MODULE 2

Internet Protocol: Connectionless Datagram Delivery (IPv4, IPv6)

Connectionless Delivery System Characteristics

The most fundamental Internet service consists of a packet delivery system. Technically, the service is defined as an unreliable, best-effort, connectionless packet delivery system. We use the technical term unreliable to mean that delivery is not guaranteed. A packet may be lost, duplicated, delayed, or delivered out of order. The connectionless service will not detect such conditions, nor will it inform the sender or receiver. The basic service is classified as connectionless because each packet is treated independently from all others. A sequence of packets sent from one computer to another may travel over different paths, or some may be lost while others are delivered. Finally, the service is said to use best-effort delivery because the Internet software makes an earnest attempt to deliver packets. That is, the Internet does not discard packets capriciously; unreliability arises only when resources are exhausted or underlying networks fail.

Purpose And Importance Of The Internet Protocol

The protocol that defines the unreliable, connectionless delivery mechanism is called the Internet Protocol (IP).

The Internet Protocol provides three important specifications.

First, IP defines the basic unit of data transfer used throughout a TCP/IP internet.

Second, IP software performs the forwarding function, choosing a path over which a packet will be sent.

Third, in addition to the precise, formal specification of data formats and forwarding, IP includes a set of rules that embody the basis of unreliable delivery.

The IP Datagram

The Internet calls its basic transfer unit an Internet datagram, usually abbreviated IP datagram. In fact, TCP/IP technology has become so successful that when someone uses the term datagram without any qualification, it is generally accepted to mean IP datagram.

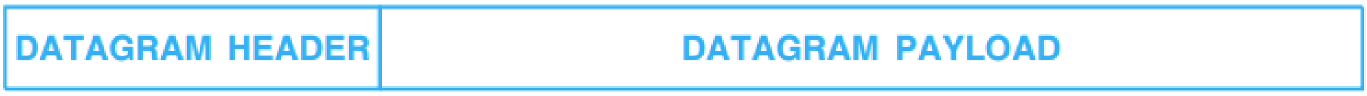


Figure 7.2 General form of an IP datagram, the Internet analogy of a network frame.

IPv4 Datagram Format

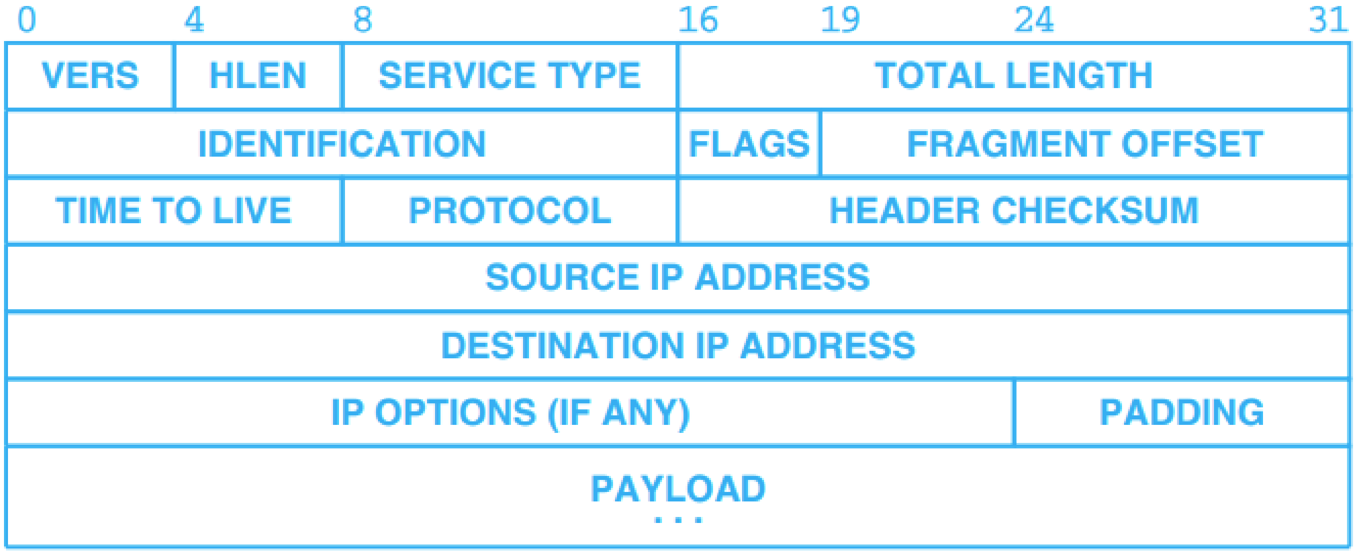


Figure 7.3 Format of an IPv4 datagram, the basic unit of transfer in a TCP/IP internet.

VERSION: Version of the IP protocol (4 bits), which is 4 for IPv4
HLEN: IP header length (4 bits), which is the number of 32 bit words in the header. The minimum value for this field is 5 and the maximum is 15.
Type of service: Low Delay, High Throughput, Reliability (8 bits)
Total Length: Length of header + Data (16 bits), which has a minimum value 20 bytes and the maximum is 65,535 bytes.
Identification: Unique Packet Id for identifying the group of fragments of a single IP datagram (16 bits)
Flags: 3 flags of 1 bit each : reserved bit (must be zero), do not fragment flag, more fragments flag (same order)
Fragment Offset: Represents the number of Data Bytes ahead of the particular fragment in the particular Datagram. Specified in terms of number of 8 bytes, which has the maximum value of 65,528 bytes.

Time to live: Datagram's lifetime (8 bits), It prevents the datagram to loop through the network by restricting the number of Hops taken by a Packet before delivering to the Destination.
Protocol: Name of the protocol to which the data is to be passed (8 bits)
Header Checksum: 16 bits header checksum for checking errors in the datagram header
Source IP address: 32 bits IP address of the sender
Destination IP address: 32 bits IP address of the receiver
Option: Optional information such as source route, record route. Used by the Network administrator to check whether a path is working or not.

IPv6 Datagram Format

Instead of trying to specify all details in a single header, IPv6 uses an extension capability that allows the IETF to adapt the protocol. an IPv6 datagram begins with a fixed-size base header followed by zero or more extension headers, followed by a payload.

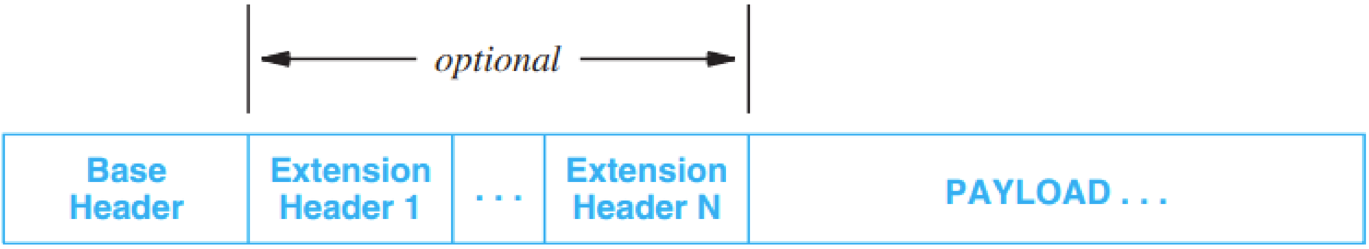


Figure 7.4 The general form of an IPv6 datagram with a base header followed by optional extension headers.

Each IPv6 header contains a NEXT HEADER field that specifies the type of the header that follows. The final header uses the NEXT HEADER field to specify the type of the payload.

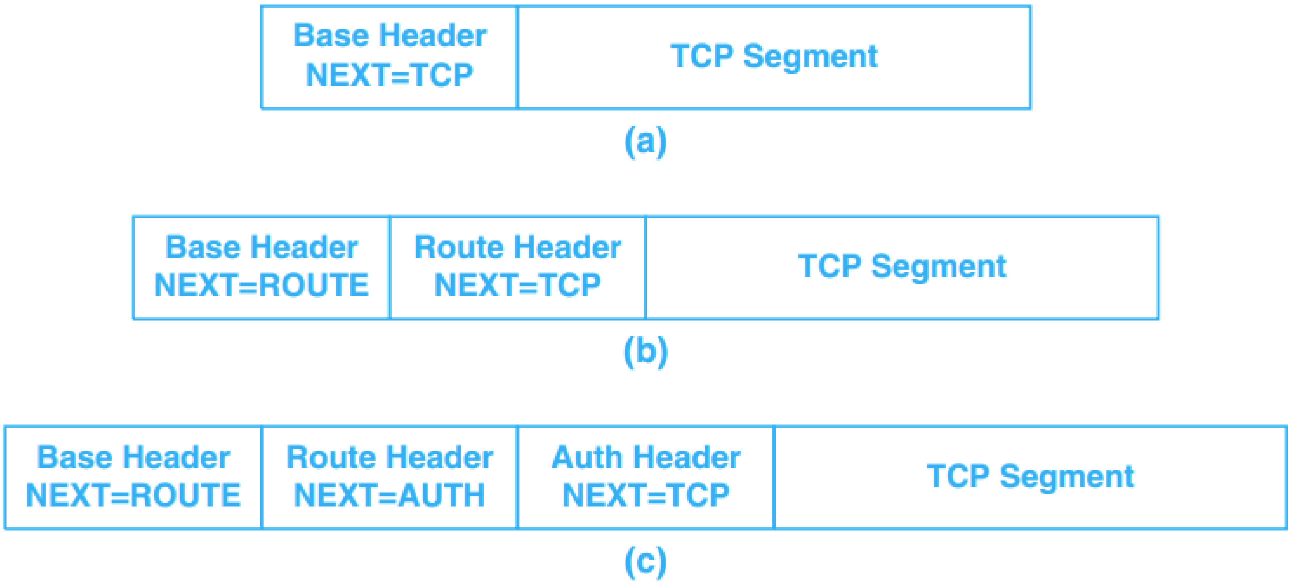


Figure 7.5 Illustration of the *NEXT HEADER* fields in IPv6 datagrams with (a) only a base header, (b) a base header and one extension, and (c) a base header and two extension headers.

IPv6 Base Header Format

Each IPv6 datagram begins with a 40-octet base header. twice as large as a typical IPv4 datagram header, the IPv6 base header contains less information because fragmentation information has been moved to extension headers.

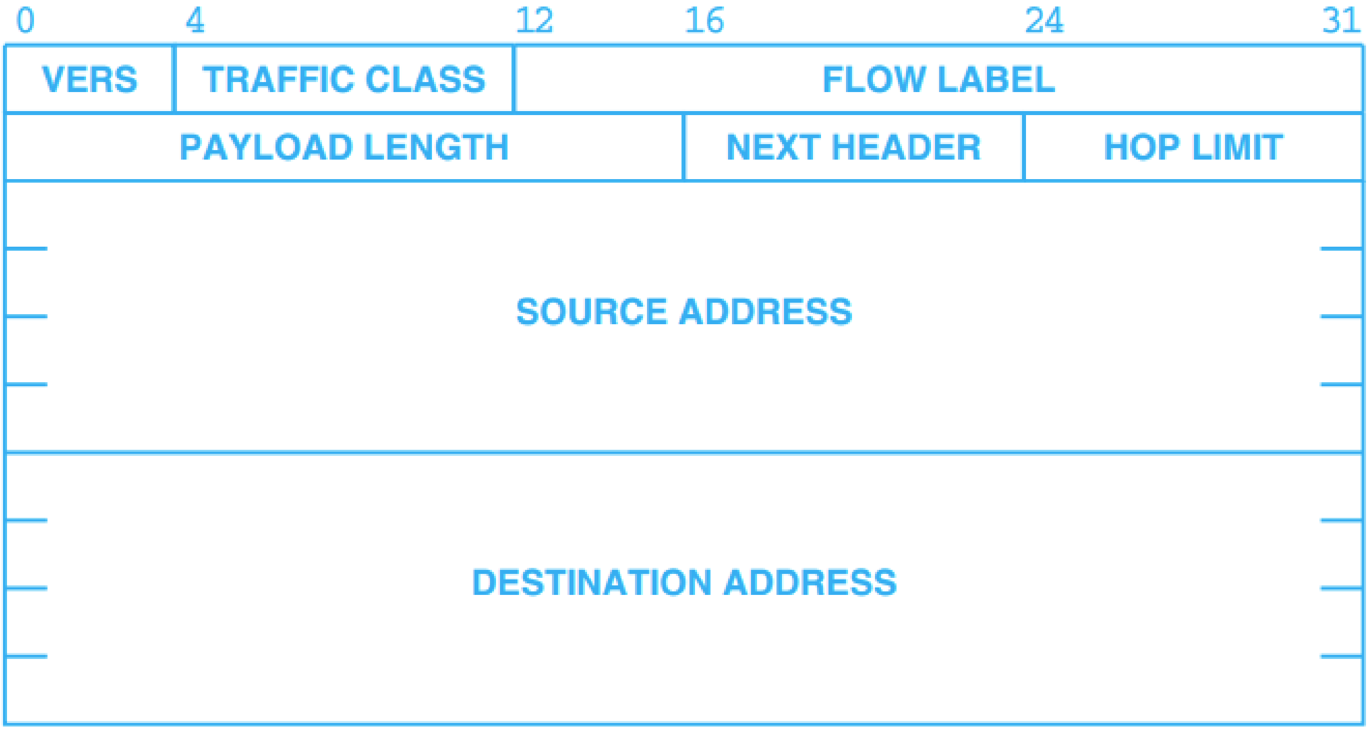


Figure 7.6 The IPv6 base header format; the size is fixed at 40 octets.

- **Version:** The size of the Version field is 4 bits. The Version field shows the version of IP and is set to 6.

- **Traffic Class:** The size of Traffic Class field is 8 bits. Traffic Class field is similar to the IPv4 Type of Service (ToS) field. The Traffic Class field indicates the IPv6 packet's class or priority.

- **Flow Label:** The size of Flow Label field is 20 bits. The Flow Label field provide additional support for real-time datagram delivery and quality of service features. The purpose of Flow Label field is to indicate that this packet belongs to a specific sequence of packets between a source and destination and can be used to prioritized delivery of packets for services like voice.

- **Payload Length:** The size of the Payload Length field is 16 bits. The Payload Length field shows the length of the IPv6 payload, including the extension headers and the upper layer protocol data

- **Next Header:** The size of the Next Header field is 8 bits. The Next Header field shows either the type of the first extension (if any extension header is available) or the protocol in the upper layer such as **TCP**, **UDP**, or ICMPv6.

- **Hop Limit:** The size of the Hop Limit field is 8 bits The Hop Limit field shows the maximum number of routers the IPv6 packet can travel. This Hop Limit field is similar to **IPv4 Time to Live (TTL) field**.

Datagram Type Of Service And Differentiated Services

Informally called Type Of Service (TOS), the 8-bit SERVICE TYPE field in an IPv4 header and the TRAFFIC CLASS field in an IPv6 header specify how the datagram should be handled.

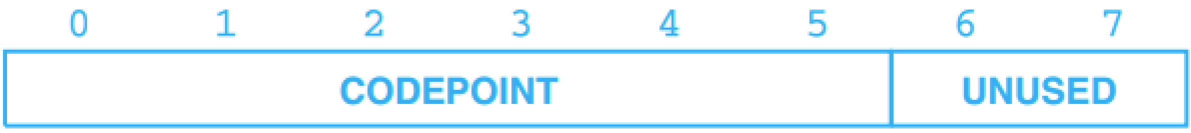
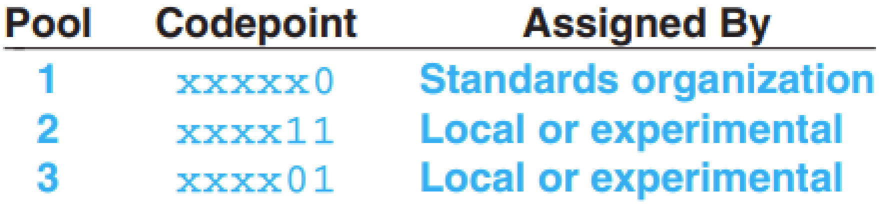


Figure 7.7 The differentiated services (DiffServ) interpretation of bits in the IPv4 *SERVICE TYPE* and IPv6 *TRAFFIC CLASS* header fields.

For example, a router might be configured with a voice service, a video service, a network management service, and a normal data service.



Pool	Codepoint	Assigned By
1	xxxxx0	Standards organization
2	xxxx11	Local or experimental
3	xxxx01	Local or experimental

Figure 7.8 The three administrative pools of DiffServ codepoint values.

We regard the service type specification as a hint to the forwarding algorithm that helps it choose among various paths to a destination based on local policies and its knowledge of the hardware technologies available on those paths. An internet does not guarantee to provide any particular type of service.

Datagram Encapsulation

The idea of carrying one datagram in one network frame is called encapsulation, and is used with both IPv4 and IPv6.

To the underlying network, a datagram is like any other message sent from one machine to another, the network hardware does not recognize the datagram format, nor does it understand the IP destination address.

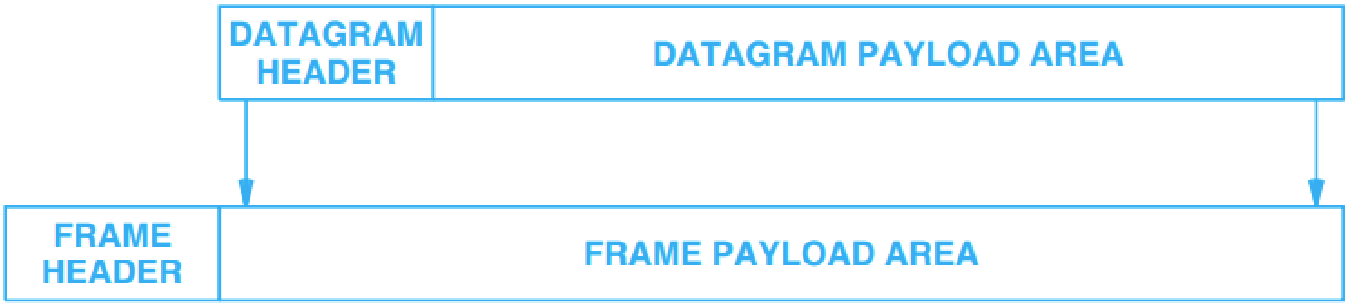


Figure 7.9 The encapsulation of an IP datagram in a frame. The underlying network treats the entire datagram, including the header, as data.

Datagram Size, Network MTU and Fragmentation

For example,Ethernet limits transfers to 1500 octets of data†. We refer to the size limit as the *network's maximum transfer unit, maximum transmission unit or MTU.*

Limiting datagrams to fit the smallest possible MTU in the internet makes transfers inefficient.

However, choosing a large size causes another problem.

Two overarching internet design principles help us understand the dilemma:

- The internet technology should accommodate the greatest possible variety of network hardware.
- The internet technology should accommodate the greatest possible variety of network applications.

when transferring a datagram, check the size to see if the datagram is less than the MTU. If the datagram does not fit into a frame, divide the datagram into smaller pieces called fragments.

Choose the fragment size such that each fragment can be sent in a network frame. ***The process of dividing a datagram is known as fragmentation.***

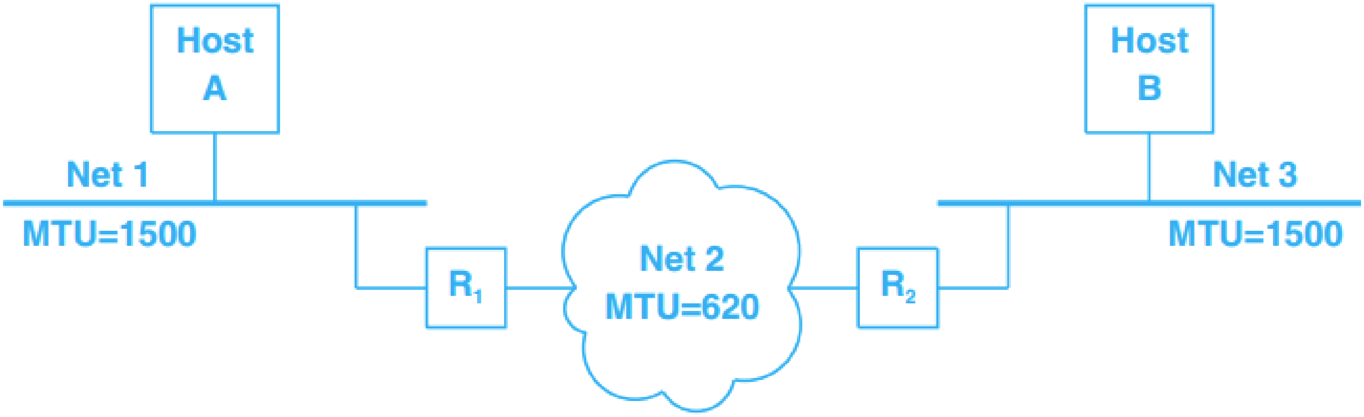


Figure 7.10 An illustration of IPv4 fragmentation. Each router may need to fragment datagrams before sending across network 2.

If an application running on host A sends a 1500-octet datagram to B, the datagram can travel across network 1 in a single frame. However, because network 2 has an MTU of 620, fragmentation is required for the datagram to travel across network 2. In addition to defining the MTU of each individual network, it will be important to consider the MTU along a path through an internet. The path MTU is defined to be the minimum of the MTUs on networks along the path.

IPv4 Datagram Fragmentation

IPv4 fragmentation occurs automatically at any point along the path when a datagram is too large for a network over which it must pass; the source only needs to insure that datagrams can travel over the first hop.

Each fragment is converted to a packet and the following changes happen in the datagram header:

1. The total length field is changed to the size of the fragment.
2. The More Fragment bit (MF bit) is set for all the fragment packets except the last one.
3. The fragment offset field is set, based on the number of fragment that is being set and the MTU.
4. Header Checksum is re-calculated

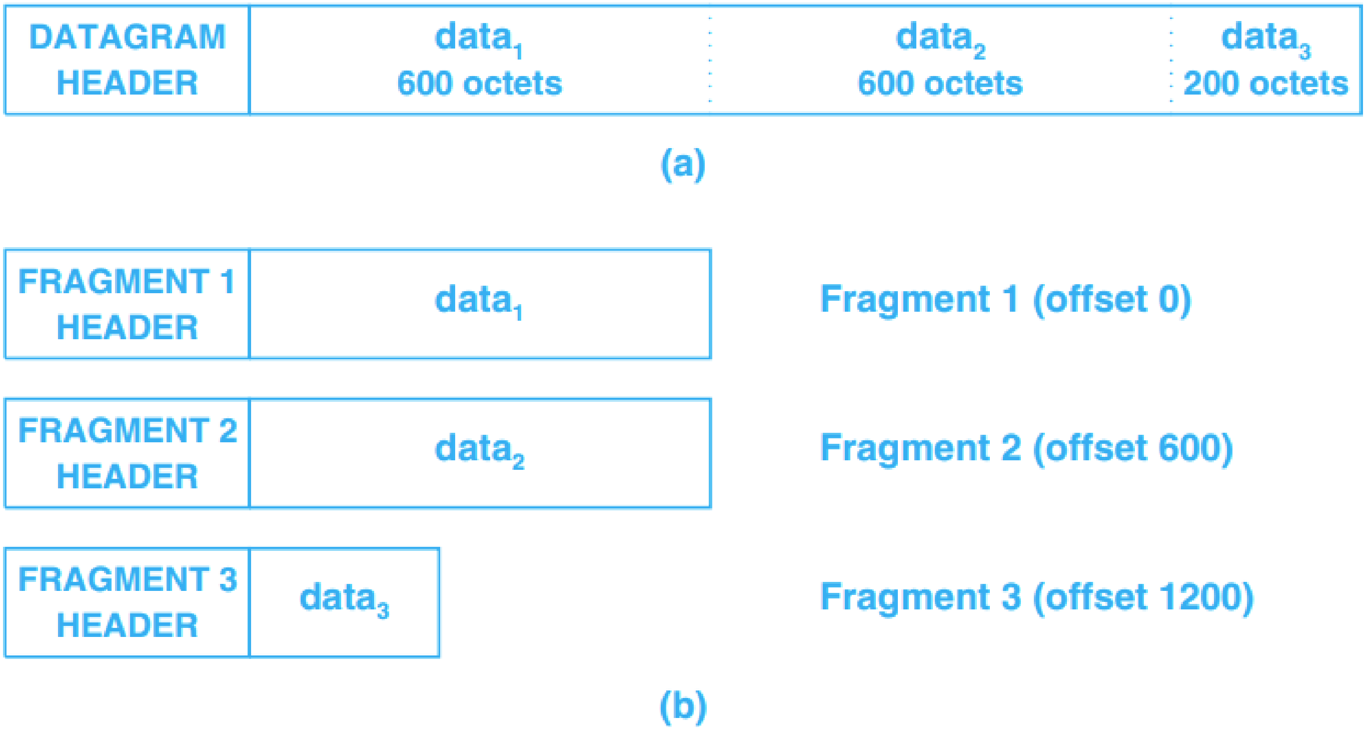


Figure 7.11 (a) An original IPv4 datagram carrying 1400 octets of data and (b) three fragments for an MTU of 620.

IPv6 Fragmentation And Path MTU Discovery (PMTUD)

Instead of delayed fragmentation, IPv6 uses a form of early binding: the original source host is required to find the minimum MTU along the path to the destination and fragment each datagram according to the path it will take. IP routers along the path are not permitted to fragment IPv6 datagrams; if a datagram does not fit into the MTU of a network, the router sends an error message to the original source and drops the datagram.

Because networking technologies used in the Internet do not inform a host about the path MTU, a host must engage in a trial-and-error mechanism to determine the path MTU. Known as ***Path MTU Discovery (PMTUD).***

The mechanism consists of sending an IPv6 datagram that fits in the MTU of the directly-connected network. If a network along the path has a smaller MTU, a router will send an ICMP error message to the original source that specifies the smaller MTU. The host fragments datagrams according to the new path MTU and tries again. If a later network along the path has an MTU that is even smaller, another router will send an error message. By repeatedly probing, a host will eventually find the smallest MTU along the path.

The IPv6 base header does not include fields to specify fragmentation. Therefore, when it fragments an IPv6 datagram, a source inserts a Fragment Extension Header into each fragment.

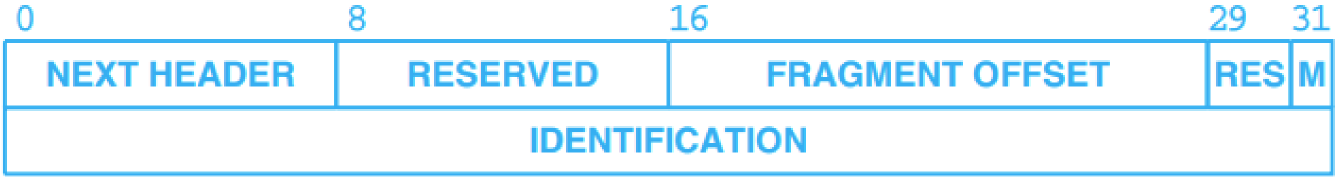


Figure 7.12 The format of an IPv6 *Fragmentation Extension Header*.

- The extension header includes the required **NEXT HEADER** field.
- It also includes two fields that are reserved for future use.
- The remaining three fields have the same meaning as IPv4 fragmentation control fields.
- A 13-bit **FRAGMENT OFFSET** field specifies where in the original datagram this fragment belongs, the **M** bits is a more fragments bit that specifies whether a fragment is the final (rightmost) fragment of the original datagram
- The **IDENTIFICATION** field contains a unique datagram ID that is shared by all the fragments of a datagram.

Datagram Reassembly

Fragments must be reassembled to produce a complete copy of the original datagram. The question arises: Should a datagram be reassembled when it reaches a network with a larger MTU, or should the datagram remain fragmented and the fragments be transported to the ultimate destination.

In a TCP/IP internet, once a datagram has been fragmented, the fragments travel as separate datagrams all the way to the ultimate destination where they are reassembled. Preserving fragments all the way to the ultimate destination may seem odd because the approach has two disadvantages.

First, if only one network along the path has a small MTU, sending small fragments over the other networks is inefficient, because transporting small packets means more overhead than transporting large packets.

Second, if any fragments are lost, the datagram cannot be reassembled. The mechanism used to handle fragment loss consists of a reassembly timer. The ultimate destination starts a timer when a fragment arrives for a given datagram. If the timer expires before all fragments arrive, the receiving machine discards the surviving fragments. The source must retransmit the entire datagram; there is no way for the receiver to request individual fragments. Thus, the probability of datagram loss increases when fragmentation occurs because the loss of a single fragment results in loss of the entire datagram.

In the Internet, the ultimate destination reassembles fragments. The design means that routers do not need to store fragments or keep other information about packets.

Header Fields Used For Datagram Reassembly

Three fields in an IPv4 datagram header or an IPv6 Fragment Extension Header control reassembly of datagrams: **IDENTIFICATION, FLAGS (M in IPv6), and FRAGMENT OFFSET**

IDENTIFICATION contains a unique integer that identifies the datagram. That is, each datagram sent by a given source has a unique ID. Thus, each fragment has exactly the same IDENTIFICATION number as the original datagram.

FRAGMENT OFFSET field specifies the offset in the original datagram of the payload being carried in the fragment, measured in units of 8 octets.

In IPv4, the low-order two bits of the **3-bit FLAGS** field control fragmentation

The following can be their possible configuration:

Bit 0: is reserved and has to be set to zero

Bit 1: means do not fragment

Bit 2: means more fragments.

Time To Live (IPv4) And Hop Limit (IPv6)

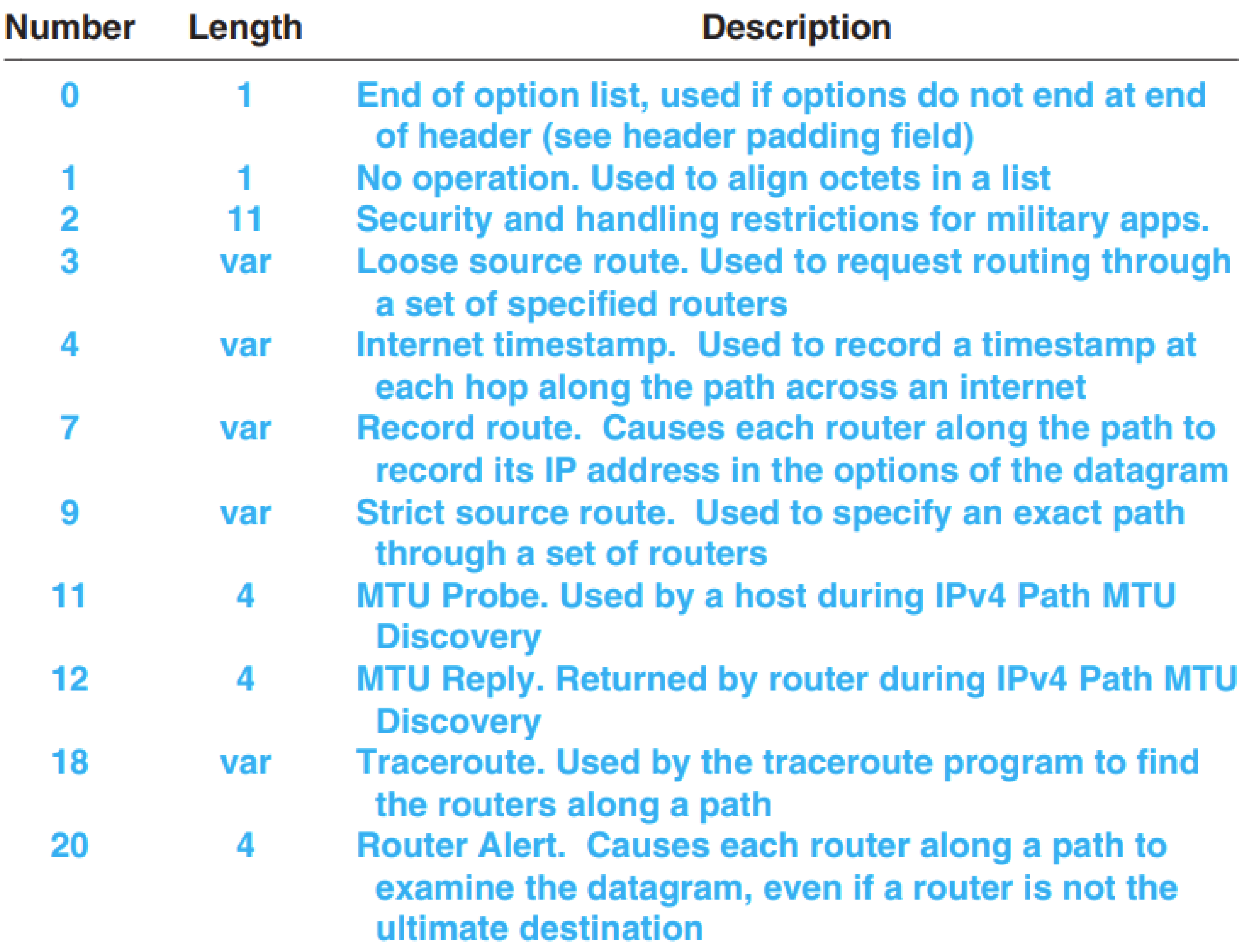
Each network that a datagram traverses counts as one network hop. Thus, in practice, the *TTL field* is now used to specify how many hops a datagram may traverse before being discarded. IPv6 includes the exact same concept. To clarify the meaning, IPv6 uses the name *HOP LIMIT* in place of *TIME-TO-LIVE*.

IP software in each machine along a path from source to destination decrements the field known as TIME-TO-LIVE (IPv4) or HOP LIMIT (IPv6). When the field reaches zero the datagram is discarded.

Optional IP Items

Both IPv4 and IPv6 define optional items that can be included in a datagram. In IPv4, the IP OPTIONS field that follows the destination address is used to send optional items. In IPv6, each of the extension headers is optional, and a given datagram may include multiple extensions.

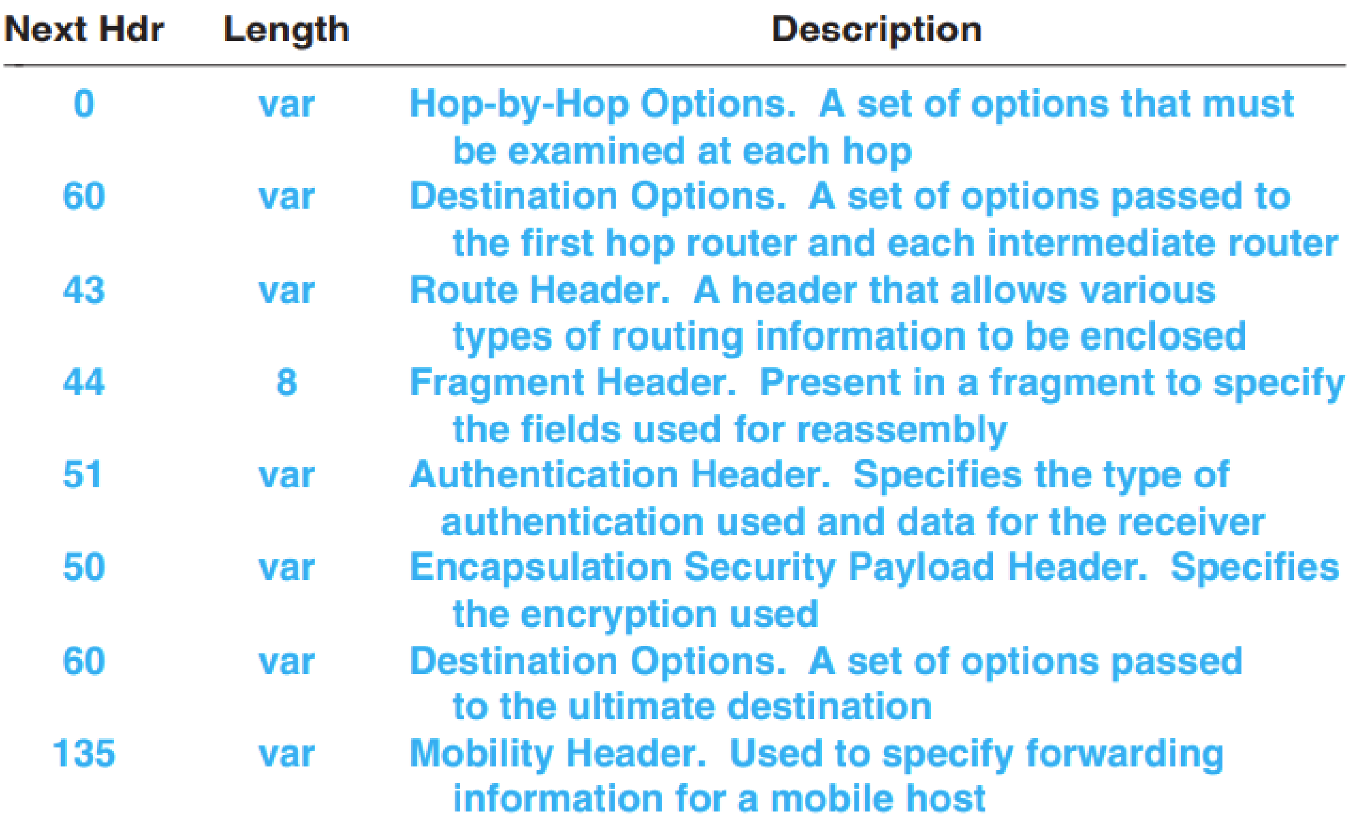
IPv4 Options



Number	Length	Description
0	1	End of option list, used if options do not end at end of header (see header padding field)
1	1	No operation. Used to align octets in a list
2	11	Security and handling restrictions for military apps.
3	var	Loose source route. Used to request routing through a set of specified routers
4	var	Internet timestamp. Used to record a timestamp at each hop along the path across an internet
7	var	Record route. Causes each router along the path to record its IP address in the options of the datagram
9	var	Strict source route. Used to specify an exact path through a set of routers
11	4	MTU Probe. Used by a host during IPv4 Path MTU Discovery
12	4	MTU Reply. Returned by router during IPv4 Path MTU Discovery
18	var	Traceroute. Used by the traceroute program to find the routers along a path
20	4	Router Alert. Causes each router along a path to examine the datagram, even if a router is not the ultimate destination

Figure 7.13 Examples of IPv4 options along with their length and a brief description of each.

IPv6 Optional Extensions



Next Hdr	Length	Description
0	var	Hop-by-Hop Options. A set of options that must be examined at each hop
60	var	Destination Options. A set of options passed to the first hop router and each intermediate router
43	var	Route Header. A header that allows various types of routing information to be enclosed
44	8	Fragment Header. Present in a fragment to specify the fields used for reassembly
51	var	Authentication Header. Specifies the type of authentication used and data for the receiver
50	var	Encapsulation Security Payload Header. Specifies the encryption used
60	var	Destination Options. A set of options passed to the ultimate destination
135	var	Mobility Header. Used to specify forwarding information for a mobile host

Figure 7.14 Example options headers used with IPv6 and the *NEXT HEADER* value assigned to each.

Options Processing During Fragmentation

If an option must be processed by intermediate routers, the option is copied into each fragment. However, if the option is only used at the ultimate destination, the option is copied into the header of the first fragment but not the rest. Omitting unnecessary options from later fragments reduces the total number of bits transmitted.

IPv4 Processing Options During Fragmentation

- The IP standard specifies that the record route option should only be copied into one of the fragments.
- The standard specifies that a source route option must be copied into all fragments.

IPv6 Processing Options During Fragmentation

IPv6 divides a datagram into two conceptual pieces: an initial piece that is classified as unfragmentable and the remainder, which is classified as fragmentable.

In particular, the *Hop-By-Hop Header and Route Header* are not fragmentable; other extension headers are fragmentable.

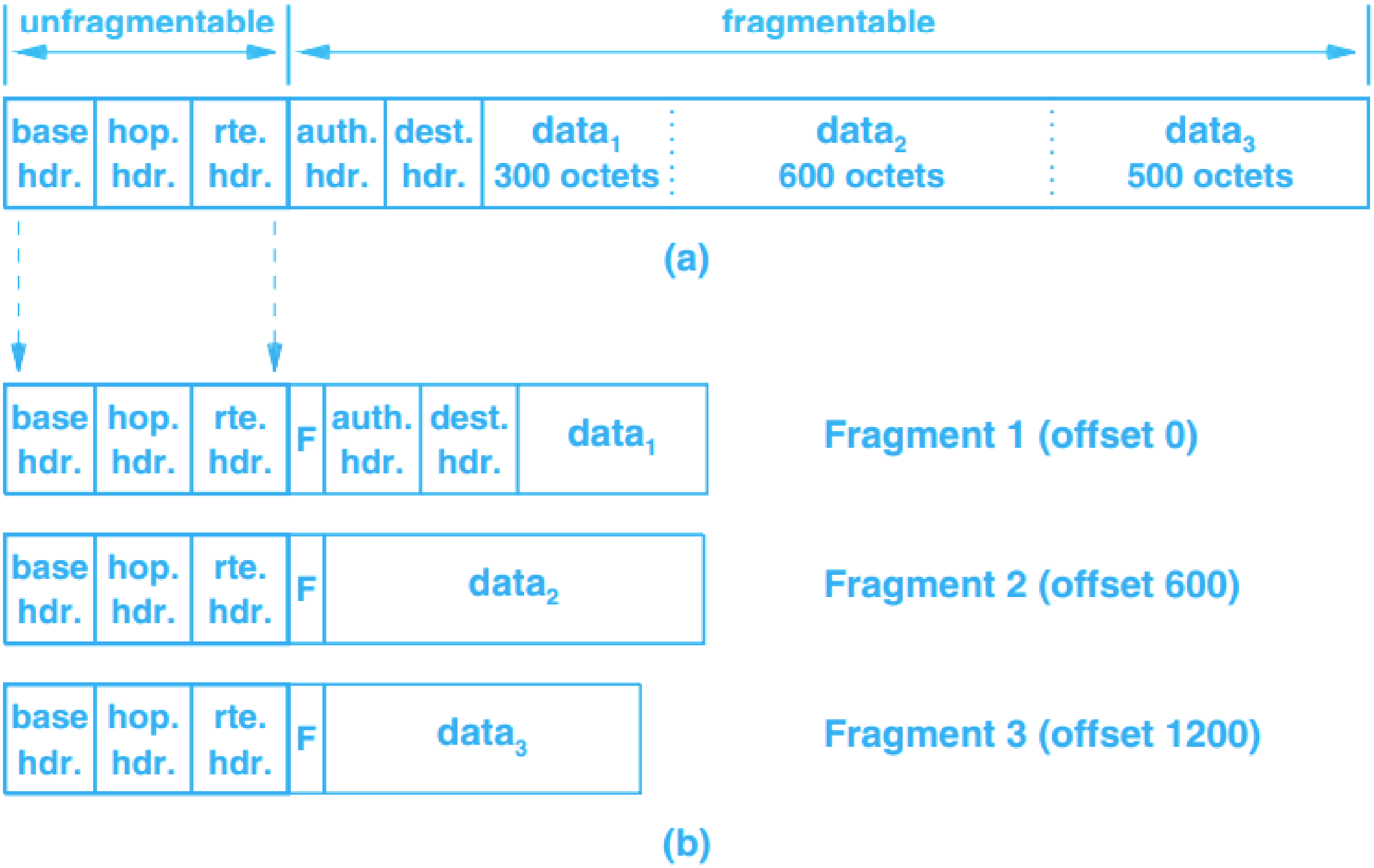


Figure 7.16 IPv6 fragmentation with (a) an IPv6 datagram with extension headers divided into fragmentable and unfragmentable pieces, and (b) a set of fragments.

Internet Protocol: Error And Control Messages (ICMP)

The Internet Control Message Protocol

The Internet Control Message Protocol allows routers to send error or control messages back to the source of a datagram that caused a problem. ICMP messages are not usually delivered to applications. We think of ICMP as providing communication between an ICMP module on one machine and an ICMP module on another.

Chief advantage of allowing hosts to use ICMP is that it provides a single mechanism used for all control and information messages.

Error Reporting Vs. Error Correction

Technically, ICMP is an error reporting mechanism.

When a datagram causes an error, ICMP can only report the error condition back to the original source of the datagram; the source must relate the error to an individual application program or take other action to correct the problem.

ICMP Message Delivery

Since each ICMP message travels in an IP datagram, two levels of encapsulation are required. Figure 9.1 illustrates the concept.

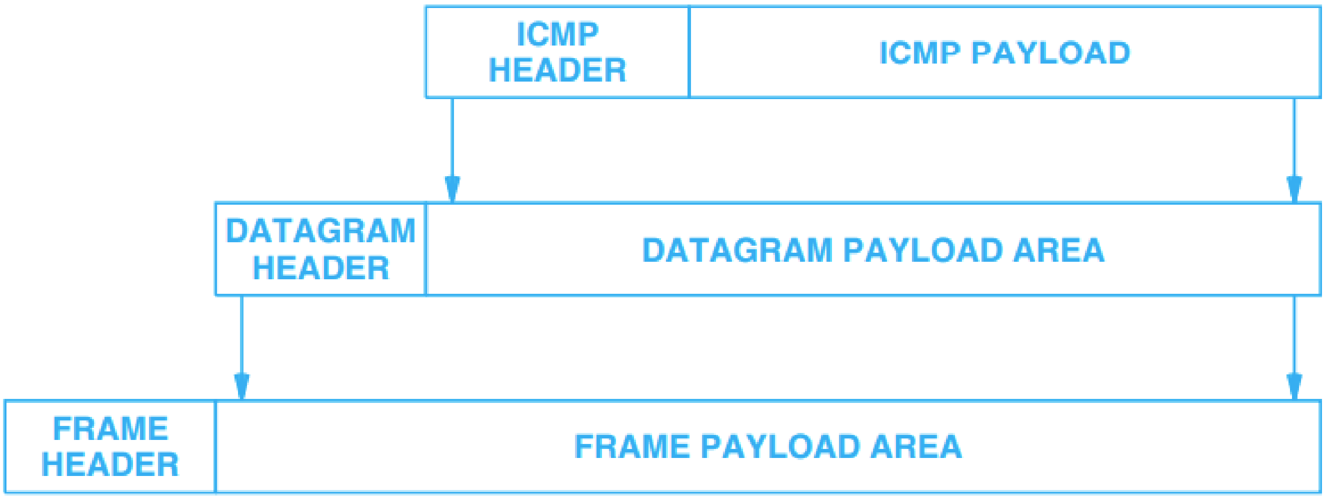


Figure 9.1 The two levels of encapsulation used when an ICMP message is sent across a network.

IPv4 uses the PROTOCOL field in the datagram header as a type field. When an ICMP message is carried in the payload area of an IPv4 datagram, the PROTOCOL field is set to 1.

IPv6 uses the NEXT HEADER field to specify the type of the item being carried.

Conceptual Layering

ICMP represents an important exception. Although each ICMP message is encapsulated in an IP datagram, ICMP is not considered a higher-level protocol. Instead, ICMP is a required part of IP, which means ICMP is classified as a Layer 3 protocol.

ICMP must send error reports to the original source, so an ICMP message must travel across multiple underlying networks to reach its final destination. Thus, ICMP messages cannot be delivered by a Layer 2 transport alone.

ICMP Message Format

The standards define two sets of ICMP messages: a set for IPv4 and a larger set for IPv6. In both versions of IP, each ICMP message has its own format. However, all ICMP messages begin with the same three fields.

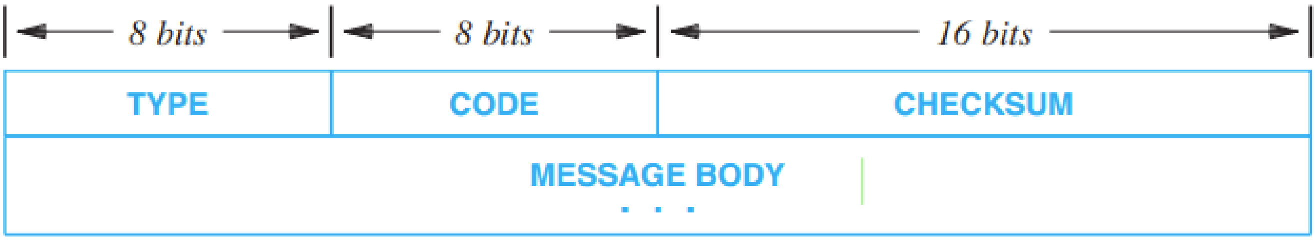


Figure 9.2 The first three fields in each ICMP message.

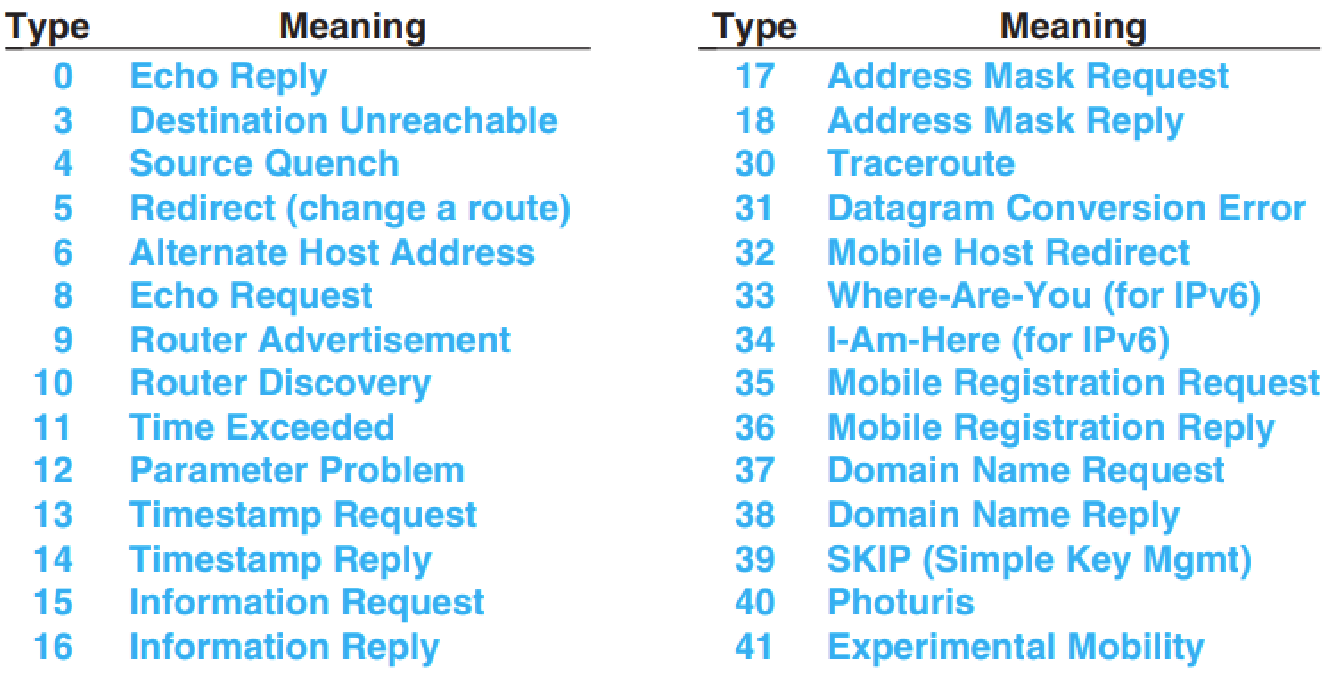
TYPE field identifies the specific ICMP message that follows.

CODE field in an ICMP message provides further information about the message type.

The third field in each ICMP message consists of a 16-bit **CHECKSUM** that is computed over the entire ICMP message.

The **message body** always includes the header plus additional octets from the datagram that caused the problem

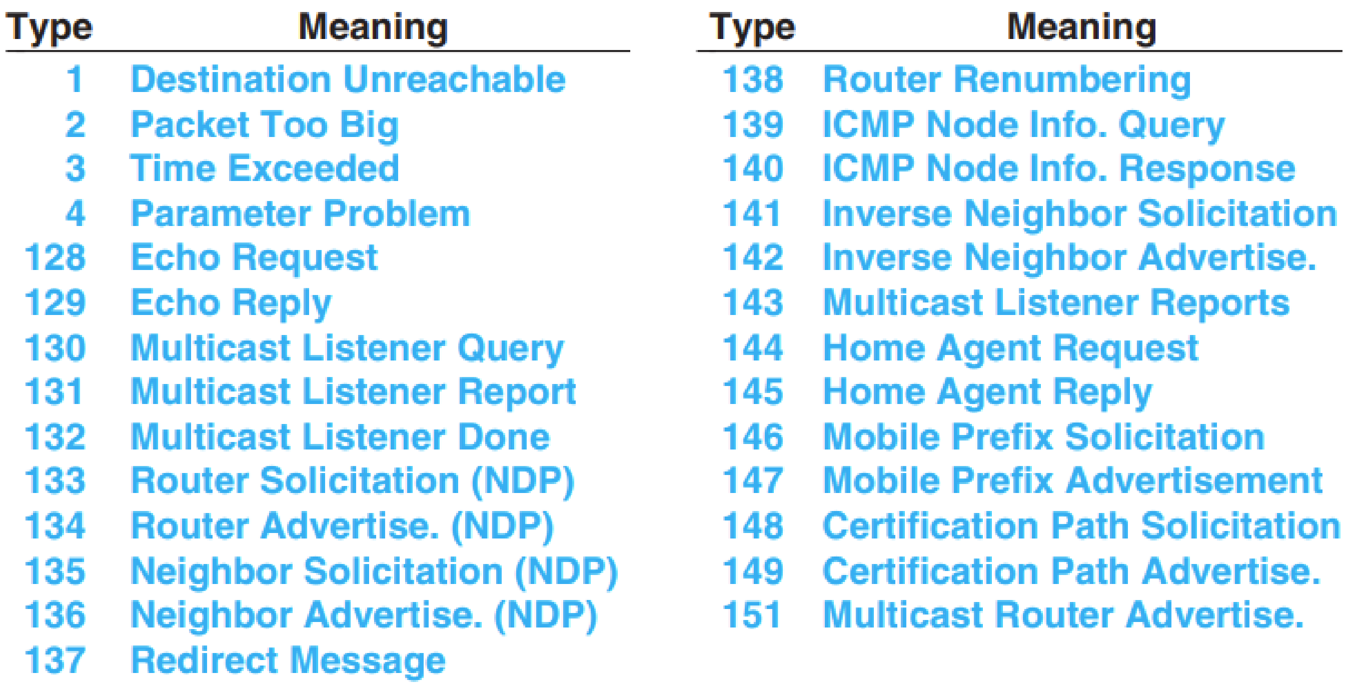
ICMP Message Types Used With IPv4



Type	Meaning	Type	Meaning
0	Echo Reply	17	Address Mask Request
3	Destination Unreachable	18	Address Mask Reply
4	Source Quench	30	Traceroute
5	Redirect (change a route)	31	Datagram Conversion Error
6	Alternate Host Address	32	Mobile Host Redirect
8	Echo Request	33	Where-Are-You (for IPv6)
9	Router Advertisement	34	I-Am-Here (for IPv6)
10	Router Discovery	35	Mobile Registration Request
11	Time Exceeded	36	Mobile Registration Reply
12	Parameter Problem	37	Domain Name Request
13	Timestamp Request	38	Domain Name Reply
14	Timestamp Reply	39	SKIP (Simple Key Mgmt)
15	Information Request	40	Photuris
16	Information Reply	41	Experimental Mobility

Figure 9.3 Example ICMPv4 message types and the meaning of each. Values not listed are unassigned or reserved.

ICMP Message Types Used With IPv6



Type	Meaning	Type	Meaning
1	Destination Unreachable	138	Router Renumbering
2	Packet Too Big	139	ICMP Node Info. Query
3	Time Exceeded	140	ICMP Node Info. Response
4	Parameter Problem	141	Inverse Neighbor Solicitation
128	Echo Request	142	Inverse Neighbor Advertise.
129	Echo Reply	143	Multicast Listener Reports
130	Multicast Listener Query	144	Home Agent Request
131	Multicast Listener Report	145	Home Agent Reply
132	Multicast Listener Done	146	Mobile Prefix Solicitation
133	Router Solicitation (NDP)	147	Mobile Prefix Advertisement
134	Router Advertise. (NDP)	148	Certification Path Solicitation
135	Neighbor Solicitation (NDP)	149	Certification Path Advertise.
136	Neighbor Advertise. (NDP)	151	Multicast Router Advertise.
137	Redirect Message		

Figure 9.4 Example ICMPv6 message types and the meaning of each.