Chapter 2

The IPv6 Protocol

This chapter describes the IPv6 protocol. It examines the fields in the IPv4 header and explores both the similarities and the differences. It explains how IPv6 offers much more than just a larger address space. It's a new protocol designed to be more flexible and more efficient.

The structure of the IPv6 header is discussed in RFC 2460, Internet Protocol, Version 6 (IPv6) Specification. In addition to the main IPv6 header, a new type of IPv6 header known as an Extension header is also examined. The end of this chapter contains a summary that includes the differences between the IPv4 and IPv6 headers.

IPv4 Header

To help you better understand the IPv6 header, we will first take a look at the IPv4 header. This can be a review for you, or it can provide a better understanding of the IPv4 header. In either case, it will clarify the differences in the IPv6 header. Figure 2-1 shows the structure of the IPv4 header as defined in RFC 791, Internet Protocol, DARPA Internet Program, Protocol Specification. The IPv6 header is shown in Figure 2-2, later in this chapter. A quick comparison of the two reveals that the IPv6 header is a simpler protocol—with fewer fields. This makes for a leaner protocol and more efficient processing.

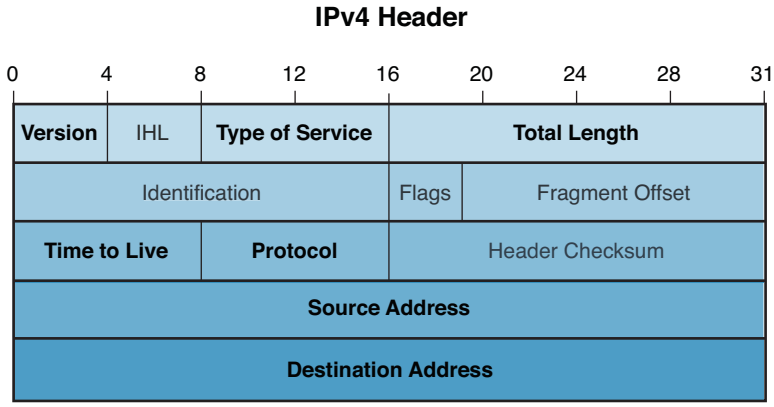

Figure 2-1 *IPv4 Header*

Note The next section examines the individual fields in the IPv4 header and is included for continuity. This section serves as a reference and a review of IPv4 for those who might need this data. This information can be helpful during the comparison of the IPv4 and IPv6 headers discussed later in this chapter.

The fields in the IPv4 header are as follows:

- **Version (4 bits):** This field contains the version number of the IP (Internet Protocol) header. In IPv4, this field is always the value of 4.
- **Internet Header Length (4 bits):** This is the length of the IP header in 32-bit words, including any optional fields. In effect, this points to where the IP header ends and the data or payload begins. The minimum value is 5 (5 * 32-bit words = 160 bits or 20 octets [bytes]). This is equal to the minimum size of an IPv4 header, excluding any options or padding.
- **Type of Service (8 bits):** The Type of Service, or ToS, field specifies what type of treatment the packet should receive from routers. The ToS information helps provide quality of service (QoS) features by offering different degrees of precedence. When multiple packets are queued to be transmitted out the same interface, the ToS value can be used to decide which is sent first. The ToS field was not widely used as originally designed, so in 1998, the Internet Engineering Task Force (IETF) redefined it further in RFC 2474 using a technique called Differentiated Services (DS). Differentiated Services Code Point (DSCP) is discussed in more detail later in this chapter, but ToS and DS are beyond the scope of this book.
- **Total Length (16 bits):** This is the length of the IP packet, measured in octets (bytes), including the IP header and the data. This is a 16-bit field, so the maximum size of an IPv4 packet is 65,535 bytes. Most IPv4 packets are much smaller.

The next three fields are used for packet fragmentation and reassembly. IP was designed for a wide variety of transmission links. Most transmission links enforce a maximum packet length known as the MTU (maximum transmission unit). The design of IP accommodates MTU differences by allowing routers to fragment IP packets when an MTU along the path is smaller than the sender's MTU. If a router receives an IPv4 packet that is larger than the MTU of the outgoing interface, this packet can be fragmented depending upon the options in the IPv4 header. Sometimes packets are fragmented into multiple packets at the source. The final destination of the IP packet is responsible for reassembling the fragments into the original full-size IP packet.

- **Identification (16 bits):** Most messages sent over the network consist of many packets. Each packet within the message has a unique value using this 16-bit Identification field. When a packet needs to be fragmented into two or more packets, this Identification field is common in all the fragmented packets to help the receiver in reassembling these fragments.
- **Flags (3 bits):** The first bit is 0, which means it is reserved or not used. The second bit is known as the DF, or Don't Fragment, bit. When set to 1, it means that this packet should not be fragmented. However, most protocols don't care about the fragmentation process and set this flag to 0, which means that this packet can be fragmented if needed. The third bit is the More Fragments Flag and is used to indicate whether this is the last fragment (0 bit) or whether there are more fragments to follow (1 bit). If a packet is not fragmented, there is only one fragment—the entire packet—and this flag is set to 0.

Note The DF flag is very useful when testing the MTU of a path between the source and the destination. If the DF flag is set to 1, the packet should not be fragmented. Any router along the path whose MTU is smaller than the packet will drop the packet and send an Internet Control Message Protocol (ICMP) "Destination Unreachable" message back to the source. The ICMP message will include the MTU of the router's egress interface. Path MTU Discovery for IPv4 is beyond the scope of our discussion. RFC 1191, Path MTU Discovery, explains this process if you are interested in learning more. Path MTU Discovery for IPv6 is discussed later in this chapter.

- **Fragment Offset (13 bits):** When a packet is fragmented, this field specifies the offset or position where this data goes in units of 8 octets (64 bits). Basically the fragment offset field notifies the receiver where to align this fragmented packet in relation to the other fragmented packets. The first fragment has offset zero. If the packet is not fragmented, this value is 0.
- **Time to Live (8 bits):** The Time to Live (TTL) field ensures that packets do not live in the network for an indefinite period of time as in the case of a routing loop. The TTL is decremented by 1 each time a router receives the packet. When the field contains the value of 0, the packet is discarded and an ICMPv4 (Internet Control Message

Protocol version 4) Time Exceeded (Type 11) message is sent to the source of the packet.

Note The Time to Live field was originally intended to represent the actual maximum amount of time that the packet is allowed to traverse the network and not the necessarily the number of router hops. RFC 791 stated, "Even if no local information is available on the time actually spent, the field must be decremented by 1. The time is measured in units of seconds (i.e., the value 1 means one second). Thus, the maximum time to live is 255 seconds or 4.25 minutes." Instead of calculating the amount of time, routers just decrement the TTL by 1, in effect making it the number of hops.

- **Protocol (8 bits):** This field indicates the protocol carried in the data portion of the IP packet. The values for various protocols are specified in RFC 1700, Assigned Numbers, and were later replaced by an online database maintained by IANA at www.iana.org/assignments/protocol-numbers/protocol-numbers.xml. Some of the more common values are 1 for ICMP, 6 for TCP, and 17 for UDP.

Note At times this field is referred to as carrying an upper-layer protocol. This can be misleading because the protocol being carried as the data or payload might be another Layer 3 protocol such as ICMP or even IP.

- **Header Checksum (16 bits):** A checksum for the IP header is provided for protection against any corruption in transit. This is not the more complex CRC (Cyclic Redundancy Check) used by Ethernet but a much simpler 16-bit checksum performed only on the IP header. Each router along the path verifies and recomputes this field. If the checksum fails, the router discards the packet.
- **Source Address (32 bits):** This is the 32-bit IP address of the originator of the packet.
- **Destination Address (32 bits):** This is the 32-bit IP address of the final destination or recipient of the packet. This address is used by routers to forward the packet along its path toward its ultimate destination.

Note Network Address Translation (NAT) can change either the source or destination address to one of the translator's addresses, typically an RFC 1918 private IP address. NAT is outside the scope of this book. See RFC 4787 for more information.

- **Options (variable length):** This field is optional, so it might or might not appear in the IP packet. It is variable in size and not included in most packets. Some of the options contain record route, timestamp, and traceroute used as an enhancement to the traceroute utility and described in RFC 1393, Traceroute Using an IP Option.

- **Padding (variable length):** If one or more options are used, and the size of the IP header is no longer a multiple of 32 bits, 0 bits are added to pad out the header so that it ends on a 32-bit boundary.
- **Data (variable length):** This is the data to be transmitted in the IP packet and identified by the Protocol field. The data can be another Layer 3 protocol such as ICMP, or a higher-layer protocol such as TCP or UDP.

IPv6 Header

IPv6 is defined in RFC 2460, Internet Protocol, Version 6 (IPv6) Specification. Figure 2-2 shows the basic structure of the IPv6 header or what is sometimes referred to as the main IPv6 header. The main IPv6 header can also include one or more IPv6 extension headers. Extension headers are explained later in this chapter.

The IPv6 header and its fields are examined in Figure 2-2.

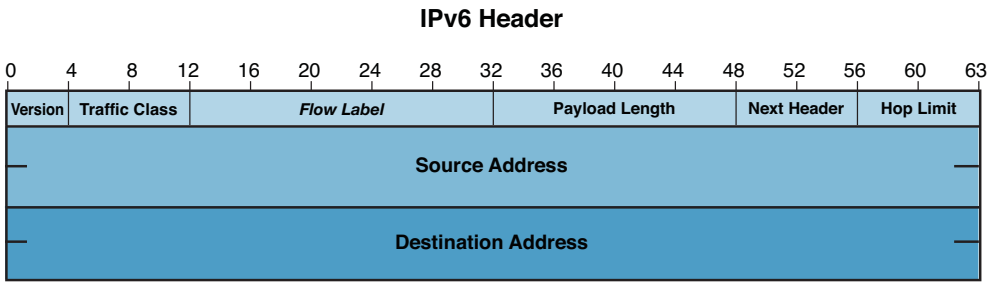

Figure 2-2 *IPv6 Header*

The IPv6 main header is required and consists of the following fields:

- **Version (4 bits):** This field contains the version number of the IP (Internet Protocol) header. In IPv6, this field is always the value of 6.
- **Traffic Class (8 bits):** This field has similar functions to the Type of Service (ToS) field in the IPv4 header. It is the same size as the IPv4 ToS field; only the name has changed. The Traffic Class field is used to identify and distinguish between different classes or priorities of IPv6 packets. IPv6 uses the Differentiated Services technique specified in RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. Using 6 bits for the DSCP (Differentiated Services Code Point) allows a possibility of 64 markings. This provides much more granularity in priority selection than the original 3 bits for IPv4 Precedence, with its 8 values.

Note DSCP and IP Precedence are beyond the scope of this book. One interesting note is that the IP Precedence value is actually the first 3 bits of the DSCP value, as shown in Figure 2-3. Therefore, both values cannot be used simultaneously. If DSCP with its additional 3 bits is used, it supersedes IP Precedence.

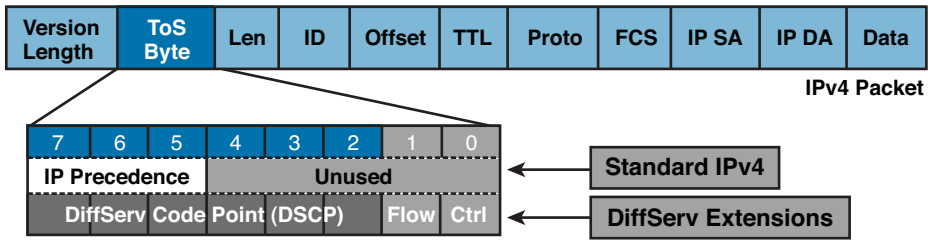

Figure 2-3 *IPv4 ToS Byte*

- **Flow Label (20 bits):** The Flow Label field is used to tag a sequence or flow of IPv6 packets sent from a source to one or more destination nodes. This flow can be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as "real-time" service. The Flow Label field is used to help identify all the packets within the same flow to ensure that all the packets receive the same type of handling by the IPv6 routers. Flow Label usage is described in RFC 6437, IPv6 Flow Label Specification. Routers keep track of these individual packet flows. Because routers do not have to independently process each packet's header, these multipacket flows are processed more efficiently. At the time of this writing, this field is still somewhat experimental.

- **Payload Length (16 bits):** This is the length in octets of the payload following the main IP header or, in other words, the data portion of the packet. If the IPv6 packet has one or more extension headers, they are included in the number of bytes contained in the Payload Length field. Extension headers are considered part of the payload. The IPv6 Payload Length field is similar to the Total Length field in the IPv4 header, except for one important difference. IPv4's Total Length field includes both the IPv4 header and the data, whereas the IPv6 Payload Length field only specifies the number of bytes of data; it does not include the main IPv6 header. The IPv4 header can vary in length because of Padding and Options fields, whereas the IPv6 header is fixed at 40 bytes.

  The Payload Length field is 16 bits, allowing a maximum payload size of 65,535 bytes. IPv6 has a Jumbogram extension header to support larger packet sizes if needed. RFC 2675, IPv6 Jumbograms, specifies an additional 32-bit field to allow the transmission of IPv6 packets with payloads between 65,536 and 4,294,967,295 bytes. Extension headers along with the Jumbo Payload Options are discussed later in this chapter.

- **Next Header (8 bits):** This field has two benefits. In a situation when there is only the main IPv6 header and no extension headers, the Next Header field specifies the protocol carried in the data portion of the IPv6 packet. This is similar to the Protocol field in the IPv4 header. The same values used in the IPv4 Protocol field are used in the IPv6 Next Header field along with additional values. Table 2-1 shows some of the significant IPv6 Next Header values. A complete listing can be found at www.iana.org/assignments/protocol-numbers/protocol-numbers.xml. You might recognize some of these, as many are the same values used by the Protocol field in IPv4 such as 6 for UDP and 17 for TCP. Figure 2-4 shows several examples including the Next Header field, indicating that the data portion of the IPv6 packet is a TCP segment.

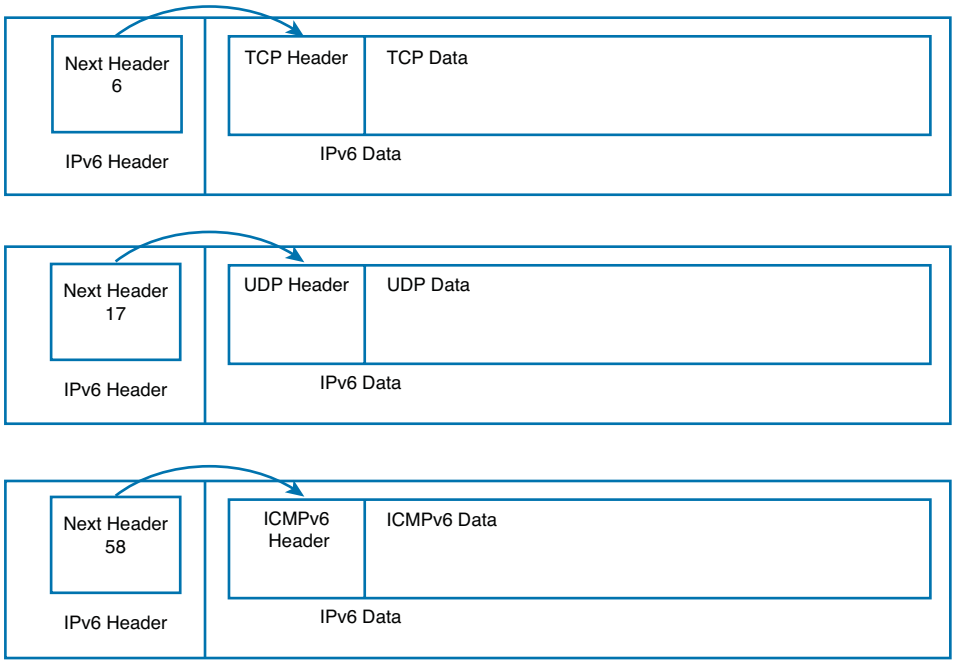

Figure 2-4 *Next Header Field*

Table 2-1 *Significant IPv6 Next Header Values*

Next Header Value (Decimal)	Next Header Value (Hexadecimal)	Description
0	0	Hop-by-hop options extension header for IPv6
1	1	Internet Control Message Protocol version 4 (ICMPv4)
2	2	Internet Group Management Protocol version 4 (IGMPv4)
4	4	IPv4 encapsulation
5	5	Internet Stream Protocol (ST)
6	6	Transmission Control Protocol (TCP)
8	8	Exterior Gateway Protocol (EGP)
17	11	User Datagram Protocol (UDP)
41	29	IPv6 encapsulation
43	2B	Routing extension header for IPv6
44	2C	Fragment header for IPv6

Next Header Value (Decimal)	Next Header Value (Hexadecimal)	Description
46	2E	Resource Reservation Protocol (RSVP)
47	2F	Generic Routing Protocol (GRE)
50	32	Encapsulation Security Protocol (ESP)
51	33	Authentication Header (AH)
58	3A	Internet Control Message Protocol version 6 (ICMPv6)
59	3B	No Next Header for IPv6
60	3C	Destinations options extension header for IPv6
88	58	Enhanced Interior Gateway Routing Protocol (EIGRP)
89	58	Open Shortest Path First (OSPF)

Note Tunneling is discussed in Chapter 10, "Dual-Stack and Tunneling," but for now, notice that an IP packet can encapsulate another IP packet. The Next Header field contains the value 4 when the following header is an IPv4 header, and 41 to indicate another IPv6 header.

- **Hop Limit (8 bits):** The Hop Limit field is equivalent to the Time to Live (TTL) field in the IPv4 header. Its name in IPv6 is more reflective of the way that routers treat this field by decrementing the hop limit by 1. Just as with the IPv4 TTL field, if the router decrements the hop limit from 1 to 0, the packet is discarded. In IPv6, an ICMPv6 Time Exceeded message (Type 3, Code 0) is sent to notify the source of the packet that the packet has been dropped. I will discuss ICMPv6 in Chapter 5, "ICMPv6 and Neighbor Discovery Protocol."
- **Source Address (128 bits):** This field contains the 128-bit IP address of the originator of the IPv6 packet. As with IPv4, this is the address of the node that originally sent the packet. The source address must be a unicast address.
- **Destination Address (128 bits):** This is the 128-bit IP address of the intended final destination or recipient of the IPv6 packet. It represents the ultimate destination, which can be a unicast or multicast address. Unlike IPv4, there is no broadcast address; however, there is an all-nodes multicast address. The details of IPv6 addresses are discussed in Chapter 3, "IPv6 Addressing," and Chapter 4, "IPv6 Address Types."

Packet Analysis Using Wireshark

Let's take a look at an IPv6 packet using a packet analyzer such as Wireshark. Based on the network setup shown in Figure 2-5, we will do a simple ping from PC1 to PC2.

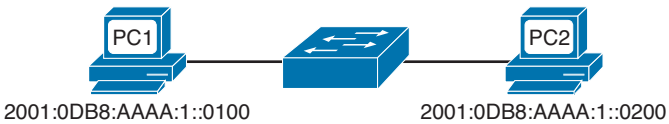

Figure 2-5 *PC1 Pinging an IPv6 Address on PC2*

```
PC1> ping 2001:0db8:aaaa:0001::0200

Pinging 2001:db8:aaaa:1::200 from 2001:db8:aaaa:1::100 with 32 bytes of data:
Reply from 2001:db8:aaaa:1::200: time<1ms
Reply from 2001:db8:aaaa:1::200: time<1ms
Reply from 2001:db8:aaaa:1::200: time<1ms
Reply from 2001:db8:aaaa:1::200: time<1ms

Ping statistics for 2001:db8:aaaa:1::200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC1>
```

Notice that some of the 0s in the addresses have been omitted in the output. IPv6 addresses and compressed formats are also discussed in Chapter 3. For now, be assured that these are the same addresses.

The IPv6 addresses are 128-bit addresses and written in hexadecimal. They might look a little strange right now, but don't worry, you will become very familiar with these addresses beginning in Chapter 3. Figure 2-6 shows the Wireshark capture of the ICMPv6 Echo Request (ping), which is further described in Table 2-2. ICMPv6 will be explained in more detail in Chapter 5.

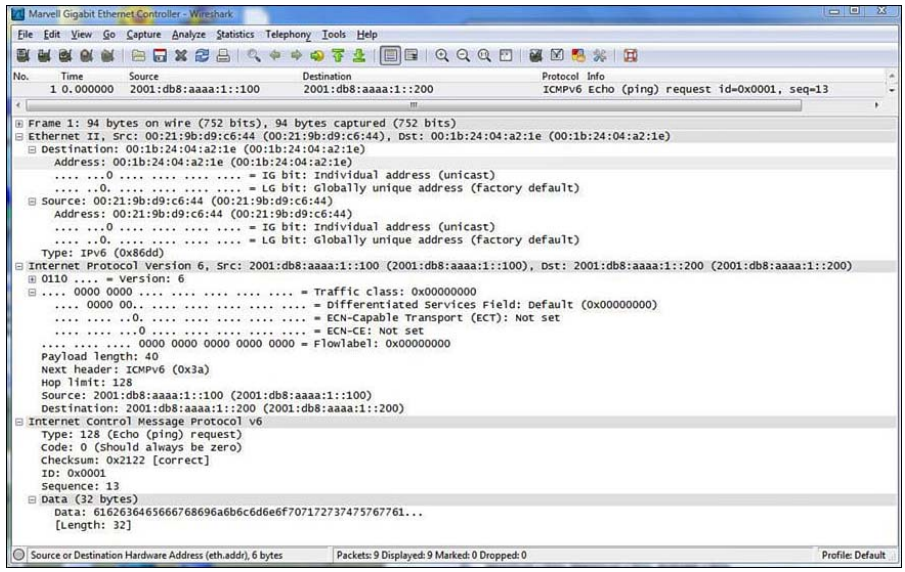

Figure 2-6 *Wireshark Capture of an IPv6 Packet*

Table 2-2 *Analysis of an IPv6 Packet*

	Field Name	Size (Bits)	Value – Description
IPv6 Header (40 Bytes)	Version	4	6 – Indicates IP version 6
	Traffic Class	8	0 – Default of 0
	Flow Label	20	0 – Default of 0
	Payload Length	16	40 bytes – This indicates the size of the data, which in this case is an ICMPv6 message. Notice that the ICMPv6 message has a length of 40 bytes.
	Next Header	8	58 – This identifies the following header as an ICMPv6 header. See Table 2-1 for a partial listing of IPv6 Next Header values.
	Hop Limit	8	128 – The maximum number of routers that this packet can traverse before being discarded.
	Source Address	128	2001:0db8:aaaa:1::100 – The source IPv6 address in hexadecimal.
	Destination Address	128	2001:0db8:aaaa:1::200 – The destination IPv6 address in hexadecimal.

	Field Name	Size (Bits)	Value – Description
ICMPv6 Header (40 Bytes)	Type	8	128 – Identifies this as an ICMPv6 Echo Request message.
	Code	8	0 – Not used; default is 0.
	Checksum	16	0x2122 – 16-bit checksum that is used to verify the integrity of the ICMPv6 header.
	ID	16	0x0001 – Used to help match ICMPv6 Echo Request and Echo Reply messages.
	Sequence	16	13 – Used to help match ICMPv6 Echo Request and Echo Reply messages.
	Data	256	Optional data of variable length depending upon the type of ICMPv6 message.

Note Wireshark is a network protocol analyzer for both IPv4 and IPv6. Wireshark is available for several operating systems and is a free downloaded at www.wireshark.org.

Extension Headers

Extension headers can be difficult to understand, so this chapter guides you with a step-by-step process. Some of them are fairly easy to grasp while others can be a little more difficult. So, don't be too concerned if some of this seems a little vague. The purpose of this section is to familiarize you with the concept of what extension headers are and how they are used.

Extension headers are optional and follow the main IPv6 header. As discussed previously, the IPv6 header includes a Next Header field, which has one of two purposes:

- To identify the protocol carried in the data portion of the packet
- To identify the presence of an extension header

As explained previously, the Next Header field can indicate which protocol is being carried in the data portion of the IPv6 packet, similar to IPv4's Protocol field and illustrated in Figure 2-5.

The second purpose is one of the more important additions to the IPv6 header, the indication of an additional header known as an extension header. Immediately following the mandatory main IPv6 header, there can be zero, one, or several extension headers. A field common in all extension headers is another Next Hop field, which indicates

whether another extension header is to follow, or the protocol of the data (payload) like a TCP segment. Therefore, the last extension header will always specify which protocol is encapsulated as the data or payload—again, similar to the Protocol field in IPv4.

The intention of extension headers is to provide flexibility to the main IPv6 header for future enhancements without having to redesign the entire protocol. This also allows the main IPv6 header to have a fixed size for more efficient processing.

There are currently six extension headers as defined in RFC 2460 and summarized in Table 2-3. If you remember, in IPv4 there is a little-used Options field of variable length to provide some additional flexibility. Two of the extension headers in IPv6 provide a similar function: the Hop-by-Hop Options and the Destination Options headers. Figure 2-7 shows an example of an IPv6 packet using two extension headers.

- The main IPv6 header has all the fields we have discussed, including the source and destination addresses. The Next Hop field is also in this header with the value of 0, indicating that a Hop-by-Hop extension header immediately follows.
- The Hop-by-Hop extension header follows the main IPv6 header. Extension headers will be explained in more detail in the next section, but for now, notice that this field also contains its own Next Header field. Its value of 51 signifies that there is yet another extension header to follow, the Authentication Header (AH).
- The final extension header is the Authentication Header. Its Next Header field has a value of 6, indicating that a TCP upper-layer header is to follow. This also means that there are no more extension headers in this packet.

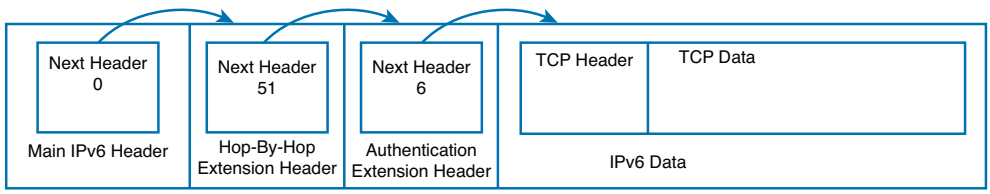

Figure 2-7 *Use of the Next Header Field in Extension Headers*

Note The Next Header field is used to chain together multiple IPv6 headers and the data portion of the packet at the end of the chain.

Table 2-3 IPv6 Extension Headers

Next Header Value (Decimal)	Extension Header Name	Extension Header Length (Bytes)	Variable-Length Options (TLV) Used?	Extension Header Description
0	Hop-by-Hop Options	Variable	Yes	Used to carry optional information, which must be examined by every router along the path of the packet.
43	Routing	Variable	No	Allows the source of the packet to specify the path to the destination.
44	Fragment	8	No	Used to fragment IPv6 packets.
50	Encapsulating Security Payload (ESP)	Variable	No	Used to provide authentication, integrity, and encryption.
51	Authentication Header (AH)	Variable	No	Used to provide authentication and integrity.
60	Destination Options	Variable	Yes	Used to carry optional information that only needs to be examined by a packet's destination node(s).

RFC 2460 recommends that when multiple extension headers are used in the same packet, those headers appear in the following order:

1. Main IPv6 header
2. Hop-by-Hop Options header
3. Destination Options header
4. Routing header
5. Fragment header
6. Authentication header
7. Encapsulating Security Payload header
8. Destination Options header
9. Upper-layer protocol header

Hop-by-Hop Options Extension Header

The Hop-by-Hop Options header is used to carry optional information that must be examined by every router along the path of the packet. The Hop-by-Hop Options header is one of two extension headers that contains a variable-length options field similar to IPv4. As the name implies, this type of option is to be examined by every router, at every hop along the path.

Note The Destination Options is the other extension header that uses options. And as its name implies, it contains information intended only for the final destination. Destination Options are discussed at the end of this section.

Let's examine the use of these options. Options provide flexibility, allowing IPv6 packets to be supplemented with sets of values that are not defined in the standard group of extension headers. These sets of values are also referred to as Type-Length-Value (TLV) triplets. It has already been established that two extension headers use these options, Hop-by-Hop Options and Destination Options. As shown in Figure 2-8, these two types of extension headers have a Next Header and a Header Extension Length field, followed by one or more sets of options. Each option contains a set of Options Type, Options Length, and Options Data field (Type, Length, Value, or TVL).

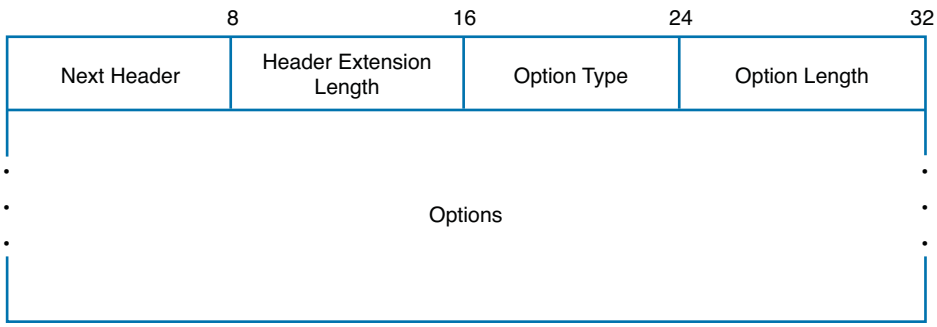

Figure 2-8 *Extension Header Options*

Figure 2-9 shows a Hop-by-Hop extension header referencing a Jumbo Payload Option. The Jumbo Payload Option is used to indicate that the size of this IPv6 packet is larger than 65,535 bytes. Because it is a Hop-by-Hop Option, this information is to be examined by each router along the path.

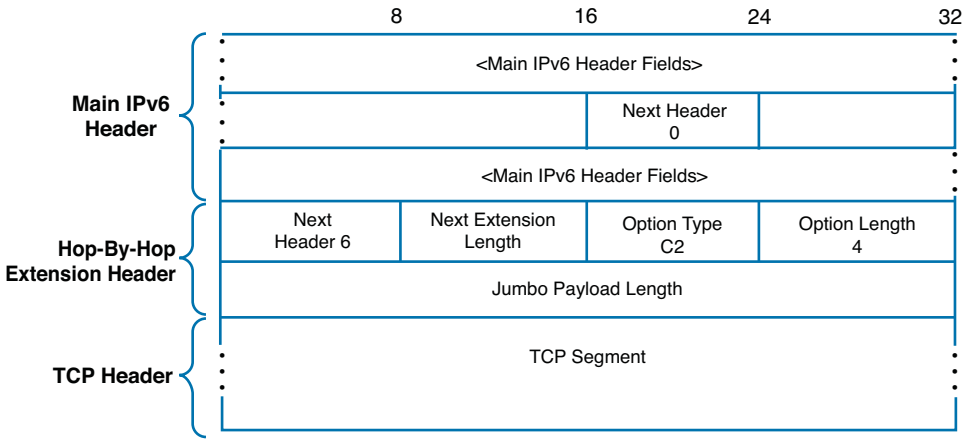

Figure 2-9 *Hop-by-Hop Extension Header with Jumbo Payload Option*

The following describes the fields associated with the Hop-by-Hop extension header:

- **The main IPv6 header: Next Header (8 bits):** Along with other information in the main IPv6 header, there is a Next Header field with the value of 0. This indicates that a Hop-by-Hop Options extension header follows this main header.
- **The Hop-by-Hop extension header:**
  - **Next Header (8 bits):** The value of 6 in the Next Header field indicates that a TCP header follows this header and that there are no other extension headers.
  - **Header Extension Length (8 bits):** This is the length of the Hop-by-Hop Options header in 8-octet units, not including the first 8 octets. There can be multiple options, with each option consisting of a TLV (Options Type, Options Length, and Options Data field).
  - **Option Type (8 bits):** This is the type of option carried in this header. A hexadecimal value of C2 indicates that this is a Jumbo Payload Option.
  - **Option Data Length (8 bits):** This is the number of bytes in the Option Data field. A value of 4 indicates that the Option data is 4 bytes (32 bits) long.
  - **Option Data (variable-length):** The data in this example is the Jumbo Payload Length. The Jumbo Payload Length is a 32-bit field indicating the size of the IPv6 packet in bytes, excluding the IPv6 header but including the Hop-by-Hop Options header and any other extension headers present. Jumbo Payload Length must be greater than 65,535 and can be up to 4,294,967,295 bytes.
- **TCP Segment:** Because there was only one option and no other extension headers, a TCP segment follows as indicated by the Next Header value of 6 in the previous Hop-by-Hop extension header.

If a Hop-by-Hop Options extension header is used, it will immediately follow the main IPv6 header.

Routing Extension Header

The Routing extension header allows the source of the packet to specify the path to the destination. This header contains a list of one or more intermediate routers on the way to a packet's destination. This function is very similar to the Loose Source option used in IPv4. The Routing header is identified by a Next Header value of 43 in the immediately preceding header.

Figure 2-10 shows the structure of a Type 2 Routing header used for mobility support in IPv6. This extension header allows the packet to be routed directly from a correspondent to the mobile node's care-of address, which provides information about the mobile node's current location.

The following describes the fields associated with the Routing Extension Header:

- **Next Header (8 bits):** Identifies the type of header immediately following the Routing header. This will be either another extension header or the protocol of the payload.
- **Header Extension Length (8 bits):** This is the length of the Routing header in 8-octet units, not including the first 8 octets.
- **Routing Type (8 bits):** This value is 2.
- **Segments Left (8 bits):** This value is 1.
- **Reserved (32 bits):** This field is reserved. Initialized to 0 for transmission and ignored on reception.
- **Home Address (128 bits):** This is the Home Address of the destination mobile node.

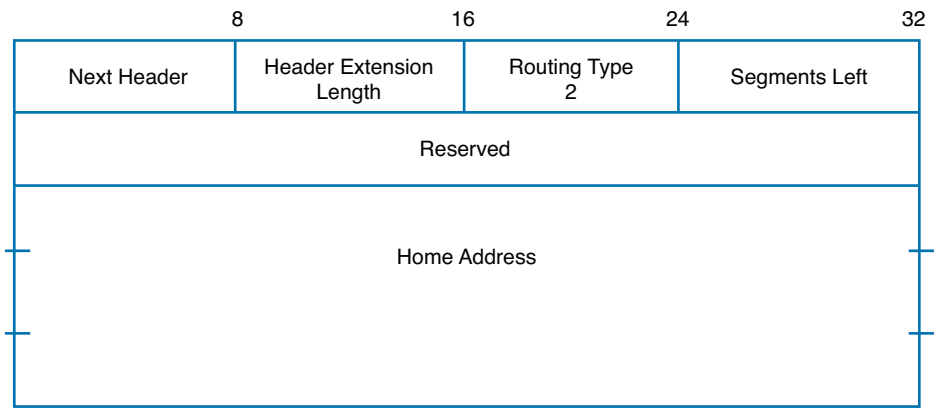

Figure 2-10 *Type 2 Routing Header*

Note The details of how the Routing extension header is processed by each router are beyond the scope of this book. If you are interested, refer to RFC 2460 or the Cisco Press book *Cisco Self-Study: Implementing Cisco IPv6 Networks*. Many ISPs cast a suspicious eye if the source node offers to help with packet next-hop selection, and it is common for packets containing routing extensions to be administratively blocked. Currently the only valid routing header is a Type 2, which is used for mobility in IPv6.

Fragment Extension Header

The Fragment extension header, as shown in Figure 2-11, is similar to that in IPv4 fields used for fragmentation. This method is used when the source of the IPv6 packet needs to divide the packet into fragments and send each fragment as a separate packet. The recipient of the packet reassembles the fragments, each with its own main IPv6 header and a Fragment extension header.

Unlike IPv4, IPv6 routers do not fragment packets unless it is the source of the packet. Intermediate nodes do not perform fragmentation. Only the source of the packet can fragment it. If a router receives an IPv6 packet that is larger than the MTU of the egress interface, the router will discard the packet and send an ICMPv6 Packet Too Big error message back to the source.

Similar to IPv4, for every packet to be fragmented, the source generates a unique Identification value. This value is included in each of the fragment packets. The Identification value ensures that fragments from the original packet are properly reassembled. If the source needs to fragment additional packets within the same message, different Identification values are used.

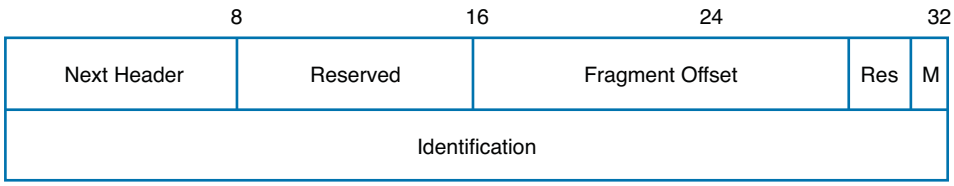

Figure 2-11 *Fragment Header*

The following fields are part of the Fragment header, as shown in Figure 2-11:

- **Next Header (8 bits):** Identifies the protocol number of the data, the fragmented part of the original packet.
- **Reserved (8 bits):** This field is reserved. Initialized to 0 for transmission and ignored on reception.
- **Fragment Offset (13 bits):** This is the relative offset or position, in 8-octet units, of the fragmented data following this header, relative to the original packet. Similar to the Fragment Offset field in IPv4, this informs the receiver where to align this fragmented packet in relation to the other fragmented packets.

- **Res (2 bits):** This field is reserved. Initialized to 0 for transmission and ignored on reception.
- **M flag (1 bit):** The M, or More Fragments, flag is used to indicate whether this is the last fragment (0 bit) or whether there are more fragments to follow (1 bit). This is similar to IPv4's More Fragments flag.
- **Identification (32 bits):** Similar to the same field in the IPv4 header, this is used to uniquely identify all fragmented packets within the same original packet. This field has been expanded from 16 bits in IPv4 to 32 bits in IPv6.

IPsec: AH and ESP Extension Headers

The following extension headers are used to implement two core security protocols in IPsec:

- Authentication Header (AH)
- Encapsulating Security Payload (ESP)

IPsec

Before examining the AH and ESP extension headers, I'll briefly discuss IPsec and the function of these two security protocols. This is not meant to give you a thorough understanding of IPsec, AH, and ESP, but will give you enough information to understand their importance and use in IPv6.

IPsec is a suite of protocols for securing delivery of packets of IP networks. Authentication Header (AH) and Encapsulating Security Payload (ESP) are two main protocols used to provide authentication and integrity for all or part of an IPv6 packet. ESP includes the additional feature of encryption.

Note IPsec is part of both IPv4 and IPv6. IPsec is not required for devices that implement the IPv4 stack. Earlier RFCs state that IPsec is mandatory for all IPv6 implementations, stating that IPsec "must be supported." RFC 6434, IPv6 Node Requirements, relaxes this requirement by declaring that IPsec "should be implemented."

The Authentication Header (AH) is used to guarantee the authenticity and integrity of the packet. *Authentication* means that it ensures that the sender and receiver of the packet are really who they say they are. *Integrity* guarantees that the data was not altered in transit. The Authentication Header provides authentication and integrity but does not provide encryption. *Encryption* is the process of transforming information (usually in plain text) by using an algorithm (called a cipher) to make it unreadable to anyone except those possessing special information (usually referred to as a key).

The Encapsulating Security Payload (ESP) provides authentication integrity *and* encryption. ESP not only protects the packet from being altered by intermediate devices but also protects the content of the packet from being viewed. ESP has its own authentication scheme, or it can be used in conjunction with AH. In summary, AH only provides authentication and integrity, while ESP provides both of these and encrypts the packet as well.

What hasn't been discussed yet is how much of the packet is authenticated or encrypted. The answer to that depends on whether transport mode or tunnel mode is used in IPsec.

Transport and Tunnel Modes

As the name suggests, Transport mode protects the transport layer and higher. The initial IP header is still used. Because the original source and destination IP addresses are in the main IP header intermediate devices, routers recognize the IPsec participants. Transport mode is typically used for host-to-host communications.

Tunnel mode is employed to protect the entire contents of the IP packet, including the IP header. This is accomplished by encapsulating the original IP packet, including the IP header, in a new IP header with the tunnel endpoints used as the new source and destination IP addresses. The tunnel endpoints can be routers or can be the hosts themselves. Tunnel mode protects the entire IP packet while Transport mode does not. Figure 2-12 illustrates the differences between Transport and Tunnel mode.

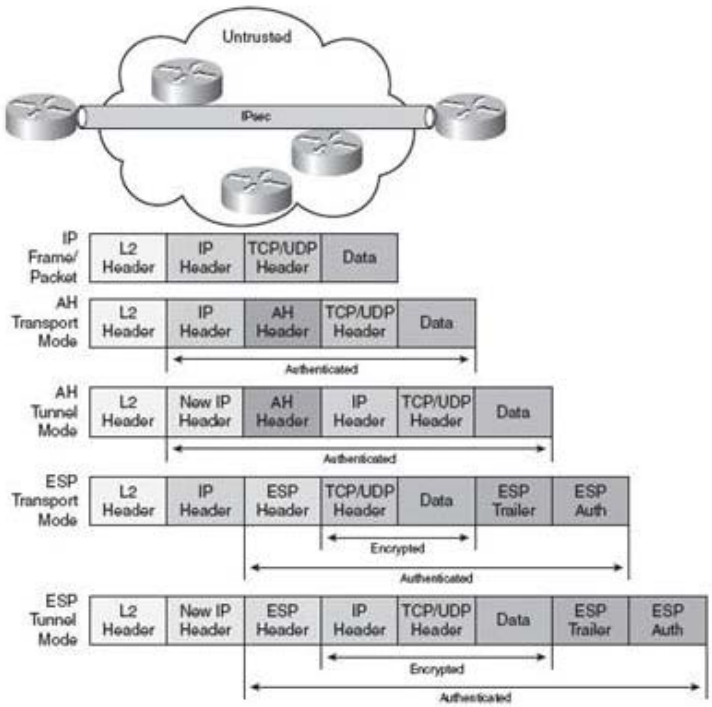

Figure 2-12 *Transport and Tunnel Mode*

Let's return to the extension headers used for AH and ESP. If you are new to IPsec, it is perfectly understandable if some of the details of AH and ESP are a not perfectly clear.

Encapsulating Security Payload (ESP) Extension Header

The Encapsulating Security Payload (ESP) is a variable-length extension header. As discussed previously, this header is used to provide authentication, integrity, and encryption. The ESP extension header is identified by a Next Header value of 50 in the immediately preceding header.

Figure 2-13 shows the ESP extension header. The ESP header can be divided into four parts:

- **ESP Header:** SPI and Sequence Number fields
- **Payload:** ESP Payload Data field
- **ESP Trailer:** Padding, Pad Length, and Next Header fields
- **ESP Authentication Data**

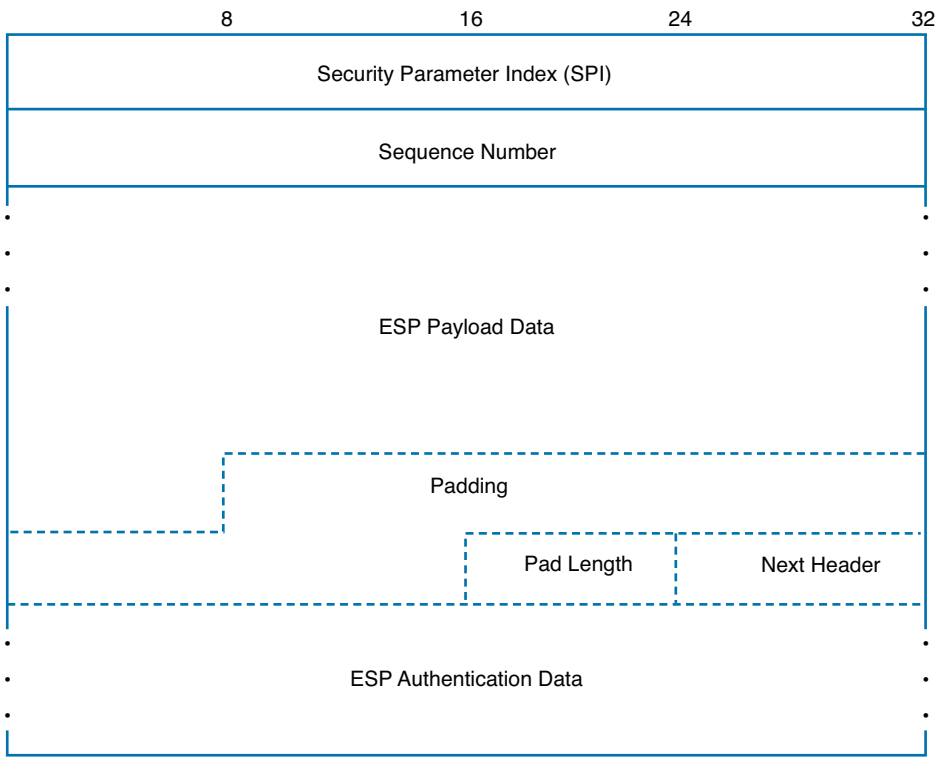

Figure 2-13 *ESP Extension Header*

Figure 2-13 illustrates the fields in the ESP extension header. ESP is viewed as end-to-end communications; in other words, it is not processed by routers along the path. Remember, ESP provides authentication integrity and confidentiality of the original packet. Therefore, the ESP extension header is encapsulated after the main IPv6 header, and after the Hop-by-Hop, Routing, and Fragment extension headers, as shown in Figure 2-14. For IPv6, encryption covers the entire transport-level segment plus the ESP trailer and the Destination Options extension header if it occurs after the ESP header. The Destination Options extension header could appear before or after the ESP header, depending upon the intent desired.

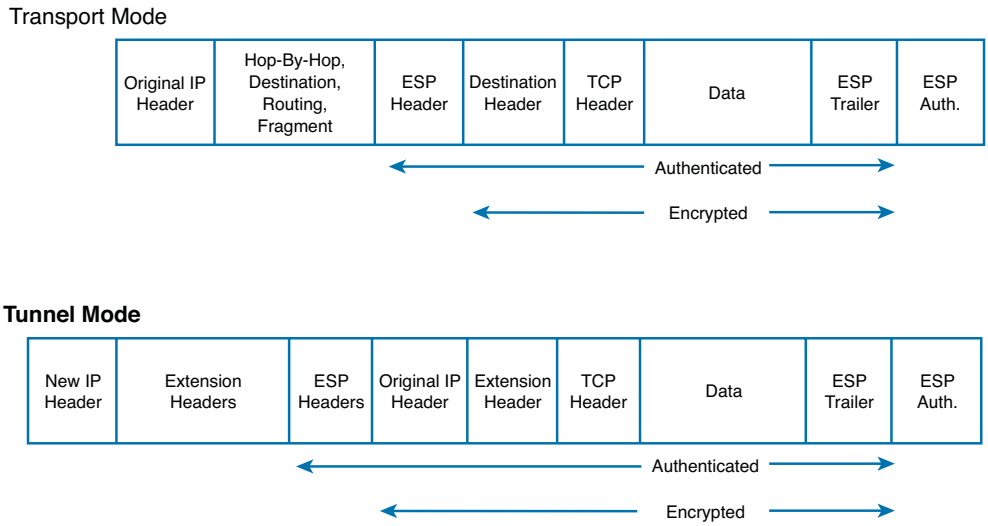

Figure 2-14 *ESP – Transport and Tunnel Modes*

Unlike the previous extension headers, the details of each field are beyond the scope of this book and require a better understanding of IPsec. If you are new to IPsec, it is understandable if some of this might be a little vague. For more information about IPsec, the Cisco Press book *IPsec Virtual Private Network Fundamentals*, by James Henry Carmouche, is an excellent resource.

Authentication Header (AH) Extension Header

The Authentication Header (AH) is also a variable-length extension header. Unlike the ESP, AH only provides authentication and integrity, and does not use encryption to provide confidentiality. The AH extension header is identified by a Next Header value of 51 in the preceding header.

Figure 2-15 illustrates the fields in the Authentication Header. Like ESP, AH is viewed as end-to-end communications. Remember, AH only provides data integrity, a feature to ensure that the participants are who they say they are and that any alteration in the packet's content in transit is detected by the recipient. Similar to ESP, the Authentication Header is encapsulated after the main IPv6 header, and after the Hop-by-Hop, Routing,

and Fragment extension headers, as shown in Figure 2-16. The Destination Options extension header can appear before or after the AH header, depending upon the intent desired.

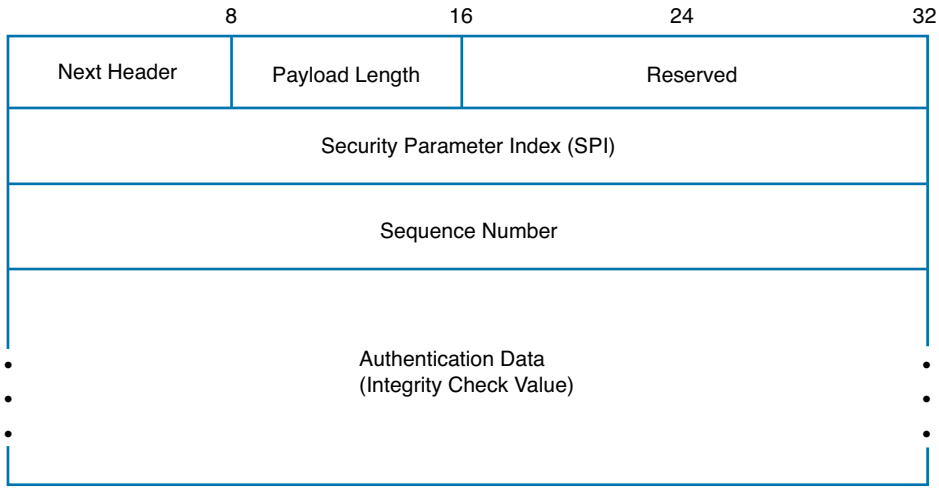

Figure 2-15 *AH Extension Header*

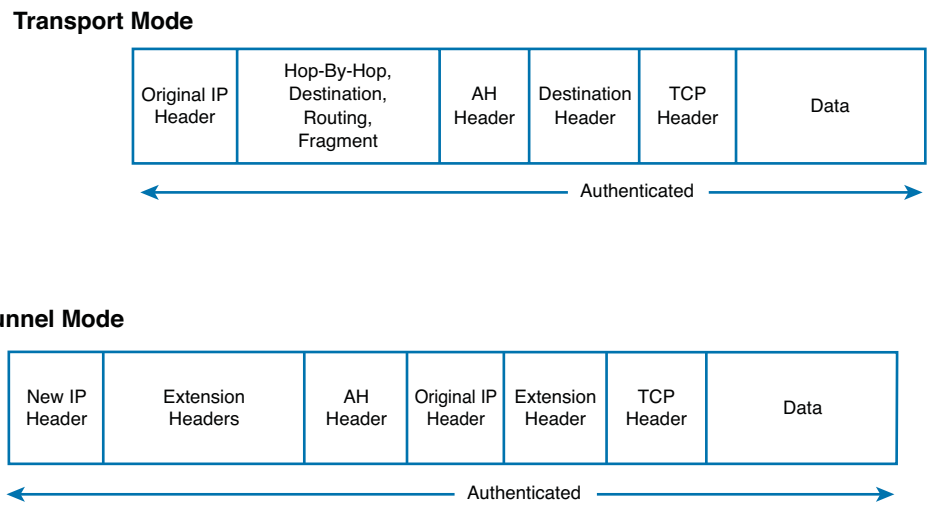

Figure 2-16 *AH – Transport and Tunnel Modes*

As with the other IPsec extension header, the details of AH and these fields are beyond the scope of this book and require a more thorough understanding of IPsec.

Destination Options Extension Header

The Destination Options header is used to carry optional information that only needs to be examined by a packet's destination node(s). The Destination Options header is the other extension header that uses options. (The Hop-by-Hop header is the other one.) The Destination Options header is identified by a Next Header value of 60 in the preceding header. As illustrated in Figure 2-17, the Destination Options header has the following format:

- **Next Header (8 bits):** Identifies the type of header immediately following the Destination Options header. This will be either another extension header or the protocol of the payload.
- **Header Extension Length (8 bits):** This is the length of the Destination Options header in 8-octet units, not including the first 8 octets.
- **Options (variable-length):** This field contains one or more TLV-encoded options, Option Type, Option Data Length, and Option Data:
  - **Option Type (8 bits):** This is the type of option carried in this header.
  - **Option Data Length (8 bits):** This signifies the number of bytes of the Option Data field.
  - **Option Data (variable-length):** This is the data content depending upon the use of this field.

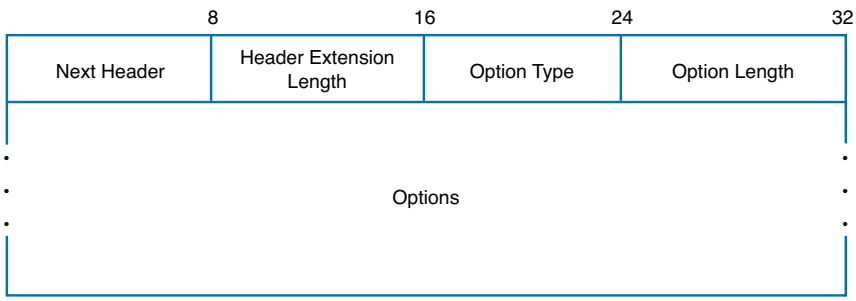

Figure 2-17 *Destination Options Extension Header*

Note One of the proposed uses for the Destination Options extension header is mobility support, as described in RFC 6275, Mobility Support in IPv6:

"Each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a care-of address, which provides information regarding the mobile node's current location. IPv6 packets addressed to a mobile node's home address are transparently routed to its care-of address. The protocol enables IPv6 nodes to cache the binding of a mobile node's home address with its care-of address, and to then send any packets destined for the mobile node directly to it at this care-of address. To support this operation, Mobile IPv6 defines a new IPv6 protocol and a new destination option. All IPv6 nodes, whether mobile or stationary, can communicate with mobile nodes."

No Next Header

The Next Header value of 59 indicates that there is no data following this header. This is just a placeholder indicating that there is nothing after this header. If the payload length indicates that there are additional bytes beyond the header, those bytes will be ignored.

Comparing IPv4 and IPv6

After examining the details of the IPv4 and IPv6 headers, it's easy to miss some of the important differences between the two protocols. There was a lot of information to digest, so let's summarize some of these differences, referring back to Figures 2-1 and 2-2.

IPv4 and IPv6 Header Comparisons

These IPv4 field names are the same as those in IPv6:

- **Version (IPv4 and IPv6):** This is an easy one; the value is 4 in IPv4 and 6 in IPv6.
- **Source Address and Destination Address (IPv4 and IPv6):** Probably the most noticeable differences are the 32-bit IPv4 source and destination addresses, which have been increased to 128 bits in IPv6.

IPv4 field names changed in IPv6 with functional differences in some cases:

- **Type of Service (IPv4) → Traffic Class (IPv6):** IPv4 can use either the 3-bit IP Precedence field along with another 3 bits for delay, throughput, and reliability, or the 6-bit Differentiated Services technique. IPv6 was designed to use the 6-bit DS method.
- **Total Length (IPv4) → Payload Length (IPv6):** IPv4's Total Length field includes both the IPv4 header and the data, whereas the IPv6 Payload Length field only specifies the number of bytes of data (payload), including any extension headers, and does not include the main IPv6 header.
- **Time to Live (IPv4) → Hop Limit (IPv6):** This has the same function in both IPv4 and IPv6, with the name being more reflective of its actual use in IPv6.
- **Protocol (IPv4) → Next Header (IPv6):** In IPv4, this indicates the protocol being carried in the IPv4 data or payload. This same function exists in the Next Header field in IPv6 but can also indicate the existence of an extension header following the main IPv6 header.

IPv4 fields removed from IPv6:

- **Internet Header Length (IPv4):** This field is not needed in IPv6 because the main IPv6 header has a fixed length of 40 bytes. Any additional headers are linked as indicated in the Next Header field.

- **Identification (IPv4), Flags (IPv4), and Fragment Offset (IPv4):** These fields are used for fragmentation in IPv4. Fragmentation is handled differently in IPv6 using the Fragment extension header.
- **Header Checksum (IPv4):** Layer 2 data link layer technologies such as Ethernet perform their own checksum and error control. Upper-layer protocols such as TCP and UDP also have their own checksums and therefore a checksum at Layer 3 becomes redundant and unnecessary. A UDP checksum, which is optional in IPv4, is mandatory in IPv6.
- **Options (IPv4):** Options in IPv4 are now handled using extension headers in IPv6. Two IPv6 extension headers, Hop-by-Hop Options and Destination Options, contain their own set of TLV options.
- **Padding (IPv4):** Because IPv6 has a fixed length of 40 bytes, it is unnecessary to extend the header to make sure that it falls on a 32-bit boundary.

New field in IPv6:

- **Flow Label (IPv6):** This is a new field to IPv6, and the specifications of its use are still being determined by the IETF. RFC 2460 does discuss using the Flow Label field to label sequences of packets for needing special handling by IPv6 routers for "real-time" service. RFC 6437, IPv6 Flow Label Specification, contains additional details on the Flow Label field.

Other Differences

There are several other important differences in the two protocols. The uses of the Hop-by-Hop extension header and the Jumbo Payload Options increase the potential size of an IP packet from 65,535 bytes in IPv4 to 4,294,967,295 bytes in IPv6.

Larger Maximum Transmission Unit (MTU)

IPv4 requires that every node must be able to forward an IP packet of 68 bytes without any further fragmentation. This is because an IPv4 header can be as large as 60 bytes in length or have a minimum fragment size of 8 bytes. Every IPv4 node that is the final destination of the IPv4 packet must be able to receive an IPv4 packet of a minimum size of 576 bytes, which can be all or fragments of the original packet.

IPv6 requires that every link have a minimum MTU of 1280 bytes, with a recommended MTU of 1500 bytes, compared to 68 bytes in IPv4.

Note RFC 1981, Path MTU Discovery for IP version 6, suggests that IPv6 devices should perform PTMU (Path Maximum Transmission Unit) discovery to avoid fragmentation.

User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) Checksum field in IPv4 is optional. Although the protocol remains the same in IPv6, the Checksum field is mandatory. This is because the IPv4 header has its own Checksum field but has been removed in the IPv6 header. The Checksum field is used to verify the integrity of the UDP header and data.

Note The Checksum field in Transmission Control Protocol (TCP) is mandatory for both IPv4 and IPv6. TCP and UDP run on top of IPv6 without any structural modifications to these protocols. TCP, UDP, and other upper-layer protocols are examined in Chapter 9, "DHCPv6 (Dynamic Host Configuration Protocol version 6)."

Fragmentation

When the Fragment extension header was discussed earlier, it was noted that unlike IPv4, IPv6 routers do not fragment packets unless the router is the source of the packet. Only the source node of the IPv6 packet can perform fragmentation. If an intermediate node such as a router receives an IPv6 packet that needs to be fragmented, it will discard the packet and send an ICMPv6 Packet Too Big error message back to the source. Fragmentation and Path MTU discovery are discussed in Chapter 5.

Summary

This chapter examined both the IPv4 header and the IPv6 header. It compared the similarities and the differences between the two protocols. The IPv6 header has fewer fields, and in many respects, IPv6 is a simpler protocol. Some of the fields moving from IPv4 to IPv6 remained the same, some had name changes with functional differences, others were removed completely, and there was a new Flow Label field added.

Extension headers were introduced. They provide more flexibility and better efficiency for IPv6. The impact of IPv6 on User Datagram Protocol (UDP) and maximum transmission unit (MTU) was explained.

Chapter 3 allows you to become familiar with IPv6 notation and the general structure of an IPv6 unicast address.

References

RFCs:

RFC 791, *Internet Protocol, DARPA Internet Program Protocol Specification*, USC, www.ietf.org/rfc/rfc791.txt, September 1981

RFC 1191, *Path MTU Discovery*, J. Mogul, Stanford University, www.ietf.org/rfc/rfc1191.txt, November 1990

RFC 1393, *Traceroute Using an IP Option*, G. Malkin, Xylogics, Inc., www.ietf.org/rfc/rfc1393.txt, January 1993

RFC 1700, *Assigned Numbers*, J. Reynolds, ISI, IETF, www.ietf.org/rfc/rfc1700.txt, October 1994

RFC 1981, *Path MTU Discovery for IP version 6*, J. McCann, Digital Equipment Corporation, www.ietf.org/rfc/rfc1981, August 1996

RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*, S. Deering, Cisco Systems, IETF, www.ietf.org/rfc/rfc2460.txt, December 1998

RFC 2474, *Using a Technique Called Differentiated Services (DS)*, K. Nichols, Cisco Systems, www.ietf.org/rfc/rfc2474.txt, December 1998

RFC 2675, *IPv6 Jumbograms*, D. Borman, Berkeley Software Design, www.ietf.org/rfc/rfc2675.txt, August 1999

RFC 3775, *Mobility Support in IPv6*, D. Johnson, Rice University, www.ietf.org/rfc/rfc3775.txt, June 2004

RFC 6434, *IPv6 Node Requirements*, E. Jankiewicz, SRI International, www.ietf.org/rfc/rfc6434, December 2011

RFC 6437, *IPv6 Flow Label Specification*, S. Amante, Level 3, www.ietf.org/rfc/rfc6437, November 2011

Website:

www.iana.org/assignments/protocol-numbers/protocol-numbers.xml