Help for Network Administrators

IPv6 Essentials

O'REILLY®

Silvia Hagen

IPv6 Essentials

Silvia Hagen

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

CHAPTER 2

The Structure of the IPv6 Protocol

This chapter explains the structure of the IPv6 header and compares it to the IPv4 header. It also discusses Extension headers, which are new in IPv6.

The header structure of an IPv6 packet is specified in RFC 2460. The header has a fixed length of 40 bytes. The two fields for source and destination addresses each use 16 bytes (128 bits), so there are only 8 bytes for general header information.

General Header Structure

In IPv6, five fields from the IPv4 header have been removed:

- Header Length
- Identification
- Flags
- Fragment Offset
- Header Checksum

The Header Length field was removed because it is not needed in a header with a fixed length. In IPv4 the minimum header length is 20 bytes, but if options are added, it can be extended in 4-byte increments up to 60 bytes. Therefore, with IPv4, the information about the total length of the header is important. In IPv6 options are defined by Extension headers (covered later in this chapter).

The Identification field, the Flags field, and the Fragment Offset field handle fragmentation of a packet in the IPv4 header. Fragmentation happens if a large packet has to be sent over a network that only supports smaller packet sizes. In that case, the IPv4 router splits the packet into smaller slices and forwards multiple packets. The destination host collects the packets and

reassembles them. If only one packet is missing or has an error, the whole transmission has to be redone; this is very inefficient. In IPv6, a host learns the Path Maximum Transmission Unit (MTU) size through a procedure called Path MTU Discovery. If a sending IPv6 host wants to fragment a packet, it will use an Extension header to do so. IPv6 routers along the path of a packet do not provide fragmentation, as they did with IPv4. So the Identification, Flags, and Fragment Offset fields were removed from the IPv6 header and will be inserted as an Extension header, if needed. Extension headers are explained later in this chapter.

Path MTU Discovery is explained in Chapter 4.

The Header Checksum field was removed to improve processing speed. If routers do not have to check and update checksums, processing becomes much faster. Checksumming is done at the media access level, too, and the risk for undetected errors and misrouted packets is minimal. There is a checksum field at the transport layer (UDP and TCP). IP is a best-effort delivery protocol; it is the responsibility of upper layer protocols to insure integrity.

The Type of Service field was replaced by the TrafficClass field. IPv6 has a different mechanism to handle preferences. Refer to Chapter 6 for more information. The Protocol Type and the Time-to-Live (TTL) fields were renamed and slightly modified. A Flow Label field was added.

The Fields in the IPv6 Header

By becoming familiar with the fields of the IPv6 header, you will better understand how IPv6 works.

For a detailed description of all the fields in an IPv4 header, refer to *Novell's Guide to Troubleshooting TCP/IP* (John Wiley & Sons) by Silvia Hagen and Stephanie Lewis.

Figure 2-1 provides an overview of the IPv6 header. The fields are discussed in detail in the following paragraphs.

Figure 2-1 shows that even though the header has a total size of 40 bytes, which is twice as long as a default IPv4 header, it has actually been streamlined because most of the header is taken by the two 16-byte IPv6 addresses. That leaves only 8 bytes for other header information.

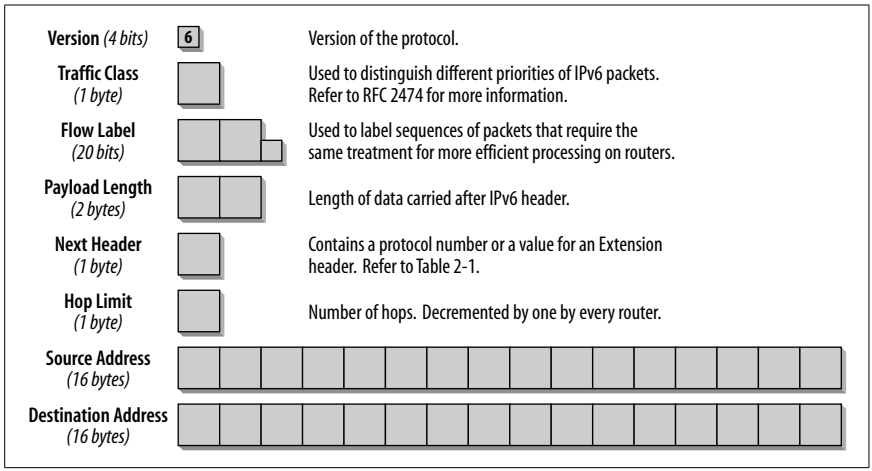

Figure 2-1. Fields in the IPv6 header

Version (4 Bits)

This is a 4-bit field and contains the version of the protocol. In the case of IPv6, the number is 6. The version number 5 could not be used because it had already been assigned an experimental stream protocol (ST2, RFC 1819).

Traffic Class (1 Byte)

This field replaces the Type of Service field in IPv4. This field facilitates the handling of real-time data and any other data that requires special handling. This field can be used by sending nodes and forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets.

RFC 2474, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," explains how the Traffic Class field in IPv6 can be used. RFC 2474 uses the term DS Field to refer to the Type of Service field in the IPv4 header, as well as to the Traffic Class field in the IPv6 header.

Flow Label (20 Bits)

This field distinguishes packets that require the same treatment, in order to facilitate the handling of real-time traffic. A sending host can label sequences of packets with a set of options. Routers keep track of flows and can process packets belonging to the same flow more efficiently because they do not have to reprocess each packet's header. A flow is uniquely identified by the flow label and the address of the source node. Nodes that do not support the functions of the Flow Label field are required to pass the field

unchanged when forwarding a packet and to ignore the field when receiving a packet. All packets belonging to the same flow must have the same source and destination IP address.

The use of the Flow Label field is experimental and still under discussion at the IETF at the time of this writing. Refer to Chapter 6 for more information.

Payload Length (2 Bytes)

This field specifies the payload—i.e., the length of data carried after the IP header. The calculation in IPv6 is different from the one in IPv4. The Length Field in IPv4 includes the length of the IPv4 header, whereas the Payload Length field in IPv6 contains only the data following the IPv6 header. Extension headers are considered part of the payload and are therefore included in the calculation.

The fact that the Payload Length field has 2 bytes limits the maximum packet payload size to 64 KB. IPv6 has a Jumbogram Extension header, which supports bigger packet sizes, if needed. Jumbograms are relevant only when IPv6 nodes are attached to links that have a link MTU greater than 64 KB. Jumbograms are specified in RFC 2675.

Next Header (1 Byte)

In IPv4, this field is the Protocol Type field. It was renamed in IPv6 to reflect the new organization of IP packets. If the next header is UDP or TCP, this field will contain the same protocol numbers as in IPv4—for example, protocol number 6 for TCP or 17 for UDP. But if Extension headers are used with IPv6, this field contains the type of the next Extension header. That header is located between the IP header and the TCP or UDP header. Table 2-1 lists possible values in the Next Header field.

Table 2-1. Values in the Next Header field

Value	Description
0	In an IPv4 header: reserved and not used
	In an IPv6 header: Hop-by-Hop Option Header following
1	Internet Control Message Protocol (ICMPv4)—IPv4 support
2	Internet Group Management Protocol (IGMPv4)—IPv4 support
4	IP in IP (encapsulation)
6	TCP

Table 2-1. Values in the Next Header field (continued)

Value	Description
8	Exterior Gateway Protocol (EGP)
9	IGP - any private interior gateway (used by Cisco for their IGRP)
17	UDP
41	IPv6
43	Routing header
44	Fragmentation header
45	Interdomain Routing Protocol (IDRP)
46	Resource Reservation Protocol (RSVP)
50	Encrypted Security Payload header
51	Authentication header
58	ICMPv6
59	No Next Header for IPv6
60	Destination Options header
88	EIGRP
89	OSPF
108	IP Payload Compression Protocol
115	Layer 2 Tunneling Protocol (L2TP)
132	Stream Control Transmission Protocol (SCTP)
134-254	Unassigned
255	Reserved

Header type numbers derive from the same range of numbers as protocol type numbers and should therefore not conflict with them.

The complete list of protocol numbers can be found in the appendix. For the most current list, go to IANA's web site at *http://www.iana.org/assignments/protocol-numbers*.

Hop Limit (1 Byte)

This field is analogous to the TTL field in IPv4. The TTL field contains a number of seconds, indicating how long a packet can remain in the network before being destroyed. Most routers simply decremented this value by one at each hop. This field was renamed to Hop Limit in IPv6. The value in this field now expresses a number of hops and not a number of seconds. Every forwarding node decrements the number by one.

Source Address (16 Bytes)

This field contains the IP address of the originator of the packet.

Destination Address (16 Bytes)

This field contains the IP address of the intended recipient of the packet. With IPv4, this field always contains the address of the ultimate destination of the packet. With IPv6, this field might not contain the IP address of the ultimate destination if a Routing header is present.

Figure 2-2 shows the IPv6 header in the trace file.

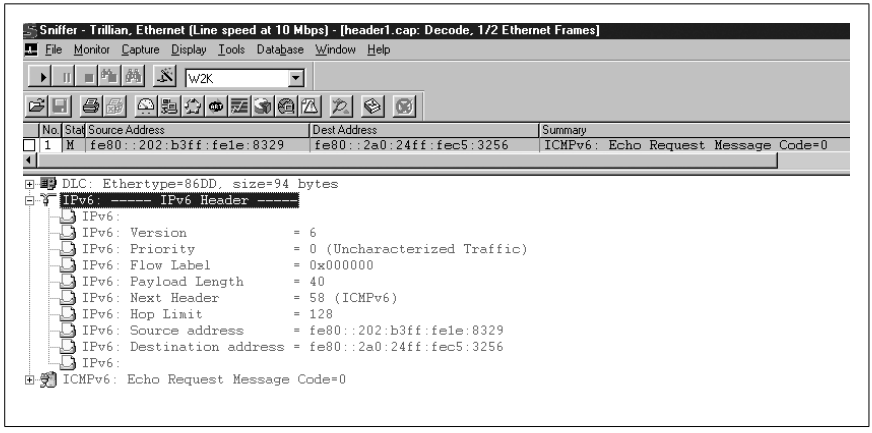

Figure 2-2. The IPv6 header in a trace file

This trace file shows all of the header fields I have discussed and how they are presented in a trace file. The Version field is set to 6 for IPv6. The Priority and the Flow Label fields are not used in this packet and are set to zero. The Payload Length is 40 and the Next Header value is set to 58 for ICMPv6. The Hop Limit is set to 128 and the Source and Destination addresses contain the link local addresses of my IPv6 nodes.

Extension Headers

The IPv4 header can be extended from a minimum of 20 bytes to 60 bytes in order to specify options such as Security Options, Source Routing, or Timestamping. This capacity has rarely been used because it causes a performance hit. For example, IPv4 hardware forwarding implementations have to pass the packet containing options to the main processor (software handling).

The simpler a packet header, the faster the processing. IPv6 has a new way to deal with options that has substantially improved processing. It handles options in additional headers called Extension headers.

The current IPv6 specification (RFC 2460) defines six Extension headers:

- Hop-by-Hop Options header
- Routing header
- Fragment header
- Destination Options header
- Authentication header
- Encrypted Security Payload header

There can be zero, one, or more than one Extension header between the IPv6 header and the upper-layer protocol header. Each Extension header is identified by the Next Header field in the preceding header. The Extension headers are examined or processed only by the node identified in the Destination Address field of the IPv6 header. If the address in the Destination Address field is a multicast address, the Extension headers are examined and processed by all the nodes belonging to that multicast group. Extension headers must be strictly processed in the order they appear in the packet header.

There is an exception to the above rule: only the destination node will process an Extension header. If the Extension header is a Hop-by-Hop Options header, the information it carries must be examined and processed by every node along the path of the packet. The Hop-by-Hop Options header, if present, must immediately follow the IPv6 header. It is indicated by the value zero in the Next Header field of the IPv6 header (see Table 2-1, earlier in this chapter).

The first four Extension headers are described in RFC 2460. The Authentication header is described in RFC 2402 and the Encrypted Security Payload header in RFC 2406.

Figure 2-3 shows how Extension headers are used.

Each Extension header is a multiple of 8 octets long. That way, subsequent headers can always be aligned. If a node is required to process the Next Header but cannot identify the value in the Next Header field, it is required to discard the packet and send an ICMPv6 Parameter Problem message back to the source of the packet. For details on ICMPv6 messages, refer to Chapter 4.

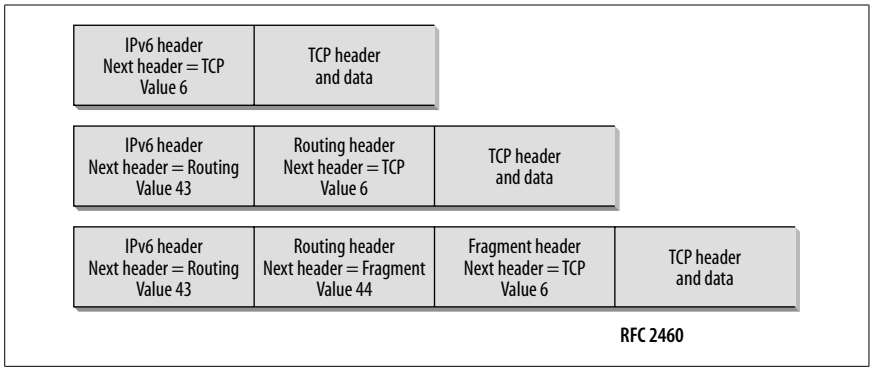

Figure 2-3. The use of Extension headers

If more than one Extension header is used in a single packet, the following header order should be used (RFC 2460):

1. IPv6 header
2. Hop-by-Hop Options header
3. Destination Options header (for options to be processed by the first destination that appears in the IPv6 Destination address field, plus subsequent destinations listed in the Routing header)
4. Routing header
5. Fragment header
6. Authentication header
7. Encapsulating Security Payload header
8. Destination Options header (for options to be processed only by the final destination of the packet)
9. Upper-Layer header

In cases when IPv6 is encapsulated in IPv4, the Upper-Layer header can be another IPv6 header and can contain Extension headers that have to follow the same rules.

Hop-by-Hop Options Header

The Hop-by-Hop Options Extension header carries optional information that must be examined by every node along the path of the packet. It must follow the IPv6 header immediately and is indicated by a Next Header value of zero. For example, the Router Alert (RFC 2711) uses the Hop-by-Hop Extension header for protocols like Resource Reservation Protocol (RSVP)

or Multicast Listener Discovery (MLD) messages. With IPv4, the only way for a router to determine if it needs to examine a datagram is to, at least partially, parse upper layer data in all datagrams. This slows down the routing process substantially. With IPv6, in the absence of a Hop-by-Hop Extension header, a router knows that it does not need to process router-specific information and can route the packet immediately to the final destination. If there is a Hop-by-Hop Extension header, the router only needs to examine this header and not look further into the packet.

The format of the Hop-by-Hop Options header is shown in Figure 2-4.

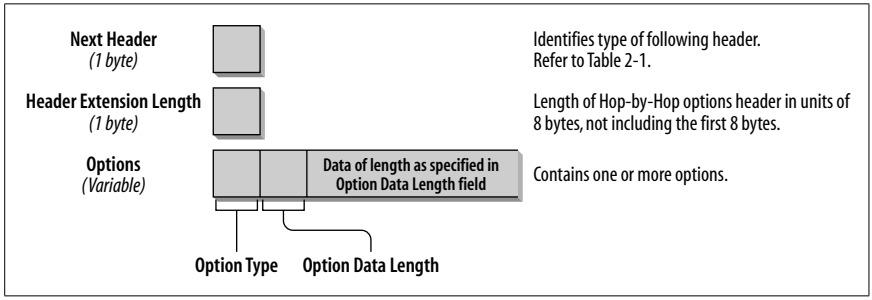

Figure 2-4. Format of the Hop-by-Hop Options header

The following list describes each field:

Next Header (1 byte)
: The Next Header field identifies the type of header that follows the Hop-by-Hop Options header. The Next Header field uses the values listed in Table 2-1, earlier in this chapter.

Header Extension Length (1 byte)
: This field identifies the length of the Hop-by-Hop Options header in 8-byte units. The length calculation does not include the first 8 bytes.

Options (variable size)
: There can be one or more options. The length of the options is variable and determined in the Header Extension Length field.

The Option Type Field, the first byte of the Options fields, contains information about how this option must be treated in case the processing node does not recognize the option. The value of the first two bits specifies the actions to be taken:

- Value 00: skip and continue processing.
- Value 01: discard the packet.

- Value 10: discard the packet and send ICMP Parameter Problem, Code 2 message to the packet's source address, pointing to the unrecognized option type.
- Value 11: discard the packet and send ICMP Parameter Problem, Code 2 message to the packet's source address only if the destination is not a multicast address.

The third bit of the Options Type field specifies whether the option information can change en route (value 01) or does not change en route (value 00).

Routing Header

The Routing header is used to give a list of one or more intermediate nodes that should be visited on the packet's path to its destination. In the IPv4 world, this is called the Loose Source and Record Route option. The Routing header is identified by a Next Header value of 43 in the immediately preceding header. Figure 2-5 shows the format of the Routing header.

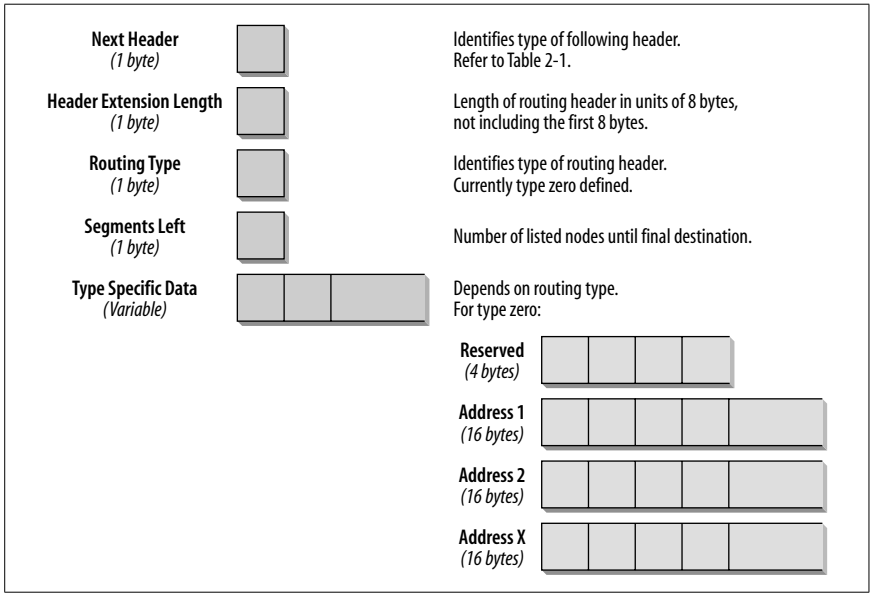

Figure 2-5. Format of the Routing header

The following list describes each field:

Next Header (1 byte)
: The Next Header field identifies the type of header that follows the Routing header. It uses the same values as the IPv4 Protocol Type field (see Table 2-1, earlier in this chapter).

Header Extension Length (1 byte)
: This field identifies the length of the Routing header in 8-byte units. The length calculation does not include the first 8 bytes.

Routing Type (1 byte)
: This field identifies the type of Routing header. RFC 2460 describes Routing Type zero.

Segments Left (1 byte)
: This field identifies how many nodes are left to be visited before the packet reaches its final destination.

Type-Specific Data (Variable-length)
: The length of this field depends on the Routing Type. The length will always make sure that this complete header is a multiple of 8 bytes.

If a node processing a Routing header cannot identify a Routing Type value, the action taken depends on the content of the Segments Left field. If the Segments Left field does not contain any nodes to be visited, the node must ignore the Routing header and process the next header in the packet, determined by the Next Header field value. If the Segments Left field is not zero, the node must discard the packet and send an ICMP Parameter Problem, Code 0 message to the packet's source address, pointing to the unrecognized Routing Type. If a forwarding node cannot process the packet because the next link MTU size is too small, it discards the packet and sends an ICMP Packet Too Big message back to the source of the packet.

The only Routing Type described in RFC 2460 is a Type Zero Routing header. The first node that processes the Routing header is the node addressed by the Destination address field in the IPv6 header. This node decrements the Segments Left field by one and inserts the next address field from within the Routing header in the IPv6 header Destination address field. Then the packet is forwarded to the next hop that will again process the Routing header as described until the final destination is reached. The final destination is the last address in the Routing Header Data field. For example, Mobile IPv6 uses the Routing header. Any node sending a packet to a mobile node will send the packet to the mobile node's care-of-address. It will include a Routing header with one entry, the mobile node's home address. The mobile node swaps the Destination address in the IPv6 header with the entry in the Routing header and will reply with its home address as a source address as if it received the packet attached to its home network. For further discussion and definition of terms regarding Mobile IPv6, refer to Chapter 7. Figure 2-6 shows the routing header in a trace file.

The Next Header field within the IPv6 header shows the value 43 for the Routing header. The Source and Destination addresses have the prefix 2002:,

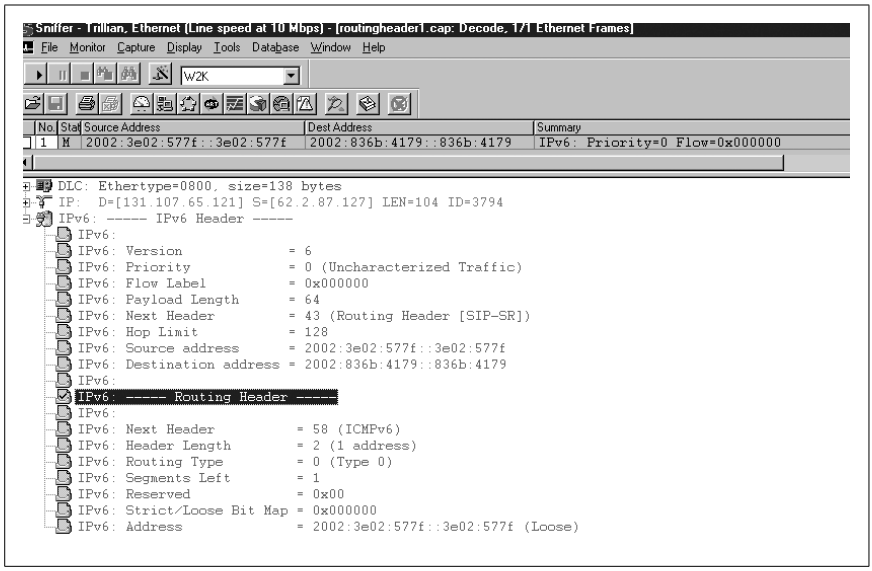

Figure 2-6. Routing header in a trace file

which is allocated to 6to4 sites. The Routing header contains the fields discussed earlier in this section. Next Header will be ICMPv6, value 58. The Header Length is two 8-byte units, which calculates to a total length of 16 bytes. The Segments Left field contains the value 1 because there is one address entry in the Options Fields. Finally, the Options field lists the addresses to be visited. In this case, there is only one entry. If a number of hosts is listed here, every forwarding node (that is, the destination IP address in the IPv6 header) takes the next entry from this host list, uses it as a new destination IP address in the IPv6 header, decrements the Segments Left field by one, and forwards the packet. This is done until the last host in the list is reached. RFC 2460 shows an example.

A source node S sends a packet to destination node D using a Routing header to send the packet through the intermediate nodes I1, I2, and I3. The Routing header changes are shown in Table 2-2.

Table 2-2. Processing the Routing header

	IPv6 Header	Routing Header
Packet from S to I1	Source address S	Segments Left 3
	Destination address I1	Address (1) = I2
		Address (2) = I3
		Address (3) = D

Table 2-2. Processing the Routing header (continued)

	IPv6 Header	Routing Header
Packet from I1 to I2	Source address S	Segments Left 2
	Destination address I2	Address (1) = I1
		Address (2) = I3
		Address (3) = D
Packet from I2 to I3	Source address S	Segments Left = 1
	Destination address I3	Address (1) = I1
		Address (2) = I2
		Address (3) = D
Packet from I3 to D	Source address S	Segments Left = 0
	Destination address D	Address (1) = I1
		Address (2) = I2
		Address (3) = I3

Fragment Header

An IPv6 host that wants to send a packet to an IPv6 destination uses Path MTU discovery to determine the maximum packet size that can be used on the path to that destination. If the packet to be sent is larger than the supported MTU, the source host fragments the packet. Unlike IPv4, with IPv6, a packet does not get fragmented by a router along the path. Fragmentation only occurs on the source host sending the packet. The destination host handles reassembly. A Fragment header is identified by a Next Header value of 44 in the preceding header. The format of the Fragment header is shown in Figure 2-7.

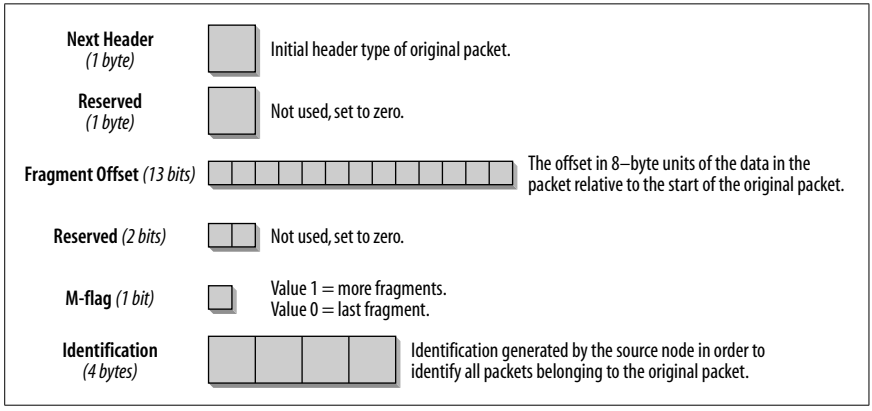

Figure 2-7. Format of the Fragment header

The following list describes each field:

Next Header (1 byte)
: The Next Header field identifies the type of header that follows the Fragment header. It uses the same values as the IPv4 Protocol Type field. (See Table 2-1).

Reserved (1 byte)
: Not used; set to zero.

Fragment Offset (13 bits)
: The offset in 8-byte units of the data in this packet relative to the start of the data in the original packet.

Reserved (2 bits)
: Not used; set to zero.

M-Flag (1 bit)
: Value 1 indicates more fragments; value zero indicates last fragment.

Identification (4 Bytes)
: Generated by the source host in order to identify all packets belonging to the original packet. This field is usually implemented as a counter, increasing by one for every packet that needs to be fragmented by the source host.

The initial unfragmented packet is referred to as the original packet. It has an unfragmentable part that consists of the IPv6 header, plus any Extension headers that must be processed by nodes along the path to the destination (i.e., Hop-by-Hop Options). The fragmentable part of the original packet consists of any Extension headers that need only to be processed by the final destination, plus the Upper-Layer headers and any data. Figure 2-8 (RFC 2460) illustrates the fragmenting process.

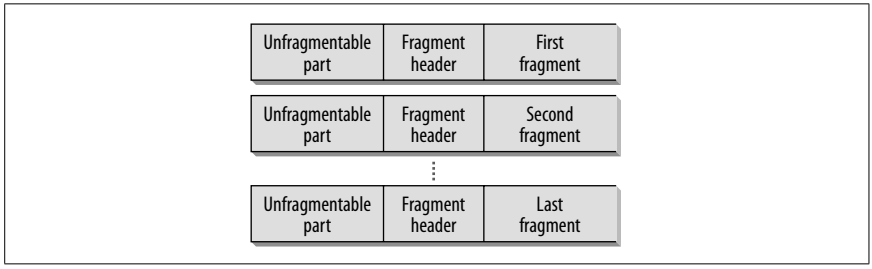

Figure 2-8. Fragmentation with IPv6

The unfragmentable part of the original packet appears in every fragment, followed by the Fragmentation header, and then the fragmentable data. The IPv6 header of the original packet has to be slightly modified. The length

field reflects the length of the fragment (excluding the IPv6 header) and not the length of the original packet.

The destination node collects all the fragments and reassembles them. The fragments must have identical Source and Destination addresses and the same identification value in order to be reassembled. If all fragments do not arrive at the destination within 60 seconds after the first fragment, the destination will discard all packets. If the destination has received the first fragment (offset = zero), it sends back an ICMPv6 Fragment Reassembly Time Exceeded message to the source.

Figure 2-9 shows a Fragment header.

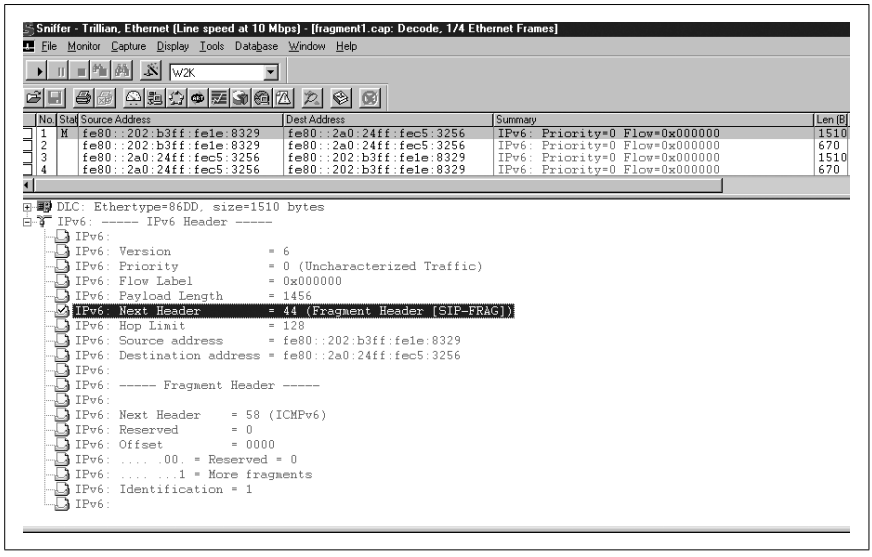

Figure 2-9. Fragment header in a trace file

I created this Fragment header by generating an oversized ping from *Marvin* to *Ford* (Win2000 to Linux). The whole fragment set consists of two packets, the first of which is shown in Figure 2-9. In the IPv6 header, the Payload Length field has a value of 1456, which is the length of the fragmentation header and this one fragment, not the length of the whole original packet. The Next Header field specifies the value 44, which is the value for the Fragment header. This field is followed by the Hop Limit field and by the Source and Destination IP addresses. The first field in the Fragment header is the Next Header field. Because this is a ping, it contains the value 58 for ICMPv6. And because this is the first packet in the fragment set, the value in the Offset field is zero and the M-Flag is set to one, which means there are more fragments to come. The Identification field is set to

one and has to be identical in all packets belonging to this fragment set. Figure 2-10 shows the second packet of the fragment set.

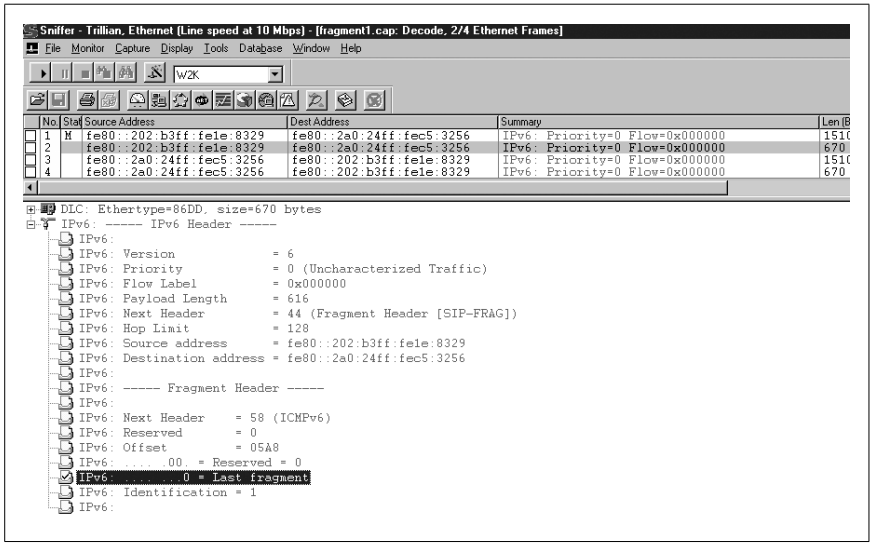

Figure 2-10. The last packet in the fragment set

The second and last packet of this fragment set has an Offset value of `0x05A8`, which translates to 1448 in decimal, the length of the first fragment. The M-Flag is set to zero. This indicates that it is the last packet and tells the receiving host that it is time to reassemble the fragments. The Identification field is set to one in both packets.

Destination Options Header

A Destination Options header carries optional information that is examined by the destination node only. The Next Header value identifiying this type of header is the value `60`. Figure 2-11 shows the format of the Destination Options header.

The following list describes each field:

Next Header (1 byte)
: The Next Header field identifies the type of header that follows the Destination Options header. It uses the same values listed in Table 2-1, earlier in this chapter.

Header Extension Length (1 byte)
: This field identifies the length of the Destination Options header in 8-byte units. The length calculation does not include the first 8 bytes.

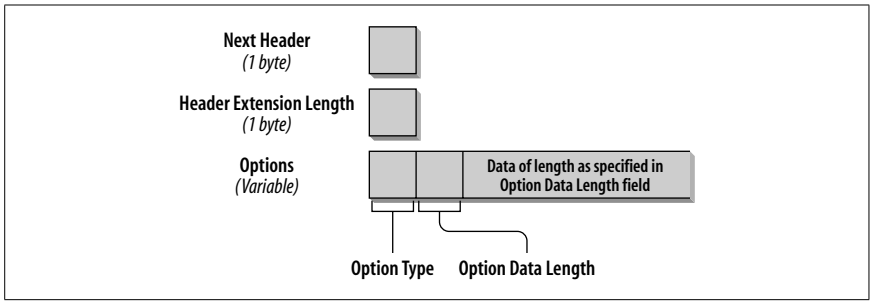

Figure 2-11. Format of the Destination Options header

Options (variable size)

There can be one or more options. The length of the options is variable and determined in the Header Extension Length field.

The Options field is used in the same way as the Hop-by-Hop Options header, which I discussed earlier in this chapter. An example of the Destination Options header is Mobile IPv6. A mobile IPv6 node connected to a foreign network can send packets with its care-of-address as a source address and its home address in a home address destination option. According to the current Mobile IPv6 draft, the ability to correctly process a home address in a Destination Option is required in all IPv6 nodes. For a detailed explanation of Mobile IPv6, refer to Chapter 7 or to the current draft of Mobile IPv6 at *http://www.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-17.txt*. Note that the draft number may have increased by one or more when you follow this link.