

1. Introduction

What is IPv6?

When we read email or browse a website we're really seeing a host of electrical signals broken into packets by the sending computer, transported in milliseconds through networks across the world, and re-assembled perfectly at our end.

This magic only succeeds if the sender, receiver, and everyone in between agrees on the format of those packets and how to handle them. To navigate through the networks every packet needs a header with sender and destination addresses, and the blueprint for that header is called the Internet Protocol.

The old version of the Internet Protocol, IPv4, has been the foundation of the Internet's success for over thirty years, but the total number of addresses IPv4 can use has a hard limit, already surpassed by today's massive diversity of connected devices.

IPv6 (Internet Protocol version 6) is a vastly improved addressing system for Internet-connected devices. It is not only the glue for an almost inconceivably larger Internet, but helps build safer and more resilient network infrastructure, which in turn drives efficiency, economy and innovation for both business and society.

IPv6 adoption world-wide is essential so that the Internet may continue to expand and innovate into the future. This book provides a quick reference to the technical definitions of IPv6.

2. Six Benefits of IPv6

Almost Unlimited Address Abundance

IPv6 has 3.4×10^{38} possible IPv6 addresses = 340 trillion trillion trillion – about 670 quadrillion addresses per square millimetre of the Earth's surface. (IPv4 has only 4.29×10^9 addresses = 4.3 billion – far less than even a single IP address per person on the planet.)

Easier and Cheaper Network Management

IPv6 networks have simpler, flatter and more manageable architectures, including auto-configuration capabilities. Greater simplicity and manageability lead to greater reliability, security and economy.

Return of End-to-End Connectivity

IPv6's vast address space means direct peer-to-peer addressing, which is better for performance, security and troubleshooting, and removes the need for stopgap NAT techniques.

Mandated Security Features

IPSec support is mandatory in IPv6, providing authentication and encryption capabilities for use with a suitable key infrastructure. (In IPv4, IPSec support was an optional feature.)

Integrated Mobility and Interoperability

Interoperability and mobility capabilities are improved in IPv6 and are already widely embedded in network devices. (In IPv4, constraints from network topologies limit such capabilities.)

A Platform for Innovation and Scaleability

Huge size, together with scalability and flexibility of IPv6 networks, fosters innovation, collaboration, streamlined processes and massive-scale real-time reporting of environmental or business conditions.

3. IPv6 Addresses

IPv6 addresses are written in the hexadecimal number system. Below, from 0 to 31 in binary (machine) format, decimal (IPv4) and hexadecimal:

Binary	Decimal	Hex	Binary	Decimal	Hex
0000	0	0	0001 0000	16	10
0001	1	1	0001 0001	17	11
0010	2	2	0001 0010	18	12
0011	3	3	0001 0011	19	13
0100	4	4	0001 0100	20	14
0101	5	5	0001 0101	21	15
0110	6	6	0001 0110	22	16
0111	7	7	0001 0111	23	17
1000	8	8	0001 1000	24	18
1001	9	9	0001 1001	25	19
1010	10	a	0001 1010	26	1a
1011	11	b	0001 1011	27	1b
1100	12	c	0001 1100	28	1c
1101	13	d	0001 1101	29	1d
1110	14	e	0001 1110	30	1e
1111	15	f	0001 1111	31	1f

Bit (binary digit) = 0 or 1

Nibble = 4 bits, e.g. 1011

Byte = 8 bits, 1 octet, e.g. 00010110

Word = 32 bits, 4 bytes

IPv4 Addresses: 32 bits in 4 bytes

32 bits in 4 binary bytes = 11000000 10101000 00000001 00000000

Same IPv4 address in decimal format = 192 . 168 . 1 . 0

Maximum number of IPv4 addresses possible: **4,294,967,296**

IPv6 Addresses: 128 bits in 16 bytes

128 bits in 16 binary bytes = 00100000 00000001 00001101 10111000
00000000 00000000 00000000 00000000 00010010 00110100 00000000
00000000 00000000 00000000 00000000 00000001

Same IPv6 address in hexadecimal format = 2001:db8:0:0:1234:0:0:1

Maximum number of IPv6 addresses possible:

340,282,366,920,938,463,463,374,607,431,768,211,456

Easy IPv6: The Lookup Book

10. IPv6 Transition

Today: if IPv6 is available on end-user systems, a variety of transport techniques over IPv4 can be used to access the IPv6 Internet.

Transition: the ideal is full dual-stacking (IPv4 and IPv6 on all devices, and OSs and applications choose preferred protocol), but in practice a mixture of techniques will probably remain during transition.

Future: IPv6 will be the standard everywhere, transport techniques will deal with legacy IPv4 systems as they are phased out.

Dual-Stacking

Dual-stacking means that hosts and servers have both IPv4 and IPv6 protocol capabilities. It is an essential transition method, widely supported, allows staged deployment of IPv6 infrastructure.

It requires an audit and overhaul of logical and physical network elements, e.g. routing, peering, websites, databases, operations support, user access, authentication, accounting, customer premises equipment, etc. See RFC 4213. Drawbacks include:

- Need to manage and troubleshoot two logically distinct protocols.
- Increased CPU and memory demands for two stacks on devices.
- May need different firewall rules for IPv4 and IPv6.
- Dual-stacking still requires a pool of IPv4 addresses.
- Costly in time/staff/resources for large or complex networks.

Transport Techniques

IPv6 and IPv4 are incompatible, although they can coexist on the same physical infrastructure. An IPv4 network cannot deliver IPv6 packets, or vice versa, unless they are either *translated* or *encapsulated*.

Translation

Translation is the stateful mapping of one kind of address protocol to another. Currently widely used in NAT (network address translation) devices to map between public IPv4 addresses and private IPv4 space, translation can also map between IPv6 and IPv4 addresses, e.g. NAT64. Problems with NAT techniques include:

- Loss of end-to-end transparency
- Difficulties with security protocols like IPSec
- Single point of failure of stateful devices
- Complexity, scalability, performance issues
- Still require a pool of IPv4 addresses

Encapsulation

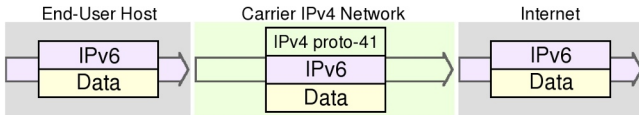
Packets of one protocol are wrapped (encapsulated) inside packets of another protocol for delivery, and unwrapped (decapsulated) at the other end. The process is also called **tunnelling**, and is used widely in telecommunications and IPv6 transition. In the case of IPv6, packets are encapsulated within IPv4 packets (protocol 41) or IPv4-UDP packets (protocol 17), and sent across the IPv4 Internet.

Encapsulation is often used with address prefixing – the construction of an IPv6 address from an IPv6 prefix and the value of an IPv4 address, often with special prefixes to denote specific protocols. See RFC 4213.

11. End-User IPv6 Tunnels

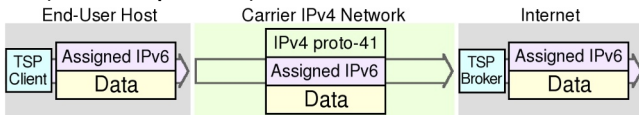
End-users may want IPv6 but their ISP/carrier has not implemented it: hence techniques for bypassing IPv4-only carriers. Techniques have limitations, e.g. *configured tunnels* (6in4, TSP) need high levels of expertise and infrastructure control. *Automatic tunnels* (6to4, Teredo), enabled by default in some Microsoft operating systems, have high failure rates due to DNS shortcomings and dependence on often-blocked protocols. The major end-user techniques are:

6in4



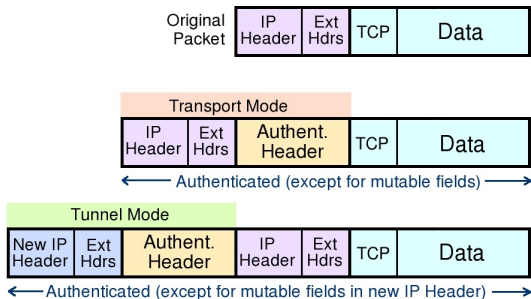
Configured tunnelling: IPv6 packets are wrapped inside IPv4 with protocol 41, then at the tunnel endpoint the IPv4 header is stripped and the packet sent to the IPv6 destination. Requires user to configure static address endpoints. See RFC 4213.

TSP (Tunnel Setup Protocol)



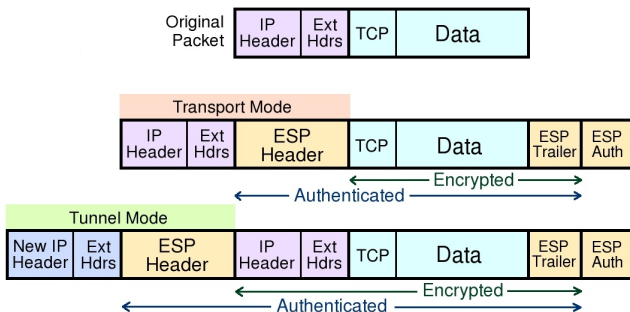
Configured tunnelling: Tunnel Setup Protocol – client software makes request to a TSP tunnel broker device to assign IPv6 address. Uses 6in4 encapsulation, relies on 3rd-party tunnel broker. See RFC 5572.

Applying IPv6 AH in Transport and Tunnel Modes



AH authenticates the packet and the outermost IPv6 addresses (except for mutable fields), but does not encrypt payloads. AH cannot be used to traverse NATs as it calculates the integrity check value over source and destination addresses: NATs translate addresses, so would invalidate ICVs.

Applying IPv6 ESP in Transport and Tunnel Modes



ESP authentication does not include the outermost IPv6 headers, but in Tunnel mode it protects the original headers. ESP is used to build virtual private network tunnels between sites. It permits NAT traversal, as it does not use the outermost address values in the ICV calculation. If AH and ESP are used together, ESP is applied first, then AH authenticates the entire new packet.

16. IPv6 Routing Security

Neighbor Discovery Protocol

NDP is new and integral to IPv6, see RFC 4861. It defines five main ICMPv6 packet types which do inverse neighbor discovery: Router Solicitation, Router Advertisement, Neighbor Solicitation, Neighbor Advertisement, and Redirect, see RFC 3122. NDP provides:

- Router discovery: hosts can locate routers on attached links.
- Prefix discovery: hosts can discover prefixes on-link for attached links.
- Parameter discovery: hosts can find link parameters (e.g. MTU).
- Address autoconfiguration: stateless configuration of network interfaces.
- Address resolution: mapping between IP and link-layer addresses.
- Next-hop determination: hosts find next-hop routers for a destination.
- Neighbor unreachability detection (NUD).
- Duplicate address detection (DAD).
- Redirect: router can inform a node about better first-hop routers.
- Recursive DNS Server (RDNS) and DNS Search List (DNSSL) assignment via router advertisement (RA) options.

NDP has weaknesses – if spoofed, it can redirect to a non-existent router, redirect traffic to wrong hosts, allow denial of service from attackers on-link, advertise a non-existent router, advertise but not route, deprecate prefixes, block autoconfiguration by spoofing DAD responses, spoof 'neighbour unreachable' responses. RFC 4861 discusses NDP security issues, advises Secure Neighbor Discovery (SEND).

Secure Neighbor Discovery

SEND secures NDP – fairly heavy-duty so most useful in high-value networks, see RFC 3971. Works in similar way to SSL – certificates held, verifiable via a trust anchor. Adds NDP messages to handle certificate checks. Uses CGA to secure endpoints.

Cryptographically Generated Addresses

CGA was developed for SEND, see RFC 3972. For IPv6 addresses, interface identifier is generated by computing a cryptographic one-way hash function from public key and auxiliary parameters. CGAs are not certified, so attackers can create new CGAs from any subnet prefix and public key but can't forge a CGA created by someone else. Protection works without a certification authority or any security infrastructure – only need an address and an algorithm, so also useful as a privacy tool.

18. Useful Information

Firefox Plugins that Show IPv6 Addresses

ShowIP displays IP address of the current webpage in the bottom corner.

ExternalIP shows the IP address the world sees you coming from.

Unix Tools for IPv6

ip -6 – show/manipulate routing, devices, tunnels etc

ifconfig -a – to see all network interfaces on a host

ping6 – host and network reachability

traceroute6 – traces route to a host

tracepath6 – traces route to host discovering MTU along path

route -6 – the current routing table

netstat – network connections, routing tables, interface statistics etc.

tcpdump – shows packet contents on network interfaces

wireshark – GUI network protocol analyzer

See **man** pages for command options.

Windows Tools for IPv6

Options below in square brackets. Use **/?** to get a list of usage options.

ipconfig /all

ipconfig /flushdns

route [print add delete]

ping [-4 -6 -i -R -S]

tracert [-4 -6 -R -S]

pathping [-4 -6]

netstat

netsh interface ipv6 show [interface address route

neighbors destination cache]

Address Testing

Check for IPv6 and IPv4 connectivity: <http://www.ipv6-test.com>

Looking glasses: publicly accessible servers for routing queries, used to troubleshoot routing issues. List of IPv6-capable looking glasses:

<http://www.bgp4.as/looking-glasses>

IPv6 Addresses in URLs

To use a website's IPv6 address rather than its URL, the address must be enclosed in square brackets, e.g., [http://\[2001:db8::2e:0:7348\]](http://[2001:db8::2e:0:7348])

Port numbers go outside the brackets – [http://\[2001:db8::2e:a0:57ab\]:80](http://[2001:db8::2e:a0:57ab]:80)