any other location on the Internet.

Network prefix tracking is less of a concern if the user's ISP assigns a dynamic network prefix via DHCP.[[14]](#cite_note-14)[[15]](#cite_note-15) Privacy extensions do little to protect the user from tracking if the ISP assigns a static network prefix. In this scenario, the network prefix is the unique identifier for tracking and the interface identifier is secondary.

In IPv4 the effort to conserve address space with [network address translation](/wiki/Network_address_translation) (NAT) obfuscates network address spaces, hosts, and topologies. In IPv6 when using address auto-configuration, the Interface Identifier ([MAC address](/wiki/MAC_address)) of an interface port is used to make its public IP address unique, exposing the type of hardware used and providing a unique handle for a user's online activity.

It is not a requirement for IPv6 hosts to use address auto-configuration, however. Yet, even when an address is not based on the MAC address, the interface's address is globally unique, in contrast to NAT-masqueraded private networks. Privacy extensions for IPv6 have been defined to address these privacy concerns,[[16]](#cite_note-16) although [Silvia Hagen](/wiki/Silvia_Hagen) describes these as being largely due to "misunderstanding".[[17]](#cite_note-17) When privacy extensions are enabled, the operating system generates random host identifiers to combine with the assigned network prefix. These ephemeral addresses are used to communicate with remote hosts making it more difficult to track a single device.[[18]](#cite_note-18) Privacy extensions are enabled by default in Windows (since XP SP1), OS X (since 10.7), and iOS (since version 4.3).[[19]](#cite_note-19)[[20]](#cite_note-20) Some Linux distributions have enabled privacy extensions as well.[[21]](#cite_note-21) In addition to the "temporary" addresses mentioned above, there are also "stable" addresses:[[22]](#cite_note-22) Interface Identifiers are generated such that they are stable for each subnet, but change as a host moves from one network to another. In this way it is difficult to track a host as it moves from network to network, but with-in a particular network it will always have the same address (unless the state used in generating the address is reset and the algorithm is run again) so that network access controls and auditing can be potentially be configured.

Privacy extensions do not protect the user from other forms of activity tracking, such as [tracking cookies](/wiki/Tracking_cookie) or [browser fingerprinting](/wiki/Browser_fingerprinting).

## Packet format[[edit](/index.php?title=(none)&action=edit&section=15)]

[Template:Main](/wiki/Template:Main) [thumb|IPv6 packet header](/wiki/File:Ipv6_header.svg)

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](/wiki/Octet_(computing)) (320 bits) of the IPv6 packet. It contains the source and destination addresses, traffic classification options, a hop counter, 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](/wiki/Payload_(computing)).

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](/wiki/IPsec) framework.

Without special options, a payload must be less than [Template:Gaps](/wiki/Template:Gaps). 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](/wiki/Path_MTU_Discovery) to make their packets small enough to reach the destination without needing to be fragmented. See [IPv6 packet fragmentation](/wiki/IPv6_Packet#Fragmentation).

## Addressing[[edit](/index.php?title=(none)&action=edit&section=16)]

[IPv6 addresses](/wiki/IPv6_address) have 128 bits. The design of the IPv6 address space implements a very 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.[[23]](#cite_note-23) The identifier is only unique within the subnet to which a host is connected. IPv6 has a mechanism for automatic address detection,[[24]](#cite_note-24) so that address autoconfiguration always produces unique assignments.

### Address representation[[edit](/index.php?title=(none)&action=edit&section=17)]

The 128 bits of an IPv6 address are represented in 8 groups of 16 bits each. Each group is written as four hexadecimal digits and the groups are separated by colons (:). An example of this representation is 2001:0db8:0000:0000:0000:ff00:0042:8329.

For convenience, an IPv6 address may be abbreviated to shorter notations by application of the following rules.

* One or more [leading zeroes](/wiki/Leading_zero) from any groups of hexadecimal digits are removed; this is usually done to either all or none of the leading zeroes. For example, the group *0042* is converted to *42*.
* Consecutive sections of zeroes are replaced with a double colon (::). The double colon may only be used once in an address, as multiple use would render the address indeterminate. RFC 5952 recommends that a double colon must not be used to denote an omitted single section of zeroes.<ref name=rfc5952sec422>RFC 5952, *A Recommendation for IPv6 Address Text Representation*, S. Kawamura (August 2010), section 4.2.2: <http://tools.ietf.org/html/rfc5952#section-4.2.2</ref>>

An example of application of these rules:

Initial address: 2001:0db8:0000:0000:0000:ff00:0042:8329

After removing all leading zeroes in each group: 2001:db8:0:0:0:ff00:42:8329

After omitting consecutive sections of zeroes: 2001:db8::ff00:42:8329

The loopback address, 0000:0000:0000:0000:0000:0000:0000:0001, may be 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](/wiki/IPv6_address#Recommended_representation_as_text).<ref name=rfc5952>RFC 5952, *A Recommendation for IPv6 Address Text Representation*, S. Kawamura (August 2010)</ref>

### Address uniqueness[[edit](/index.php?title=(none)&action=edit&section=18)]

Hosts verify the uniqueness of addresses assigned by sending a neighbor solicitation message asking for the Link Layer address of the IP address. If any other host is using that address, it responds. However, MAC addresses are designed to be unique on each network card which minimizes chances of duplication.[[25]](#cite_note-25) The host first determines if the network is connected to any routers at all, because if not, then all nodes are reachable using the link-local address that already is assigned to the host. The host will send out a Router Solicitation message to the all-routers<ref name=rfc4862sec551>RFC 4862, *IPv6 Stateless Address Autoconfiguration*, S.Thomson (September 2007), section 5.5.1: <http://tools.ietf.org/html/rfc4862#section-5.5.1</ref><ref> name=rfc4861sec637>RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*, T.Narten (September 2007), section 6.3.7: <http://tools.ietf.org/html/rfc4861#section-6.3.7</ref>> 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 from a router, there will be network information inside that is needed to create a globally unique address. There are also two flag bits that tell the host whether it should use DHCP to get further information and addresses:

* The Manage bit, that indicates whether or not the host should use DHCP to obtain additional addresses
* The Other bit, that 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:[[25]](#cite_note-25)\*\* 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 is the flag that tells the host to actually create a global address.

### Link local address[[edit](/index.php?title=(none)&action=edit&section=19)]

All interfaces of IPv6 hosts require a link-local address. A link-local address is derived from the MAC address of the interface and the prefix fe80::/10. The process involves filling the address space with prefix bits left-justified to the most-significant bit, and filling the MAC address in EUI-64 format into the least-significant bits. If any bits remain to be filled between the two parts, those are set to zero.[[24]](#cite_note-24) The uniqueness of the address on the subnet is tested with the [Duplicate Address Detection](/wiki/IPv6_address#Duplicate_address_detection) (DAD) method.[[26]](#cite_note-26) Many of these transition mechanisms use tunneling to encapsulate IPv6 traffic within IPv4 networks. This is an imperfect solution, which reduces the [maximum transmission unit](/wiki/Maximum_transmission_unit) (MTU) of a link and therefore complicates [Path MTU Discovery](/wiki/Path_MTU_Discovery), and may increase [latency](/wiki/Latency_(engineering)#Communication_latency).[[29]](#cite_note-29) [Tunneling protocols](/wiki/Tunneling_protocol) are a temporary solution for networks that do not support native dual-stack, where both IPv6 and IPv4 run independently.

### Dual IP stack implementation[[edit](/index.php?title=(none)&action=edit&section=23)]

Dual-stack (or *native dual-stack*) IP implementations provide complete IPv4 and IPv6 protocol stacks in the same network node. This facilitates native communications between nodes using either protocol. The method is defined in RFC 4213.[[30]](#cite_note-30) This is the most desirable IPv6 implementation during the transition from IPv4 to IPv6, as it avoids the complexities of tunneling, such as security, increased latency, management overhead, and a reduced [PMTU](/wiki/Maximum_transmission_unit#Path_MTU_Discovery).[[31]](#cite_note-31) However, it is not always possible, since outdated network equipment may not support IPv6.

Dual-stack software design is a transitional technique to facilitate the adoption and deployment of IPv6. However, it might introduce more security threats as hosts could be subject to attacks from both IPv4 and IPv6. It has been argued that dual-stack could ultimately overburden the global networking infrastructure by requiring routers to deal with IPv4 and IPv6 routing simultaneously.[[32]](#cite_note-32)

### Tunneling[[edit](/index.php?title=(none)&action=edit&section=24)]

Many current Internet users do not have IPv6 dual-stack support, and thus cannot reach IPv6 sites directly. Instead, they must use IPv4 infrastructure to carry IPv6 packets. This is done using a technique known as [*tunneling*](/wiki/Tunneling_protocol), which encapsulates IPv6 packets within IPv4, in effect using IPv4 as a link layer for IPv6.

IP protocol 41 indicates IPv4 packets which encapsulate IPv6 datagrams. Some routers or network address translation devices may block protocol 41. To pass through these devices, UDP packets may be used to encapsulate IPv6 datagrams. Other encapsulation schemes, such as [AYIYA](/wiki/AYIYA) or [Generic Routing Encapsulation](/wiki/Generic_Routing_Encapsulation), are also popular.

Conversely, on IPv6-only Internet links, when access to IPv4 network facilities is needed, tunneling of IPv4 over IPv6 protocol occurs, using the IPv6 as a link layer for IPv4.

#### Automatic tunneling[[edit](/index.php?title=(none)&action=edit&section=25)]

*Automatic tunneling* refers to a technique by which the routing infrastructure automatically determines the tunnel endpoints. Some automatic tunneling techniques are below.

[6to4](/wiki/6to4) is recommended by RFC 3056. It uses protocol 41 encapsulation.<ref name=rfc3056>RFC 3056, *Connection of IPv6 Domains via IPv4 Clouds*, B. Carpenter, February 2001.</ref> Tunnel endpoints are determined by using a well-known IPv4 anycast address on the remote side, and embedding IPv4 address information within IPv6 addresses on the local side. 6to4 is the most common tunnel protocol currently deployed.

[Teredo](/wiki/Teredo_tunneling) is an automatic tunneling technique that uses UDP encapsulation and can allegedly cross multiple NAT nodes.<ref name=rfc4380>RFC 4380, *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)*, C. Huitema, Februari 2006</ref> IPv6, including 6to4 and Teredo tunneling, are enabled by default in [Windows Vista](/wiki/Windows_Vista)<ref name=vista>[Template:Cite web](/wiki/Template:Cite_web)</ref> and [Windows 7](/wiki/Windows_7). Most Unix systems implement only 6to4, but Teredo can be provided by third-party software such as [Miredo](/wiki/Miredo).

[ISATAP](/wiki/ISATAP) (Intra-Site Automatic Tunnel Addressing Protocol)<ref name=rfc5214>RFC 5214, *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*, F. Templin, T. Gleeson, D. Thaler, March 2008.</ref> uses the IPv4 network as a virtual IPv6 local link, with mappings from each IPv4 address to a link-local IPv6 address. Unlike 6to4 and Teredo, which are *inter-site* tunneling mechanisms, ISATAP is an *intra-site* mechanism, meaning that it is designed to provide IPv6 connectivity between nodes within a single organization.

#### Configured and automated tunneling (6in4)[[edit](/index.php?title=(none)&action=edit&section=26)]

[6in4](/wiki/6in4) tunneling requires the tunnel endpoints to be explicitly configured, either by an administrator manually or the operating system's configuration mechanisms, or by an automatic service known as a [tunnel broker](/wiki/Tunnel_broker);<ref name=rfc3053>RFC 3053, *IPv6 Tunnel Broker*, A. Durand, P. Fasano, I. Guardini, D. Lento (January 2001)</ref> this is also referred to as *automated tunneling*. Configured tunneling is usually more deterministic and easier to debug than automatic tunneling, and is therefore recommended for large, well-administered networks. Automated tunneling provides a compromise between the ease of use of automatic tunneling and the deterministic behavior of configured tunneling.

Raw encapsulation of IPv6 packets using [IPv4](/wiki/Internet_Protocol) protocol number 41 is recommended for configured tunneling; this is sometimes known as [6in4](/wiki/6in4) tunneling. As with automatic tunneling, encapsulation within UDP may be used in order to cross NAT boxes and firewalls.

### Proxying and translation for IPv6-only hosts[[edit](/index.php?title=(none)&action=edit&section=27)]

After the [regional Internet registries](/wiki/Regional_Internet_registry) have exhausted their pools of available IPv4 addresses, it is likely that hosts newly added to the Internet might only have IPv6 connectivity. For these clients to have backward-compatible connectivity to existing IPv4-only resources, suitable [IPv6 transition mechanisms](/wiki/IPv6_transition_mechanisms) must be deployed.

One form of address translation is the use of a dual-stack application-layer [proxy server](/wiki/Proxy_server), for example a web proxy.

NAT-like techniques for application-agnostic translation at the lower layers in routers and gateways have been proposed. The NAT-PT standard was dropped because of criticisms;<ref name=rfc4966>RFC 4966, *Reasons to Move the Network Address Translator-Protocol Translator (NAT-PT) to Historic Status*</ref> however, more recently, the continued low adoption of IPv6 has prompted a new standardization effort of a technology called [NAT64](/wiki/IPv6_transition_mechanisms#NAT64).

## IPv6 readiness[[edit](/index.php?title=(none)&action=edit&section=28)]

Compatibility with IPv6 networking is mainly a software or firmware issue. However, much of the older hardware that could in principle be upgraded is likely to be replaced instead. The [American Registry for Internet Numbers](/wiki/American_Registry_for_Internet_Numbers) (ARIN) suggested that all Internet servers be prepared to serve IPv6-only clients by January 2012.[[33]](#cite_note-33)

### Software[[edit](/index.php?title=(none)&action=edit&section=29)]

Host software may have only IPv4 or only IPv6 networking software, or it may support dual-stack, or hybrid dual-stack operation. The majority of personal computers running recent [operating system](/wiki/Operating_system) versions support IPv6. Many popular applications with networking capabilities are compliant.

Some software transitioning mechanisms are outlined in RFC 4038, RFC 3493, and RFC 3542.

#### IPv4-mapped IPv6 addresses[[edit](/index.php?title=(none)&action=edit&section=30)]

Hybrid dual-stack IPv6/IPv4 implementations recognize a special class of addresses, the IPv4-mapped IPv6 addresses. These addresses consist of an 80-bit prefix of zeros, the next 16 bits are one, and the remaining, least-significant 32 bits contain the IPv4 address. 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](/wiki/Dot-decimal_notation) of IPv4. For example, ::ffff:192.0.2.128 represents the IPv4 address 192.0.2.128. A deprecated format for IPv4-compatible IPv6 addresses is ::192.0.2.128.[[34]](#cite_note-34) Because of the significant internal differences between IPv4 and IPv6, 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](/wiki/Microsoft_Windows) 2000, XP, and Server 2003), or because of security concerns ([OpenBSD](/wiki/OpenBSD)).<ref name=openbsd-mapped-addr>[Template:Cite web](/wiki/Template:Cite_web)</ref> 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](/wiki/Linux_kernel), [NetBSD](/wiki/NetBSD), and [FreeBSD](/wiki/FreeBSD), this feature is controlled by the socket option IPV6\_V6ONLY, as specified in RFC 3493.[[35]](#cite_note-35)

### Hardware and embedded systems[[edit](/index.php?title=(none)&action=edit&section=31)]

Basic infrastructure equipment, such as [network adapters](/wiki/Network_adapter) and [network switches](/wiki/Network_switch), may not be affected by the change, since they transmit link layer frames without inspecting the contents. Most equipment may be IPv6 capable with a software or firmware update if the device has sufficient storage and memory space for the new IPv6 stack.

In some cases, non-compliant equipment needs to be replaced because the manufacturer no longer exists or software updates are not possible, for example, because the network stack is implemented in permanent [read-only memory](/wiki/Read-only_memory).

The [CableLabs](/wiki/CableLabs) consortium published the 160 Mbit/s [DOCSIS](/wiki/DOCSIS) 3.0 IPv6-ready specification for [cable modems](/wiki/Cable_modem) in August 2006. The widely used DOCSIS 2.0 does not support IPv6. The new 'DOCSIS 2.0 + IPv6' standard supports IPv6, which may on the cable modem side require only a firmware upgrade.[[36]](#cite_note-36)[[37]](#cite_note-37) It is expected that only 60% of cable modems' servers and 40% of cable modems will be DOCSIS 3.0 by 2011.[[38]](#cite_note-38) However, most ISPs that support DOCSIS 3.0 do not support IPv6 across their networks.

Other equipment which is typically not IPv6-ready ranges from VoIP devices to laboratory equipment and printers.[Template:Citation needed](/wiki/Template:Citation_needed)

### Shadow networks[[edit](/index.php?title=(none)&action=edit&section=32)]

One side effect of IPv6 implementation may be the emergence of so-called *shadow networks* caused by IPv6 traffic flowing into IPv4 networks when IPv6 enabled nodes are added to the existing network, and the IPv4 security in place is unable to properly identify it. This may occur with operating system upgrades, when the newer OS enables IPv6 support by default, while the older one did not. Failing to update the security infrastructure to accommodate IPv6 can lead to IPv6 traffic bypassing it.[[39]](#cite_note-39) Shadow networks have been found occurring on business networks in which enterprises are replacing [Windows XP](/wiki/Windows_XP) systems that do not have an IPv6 stack enabled by default, with [Windows 7](/wiki/Windows_7) systems, that do.[[40]](#cite_note-40) Some IPv6 stack implementors have therefore recommended to disable IPv4 mapped addresses and to instead use a dual-stack network where supporting both IPv4 and IPv6 is necessary.[[41]](#cite_note-41)

## Deployment[[edit](/index.php?title=(none)&action=edit&section=33)]

[Template:Main](/wiki/Template:Main) The 1993 introduction of [Classless Inter-Domain Routing](/wiki/Classless_Inter-Domain_Routing) (CIDR) in the routing and IP address allocation for the Internet, and the extensive use of [network address translation](/wiki/Network_address_translation) (NAT) delayed [IPv4 address exhaustion](/wiki/IPv4_address_exhaustion). The final phase of exhaustion started on 3 February 2011.[[42]](#cite_note-42) However, despite a decade long development and implementation history as a Standards Track protocol, general worldwide deployment of IPv6 is increasing slowly. [Template:As of](/wiki/Template:As_of), about 4% of domain names and 16.2% of the networks on the Internet have IPv6 protocol support.<ref name=hestat>[Template:Cite web](/wiki/Template:Cite_web)</ref>

IPv6 has been implemented on all major operating systems in use in commercial, business, and home consumer environments. Since 2008, the [domain name system](/wiki/Domain_name_system) can be used in IPv6. IPv6 was first used in a major world event during the [2008 Summer Olympic Games](/wiki/2008_Summer_Olympic_Games),[[43]](#cite_note-43) the largest showcase of IPv6 technology since the inception of IPv6.[[44]](#cite_note-44) Some governments including the [Federal government of the United States](/wiki/Federal_government_of_the_United_States) and [China](/wiki/China) have issued guidelines and requirements for IPv6 capability.

In 2009, Verizon mandated IPv6 operation and deprecated IPv4 as an optional capability for cellular (LTE) hardware.[[45]](#cite_note-45) [Template:As of](/wiki/Template:As_of), T-Mobile USA also supports external IPv6 access.[[46]](#cite_note-46) As of 2014, IPv4 still carried more than 99% of worldwide [Internet traffic](/wiki/Internet_traffic).[[47]](#cite_note-47)[[48]](#cite_note-48) The Internet exchange in Amsterdam is the only large exchange that publicly shows IPv6 traffic statistics, which as of June 2016 is tracking at about 1.5%, growing at about 0.3% per year.[[49]](#cite_note-49) [Template:As of](/wiki/Template:As_of), the percentage of users reaching [Google](/wiki/Google) services with IPv6 reached 12.2% for the first time, growing at about 5.1% per year, although varying widely by region.[[50]](#cite_note-50) [Template:As of](/wiki/Template:As_of), deployment of IPv6 on web servers also varied widely, with over half of web pages available via IPv6 in many regions, with about 14% of web servers supporting IPv6.[[51]](#cite_note-51)

## See also[[edit](/index.php?title=(none)&action=edit&section=34)]

[Template:Portal](/wiki/Template:Portal)

* [China Next Generation Internet](/wiki/China_Next_Generation_Internet)
* [Comparison of IPv6 support in operating systems](/wiki/Comparison_of_IPv6_support_in_operating_systems)
* [Comparison of IPv6 support in common applications](/wiki/Comparison_of_IPv6_support_in_common_applications)
* [DoD IPv6 product certification](/wiki/DoD_IPv6_product_certification)
* [List of IPv6 tunnel brokers](/wiki/List_of_IPv6_tunnel_brokers)
* [University of New Hampshire InterOperability Laboratory](/wiki/University_of_New_Hampshire_InterOperability_Laboratory)

## References[[edit](/index.php?title=(none)&action=edit&section=35)]

[Template:Reflist](/wiki/Template:Reflist)

## External links[[edit](/index.php?title=(none)&action=edit&section=36)]

[Template:Wikiversity](/wiki/Template:Wikiversity)

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

[Template:IPv6](/wiki/Template:IPv6)

[Template:Authority control](/wiki/Template:Authority_control)

[Category:IPv6](/wiki/Category:IPv6) [Category:Internet layer protocols](/wiki/Category:Internet_layer_protocols) [Category:Internet Protocol](/wiki/Category:Internet_Protocol) [Category:Network layer protocols](/wiki/Category:Network_layer_protocols) [Category:1996 introductions](/wiki/Category:1996_introductions)