CHAPTER 7

LAYER 3 CONNECTIVITY: IPv6 TECHNOLOGIES FOR THE IoT

7.1 OVERVIEW AND MOTIVATIONS

Internet Protocol Version 6 (IPv6) is a newer version of the network layer protocol that is designed to coexist (but not directly interwork) with IPv4. In the long term, IPv6 is expected to replace IPv4, but that will not happen overnight. IPv6 provides improved internetworking capabilities compared to what is presently available with IPv4. The current IPv4 has been in use for over 30 years, but it exhibits some challenges in supporting emerging demands for address space cardinality, high-density mobility, multimedia, and strong security. IPv6 offers the potential of achieving scalability, reacheability, end-to-end interworking, quality of service (QoS), and commercial-grade robustness that is needed for contemporary and emerging web services, data services, mobile video, and Internet of things (IoT) applications.

We retain the position stated in Chapter 1 that *IoT may well become the "killer-app" for IPv6.* Using IPv6 with its abundant address spaces, globally unique object (thing) identification and connectivity can be provided in a standardized manner without additional status or address (re)processing; hence, its intrinsic advantage over IPv4 or other schemes. We are not implying in this text that IPv6 is strictly and uniquely required to support IoT, just that it provides an ideal, future-proof, scalable mechanism for such services, whether in a terrestrial mode or in a satellite-based mode (1, 2).

IP was designed as a packet-based technology (protocol) in the late 1970s–early 1980s for the purpose of connecting computers that were in separate geographic locations. Starting in the early 1990s, developers realized that the communication needs of the twenty-first century needed a protocol with some new features and capabilities, while at the same time retaining the useful features of the existing protocol. IPv6 was initially developed in the early 1990s because of the anticipated need for more end-system addresses based on anticipated Internet growth, encompassing mobile phone deployment, smart home appliances, and billions of new users in developing countries (e.g., BRIC: Brazil, Russia, India, China). Technologies and applications such as voice over IP (VoIP), "always-on access" (e.g., cable modems), broadband and/or ethernet-to-the-home, converged networks, evolving ubiquitous computing applications, and IoT will be driving this need even more in the next few years (3).

IPv6 is now being slowly deployed worldwide: there is documented institutional and commercial interest and activity in Europe and Asia, and there also is evolving interest in the United States. The expectation is that in the next few years deployment of this new protocol will occur worldwide. For example, the U.S. Department of Defense (DoD) announced that from October 1, 2003, all new developments and procurements needed to be IPv6 capable; the DoD's goal was to complete the transition to IPv6 for all intra- and internetworking across the agency by 2008, which was accomplished. The U.S. Government Accountability Office (GAO) has recommended that all agencies become proactive in planning a coherent transition to IPv6. The current expectation is that IPv4 will continue to exist for the foreseeable future, while IPv6 will be used for new broad-scale applications. The two protocols are not directly interworkable, but tunneling and dual-stack techniques allow coexistence and co-working.

While the basic function of the network layer internetworking protocol is to move information across networks, IPv6 has more capabilities built into its foundation than IPv4. Link-level communication does not generally require a node identifier (address) since the device is intrinsically identified with the link-level address; however, communication over a group of links (a network) does require unique node identifiers (addresses). The IP address is an identifier that is applied to each device connected to an IP network. In this setup, different entities taking part in the network (servers, routers, user computers, and so on) communicate among each other using their IP address, as an entity identifier. The current IPv4 naming scheme was developed in the 1970s and had capacity for about 4.3 billion addresses, which were grouped into 255 blocks of 16 million addresses each. In IPv4, addresses consist of four octets. With IPv4, the 32-bit address can be represented as **AdrClass|netID|hostID**. The network portion can contain either a network ID or a network ID and a subnet. Every network and every host or device has a unique address, by definition. For ease of human conversation, IP addresses are represented as separated by periods, for example: 166.74.110.83, where the decimal numbers are a shorthand and corresponds to the binary code described by the byte in question (an 8-bit number takes a value in the 0–255 range). Since the IPv4 address has 32 bits, there are nominally 2^{32} different IP addresses (as noted, approximately 4.3 billion nodes, if all combinations are used).

TABLE 7.1 Projected RIR Unallocated Address Pool Exhaustion (as of April 2011)

RIR	Assigned Addresses (/8s)	Remaining Addresses (/8s)
AFRINIC	8.3793	4.6168
APNIC	53.7907	1.2093
ARIN	77.9127	6.0130
LACNIC	15.6426	4.3574
RIPE NCC	45.0651	3.9349

RIR, regional Internet registry; AFRINIC, African Network Information Centre; ARIN, American Registry for Internet Numbers; APNIC, Asia-Pacific Network Information Centre; LACNIC, Latin America and Caribbean Network Information Centre; RIPE NCC, Réseaux IP Européens Network Coordination Centre (the RIR for Europe, the Middle East, and parts of Central Asia).

IPv4 has proven, by means of its long life, to be a flexible and powerful networking mechanism. However, IPv4 is starting to exhibit limitations, not only with respect to the need for an increase of the IP address space, driven, for example, by new populations of users in countries such as China and India; by new technologies with "always connected devices" (e.g., cable modems, networked PDAs, 3G/4G mobile smartphones, and so on); and by new services such as global rollout of VoIP, IP Television (IPTV), and social networking. A full deployment of IoT applications will certainly stress the IPv4 environment. A regional Internet registry (RIR) manages the allocation and registration of Internet resources such as IPv4 addresses, IPv6 addresses, and autonomous system (AS) numbers, in a specific region of the world. As of February 1, 2011, only 1% of all possible IPv4 addresses were left unassigned. This has led to a predicament known as *IPv4 Run-Out*. The entire address space was expected to be more or less exhausted by September 2011, according to the IPv4 Address Report (see Table 7.1) (4, 5). The IPv4 address allocation is based on the following hierarchy:

Internet assigned numbers authority (IANA) $\rightarrow$ RIRs $\rightarrow$ internet service providers (ISPs) $\rightarrow$ the public (including businesses).

Thus, a key desirable capability is the increase in address space such that it is able to cover all elements of the universe set under consideration. For example, all computing devices could have a public IP address, so that they can be uniquely tracked[1]; today inventory management of dispersed IT assets cannot be achieved with IP mechanisms alone. With IPv6, one can use the network to verify that such equipment is deployed in place and active; even non-IT equipment in the field can be tracked by having an IP address permanently assigned to it. IPv6 creates a new IP address format, such that the number of IP addresses will not exhaust for several decades or longer, even though an entire new crop of devices are expected to connect to Internet over the coming years.

[1]Note that this has some potential negative security issues as attackers could be able to own a machine and then exactly know how to go back to that same machine again. Therefore, reliable security mechanisms need to be understood and put in place in IPv6 environments.

IPv6 also adds improvements in areas such as routing and network configuration. IPv6 has extensive automatic configuration (autoconfiguration) mechanisms and reduces the IT burden, making configuration essentially plug-and-play. Specifically, new devices that connect to intranet or Internet will be “plug-and-play” devices. With IPv6, one is not required to configure dynamic non-published local IP addresses, the gateway address, the subnetwork mask, or any other parameters. The equipment automatically obtains all requisite configuration data when it connects to the network. Autoconfiguration implies that a dynamic host configuration protocol (DHCP) server is not needed and/or does not have to be configured (2, 6, 7).

IPv6 was originally defined in RFC 1883 that was then obsolete by RFC 2460, “Internet Protocol, Version 6 (IPv6) Specification,” S. Deering, R. Hinden (December 1998).[2] A large body of additional RFCs has emerged in recent years to add capabilities and refine the concept.

The advantages of IPv6, some of which we already noted in Chapter 1, can be summarized as follows:

- Scalability and expanded addressing capabilities: IPv6 has 128-bit addresses versus 32-bit IPv4 addresses. With IPv4, the theoretical number of available IP addresses is $2^{32} \sim 10^{10}$. IPv6 offers a 2^{128} space. Hence, the number of available unique node addressees is $2^{128} \sim 10^{39}$. IPv6 has more than 340 undecillion (340,282,366,920,938,463,463,374,607,431,768,211,456) addresses, grouped into blocks of 18 quintillion addresses.
- “Plug-and-play”: IPv6 includes a “plug-and-play” mechanism that facilitates the connection of equipment to the network. The requisite configuration is automatic; it is a serverless mechanism.
- IPv6 makes it easy for nodes to have multiple IPv6 addresses on the same network interface. This can create the opportunity for users to establish overlay or communities of interest (COI) networks on top of other physical IPv6 networks. Department, groups, or other users and resources can belong to one or more COIs, where each can have its own specific security policy (8).
- Security: IPv6 includes security in its specifications such as payload encryption and authentication of the source of the communication. It calls for end-to-end security, with built-in, strong IP-layer encryption and authentication (embedded security support with mandatory IP Security [IPsec] implementation). It follows that IPv6 network architectures can easily adapt to an end-to-end security model where the end hosts have the responsibility of providing the security services necessary to protect any data traffic between them; this results in greater flexibility for creating policy-based trust domains that are based on varying parameters including node address and application (9).
- In IPv6, creating a Virtual Private Network (VPN) is easier and more standard than in IPv4, because of the (authentication header [AH] and encapsulating

[2]The “version 5” reference was employed for another use—an experimental real-time streaming protocol—and to avoid any confusion, it was decided not to use this nomenclature.

security protocol [ESP]) extension headers. The performance penalty is lower for the VPN implemented in IPv6 compared to those built in IPv4 (10).

- Optimized protocol: IPv6 embodies IPv4 best practices but removes unused or obsolete IPv4 characteristics. This results in a better-optimized IP. Also, merging two IPv4 networks with overlapping addresses (say, if two organizations merge) is complex; it will be much easier to merge networks with IPv6.
- Real-time applications: To provide better support for real-time traffic (e.g., VoIP, IPTV), IPv6 includes "labeled flows" in its specifications. By means of this mechanism, routers can recognize the end-to-end flow to which transmitted packets belong. This is similar to the service offered by multiprotocol label switching (MPLS), but it is intrinsic with the IP mechanism rather than an add-on. Also, it preceded this MPLS feature by a number of years.
- Mobility: IPv6 includes more efficient and robust mobility mechanisms (enhanced support for mobile IP, mobile computing devices, and mobile video). Specifically, mobile IPv6 (MIPv6) as defined in RFC 3775 is now starting to be deployed (11).
- Streamlined header format and flow identification.
- Extensibility: IPv6 has been designed to be extensible and offers support for new options and extensions.

ISPs and carriers have been preparing for IP-address exhaustion for a number of years, and there are transition plans in place. The expectation is that IPv6 can make IP devices less expensive, more powerful, and even consume less power; the power issue is not only important for environmental reasons, but also improves operability (e.g., longer battery life in portable devices, such as mobile phones).

7.2 ADDRESS CAPABILITIES

7.2.1 IPv4 Addressing and Issues

IPv4 addresses can be from an officially assigned public range or from an internal intranet private (but not globally unique) block. As noted, IPv4 theoretically allows up to 2^{32} addresses, based on a four-octet address space. Hence, there are 4,294,967,296 unique values, which can be considered as a sequence of 256 "/8s," where each "/8" corresponds to 16,777,216 unique address values. Public, globally unique addresses are assigned by IANA. IP addresses are addresses of network nodes at layer 3; each device on a network (whether the Internet or an intranet) must have a unique address. In IPv4, it is a 32-bit (4-byte) binary address used to identify a host's network ID. It is represented by the nomenclature a.b.c.d (each of a, b, c, and d being from 1 to 255) (0 has a special meaning). Examples are 167.168.169.170, 232.233.229.209, and 200.100.200.100.

The problem is that during the 1980s, many public, registered addresses were allocated to firms and organizations without any consistent control. As a result, some organizations have more addresses that they actually might need, giving rise to the present dearth of available "registerable" layer 3 addresses. Furthermore, not all IP addresses can be used due to the fragmentation described above.

One approach to the issue would be a renumbering and a reallocation of the IPv4 addressing space. However, this is not as simple as it appears since it requires worldwide coordination efforts. Moreover, it would still be limited for the human population and the quantity of devices that will be connected to Internet in the medium-term future. At this juncture, and as a temporary and pragmatic approach to alleviate the dearth of addresses, network address translation (NAT) mechanisms are employed by organizations and even home users. This mechanism consists of using only a small set of public IPv4 addresses for an entire network to access the Internet. The myriad of internal devices are assigned IP addresses from a specifically designated range of Class A or Class C address that are locally unique but are duplicatively used and reused within various organizations. In some cases (e.g., residential Internet access use via Digital Subscriber Line [DSL] or cable), the legal IP address is only provided to a user on a time-lease basis, rather than permanently.

Internal intranet addresses may be in the ranges 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16. In the internal intranet private address case, a NAT function is employed to map the internal addresses to an external public address when the private-to-public network boundary is crossed. This, however, imposes a number of limitations, particularly since the number of registered public addresses available to a company is almost invariably much smaller (as small as 1) than the number of internal devices requiring an address. A number of protocols cannot travel through a NAT device, and hence the use of NAT implies that many applications (e.g., VoIP) cannot be used effectively in all instances. As a consequence, these applications can only be used in intranets. Examples include:

- Multimedia applications such as videoconferencing, VoIP, or video-on-demand/IPTV do not work smoothly through NAT devices. Multimedia applications make use of real-time transport protocol (RTP) and real-time control protocol (RTCP). These in turn use User Datagram Protocol (UDP) with dynamic allocation of ports and NAT does not directly support this environment.
- IPsec is used extensively for data authentication, integrity, and confidentiality. However, when NAT is used, IPsec operation is impacted, since NAT changes the address in the IP header.
- Multicast, although possible in theory, requires complex configuration in a NAT environment and hence, in practice, is not utilized as often as could be the case.

The need for obligatory use of NAT disappears with IPv6.

7.2.2 IPv6 Address Space

The IPv6 addressing architecture is described in RFC 4291 February 2006 (12). One of the major modifications in the addressing scheme in IPv6 is a change to the

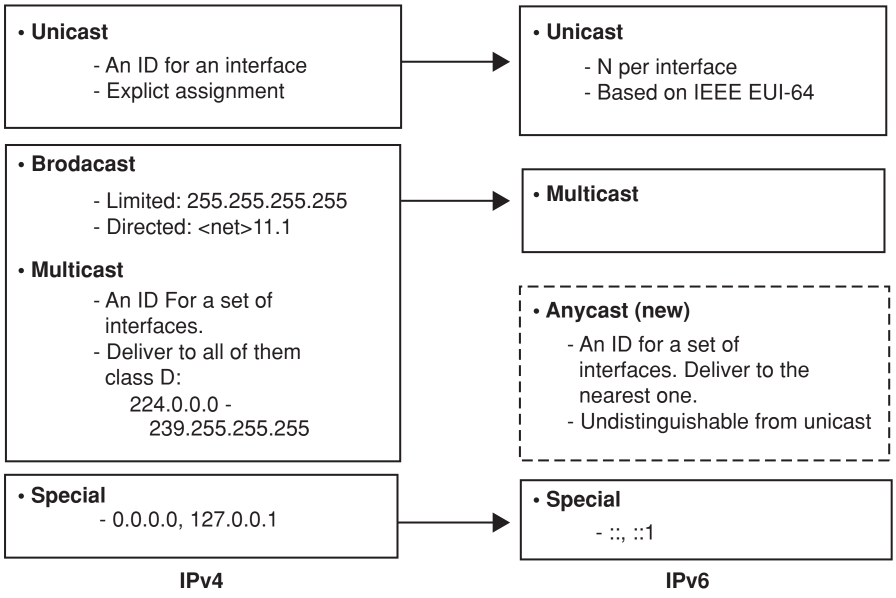


FIGURE 7.1 Address comparison between IPv4 and IPv6.

basic types of addresses and how they are utilized. *Unicast* addresses are utilized for a majority of traditional (enterprise) communications, as was the case in IPv4. However, *Broadcast* as a specific addressing type has been eliminated; in its place support for *multicast* addressing has been expanded and made a required part of the protocol. A new type of addressing called *anycast* has also been implemented. In addition, there are a number of special IPv6 addresses. Figure 7.1 compares the two address formats. Figure 7.2 provides a pictorial comparison of these three transmission (and address) modes. Logically, one can interpret the types of transmissions as follows[3]:

- Unicast transmission: "send to this one specific address"
- Multicast transmission: "send to every member of this specific group"
- Anycast transmission: "send to any one member of this specific group." Typically (motivated by efficiency goals), the transmission occurs to the closest (in routing terms) member of the group. Generally one interprets anycast to mean "send to the closest member of this specific group."

ETSI standards on the M2M system require support for anycast, unicast, multicast and broadcast communication modes; whenever possible, a global broadcast is

[3]Broadcast, by contrast, means "send this information/content to the entire universe of users in the address space."

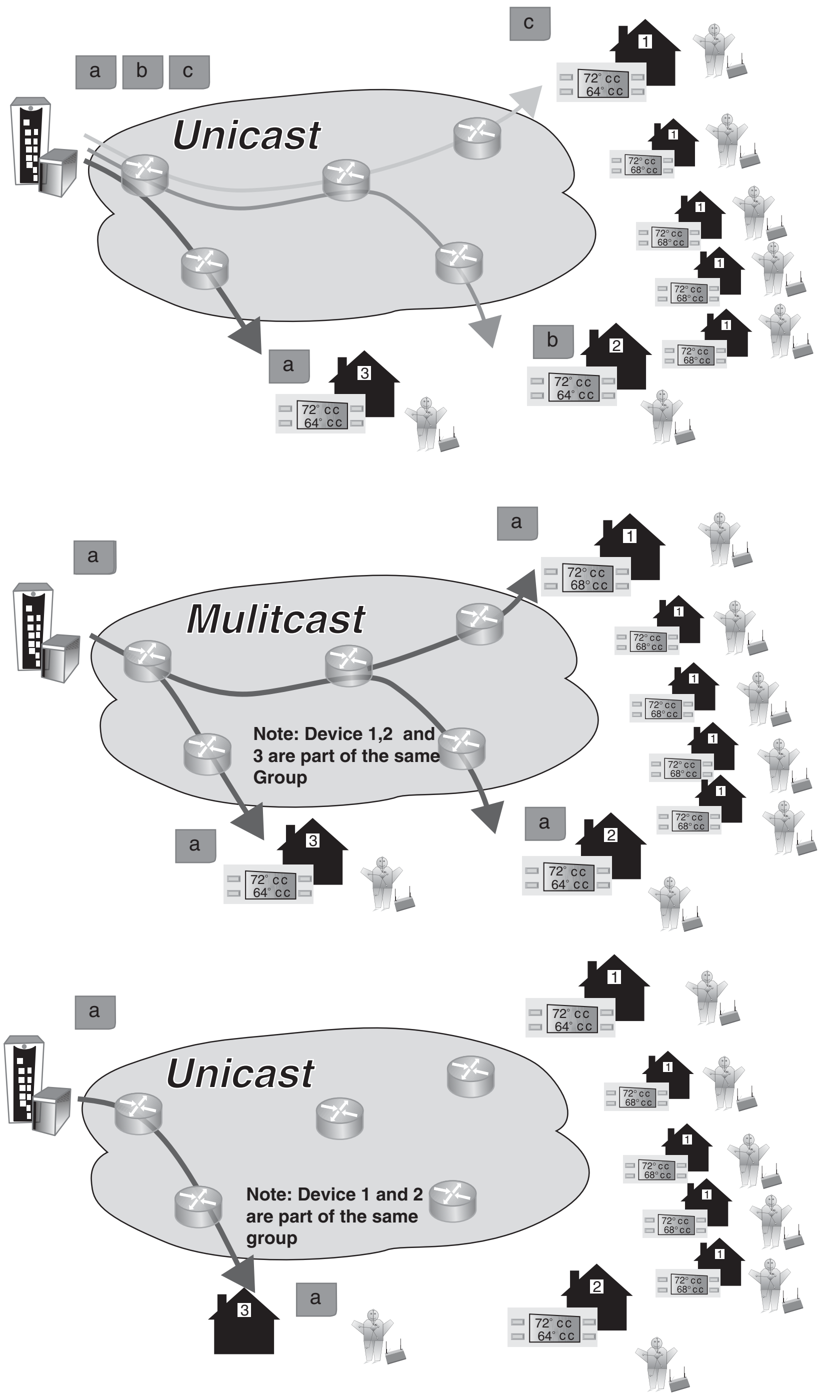

a b c
c
Unicast
b
a
1
2
3
72° cc
64° cc
72° cc
68° cc
a
a
Mulitcast
Note: Device 1,2 and 3 are part of the same Group
a
a
a
Unicast
Note: Device 1 and 2 are part of the same group
a

FIGURE 7.2 Comparison of transmissions to IPv6 nodes.

expected to be replaced by a multicast or anycast in order to minimize the load on the communication network (13).

The format of IPv6 addressing is described in RFC 2373. As noted, an IPv6 address consists of 128 bits, rather than 32 bits as with IPv4 addresses; the number of bits correlates to the address space, as follows:

IP Version	Size of Address Space
IPv6	128 bits, which allows for 2^{128} or 340,282,366,920,938,463,463,374,607,431,768,211,456 (3.4×10^{38}) possible addresses
IPv4	32 bits, which allows for 2^{32} or 4,294,967,296 possible addresses

The relatively large size of the IPv6 address is designed to be subdivided into hierarchical routing domains that reflect the topology of the modern-day Internet. The use of 128 bits provides multiple levels of hierarchy and flexibility in designing hierarchical addressing and routing. The IPv4-based Internet currently lacks this flexibility (14).

The IPv6 address is represented as eight groups of 16 bits each, separated by the ":" character. Each 16-bit group is represented by 4 hexadecimal digits, that is, each digit has a value between 0 and f (0,1, 2, ... a, b, c, d, e, f with a = 10, b = 11, and so on, to f = 15). What follows is an IPv6 address example

3223:0ba0:01e0:d001:0000:0000:d0f0:0010

An abbreviated format exists to designate IPv6 addresses when all endings are 0. For example

3223:0ba0::

is the abbreviated form of the following address:

3223:0ba0:0000:0000:0000:0000:0000:0000

Similarly, only one 0 is written, removing 0's in the left side, and four 0's in the middle of the address. For example the address

3223:ba0:0:0:0:0::1234

is the abbreviated form of the following address

3223:0ba0:0000:0000:0000:0000:0000:1234

There is also a method to designate groups of IP addresses or subnetworks that is based on specifying the number of bits that designate the subnetwork, beginning from left to right, using remaining bits to designate single devices inside the network. For example, the notation

3223:0ba0:01a0::/48

indicates that the part of the IP address used to represent the subnetwork has 48 bits. Since each hexadecimal digit has 4 bits, this points out that the part used to represent the subnetwork is formed by 12 digits, that is: "3223:0ba0:01a0." The remaining digits of the IP address would be used to represent nodes inside the network.

As noted, anycast addresses are a new type of address defined in IPv6 (as originally defined in RFC 1546). The purpose of the anycast address functionality is to enable capabilities that were difficult to implement in IPv4 environments. Datagrams sent to the anycast address are automatically delivered to the device in the network that is the easiest to reach. Anycast addresses can be used to define a group of devices, any one of which can support a service request from the user sent to a single specific IP address. One example is situations where one needs a service that can be provided by a set of different (dispersed) servers, but where one does not specifically care which one provides it; a specific example here may be an Internet or video (streaming) cache. Another example of anycast addressing is a router arrangement that allows datagrams to be transmitted to whichever router in a group of equivalent routers is closest to the point of transmission; a specific example here may be to allow load sharing between routers. It should be noted that there is no special anycast addressing format: anycast addresses are the same as unicast addresses from an address format perspective. In practicality, an anycast address is defined and created in a self-declarative manner when a unicast address is assigned to more than one device interface.

Special IPv6 addresses, as follows (see Table 7.2 for additional details) (15):

- Auto-return or loopback virtual address. This address is specified in IPv4 as the 127.0.0.1 address. In IPv6, this address is represented as ::1.
- Not specified address (::). This address is not allocated to any node since it is used to indicate absence of address.
- IPv6 over IPv4 dynamic/automatic tunnel addresses. These addresses are designated as IPv4-compatible IPv6 addresses and allow the sending of IPv6 traffic over IPv4 networks in a transparent manner. They are represented as, for example, ::156.55.23.5.
- IPv4 over IPv6 addresses automatic representation. These addresses allow for IPv4-only nodes to still work in IPv6 networks. They are designated as "mapped from IPv4 to IPv6 addresses" and are represented as ::FFFF:, for example ::FFFF.156.55.43.3.

TABLE 7.2 A Set of IPv6 Addresses of Particular Note

Node-scoped unicast	::1/128 is the loopback address (per RFC 4291) ::/128 is the unspecified address (per RFC 4291) Addresses within this block should not appear on the public Internet
IPv4-mapped addresses	::FFFF:0:0/96 are the IPv4-mapped addresses (per RFC 4291). Addresses within this block should not appear on the public Internet
IPv4-compatible addresses	::ipv4-address/96 are the IPv4-compatible addresses (per RFC4291). These addresses are deprecated and should not appear on the public Internet
Link-scoped unicast	FE80::/10 are the link-local unicast (per RFC 4291) addresses. Addresses within this block should not appear on the public Internet
Unique local	FC00::/7 are the unique-local addresses (per RFC 4193). Addresses within this block should not appear by default on the public Internet
Documentation prefix	The 2001:db8::/32 are the documentation addresses (per RFC 3849). They are used for documentation purposes such as user manuals, RFCs, and so on. Addresses within this block should not appear on the public Internet
6to4	2002::/16 are the 6to4 addresses (per RFC 3056). The 6to4 addresses may be advertised when the site is running a 6to4 relay or offering a 6to4 transit service. However, the provider of this service should be aware of the implications of running such service (per RFC 3964), which include some specific filtering rules for 6to4. IPv4 addresses disallowed in 6to4 prefixes are listed in (per RFC 3964)
Teredo	2001::/32 are the Teredo addresses (per RFC 4380). The Teredo addresses may be advertised when the site is running a Teredo relay or offering a Teredo transit service
6bone	5F00::/8 were the addresses of the first instance of the 6bone experimental network (per RFC 1897)3FFE::/16 were the addresses of the second instance of the 6bone experimental network (per RFC 2471)Both 5F00::/8 and 3FFE::/16 were returned to IANA (per RFC 3701). These addresses are subject to future allocation, similar to current unallocated address space. Addresses within this block should not appear on the public Internet until they are reallocated
ORCHID	2001:10::/28 are ORCHID addresses (per RFC 4843). These addresses are used as identifiers and are not routable at the IP layer. Addresses within this block should not appear on the public Internet
Default route	::/0 is the default unicast route address
IANA special-purpose IPv6 address block	An IANA registry (iana-ipv6-special-registry) is set (per RFC 4773) for special-purpose IPv6 address block assignments used for experiments and other purposes. Addresses within this registry should be reviewed for Internet routing considerations
Multicast	FF00::/8 are multicast addresses (per RFC 4291). They have a 4-bit scope in the address field where only some values are of global scope (per RFC 4291). Only addresses with global scope in this block may appear on the public Internet Multicast routes must not appear in unicast routing tables

7.3 IPv6 PROTOCOL OVERVIEW

Table 7.3 summarizes the core protocols that comprise IPv6. IPv6 basic protocol capabilities include the following:

- Addressing
- Anycast
- Flow Labels
- ICMPv6
- Neighbor discovery (ND)

Like IPv4, IPv6 is a connectionless datagram protocol used primarily for addressing and routing packets between hosts. Connectionless means that a session is not established before exchanging data. "Unreliable" means that delivery is not guaranteed. IPv6 always makes a best-effort attempt to deliver a packet. An IPv6 packet might be lost, delivered out of sequence, duplicated, or delayed. IPv6 *per se* does not attempt to recover from these types of errors. The acknowledgment of packets delivered and the recovery of lost packets is done by a higher-layer protocol, such as TCP (14). From a packet-forwarding perspective, IPv6 operates in a similar, nearly identical manner to IPv4.

TABLE 7.3 Key IPv6 Protocols

Protocol (Current Version)	Description
IPv6: RFC 2460 Updated by RFC 5095, RFC 5722, RFC 5871	IPv6 is a connectionless datagram protocol used for routing packets between hosts
Internet control message protocol for IPv6 (ICMPv6): RFC 4443 Updated by RFC 4884	A mechanism that enables hosts and routers that use IPv6 communication to report errors and send status messages
Multicast listener discovery (MLD): RFC 2710 Updated by RFC 3590, RFC 3810	A mechanism that enables one to manage subnet multicast membership for IPv6. MLD uses a series of three ICMPv6 messages. MLD replaces the Internet group management protocol (IGMP) v3 that is employed for IPv4
ND: RFC 4861 Updated by RFC 5942	A mechanism that is used to manage node-to-node communication on a link. ND uses a series of five ICMPv6 messages. ND replaces address resolution protocol (ARP), ICMPv4 router discovery, and the ICMPv4 redirect message ND is implemented using the neighbor discovery protocol (NDP)

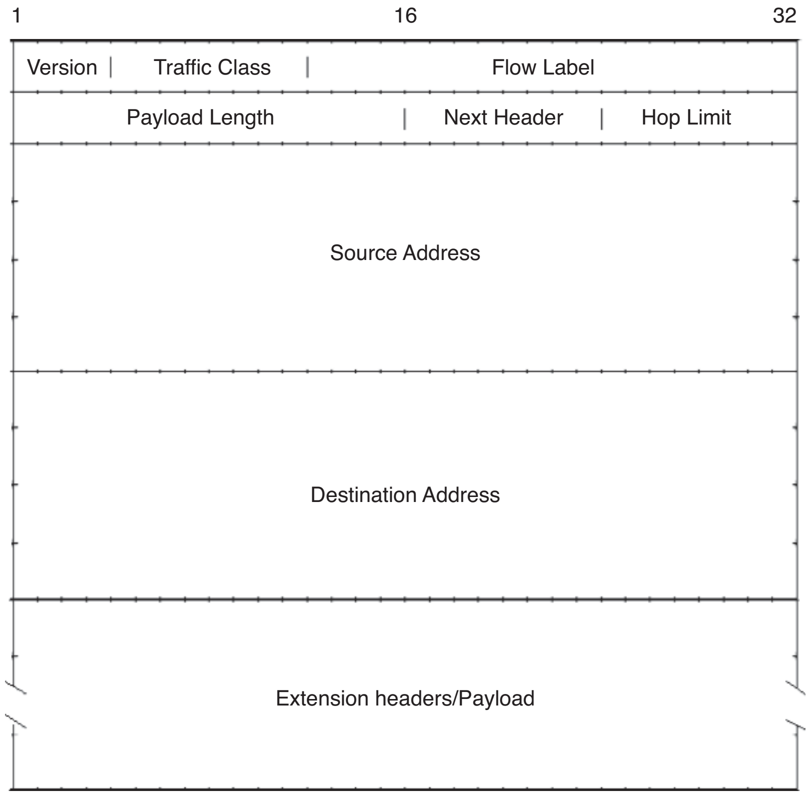
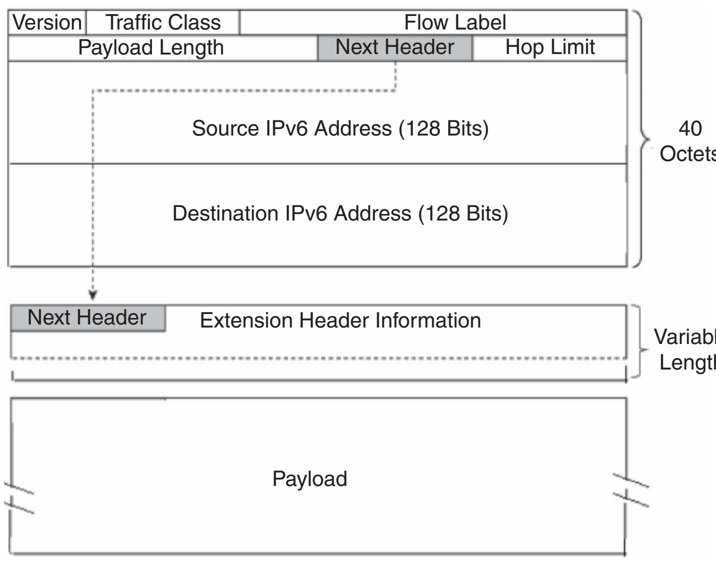


FIGURE 7.3 IPv6 packet.

An IPv6 packet, also known as an IPv6 datagram, consists of an IPv6 header and an IPv6 payload, as shown Figure 7.3. The IPv6 header consists of two parts, the IPv6 base header and optional extension headers. See Figure 7.4. Functionally, the optional extension headers and upper-layer protocols, for example TCP, are considered part of the IPv6 payload. Table 7.4 shows the fields in the IPv6 base header. IPv4 headers and IPv6 headers are not directly interoperable: hosts and/or routers must use an implementation of both IPv4 and IPv6 in order to recognize and process both header formats (see Fig. 7.5). This gives rise to a number of complexities in the migration process between the IPv4 and the IPv6 environments. The IP header in IPv6 has been streamlined and defined to be of a fixed length (40 bytes). In IPv6, header fields from the IPv4 header have been removed, renamed, or moved to the new optional IPv6 extension headers. The header length field is no longer needed since the IPv6 header is now a fixed-length entity. The IPv4 "type of service" is equivalent to the IPv6 "traffic class" field. The "total length" field has been replaced with the "payload length" field. Since IPv6 only allows for fragmentation to be performed by the IPv6 source and destination nodes, and not individual routers, the IPv4 segment control



IPv6 extension headers are optional headers that may follow the basic IPv6 header. An IPv6 PDU may include zero, one, or multiple extension headers. When multiple extension headers are used, they form a chained list of headers identified by the Next Header field of the previous header.

FIGURE 7.4 IPv6 extension headers.

fields (identification, flags, and fragment offset fields) have been moved to similar fields within the fragment extension header. The functionality provided by the "time to live (TTL[4])" field has been replaced with the "hop limit" field. The "protocol" field has been replaced with the "next header type" field. The "header checksum" field was removed, which has the main advantage of not having each relay spend time processing the checksum. The "options" field is no longer part of the header as it was in IPv4. Options are specified in the optional IPv6 extension headers. The removal of the options field from the header enables more efficient routing; only the information that is needed by a router needs to be processed (16).

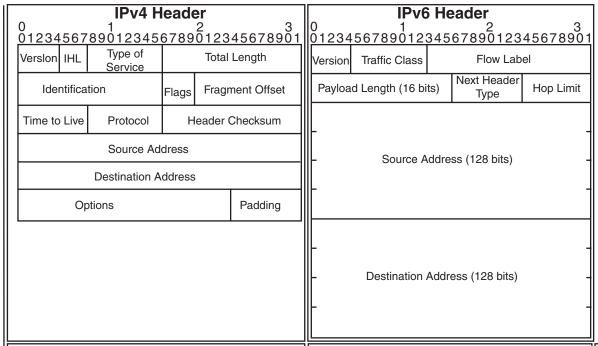
One area requiring consideration, however, is the length of the IPv6 PDU: the 40-octet header can be a problem for real-time IP applications such as VoIP and IPTV. Header compression (HC) becomes critical for many applications, as noted in Section 7.4. Also, there will be some bandwidth inefficiency in general, which could be an issue in limited-bandwidth environments or applications (e.g., wireless networks, sensor networks, IoT networks).

Stateless address autoconfiguration (described in RFC 4862) defines how an IPv6 node generates addresses without the use of a DHCP for IPv6 (DHCPv6) server (17). "Autoconfiguration" is a new characteristic of the IPv6 protocol that facilitates network management and system set-up tasks by users. This characteristic is

[4]TTL has been used in many attacks and intrusion detection system (IDS) tricks in IPv4.

TABLE 7.4 IPv6 Base Header

IPv6 Header Field	Length (bits)	Function
Version	4	Identifies the version of the protocol. For IPv6, the version is 6
Traffic class	8	Intended for originating nodes and forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets
Flow label	20	(sometimes referred to as flow ID) Defines how traffic is handled and identified. A flow is a sequence of packets sent either to a unicast or to a multicast destination. This field identifies packets that require special handling by the IPv6 node. The following list shows the ways the field is handled if a host or router does not support flow label field functions: • If the packet is being sent, the field is set to zero • If the packet is being received, the field is ignored
Payload length	16	Identifies the length, in octets, of the payload. This field is a 16-bit unsigned integer. The payload includes the optional extension headers, as well as the upper-layer protocols, for example, TCP
Next header	8	Identifies the header immediately following the IPv6 header. The following shows examples of the next header: • 00 = Hop-by-hop options • 01 = ICMPv4 • 04 = IP in IP (encapsulation) • 06 = TCP • 17 = UDP • 43 = Routing • 44 = Fragment • 50 = Encapsulating security payload • 51 = Authentication • 58 = ICMPv6
Hop limit	8	Identifies the number of network segments, also known as links or subnets, on which the packet is allowed to travel before being discarded by a router. The hop limit is set by the sending host and is used to prevent packets from endlessly circulating on an IPv6 internetwork When forwarding an IPv6 packet, IPv6 routers must decrease the hop limit by 1 and must discard the IPv6 packet when the hop limit is 0
Source address	128	Identifies the IPv6 address of the original source of the IPv6 packet
Destination address	128	Identifies the IPv6 address of intermediate or final destination of the IPv6 packet



Version (4-bit)	Version (4-bit)	IPv6 header contains a new value
Header length (4-bit)	—	Removed in IPv6, the basic IPv6 header has fixed length of 40 octets
Type of service (8-bit)	Traffic class (8-bit)	Same function for both headers
—	Flow label (20-bit)	New field added to tag a flow for IPv6 packets
Total PDU length (16-bit)	Payload length (16-bit)	Same function for both headers
Identification (16-bit)	—	Removed in IPv6 because fragmentation is no longer done by intermediate routers in the networks, but by the source node that originates the packet
Flags (3-bit)	—	Removed in IPv6 because fragmentation is no longer done by intermediate routers in the networks, but by the source node that originates the packet
Fragment offset (13-bit)	—	Removed in IPv6 because fragmentation is no longer done by intermediate routers in the networks, but by the source node that originates the packet
Time to live (8-bit)	Hop limit (8-bit)	Same function for both headers
Protocol number (8-bit)	Next header (8-bit)	Same function for both headers
Header checksum (16-bit)	—	Removed in IPv6; upper-layer protocols handle checksums
Source address (32-bit)	Source address (128-bit)	Same function, but source address is expanded in IPv6
Destination address (32-bit)	Destination address (128-bit)	Same function, but destination address is expanded in IPv6
Options (variable)	—	Removed in IPv6. Options handled differently
Padding (variable)	—	Removed in IPv6. Options handled differently
—	Extension headers	New way in IPv6 to handle Options fields, security

FIGURE 7.5 Comparison of IPv4 and IPv6 headers.

often called "plug-and-play" or "connect-and-work." Autoconfiguration facilitates initialization of user devices: after connecting a device to an IPv6 network, one or several IPv6 globally unique addresses are automatically allocated. Note, however, that an IPv6 address must be configured on a router's interface for the interface to forward IPv6 traffic. Configuring a site-local or global IPv6 address on a router's interface automatically configures a link-local address (LLA) and activates IPv6 for that interface.

DHCP allows systems to obtain an IPv4 address and other required information (e.g., default router or domain name system [DNS] server); a similar protocol, DHCPv6, has been published for IPv6. DHCP and DHCPv6 are known as stateful protocols because they maintain tables on (specialized) servers. However, IPv6 also has a new stateless autoconfiguration protocol that has no equivalent in IPv4. The stateless autoconfiguration protocol does not require a server component because there is no state to maintain (a DHCP server may typically run in a router or firewall). Every IPv6 system (other than routers) is able to build its own unicast global address

(18). “Stateless” autoconfiguration is also described as “serverless.” The acronym SLAAC is also used; it expands to *stateless address autoconfiguration*. SLAAC was originally defined in RFC 2462. With SLAAC, the presence of configuration servers to supply profile information is not required.

The host generates its own address using a combination of the information that it possesses (in its interface or network card) and the information that is supplied by the router. As noted in RFC 4941, nodes use IPv6 SLAAC to generate addresses using a combination of locally available information and information advertised by routers. Addresses are formed by combining network prefixes with an interface identifier. On an interface that contains an embedded IEEE identifier, the interface identifier is typically derived from it. On other interface types, the interface identifier is generated through other means, for example, via random number generation (19). Some types of network interfaces come with an embedded IEEE identifier (i.e., a link-layer media access control [MAC] address), and in those cases, SLAAC uses the IEEE identifier to generate a 64-bit interface identifier (12). By design, the interface identifier is likely to be globally unique when generated in this fashion. The interface identifier is in turn appended to a prefix to form a 128-bit IPv6 address. Not all nodes and interfaces contain IEEE identifiers. In such cases, an interface identifier is generated through some other means (e.g., at random), and the resultant interface identifier may not be globally unique and may also change over time. Routers determine the prefix that identifies networks associated to the link under discussion. The “interface identifier” identifies an interface within a subnetwork and is often, and by default, generated from the MAC address of the network card. The IPv6 address is built combining the 64 bits of the interface identifier with the prefixes that routers determine as belonging to the subnetwork. If there is no router, the interface identifier is self-sufficient to allow the PC to generate a “link-local” address. The “link-local” address is sufficient to allow the communication between several nodes connected to the same link (the same local network).

In summary, all nodes combine interface identifiers (whether derived from an IEEE identifier or generated through some other technique) with the reserved link-local prefix to generate LLAs for their attached interfaces. Additional addresses can then be created by combining prefixes advertised in router advertisements via ND (defined in RFC 4861) (20) with the interface identifier.

Note: As seen addresses generated using SLAAC contain an embedded interface identifier that remains constant over time. Whenever a fixed identifier is used in multiple contexts, a security exposure could theoretically result. A correlation can be performed by an attacker who is in the path between the node in question and the peer(s) to which it is communicating, and who can view the IPv6 addresses present in the datagrams. Because the identifier is embedded within the IPv6 address, which is a fundamental requirement of communication, it cannot be easily hidden. Solutions to this issue have been proposed by generating interface identifiers that vary over time (19).

IPv6 addresses are “leased” to an interface for a fixed established time (including an infinite time). When this “lifetime” expires, the link between the interface and the address is invalidated and the address can be reallocated to other interfaces. For the suitable management of addresses expiration time, an address goes through two states (stages) while is affiliated to an interface (21):

(a) At first, an address is in a “preferred” state, so its use in any communication is not restricted.
(b) After that, an address becomes “deprecated,” indicating that its affiliation with the current interface will (soon) be invalidated.

When it is in a “deprecated” state, the use of the address is discouraged, although it is not forbidden. However, when possible, any new communication (e.g., the opening of a new TCP connection) must use a “preferred” address. A “deprecated” address should only be used by applications that have already used it before and in cases where it is difficult to change this address to another address without causing a service interruption.

To ensure that allocated addresses (granted either by manual mechanisms or by autoconfiguration) are unique in a specific link, the *link duplicated address detection algorithm* is used. The address to which the duplicated address detection algorithm is being applied to is designated (until the end of this algorithmic session) as an “attempt address.” In this case, it does not matter that such address has been allocated to an interface and received packets are discarded.

Next we describe how an IPv6 address is formed. The lowest 64 bits of the address identify a specific interface, and these bits are designated as “interface identifier.” The highest 64 bits of the address identify the “path” or the “prefix” of the network or router in one of the links to which such interface is connected. The IPv6 address is formed by combining the prefix with the interface identifier.

It is possible for a host or device to have IPv6 and IPv4 addresses simultaneously. Most of the systems that currently support IPv6 allow the simultaneous use of both protocols. In this way, it is possible to support communication with IPv4-only networks as well as with IPv6-only networks and the use of the applications developed for both protocols (21).

Is it possible to transmit IPv6 traffic over IPv4 networks via tunneling methods. This approach consists of “wrapping” the IPv6 traffic as IPv4 payload data: IPv6 traffic is sent “encapsulated” into IPv4 traffic, and at the receiving end this traffic is parsed as IPv6 traffic. Transition mechanisms are methods used for the coexistence of IPv4 and/or IPv6 devices and networks. For example, an “IPv6-in-IPv4 tunnel” is a transition mechanism that allows IPv6 devices to communicate through an IPv4 network. The mechanism consists of creating the IPv6 packets in a normal way and encapsulating them in an IPv4 packet. The reverse process is undertaken in the destination machine that de-encapsulates the IPv6 packet.

There is a significant difference between the procedures to allocate IPv4 addresses, which focus on the parsimonious use of addresses (since addresses are a scare resource and should be managed with caution), and the procedures to allocate IPv6 addresses, which focus on flexibility. ISPs deploying IPv6 systems follow the RIRs' policies relating to how to assign IPv6 addressing space among their clients. RIRs are recommending ISPs and operators allocate to each IPv6 client a /48 subnetwork; this allows clients to manage their own subnetworks without using NAT. (The implication is that the *obligatory* need for NAT for intranet-based devices disappears in IPv6.)

In order to allow its maximum scalability, the IPv6 protocol uses an approach based on a basic header, with minimum information. This differentiates it from IPv4 where different options are included in addition to the basic header. IPv6 uses a header "concatenation" mechanism to support supplementary capabilities. The advantages of this approach include the following:

- The size of the basic header is always the same and is well known. The basic header has been simplified compared with IPv4, since only eight fields are used instead of 12. The basic IPv6 header has a fixed size; hence, its processing by nodes and routers is more straightforward. Also, the header's structure aligns to 64 bits, so that new and future processors (64 bits minimum) can process it in a more efficient way.
- Routers placed between a source point and a destination point (i.e., the route that a specific packet has to pass through) do not need to process or understand any "following headers." In other words, in general, interior (core) points of the network (routers) only have to process the basic header, while in IPv4 all headers must be processed. This flow mechanism is similar to the operation in MPLS, yet precedes it by several years.
- There is no limit to the number of options that the headers can support (the IPv6 basic header is 40 octets in length, while IPv4 one varies from 20 to 60 octets, depending on the options used).

In IPv6, interior/core routers do not perform packet fragmentation, but the fragmentation is performed end-to-end. That is, source and destination nodes perform, by means of the IPv6 stack, the fragmentation of a packet and the reassembly, respectively. The fragmentation process consists of dividing the source packet into smaller packets or fragments (21).

The IPv6 specification defines a number of extension headers (16) (also see Table 7.5) (22):

- Routing header—Similar to the source routing options in IPv4. The header is used to mandate a specific routing.
- AH—A security header that provides authentication and integrity.
- Encapsulating security payload (ESP) header—A security header that provides authentication and encryption.
- Fragmentation header—The Fragmentation Header is similar to the fragmentation options in IPv4.

TABLE 7.5 IPv6 Extension Headers

Header (Protocol ID)	Description
Hop-by-hop options header (protocol 0)	The hop-by-hop options header is used for Jumbogram packets and the router alert. An example of applying the hop-by-hop options header is resource reservation protocol (RSVP). This field is read and processed by every node and router along the delivery path
Destination options header (protocol 60)	This header carries optional information that is specifically targeted to a packet's destination address. The MIPv6 protocol specification makes use of the destination options header to exchange registration messages between MNs and the HA. Mobile IP is a protocol allowing MNs to keep permanent IP addresses even if they change point of attachment
Routing header (protocol 43)	This header can be used by an IPv6 source node to force a packet to pass through specific routers on the way to its destination. A list of intermediary routers may be specified within the routing header when the routing type field is set to 0
Fragment header (protocol 44)	In IPv6, the path MTU discovery (PMTUD) mechanism is recommended to all IPv6 nodes. When an IPv6 node does not support PMTUD and it must send a packet larger than the greatest MTU along the delivery path, the fragment header is used. When this happens, the node fragments the packets and sends each fragment using fragment headers; then the destination node reassembles the original packet by concatenating all the fragments
AH (protocol 51)	This header is used in IPsec to provide authentication, data integrity, and replay protection. It also ensures protection of some fields of the basic IPv6 header. This header is identical in both IPv4 and IPv6
Encapsulating security payload (ESP) header (protocol 50)	This header is also used in IPsec to provide authentication, data integrity, replay protection, and confidentiality of the IPv6 packet. Similar to the AH, this header is identical in both IPv4 and IPv6

- Destination options header—Header that contains a set of options to be processed only by the final destination node. MIPv6 is an example of an environment that uses such a header.
- Hop-by-hop options header—A set of options needed by routers to perform certain management or debugging functions.

7.4 IPv6 TUNNELING

IPv6 tunneling is used in a variety of settings, including in MIPv6. MIPv6 tunnels payload packets between the mobile node (MN) and the home agent (HA) in both directions. This tunneling uses IPv6 encapsulation discussed below.

IPv6 tunneling as defined in RFC 2473 (23) is a technique for establishing a "virtual link" between two IPv6 nodes for transmitting data packets as payloads of IPv6 packets. From the perspective of the two nodes, this "virtual link," called an *IPv6 tunnel*, appears as a point-to-point link on which IPv6 acts like a link-layer protocol. The two IPv6 nodes support specific roles. One node encapsulates original packets received from other nodes or from itself and forwards the resulting tunnel packets through the tunnel. The other node decapsulates the received tunnel packets and forwards the resulting original packets toward their destinations, possibly itself. The encapsulator node is called the tunnel entry-point node, and it is the source of the tunnel packets. The decapsulator node is called the tunnel exit point, and it is the destination of the tunnel packets. An IPv6 tunnel is a unidirectional mechanism—tunnel packet flow takes place in one direction between the IPv6 tunnel entry-point and exit-point nodes (see Fig. 7.6, top). Bidirectional tunneling is achieved by merging two unidirectional mechanisms, that is, configuring two tunnels, each in opposite direction to the other—the entry-point node of one tunnel is the exit-point node of the other tunnel (see Fig. 7.6, bottom).

Note: while tunnels between two nodes identified by unicast addresses are typical (such tunnels look like "virtual point to point links"), one can also define tunnels where the exit-point nodes are identified by anycast or multicast addresses.

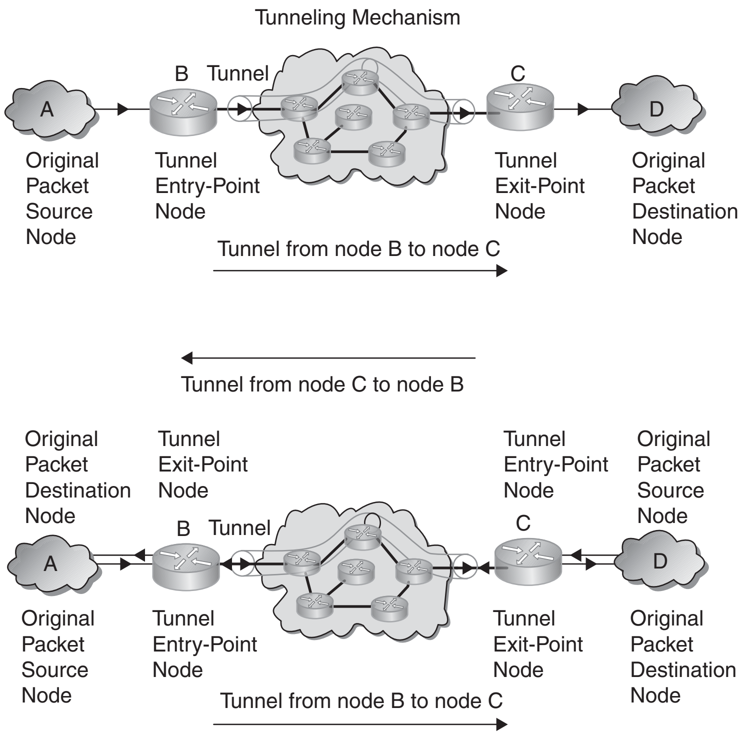

Tunneling Mechanism
B Tunnel
C
A
D
Original Packet Source Node
Tunnel Entry-Point Node
Tunnel Exit-Point Node
Original Packet Destination Node
Tunnel from node B to node C
Tunnel from node C to node B
Original Packet Destination Node
Tunnel Exit-Point Node
Tunnel Entry-Point Node
Original Packet Source Node
B Tunnel
C
A
D
Original Packet Source Node
Tunnel Entry-Point Node
Tunnel Exit-Point Node
Original Packet Destination Node
Tunnel from node B to node C

FIGURE 7.6 IPv6 tunneling. Top: Unidirectional. Bottom: Bidirectional.

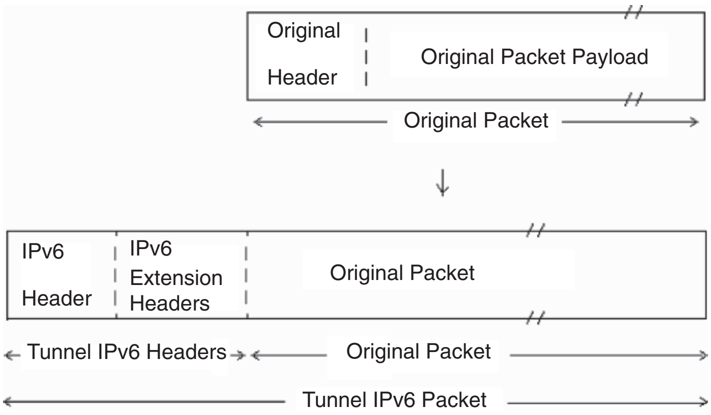


FIGURE 7.7 Encapsulating a packet.

IPv6 encapsulation entails prepending an IPv6 header to the original packet, and, optionally, a set of IPv6 extension headers, as depicted in Figure 7.7 that are collectively called tunnel IPv6 headers. The encapsulation takes place in an IPv6 tunnel entry-point node, as a result of an original packet being forwarded onto the virtual link represented by the tunnel. The original packet is processed during forwarding according to the forwarding rules of the protocol of that packet. At encapsulation, the source field of the tunnel IPv6 header is filled with an IPv6 address of the tunnel entry-point node and the destination field with an IPv6 address of the tunnel exit point. Subsequently, the tunnel packet resulting from encapsulation is sent toward the tunnel exit-point node.

IPv6 intermediate processing by intermediate nodes in the tunnel processes the IPv6 tunnel packets according to the IPv6 protocol. For example, a tunnel hop-by-hop options extension header is processed by each receiving node in the tunnel; a tunnel routing extension header identifies the intermediate processing nodes and controls at a finer granularity the forwarding path of the tunnel packet through the tunnel; a tunnel destination options extension header is processed at the tunnel exit-point node.

IPv6 decapsulation is the opposite process of encapsulation. Upon receiving an IPv6 packet destined to an IPv6 address of a tunnel exit-point node, its IPv6 protocol layer processes the tunnel headers. The strict left-to-right processing rules for extension headers are applied. When processing is complete, control is handed to the next protocol engine, which is identified by the next header field value in the last header processed. If this is set to a tunnel protocol value, the tunnel protocol engine discards the tunnel headers and passes the resulting original packet to the Internet or lower-layer protocol identified by that value for further processing. For example, in the case the next header field has the IPv6 tunnel protocol value, the resulting original packet is passed to the IPv6 protocol layer. (The tunnel exit-point node, which decapsulates the tunnel packets, and the destination node, which receives the resulting original packets, can be the same node.)

7.5 IPsec IN IPv6

As noted, IPsec provides network-level security where the application data is encapsulated within the IPv6 packet. IPsec itself is a set of two protocols: ESP, which provides integrity and confidentiality and AH, which provides integrity. IPsec utilizes the AH and/or ESP header to provide security (the AH and ESP header may be used separately or in combination). IPsec, with ESP, offers integrity and data origin authentication, confidentiality, and optional (at the discretion of the receiver) anti-replay features (using confidentiality without integrity is discouraged by the RFCs); in addition, ESP provides limited traffic flow confidentiality. Both the AH and ESP header may be employed as follows (16):

- "Tunnel mode"—The protocol is applied to the entire IP packet. This method is needed to ensure security over the entire packet, where a new IPv6 header and an AH or ESP header are wrapped around the original IP packet.
- "Transport mode"—The protocol is just applied to the transport layer (i.e., TCP, UDP, ICMP) in the form of an IPv6 header and AH or ESP header, followed by the transport protocol data (header, data). (see Fig. 7.8).

It should be noted that although the basic IPv6 standards have long been stable, considerable work continues in the IETF, particularly to resolve the issue of highly scalable multihoming support for IPv6 sites, and to resolve the problem of IP-layer interworking between IPv6-only and IPv4-only hosts. IPv6/IPv4 interworking at the application layers is handled within the original dual-stack model of IPv6 deployment: either one end of an application session will have dual-stack connectivity or a dual-stack intermediary such as a hypertext transfer protocol (HTTP) proxy or simple mail transfer protocol (SMTP) server will interface to both IPv4-only and IPv6-only hosts or applications (24).

7.6 HEADER COMPRESSION SCHEMES

Implementation of IPv6 gives rise to concerns related to expanded packet headers, especially for video and wireless (low bandwidth channel) applications. As noted in earlier sections, the packet header size doubled from 20 bytes in IPv4 to at least 40 bytes in IPv6. The use of network-layer encryption mechanism nearly doubles IP operational overhead. HC is, therefore, of interest. Currently, the use of HC in commercial networks is generally rare, but wireless and video applications (especially in an IPv6 environment) may well drive future deployment of the technology.

HC algorithms can reduce the performance and throughput impact of expanded IPv6 packet headers and protocol-imposed overhead. Consider the illustrative case where packets with constant 20 byte payloads are transmitted using a 40-byte IPv6 header. Consider a 1 Mbps link. Then during a 1-s period, about 666 kb transmitted over the link is IPv6 overhead, and only about 333 kb transmitted over the link is

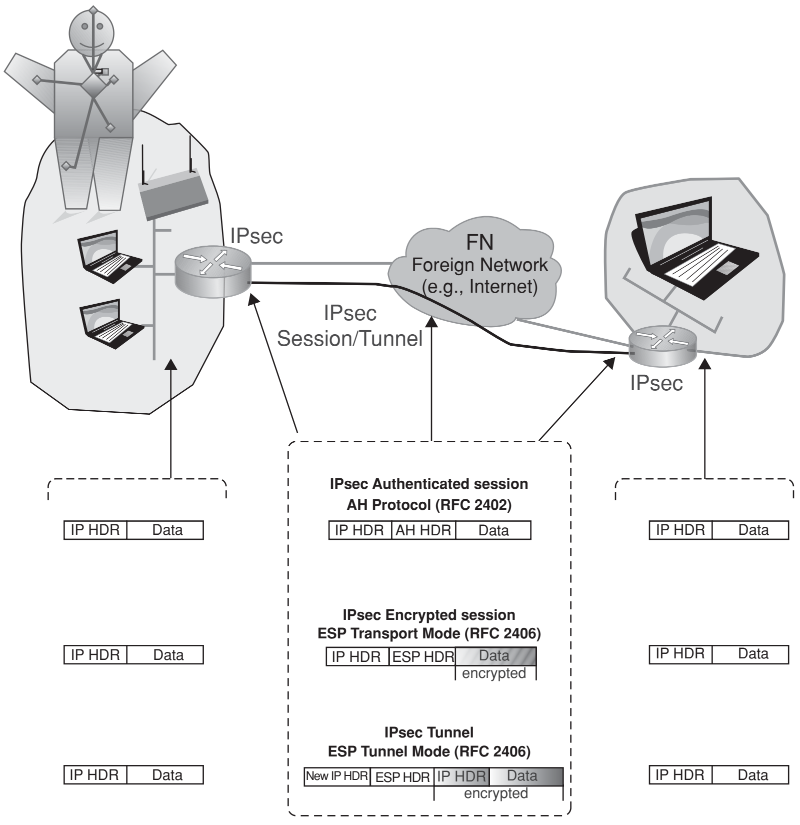


FIGURE 7.8 IPsec network environment.

actual user data. This implies that 66% of data transmitted is overhead. Now consider the case where the same packet of payload is sent with a 2-byte compressed header. Now over a 1-s period, about 90 kb transmitted is IPv6 overhead, and about 910 kb transmitted is actual user data. This implies that only 9% of data transmitted is overhead. This example shows that HC can theoretically decrease header overhead by 95%. Overhead is defined as “IP header bytes” divided by “total bytes transmitted.” Naturally, the overhead for larger packets will be less as a total percentage. Studies show that although the average packet length of packets traveling over the Internet is around 350–400 bytes, a considerable portion of the Internet traffic is short (say, 40 bytes or less) (25). Depending on the encapsulation protocol, video packets can also be small. For example, under the DVB standard (e.g., DVB-T, DVB-C, DVB-S, DVB-S2), basic packets have a length of 204 bytes. This implies a significant percentage of overhead is incurred without HC. (For illustration, a 40-byte IPv6 header on a DVB packet would result in an overhead of 40/244 = 16.39%; if one assumes that header size is reduced to 2 bytes per packet, the overhead is 2/206 = 0.97%.)

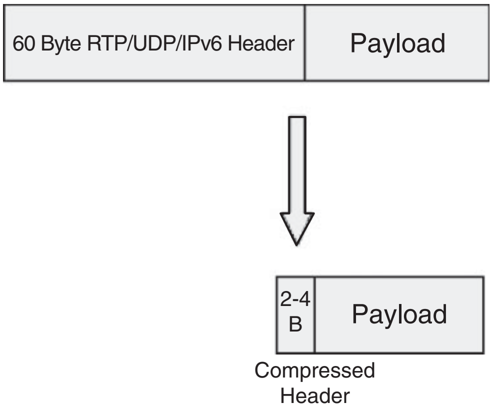


FIGURE 7.9 HC for IPv6.

There is additional protocol overhead. Applications carrying data carried within RTP will, in addition to link-layer framing, have an IPv4 header (20 octets), a UDP header (8 octets), and an RTP header (12 octets), for a total of 40 octets. With IPv6, the IPv6 header is 40 octets for a total of 60 octets. Applications transferring data using TCP have 20 octets for the transport header, for a total size of 40 octets for IPv4 and 60 octets for IPv6 (26).

Usually HC techniques are applied to a link, on a per-hop basis. Application of hop-by-hop HC techniques to network backbones is relatively rare because to achieve compression over the network, multiple compression–decompression cycles are required. This represents a scalability and resource issues on core network nodes. Developments in the IETF in the past few years provide a framework for applying HC over multiple-hop to backbone networks. For example, work has been done HC techniques to MPLS backbones and mobile ad hoc networks (MANETs) (where trade needs to be made between computational processing, power requirements, and bandwidth savings).

Traditionally, compression is applied to layer 3 (IP) and several layer 4 protocol headers; for example, RTP/UDP/IPv6 headers can be compressed from 60 bytes to 2–4 bytes. See Figure 7.9. HC algorithms can also reduce the additional overhead introduced by network-layer encryption mechanisms (e.g., IPsec). Compression algorithms that address encryption/decryption have the ability to: (i) compress inner headers before encryption and (ii) compress outer ESP/IP headers after encryption. Two compression protocols emerged from the IETF in recent years:

(i) Internet protocol header compression (IPHC), a scheme designed for low bit error rate (BER) links (compression profiles were originally defined in RFC 2507 and RFC 2508, and further discussed in RFC 4995, 4996, and 4497); it provides compression of TCP/IP, UDP/IP, RTP/UDP/IP, and ESP/IP header; “enhanced” compression of RTP/UDP/IP (ECRTP) headers is defined in RFC 3545.

(ii) Robust header compression (ROHC) is a scheme designed for wireless links that provides greater compression compared to IPHC at the cost of greater implementation complexity (compression profiles were originally defined in RFC 3095 and RFC 3096 with further developments in other RFCs) (26–28); this is more suitable for high BER, long RTT links and supports compression of ESP/IP, UDP/IP, and RTP/UDP/IP headers.

Compression is applied over a link between a source node (i.e., compressor) and a destination node (i.e., decompressor). HC algorithms make use of protocol inter-packet header field redundancies to improve overall efficiency. Both compressor and decompressor store header fields of each packet stream and associate each stream with a context identifier (CID). Upon reception of a packet with an associated context, the compressor removes the IPv6 header fields from packet header and appends a CID. Upon reception of a packet with a CID, the decompressor inserts IPv6 header fields back into packet header and transmits packet (25). IPHC and ROHC are both specified in Release 4 and Release 5 of the Third-Generation Partnership Project (3GPP). Cisco Systems router Internetwork Operating System (IOS) provides IPHC implementation.

Point-to-point protocol (PPP) (defined in RFC 1661) provides (i) a method for encapsulating datagrams over serial links; (ii) a link control protocol (LCP) for establishing, configuring, and testing the data-link connection; and (iii) a family of network control protocols (NCPs) for establishing and configuring different network-layer protocols. In order to establish communications over a point-to-point link, each end of the PPP link must first send LCP packets to configure and test the data link. After the link has been established and optional facilities have been negotiated as needed by the LCP, PPP must send NCP packets to choose and configure one or more network-layer protocols. Once each of the chosen network-layer protocols has been configured, datagrams from each network-layer protocol can be sent over the link. The link will remain configured for communications until explicit LCP or NCP packets close the link down, or until some external event occurs (power failure at the other end, carrier drop, and so on) (29).

In RFC 5072, the NCP for establishing and configuring IPv6 over PPP, called IPV6CP, is defined. In RFC 5172, the compression parameter for use in IPv6 datagram compression is defined. The configuration option described in this just-cited RFC provides a way to negotiate the use of a specific IPv6 packet compression protocol. The IPv6-compression protocol configuration option is used to indicate the ability to receive compressed packets. IPv6-compression protocol field values have been assigned in for IPHC (0061) and for ROHC (0003).

7.7 QUALITY OF SERVICE IN IPv6

ETSI standards on the M2M system require that the M2M system should be able to make use of the QoS supported by underlying networks; M2M applications or service capabilities may use QoS capabilities of the underlying networks when implemented by the system (13). QoS is supported in IPv6. The IPv6 header has two QoS-related fields:

- 20-bit flow label, usable in IntServ-based environments. In IntServ environments, performance guarantees to traffic and resource reservations are provided on per-flow basis. A guaranteed and controlled load service capability is supported. IntServ approaches have scalability issues;
- 8-bit traffic class indicator usable in DiffServ-based environments. DiffServ environments are more common. The traffic class field may be used to set specific precedence or differentiated services code point (DSCP) values. These values are used in the exact same way as in IPv4. Performance guarantees are provided to traffic aggregates rather than to flows. DiffServ classifies all the network traffic into classes. Two distinct types (per hop behaviors) are supported:
- Expedited forwarding (EF): aims at providing QoS for the class by minimizing jitter and is generally focused on providing stricter guarantees;
- Assured forwarding (AF): inserts at most four classes with at most three levels of packets dropping categories.

There are no signaling protocol for resource allocation (admission control) and QoS mechanisms control. The following priority levels are typical, but variances are possible:

- Level 0—No specify priority
- Level 1—Background traffic (news)
- Level 2—Unattended data transfer (email)
- Level 3—Reserved
- Level 4—Attended bulk transfer (FTP)
- Level 5—Reserved
- Level 6—Interactive traffic (Telnet, Windowing)
- Level 7—Control traffic (routing, network management)

7.8 MIGRATION STRATEGIES TO IPv6

7.8.1 Technical Approaches

While the infrastructure is in place for IPv4 systems and IPv6 systems to run in parallel, widespread adoption of IPv6 has been slow because the two systems are not directly compatible (IPv6 and IPv4 protocols can coexist, but they cannot intercommunicate directly), and there been so far rather limited economic incentive for providers and end-user firms to introduce the technology (4). Therefore, migration to IPv6 environments is expected to be fairly complex. However, with the growth in the number of users and the IPv4 address exhaustion, large-scale deployment will invariably happen in the near future. IoT/M2M applications are expected to provide an impetus to the deployment of IPv6. Initially, internetworking between the two environments will be critical (6). Existing IPv4-endpoints and/or nodes will need to run dual-stack nodes or convert to IPv6 systems. Fortunately, the new protocol supports an IPv4-compatible IPv6 address that is an IPv6 address employing embedded IPv4 addresses. Tunneling, which we already described in passing, will play a major role in the beginning. There are a number of requirements that are typically applicable to an organization wishing to introduce an IPv6 service (30):

- The existing IPv4 service should not be adversely disrupted (e.g., as it might be by router loading of encapsulating IPv6 in IPv4 for tunnels);
- The IPv6 service should perform as well as the IPv4 service (e.g., at the IPv4 line rate, and with similar network characteristics);
- The service must be manageable and be able to be monitored (thus tools should be available for IPv6 as they are for IPv4);
- The security of the network should not be compromised, due to the additional protocol itself or a weakness of any transition mechanism used; and
- An IPv6 address allocation plan must be drawn up.

Well-known interworking mechanisms include the following, as described in RFC 2893:

- Dual IP layer (also known as dual stack): A technique for providing complete support for both IPs—IPv4 and IPv6—in hosts and routers;
- Configured tunneling of IPv6 over IPv4: Point-to-point tunnels made by encapsulating IPv6 packets within IPv4 headers to carry them over IPv4 routing infrastructures; and
- Automatic tunneling of IPv6 over IPv4: A mechanism for using IPv4-compatible addresses to automatically tunnel IPv6 packets over IPv4 networks.

Tunneling techniques include the following approaches, as described in RFC 2893:

- IPv6-over-IPv4 tunneling: The technique of encapsulating IPv6 packets within IPv4 so that they can be carried across IPv4 routing infrastructures.
- Configured tunneling: IPv6-over-IPv4 tunneling where the IPv4 tunnel endpoint address is determined by configuration information on the encapsulating node. The tunnels can be either unidirectional or bidirectional. Bidirectional configured tunnels behave as virtual point-to-point links.
- Automatic tunneling: IPv6-over-IPv4 tunneling where the IPv4 tunnel endpoint address is determined from the IPv4 address embedded in the IPv4-compatible destination address of the IPv6 packet being tunneled.
- IPv4 multicast tunneling: IPv6-over-IPv4 tunneling where the IPv4 tunnel endpoint address is determined using ND. Unlike configured tunneling, this does

not require any address configuration, and unlike automatic tunneling it does not require the use of IPv4-compatible addresses. However, the mechanism assumes that the IPv4 infrastructure supports IPv4 multicast.

Applications (and the lower-layer protocol stack) need to be properly equipped. Some examples of interoperability techniques include dual stacks and tunneling—IPv6-in-IPv4 (e.g., 6-to-4, 6rd, protocol 41), IPv4-in-IPv6, and IPv6-in-UDP (Teredo, TSP). There are four cases, as described in RFC 4038:

Case 1: IPv4-only applications in a dual-stack node. IPv6 protocol is introduced in a node, but applications are not yet ported to support IPv6. The protocol stack is as follows:

```
+----------------------+
|        appv4         | (appv4 - IPv4-only applications)
+----------------------+
|  TCP / UDP / others  | (transport protocols - TCP,
+----------------------+  UDP, and so on)
|     IPv4 | IPv6      | (IP protocols supported/enabled
+----------------------+   in the OS)
```

Case 2: IPv4-only applications and IPv6-only applications in a dual-stack node. Applications are ported for IPv6 only. Therefore, there are two similar applications, one for each protocol version (e.g., ping and ping6). The protocol stack is as follows:

```
+----------------------+ (appv4 - IPv4-only applications)
|    appv4 | appv6     | (appv6 - IPv6-only applications)
+----------------------+
|  TCP / UDP / others  | (transport protocols - TCP,
+----------------------+  UDP, and so on)
|      IPv4 | IPv6     | (IP protocols supported/
+----------------------+  enabled in the OS)
```

Case 3: Applications supporting both IPv4 and IPv6 in a dual-stack node. Applications are ported for both IPv4 and IPv6 support. Therefore, the existing IPv4 applications can be removed. The protocol stack is as follows:

```
+----------------------+
|       appv4/v6       | (appv4/v6 - applications
+----------------------+  supporting both IPv4 and IPv6)
|  TCP / UDP / others  | (transport protocols - TCP,
+----------------------+  UDP, and so on)
|      IPv4 | IPv6     | (IP protocols supported/
+----------------------+  enabled in the OS)
```

Case 4: Applications supporting both IPv4 and IPv6 in an IPv4-only node. Applications are ported for both IPv4 and IPv6 support, but the same applications may also have to work when IPv6 is not being used (e.g., disabled from the OS). The protocol stack is as follows:

```
+----------------------+
|       appv4/v6       | (appv4/v6 - applications
+----------------------+   supporting both IPv4 and IPv6)
|  TCP / UDP / others  | (transport protocols - TCP,
+----------------------+  UDP, and so on)
|         IPv4         | (IP protocols supported/
+----------------------+  enabled in the OS)
```

The first two cases are not interesting in the longer term; only a few applications are inherently IPv4 or IPv6 specific and should work with both protocols without having to care about which one is being used.

It should be noted that the transition from a pure IPv4 network to a network where IPv4 and IPv6 coexist brings a number of extra security considerations that need to be taken into account when deploying IPv6 and operating the dual-protocol network and the associated transition mechanisms (7, 31).

Figure 7.10 depicts some basic scenarios of carrier-based IPv6 support. Case (a) and (b) represent traditional environments where the carrier link supports either a clear channel that is used to connect, say, two IPv4 routers, or is IP aware. (In each case, the “cloud” on the left could also be the IPv4 Internet or the IPv6 Internet.)

In Case (c), the carrier link is used to connect as a transparent link two IPv6 routers; the carrier link is not (does not need to be) aware that it is transferring IPv6 PDUs. In Case (d), the carrier system is IPv4 aware, so the use of that environment to support IPv6 requires IPv6 to operate in a tunneled mode over the non-IPv6 cloud, which is a capability of IPv6.

In Case (e), the carrier infrastructure needs to provide a gateway function between the IPv4 and the IPv6 world (this could entail re-packing the IP PDUs from the v4 format to the v6 format). Case (f) is the ideal long-term scenario where the “world has converted to IPv6” and “so did the carrier network.”

In Case (g), the carrier IP-aware network provides a conversion function to support both IPv4 (as a baseline) and IPv6 (as a “new technology”) handoffs. Possibly a dual-stack mechanism is utilized. In Case (h), the carrier IPv6-aware network provides a support function for IPv6 (as a baseline) and also a conversion function to support legacy IPv4 islands.

Some user organizations have expressed concerns about security in an IPv6 environment, fundamentally because of tunneling and firewall issues. The interested reader should consult Reference 7 for an extensive discussion of this topic and for tools and techniques to address the issues. Even network/security administrators that operate in a pure IPv4 environment need to be aware of IPv6-related security issues. In a standard IPv4 environment where IPv6 is not explicitly supported, any form of IPv6-based tunneling traffic must be considered abnormal, malicious traffic. For

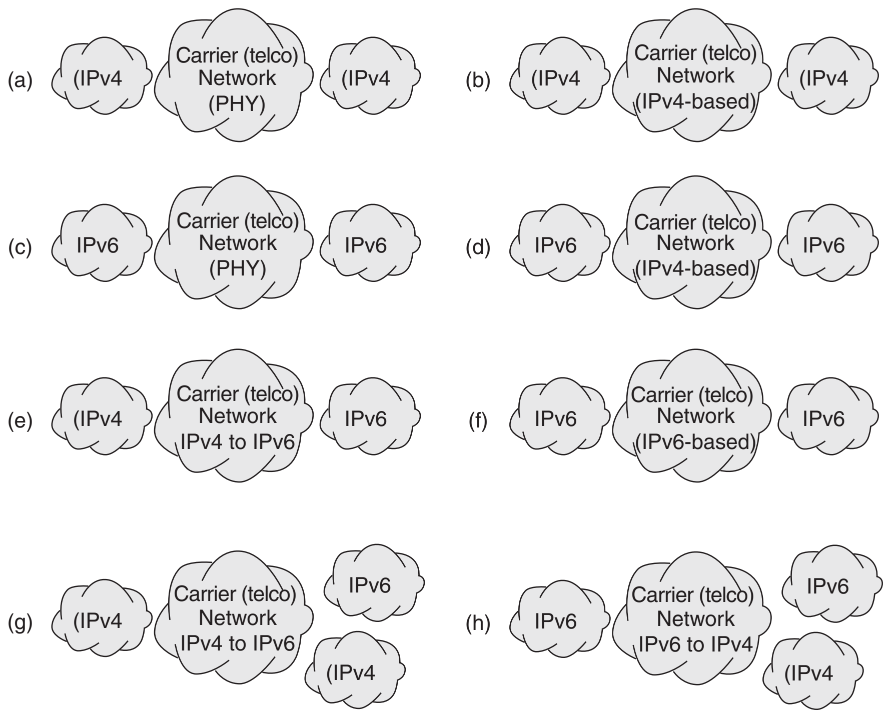

(a)
(IPv4
Carrier (telco) Network (PHY)
(IPv4
(b)
(IPv4
Carrier (telco) Network (IPv4-based)
(IPv4
(c)
IPv6
Carrier (telco) Network (PHY)
IPv6
(d)
IPv6
Carrier (telco) Network (IPv4-based)
IPv6
(e)
(IPv4
Carrier (telco) Network IPv4 to IPv6
IPv6
(f)
IPv6
Carrier (telco) Network (IPv6-based)
IPv6
(g)
(IPv4
Carrier (telco) Network IPv4 to IPv6
IPv6
(IPv4
(h)
IPv6
Carrier (telco) Network IPv6 to IPv4
IPv6
(IPv4

FIGURE 7.10 Support of IPv6 in carrier networks.

example, unconstrained 6to4-based traffic should be blocked (as noted elsewhere 6to4 is a transitional mechanism intended for individual independent nodes to connect IPv6 over the greater Internet). Most commercial-grade IPv4 firewalls block the IP protocol 41, the 6to4, and the tunnel protocol, unless it has been explicitly enabled (32).

7.8.2 Residential Broadband Services in an IPv6 Environment

One of the challenges related to the deployment of IPv6 is how to continue to support IPv4 services in residential broadband environments at the same time as the users migrate to a mixed IPv4 and IPv6 operational model. This is especially critical as IPv6 is technically incompatible with IPv4; this forces the introduction of some new concepts that change the present operation of broadband networks and has ramifications on how IPv6 can be offered to residential subscribers. Three approaches can be used, as covered in Reference 33 on which this discussion is based:

1. IPv6 support in telco environments using PPP over Ethernet (PPPoX) and/or asynchronous transfer mode (ATM) as defined in TR-187 of the Broadband Forum.

2. IPv6 support using PPPoX in conjunction with the bridged residential gateway (RG).
3. IPv6 support using IP over ethernet (IPoE) as defined in the Broadband Forum specification TR-177.

Approach 1. The introduction of IPv6 using PPPoX/layer 2 tunneling protocol (L2TP) has no implications on the access and aggregation network elements. PPP session authentication for IPv6 is identical to IPv4, using password authentication protocol/challenge handshake authentication protocol (PAP/CHAP) or option 82. IPv4 and IPv6 authentication can be done in a single authentication phase to RADIUS (remote authentication dial-in user service). Since PPPoX IPv6 control protocol (CP) is only defining the LLA, global IPv6 addresses are typically assigned using DHCP or SLAAC. To support an IPv6 routed RG using the PPP termination and aggregation/L2TP network server (PTA/LNS) model, the following mechanisms are required between the RG and the broadband network gateway/broadband remote access server (BNG/BRAS) to ensure IPv6 connectivity:

- PPPoX IPv6 CP is used for LLA assignment
- DHCPv6 prefix delegation (IA-PD—identity association for prefix delegation) is used to obtain a prefix for LAN address assignment
- Stateless DHCPv6 is used to obtain additional configuration parameters
- When the numbered RG model is deployed, stateful DHCPv6 (identity association for non-temporary addresses [IA-NA]) is used to obtain an RG management IPv6 address; in case of an unnumbered RG model, this is not required
- Route advertisements are required to assign the default gateway assignment

Approach 2. The utilization of the Bridged RG requires the following:

- PPPoX IPv6CP is used for LLA assignment
- SLAAC is used for the host to obtain a global-Unicast IPv6 address
- Stateless DHCP is used to obtain additional configuration parameters
- Route advertisements are used to assign the default gateway assignment

Therefore, to support IPv6 in a telco environment, PPPoX for IPv6 imposes no different requirements on N:1 virtual local area network (VLAN) or 1:1 VLAN architectures or on a bridged gateway model, compared to IPv4. However, PPPoX for IPv6 will always impact the BNG/BRAS, CPE, and home gateway using a routed gateway model.

Approach 3. The implications for introducing IPv6 IPoE mainly depend on the VLAN model used (1:1 or N:1) and the operational model of the home gateway (bridged or routed). The impact of IPv6 support for IPoE in a bridged RG model depends on whether DHCP or SLAAC is used to the end device. When deploying DHCP, the key difference from the routed RG IPoE model arises from the fact that there is no DHCP PD address required and only an IA address is assigned to the host. Care must be taken to ensure communication between IPv6 devices in the home remains local and is not sent through the BNG.

7.8.3 Deployment Opportunities

There was a lack of ubiquitous IPv6 utilization as of early 2013; this is partly due to the fact that the number of IPv6 nodes is rather low. However, IPv6 rollout has started to get traction. The approaching exhaustion of IPv4 address space will bring about a situation where ISPs are faced with a choice between one or more of three major alternatives (24):

1. Squeeze the use of IPv4 addresses even harder than today, using smaller and smaller address blocks per enterprise customer, and possibly trading address blocks with other ISPs.
2. Install multiple layers of NAT or share IPv4 addresses by other methods, such as address-plus-port mapping.
3. Deploy IPv6 and operate IPv4–IPv6 coexistence and interworking mechanisms.

RFC 5514 (April 2009) proposed to vastly increase the number of IPv6 hosts by transforming all social networking platforms into IPv6 networks. This would immediately add millions of IPv6 hosts to the existing IPv6 Internet.

Hosts (PCs, servers) and network infrastructure (routers, switches) are generally IPv6 ready at this time, but organizations may need to upgrade their overall end-to-end environment. Service providers such as Google have already rolled out an IPv6 site for customers already on that system. In fact, Google, Facebook, Yahoo!, Akamai, and Limelight Networks are among some of the larger companies that planed a one-day test run of IPv6 addresses as part of World IPv6 Day, on June 8, 2011, to encourage the transition to the new namespace. These organizations were planning to offer their content over IPv6 for a 24-hour “test flight,” with the goal of the Test Flight Day being to motivate organizations across the industry—ISPs, hardware makers, operating system (OS) vendors, and web companies—to prepare their services for IPv6 to ensure a successful transition as IPv4 addresses run out. Internet users did not need to do anything different on World IPv6 Day. Web services, ISPs, and OS manufacturers were planning to be updating their systems to ensure Internet users receive uninterrupted service. In rare cases, users may still experience connectivity issues when visiting participating websites. Users were able to visit an IPv6 test site to check if their connectivity was impacted. Organizations that wanted to bring their company’s website online using IPv6 during the World IPv6 Day needed to make it IPv6 accessible using dual-stack technology and provide an AAAA record for the site. Of course, IPv4 websites continued to be accessible over IPv4 during the event.

According to the Internet Society (ISOC), the World IPv6 Day saw more than 1000 major website operators switch over to IPv6-compatible main pages in the most extensive live run of the next-generation addressing protocol so far. The day turned out to be a technological success. Approximately two-thirds of the participants were reportedly so pleased with the results they left IPv6 enabled on their equipment going forward. Nonetheless, just 0.16% of Facebook users were IPv6 natives and 0.04% were using 6to4 tunneling capabilities, delivering around 1 million IPv6 visitors over the course of the day.

In DNS, host names are mapped to IPv6 addresses by AAAA (also known as Quad A) resource records (RRs). The IETF specifies the use AAAA RR for forward mapping and pointer RRs (PTRs) for reverse mapping. The IPv6 AAAA RR approach is described in RFC 3596. The forward DNS entry for an IPv6 entry in entered using AAAA. It can be entered using the full IPv6 address or by using the shorthand :: notation. PTRs are the opposite of AAAA RRs and are used in reverse map zone files to map an IPv6 address to a host name.

Tier 1 telecommunication firms have been upgrading their infrastructure over the past few years in anticipation of the eventual transition. The same has occurred for content providers. For example, Comcast has begun assigning IPv6 addresses to its cable modem customers in a "native dual-stack" configuration as of early 2011; under this configuration, customers have both IPv4 and IPv6 addresses and can access content and services over both systems. Comcast's first 25 IPv6-enabled customers went live January 11, 2011, in the Littleton, Colo. TimeWarner Cable has already signed up commercial customers on IPv6 and was planning to begin residential IPv6 trials in early 2011. TimeWarner Cable is also expected to adopt a dual-stack approach similar to that of Comcast. Domain infrastructure company VeriSign will also provide business services to assist companies with the transition in 2011 (4).

Having ISPs deploy IPv6 to customers' sites, in addition to IPv4 and without extra charge, is a way to break the existing impasse that has delayed IPv6 deployment: ISPs wait for customer demand before deploying IPv6; customers do not demand IPv6 as long as application vendors announce that their products work on existing infrastructures (that are based on IPv4 with NATs); application vendors focus their investments on NAT traversal compatibility as long as ISPs do not deploy IPv6. However, most ISPs are not willing to add IPv6 to their current offerings at no charge unless incurred investment and operational costs are small. For this, ISPs that provide router customer premise equipment (CPE) to their customers have the most favorable conditions: they can upgrade their router CPEs and can operate gateways between their IPv4 infrastructures and the global IPv6 Internet to support IPv6 encapsulation in IPv4. They then need no additional routing plans than those that already exist on these IPv4 infrastructures. Encapsulation using 6to4 methods, as specified in RFC 3056, is nearly sufficient for this: (i) it is simple; (ii) it is supported on many platforms including PC-compatible appliances; (iii) open-source portable code is available; and (iv) its stateless nature ensures good scalability. There is, however, a limitation of 6to4 that prevents ISPs from using it to offer full IPv6 unicast connectivity to their customers. While an ISP that deploys 6to4 can guarantee that IPv6 packets outgoing from its customer sites will reach the IPv6 Internet, and also can guarantee that packets coming from other 6to4 sites will reach its customer sites, it cannot guarantee that packets from native IPv6 sites will reach them. The problem is that a packet coming from a native IPv6 address needs to traverse (somewhere on its way) a 6to4 relay router to do the required IPv6/IPv4 encapsulation. There is no guarantee that routes toward such a relay exist from everywhere, nor is there a guarantee that all such relays do forward packets toward the IPv4 Internet. Also, if an ISP operates one or several 6to4 relay routers and opens IPv6 routes toward them in the IPv6 Internet, for the 6to4 prefix 2002::/16, it may receive in these relays packets destined to an unknown number of other 6to4 ISPs. If it does not forward these packets, it creates a “black hole” in which packets may be systematically lost, breaking some of the IPv6 connectivity. If it does forward them, it can no longer dimension its 6to4 relay routers in proportion to the traffic of its own customers; QoS, at least for customers of other 6to4 ISPs, will then not be guaranteed (34). To address these issues RFC 5569, *6rd—IPv6 Rapid Deployment*, also known simply as *6rd*, proposes to slightly modify 6to4 so that:

1. Packets coming from the global Internet, entering *6rd* gateways of an ISP are only packets destined to customer sites of this ISP.
2. All IPv6 packets destined to *6rd* customer sites of an ISP, and coming from anywhere else on the IPv6 Internet, traverse a *6rd* gateway of this ISP.

The principle of the RFC 5569 proposal is that to build on 6to4 and suppress its limitation, it is sufficient that:

1. 6to4 functions are modified to replace the standard 6to4 prefix 2002::/16 by an IPv6 prefix that belongs to the ISP-assigned address space, and to replace the 6to4 anycast address by another anycast address chosen by the ISP.
2. The ISP operates one or several 6rd gateways (upgraded 6to4 routers) at its border between its IPv4 infrastructure and the IPv6 Internet.
3. CPEs support IPv6 on their customer-site side and support 6rd (upgraded 6to4 function) on their provider side.

There is no guarantee that this proposal will be broadly accepted, but it represents one press-time approach for IPv6 deployment.

REFERENCES

1. Minoli D. *IP Multicast with Applications to IPTV and Mobile DVB-H*. New York: Wiley; 2008.
2. Minoli D. *Satellite Systems Engineering in an IPv6 Environment*. Francis and Taylor; 2009.
3. Minoli D. *Voice Over IPv6 – Architecting the Next-Generation VoIP*. New York: Elsevier; 2006.
4. Rashid FY. IPv4 Address Exhaustion Not Instant Cause for Concern with IPv6 in Wings, Eweek, 2011-02-01.
5. The IPv4 Address Report, Online resource, http://www.potaroo.net.

6. Minoli D, Amoss J. *Handbook of IPv4 to IPv6 Transition Methodologies For Institutional & Corporate Networks*. New York: Auerbach/CRC; 2008.
7. Minoli D, Kouns J. *Security in an IPv6 Environment*. Taylor and Francis; 2009.
8. An IPv6 Security Guide for U.S. Government Agencies—Executive Summary, The IPv6 World Report Series, Volume 4 February 2008, Juniper Networks, 1194 North Mathilda Avenue, Sunnyvale, CA 94089 USA.
9. Kaeo M, Green D, Bound J, Pouffary, Y. IPv6 Security Technology Paper. North American IPv6 Task Force (NAv6TF) Technology Report, July 22, 2006.
10. Lioy A. Security Features of IPv6Security Features of IPv6, Chapter 8 of *Internetworking IPv6 with Cisco Routers* by Silvano Gai McGraw-Hill, 1998; also available at www.ip6.com/us/book/Chap8.pdf.
11. Johnson D, Perkins C, Arkko J. Mobility Support in IPv6. RFC 3775, June 2004.
12. Hinden R, Deering S. IP Version 6 Addressing Architecture. RFC 4291, February 2006.
13. Machine-to-Machine Communications (M2M); M2M Service Requirements. ETSI TS 102 689 V1.1.1 (2010-08). 650 Route des Lucioles F-06921 Sophia Antipolis Cedex—FRANCE.
14. Microsoft Corporation, MSDN Library, Internet Protocol, 2004, http://msdn.microsoft.com.
15. Blanchet M. Special-Use IPv6 Addresses. draft-ietf-v6ops-rfc3330-for-ipv6-04.txt, January 15, 2008.
16. Hermann-Seton, P. Security Features in IPv6, SANS Institute 2002, As part of the Information Security Reading Room.
17. Thomson S, Narten T, Jinmei T. IPv6 Stateless Address Autoconfiguration. RFC 4862, September 2007.
18. Donzé F. IPv6 Autoconfiguration. The Internet Protocol Journal, June 2004;7 (2). Published Online, http://www.cisco.com.
19. Narten T, Draves R, Krishnan S. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 4941, September 2007.
20. Narten T, Nordmark E, Simpson W, Soliman H. Neighbor Discovery for IP version 6 (IPv6). RFC 4861, September 2007.
21. IPv6 Portal, http://www.ipv6tf.org.
22. Desmeules R. *Cisco Self-Study: Implementing Cisco IPv6 Networks (IPv6)*. Cisco Press; June 6, 2003.
23. Conta A, Deering S. Generic Packet Tunneling in IPv6 Specification. RFC 2473, December 1998.
24. Carpenter B, Jiang S. Emerging Service Provider Scenarios for IPv6 Deployment. RFC 6036, October 2010.
25. Ertekin E, Christou C. IPv6 Header Compression. North American IPv6 Summit, June 2004.
26. Jonsson L-E, Pelletier G, Sandlund K. The RObust Header Compression (ROHC) Framework. RFC 4995, July 2007.
27. Pelletier G, Sandlund K, Jonsson L-E, West M. RObust Header Compression (ROHC): A Profile for TCP/IP (ROHC-TCP). RFC 4996, July 2007.
28. Finking R, Pelletier G. Formal Notation for RObust Header Compression (ROHC-FN). RFC 4997, July 2007.

29. Varada S, editor. IPv6 Datagram Compression. RFC 5172, March 2008.
30. 6NET. D2.2.4: Final IPv4 to IPv6 Transition Cookbook for Organizational/ISP (NREN) and Backbone Networks. Version: 1.0 (4th February 2005), Project Number: IST-2001-32603, CEC Deliverable Number: 32603/UOS/DS/2.2.4/A1.
31. Davies E, Krishnan S, Savola P. IPv6 Transition/Co-existence Security Considerations. RFC 4942, September 2007.
32. Warfield MH. Security Implications of IPv6", 16th Annual FIRST Conference on Computer Security Incident Handling, June 13–18, 2004—Budapest, Hungary.
33. Henderickx W. Making the Move to IPv6, alcatel-lucent White Paper, September 20, 2011.
34. Despres R. 6rd - IPv6 Rapid Deployment. RFC 5569, January 2010.