

A Quick Overview of IPv6

To understand what IPv6 is and what it is not, what features to look out for, and how it fits into the TCP/IP stack, this chapter provides a rough overview.

1.1 Terminology: IP, IPv4, IPv6 and the Internet

When we talk about “traditional IP” from now on, we use the term *IPv4*, which is short for *Internet protocol, version 4* as of RFC 791 [32] and related documents.

Its successor protocol is called *IPv6*, or *Internet protocol, version 6*. It is defined in RFC 2460 [24] and related standards.

Whenever we talk about *IP*, from now on we talk about the “Internet Protocol” family in general. This includes all network layer protocols from the TCP/IP stack, as explained later on in section 1.3: IPv4, IPv6 and any future successor to both.

On a similar line, when we talk about the *Internet*, we talk about the global network connected using IP. The *Internet4* is the part of the Internet that uses IPv4 and the *Internet6* is the part that uses IPv6. The Internet4 and Internet6 are not strictly disjoint, but this distinction is very helpful when we address the issues concerned with the interoperation of both.

Finally there are *protocol families* or *address families* that denote an entire family of protocols using the same addressing scheme. The *INET* address family includes IPv4 as well as all protocols running on top of IPv4, like TCP or UDP over IPv4. Similarly, the *INET6* protocol family includes IPv6 and all other protocols using IPv6 addresses or running on top of IPv6.

1.2 The “IPv6 Sales Pitch”

What are the differences that make IPv6 superior to IPv4? The most visible differences fall into two categories: Changes that solve fundamental inade-

quacies of traditional IPv4 and new features that were first introduced with IPv6.

The features resolving fundamental problems with IPv4 that made a redesign necessary include these:

Larger address space Probably the most essential advantage of IPv6 over IPv4 is its enlarged address space. While IPv4 addresses are 32 bits long, IPv6 uses 128 bit addresses. These long addresses resolve the address scarcity issues getting more severe every day.

Abolition of NAT With IPv6 there is no need to connect multiple machines to the Internet using a single address and *network address translation* (NAT). Without NAT, *end-to-end connectivity* becomes available again, allowing machines to connect to each other without intermediate “broker” services, like mail exchangers/relays, web proxies, DNS forwarders or SIP gatekeepers, that are run by a service provider.

At first glance this doesn’t seem like much of an advantage, but at this time its consequences are barely fathomable, making services possible that are difficult even to imagine to our NAT-conditioned minds.

Simplified address structure With the large address space there is no more need for configurable network masks, thus simplifying network configuration and disposing of an ever annoying source of misconfiguration.

Simplified address configuration The large address space allows for a simplified address configuration mechanism, providing a service similar to the dynamic host configuration protocol (DHCP) but avoiding the need to maintain state information about address leases.

Replacing DHCP with a minimum-configuration, stateless mechanism simplifies network configuration even more and eliminates another common cause of network problems.

Simplified address renumbering With the address configuration mechanism it is perfectly feasible to change addresses throughout an entire network during normal operations without touching or even rebooting any machine connected.

IPv4 network renumberings put a network temporarily down and require a serious effort, thus making network reorganizations expensive and risky. This problem ties many customers to their Internet service providers (ISPs). With IPv6 it is feasible to reorganize networks or switch ISPs without disruption of network services.

Improved multicast The multicast address range has been vastly extended, making use of a wide range of “scopes” that define the domain within which an address is used. Multicasts as well as multicast routing are base features of IPv6.

Routed multicasts are a functionality necessary to build “self-configuring” network services and more efficient “intelligent broadcast” services like “Internet Radio”, among other things.

Abolition of broadcast With the extended multicast functionality IPv6 doesn’t have any further need for IPv4-style broadcasts.

This makes IPv6 invulnerable to attacks that use remote broadcasts such as “ping bounce” or “smurf” denial of service attacks, while it still supports all the “reasonable” features that IPv4 broadcasts are used for. As another advantage over broadcasts, multicasts are only processed on those nodes which have actively signalled that they are interested in the particular multicast group. This reduces the load on all other machines.

Streamlined routing tables With IPv4, address ranges were assigned in an ad-hoc style and for unlimited time. Medium to large organizations obtained *provider-independent addresses* (*PI addresses*) and then connected through one or several ISPs, leading to an excessive growth of routing table entries in the “backbone” routers at the top network service providers. With IPv4 addresses becoming ever more precious and renumberings being virtually infeasible these organizations refuse to release these addresses they hold.

IPv6 doesn’t provide PI addresses, it makes renumberings easy and far less risky, it only assigns addresses on a non-permanent basis and provides such an abundance of addresses that hoarding them doesn’t make sense. As a consequence, routing tables in the core routers are several orders of growth shorter with IPv6 than with IPv4; and even when the Internet6 grows, the routing tables will mostly stay at their current size.

All these features are deeply incorporated into the IPv6 design, making them readily available.

In addition, some more advanced features were standardized that don’t solve a problem with existing IPv4 but implement new functionalities:

Network traffic security with IPsec The standards expect a full implementation of IPv6 to include network layer encryption and authentication using IPsec as a mandatory feature. Among other advantages of fully integrated network traffic encryption this provides the means to encrypt traffic even within a local network, thus providing protection from insiders trying to sniff network traffic.

IPsec has been backported to IPv4 as an optional feature with little or no loss of functionality. More or less usable implementations are available though the key exchange protocols still show interoperation problems.

While Microsoft Windows XP (SP2) currently limits itself to the “NULL” encryption algorithm, other implementations do provide strong end-to-end encryption.

Mobile IPv6 The IPv6 standards include a feature called “Mobile IPv6”. This allows “roaming” while maintaining a “home” network address at all times, keeping all existing network connections open even while the underlying network connectivity changes. While Mobile IPv6 has a number of mind-boggling security implications, “roaming” provides the base technology for a wide range of mobile applications.

The standards for mobile IPv6 have been released fairly late. Implementations are based on preliminary drafts of the standards and should be considered experimental.

IPv4 offers a similar optional feature; it has been added to IPv4 only lately though, severely restricting its functionality compared to IPv6.

Quality of service (QoS) support Several standards addressing quality of service have been released that specify how near-realtime functionality can be incorporated into IPv6. While quality of service is still an emerging technology, near-realtime applications like IP telephony may well make good use of this feature.

Implementations are not yet readily available; with the political issues involved it remains questionable if end-to-end quality of service support will ever become generally available.

The near-realtime features defined for IPv6 haven't been backported to IPv4 and it is unlikely they will ever be.

IPsec may be considered the most mature of these features, but even IPsec isn't fully usable in a production environment. Certificate-based authentication and multicast support are still missing from implementations.

Even though mobile IPv6 and quality of service are very exciting—and scary in the case of mobile IPv6—they are neither essential to the setup and operation of IPv6 nor are they stable enough to be used in a production environment yet.

1.3 IPv6 and the TCP/IP Stack

What exactly is IPv6? You may have a reasonable idea of what the “standard” TCP/IP stack looks like. Maybe you've read the standard “TCP/IP Illustrated” by the late W. Richard Stevens [103], or any other of the wide

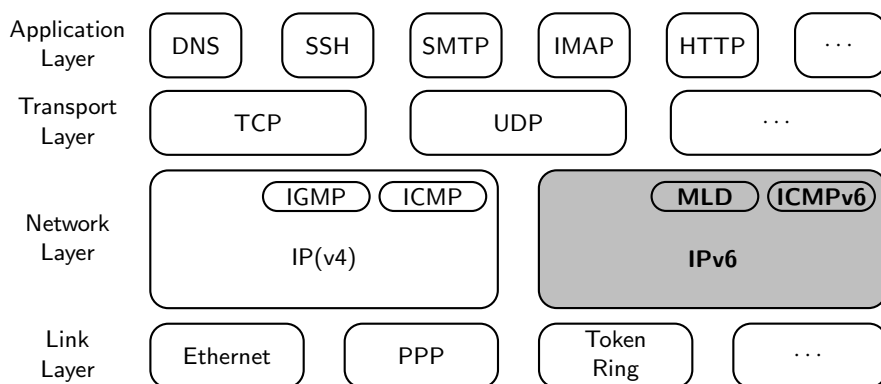


Fig. 1.1. IPv6 and its role in the traditional TCP/IP stack

range of introductory books on TCP/IP. So except for the highlighted IPv6 part, figure 1.1 may look reasonably familiar to you. If you have never seen it, this is how it works: The network stack is organized in four different layers, communicating only with the layers immediately above and below them (except in one case we'll see below). Every layer provides a specific functionality to the layers above:

Link Layer The link layer transmits data packets, called *frames*, between devices directly connected to the same physical network. The archetypical link layer is *Ethernet* in one of its many physical implementations.

Network Layer Devices connected to different physical networks can communicate through the network layer. An IP packet is sent from one device to another by being wrapped up in a link-layer frame and then being sent either to the recipient if it is connected to the same physical network, or to an intermediate device called a “router”. A router that receives a frame first unpacks the IP packet within. If the packet is not addressed to the router itself it decides where to forward the packet to—either another router or the destination device. It re-wraps the packet in another link-layer frame and sends it out the the next link-layer destination. Eventually the packet arrives at its destination.

Transport Layer While the network layer only addresses devices, like computers, the transport layer adds *port numbers* to its communication to help the destination device pass the communication to a particular process. There are two major transport layer protocols: The *transmission control protocol (TCP)* implements a virtual connection, taking care of the re-transmission of lost or damaged network layer packets and the ordering of packets. The *User Datagram Protocol (UDP)* simply sends individual packets, called *datagrams* to a destination process but doesn't provide for a connection or the handling of lost packets.

Application Layer Applications use the transport layer to implement communication between processes on different computers to provide a specific functionality. Applications access the network layer directly when they deal with IP addresses, usually when they try to address their communication peers; this is the one exception to the rule that any layer only communicates with the layers immediately above or below.

A somewhat unusual application layer protocol is the *domain name system (DNS)*. It provides a translation service turning a host name like `www.example.com` into an IP address and vice versa. Virtually all application programs use this service, so from an application developer's point of view the DNS conceptually belongs to the transport or network layer even though the protocol definition puts it in the application layer.

So how does IPv6 fit in? The figure already explains two essential properties of IPv6.

First of all, IPv6 is a network layer protocol; it doesn't interfere with the transport layer. You may sometimes read about “TCPv6”, which doesn't

really exist; usually this means “TCP over IPv6”. More important, since most application software uses the transport layer interface most of the time, it is usually fairly straightforward to make IPv4 applications support IPv6. The majority of work involved deals with the (usually minor) tweaks necessary to support the larger addresses whenever the application needs to deal with addresses directly.

Next, IPv6 runs in parallel with IPv4 even to the point that they “share” a single interface. Legacy systems that need IPv4 continue to work even when IPv6 is enabled; they just require the extra administration effort to maintain them. There won’t be a “great switchover” on a fixed “flag day” that needs to be organized all over the world. Instead, the core strategy to deploy IPv6 in any existing environment is a soft migration, introducing IPv6 in small, easily reversible steps.

<http://www.springer.com/978-3-540-24524-7>

IPv6 in Practice

A Unixer's Guide to the Next Generation Internet

Stoekelbrand, B.

2007, XXII, 390 p. 50 illus., Hardcover

ISBN: 978-3-540-24524-7