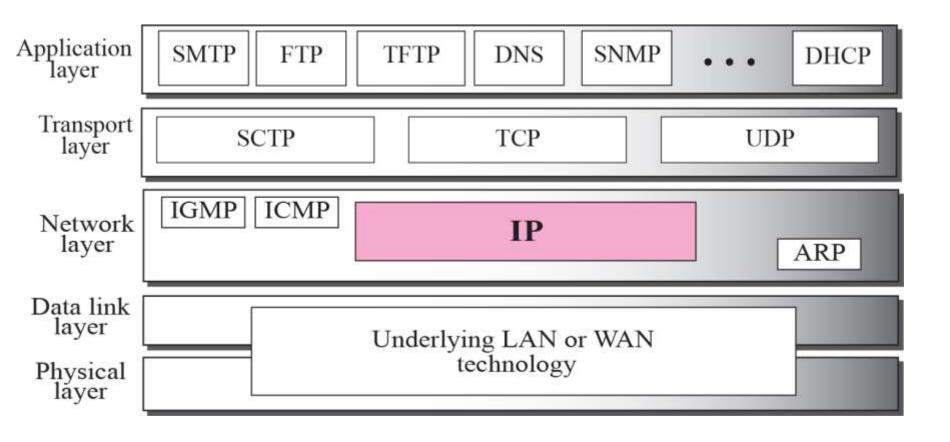
7-1 INTRODUCTION

The Internet Protocol (IP) is the transmission mechanism used by the TCP/IP protocols at the network layer.

Topics Discussed in the Section

✓ Relationship of IP to the rest of the TCP/IP Suite





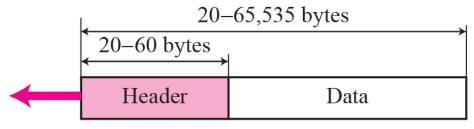
7-2 DATAGRAMS

Packets in the network (internet) layer are called datagrams. A datagram is a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery. It is customary in TCP/IP to show the header in 4-byte sections. A brief description of each field is in order.

Topics Discussed in the Section

- **✓** Format of the datagram packet
- **✓** Some examples

Figure 7.2 IP datagram

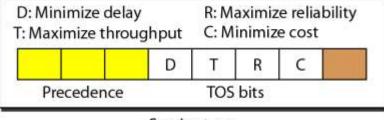


a. IP datagram

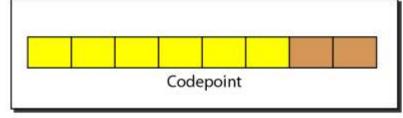
0	3	4 7	8 15	16		31
	VER 4 bits	HLEN 4 bits	Service type 8 bits			
	Identification 16 bits			Flags 3 bits	Fragmentation offset 13 bits	
	Time to live 8 bits		Protocol 8 bits	Header checksum 16 bits		
	Source IP address					
	Destination IP address					
5	Options + padding (0 to 40 bytes)					

b. Header format

Figure 20.6 Service type or differentiated services



Service type



Differentiated services

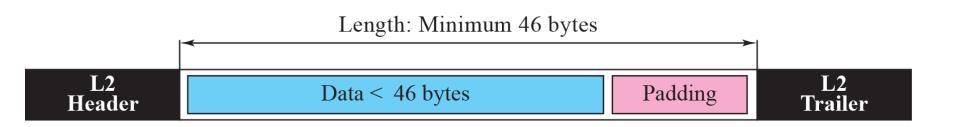
 Table 20.1
 Types of service

TOS Bits	Description		
0000	Normal (default)		
0001	Minimize cost		
0010	Maximize reliability		
0100	Maximize throughput		
1000	Minimize delay		



The total length field defines the total length of the datagram including the header.







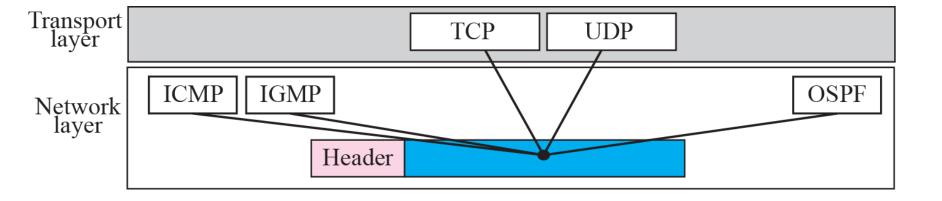




Table 7.2 Protocols

Value	Protocol	Value	Protocol
1	ICMP	17	UDP
2	IGMP	89	OSPF
6	TCP		

An IP packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

An IP packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 left-most bits (0100) show the version, which is correct. The next 4 bits (0010) show the wrong header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

In an IP packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

In an IP packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8×4 or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

In an IP packet, the value of HLEN is 5_{16} and the value of the total length field is 0028_{16} . How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5×4 or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40 – 20).

An IP packet has arrived with the first few hexadecimal digits as shown below:

45000028000100000102...

How many hops can this packet travel before being dropped? The data belong to what upper layer protocol?

Solution

To find the time-to-live field, we skip 8 bytes (16 hexadecimal digits). The time-to-live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper layer protocol is IGMP (see Table 7.2)

7-3 FRAGMENTATION

A datagram can travel through different networks. Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame. The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled. The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.

Topics Discussed in the Section

- **✓** Maximum Transfer Unit (MTU)
- **✓ Fields Related to Fragmentation**

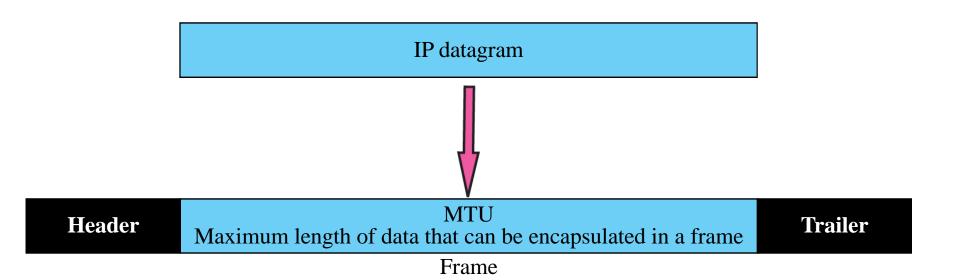


 Table 20.5
 MTUs for some networks

Protocol	MTU
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

-

Note

Only data in a datagram is fragmented.

D: Do not fragment M: More fragments





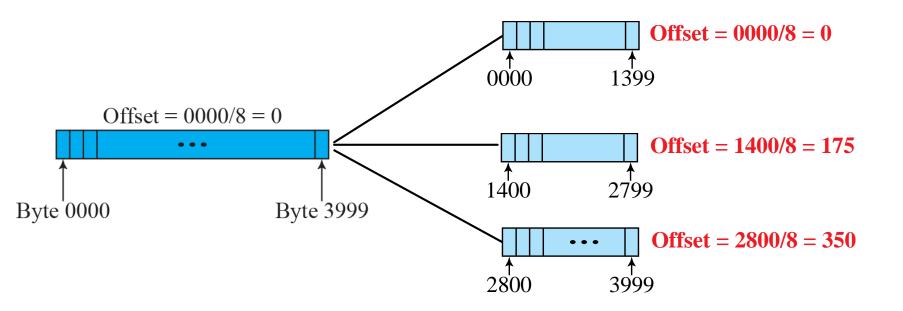
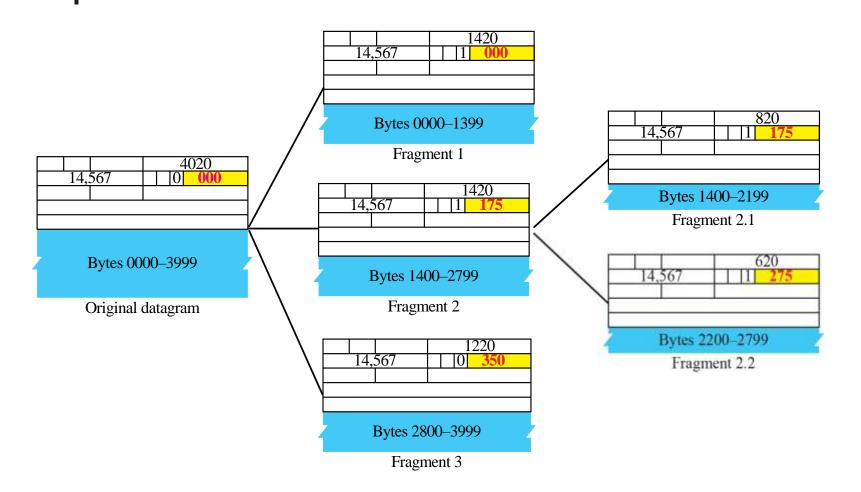


Figure 7.9 Detailed fragmentation example



A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A nonfragmented packet is considered the last fragment.

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset). See also the next example.

A packet has arrived with an M bit value of 1 and a fragmentation offset value of zero. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is 1, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment.

A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length of the data.

A packet has arrived in which the offset value is 100, the value of HLEN is 5 and the value of the total length field is 100. What is the number of the first byte and the last byte?

Solution

The first byte number is $100 \times 8 = 800$. The total length is 100 bytes and the header length is 20 bytes (5×4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.

Figure 20.13 Example of checksum calculation in IPv4

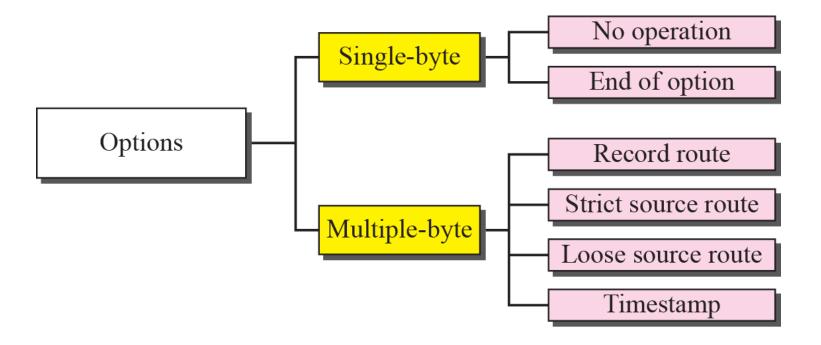
4	5	0				28	
	ì			0		0	
4		17				0	A
	10.12.14.5						
			12.6	7.9			
4, 5	, and 0	→	4	5	0	0	
	28	\longrightarrow	0	0	1	C	
	1	\longrightarrow	0	0	0	1	
(and 0	\longrightarrow	0	0	0	0	
4	and 17	\longrightarrow	0	4	1	1	
	0	\longrightarrow	0	0	0	0	
	10.12	\longrightarrow	0	Α	0	C	
	14.5	\longrightarrow	0	E	0	5	
	12.6	\longrightarrow	0	C	0	6	
	7.9	\longrightarrow	0	7	0	9	
	Sum	→	7	4	4	E	
Checksum		\longrightarrow	8	В	В	1	

7-4 OPTIONS

The header of the IP datagram is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long and was discussed in the previous section. The variable part comprises the options, which can be a maximum of 40 bytes.

Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging. Although options are not a required part of the IP header, option processing is required of the IP software.

Figure 7.11 Categories of options



changes implemented in the protocol in addition to changing address size and format.

- Better header format. IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- New options. IPv6 has new options to allow for additional functionalities.
- Allowance for extension. IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.

- Support for resource allocation. In IPv6, the type-of-service field has been removed, but two new fields, traffic class and flow label, have been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- Support for more security. The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet

27-2 PACKET FORMAT

The IPv6 packet is shown in Figure 27.1. Each packet is composed of a mandatory base header followed by the payload. The payload consists of two parts: optional extension headers and data from an upper layer. The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.

Figure 20.15 IPv6 datagram header and payload

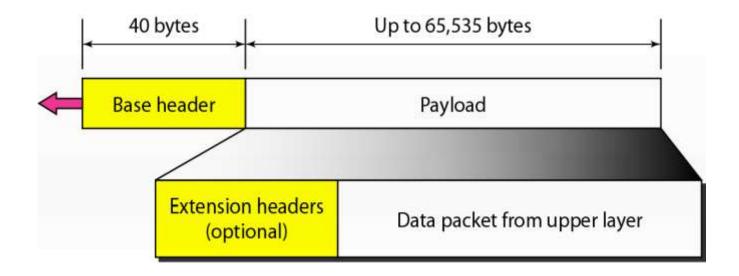


Figure 27.1 IPv6 datagram

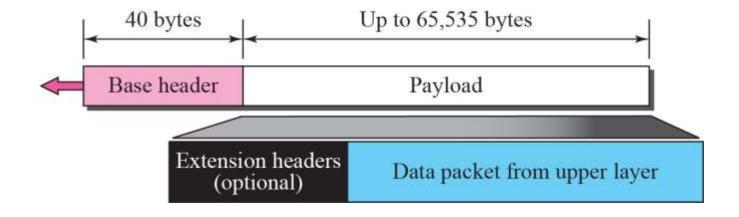
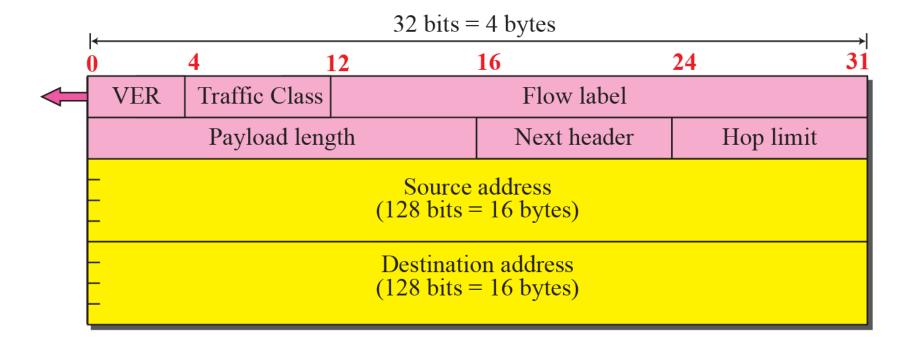


Figure 27.2 Format of the base header



- Version. The 4-bit version field defines the version number of the IP. For IPv6, the value is 6.
- Traffic class. The 8-bit traffic class field is used to distinguish different payloads with different delivery requirements. It replaces the type-of-service field in IPv4.
- Flow label. The flow label is a 20-bit field that is designed to provide special han□dling for a particular flow of data. We will discuss this field later.
- Payload length. The 2-byte payload length field defines the length of the IP datagram excluding the header. Note that IPv4 defines two fields related to the length: header length and total length. In IPv6, the length of the base header is fixed (40 bytes); only the length of the payload needs to be defined..

■ Next header. The next header is an 8bit field defining the type of the first extension header (if present) or the type of the data that follows the base header in the datagram. This field is similar to the protocol field in IPv4, but we talk more about it when we discuss the payload

- Hop limit. The 8-bit hop limit field serves the same purpose as the TTL field in IPv4.
- Source and destination addresses. The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram. The destination address field is a 16-byte (128-bit) Internet address that identifies the destination of the datagram.
- Payload. Compared to IPv4, the payload field in IPv6 has a different format



 Table 27.1
 Next Header Codes

Code	Next Header	Code	Next Header
0	Hop-by-hop option	44	Fragmentation
2	ICMP	50	Encrypted security payload
6	TCP	51	Authentication
17	UDP	59	Null (No next header)
43	Source routing	60	Destination option

Figure 27.3 Extension header format

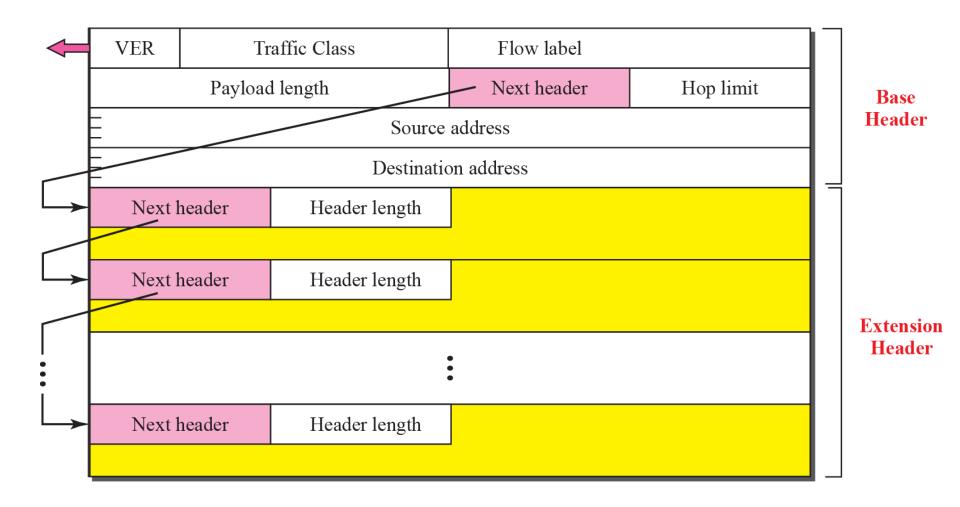


Table 20.9 Comparison between IPv4 and IPv6 packet

Comparison

- The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
- The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
- 3. The total length field is eliminated in IPv6 and replaced by the payload length field.
- The identification, flag, and offset fields are eliminated from the base header in IPv6. They
 are included in the fragmentation extension header.
- 5. The TTL field is called hop limit in IPv6.
- 6. The protocol field is replaced by the next header field.
- The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
- 8. The option fields in IPv4 are implemented as extension headers in IPv6.

Figure 20.17 Extension header

types

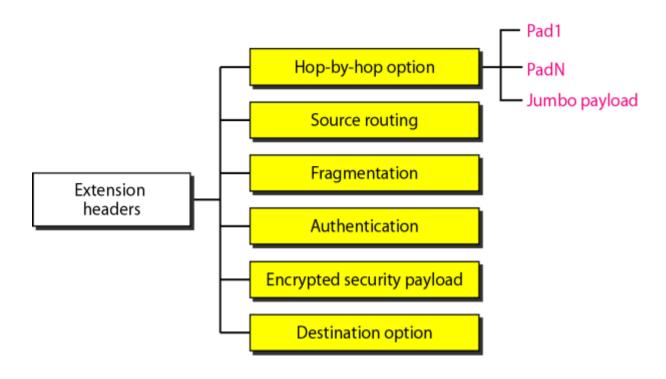


Table 20.10 Comparison between IPv4 options and IPv6 extension

Comparison

- The no-operation and end-of-option options in IPv4 are replaced by Pad1 and PadN options in IPv6.
- 2. The record route option is not implemented in IPv6 because it was not used.
- 3. The timestamp option is not implemented because it was not used.
- 4. The source route option is called the source route extension header in IPv6.
- The fragmentation fields in the base header section of IPv4 have moved to the fragmentation extension header in IPv6.
- 6. The authentication extension header is new in IPv6.
- 7. The encrypted security payload extension header is new in IPv6.

Transition from IPv4 to IPv6

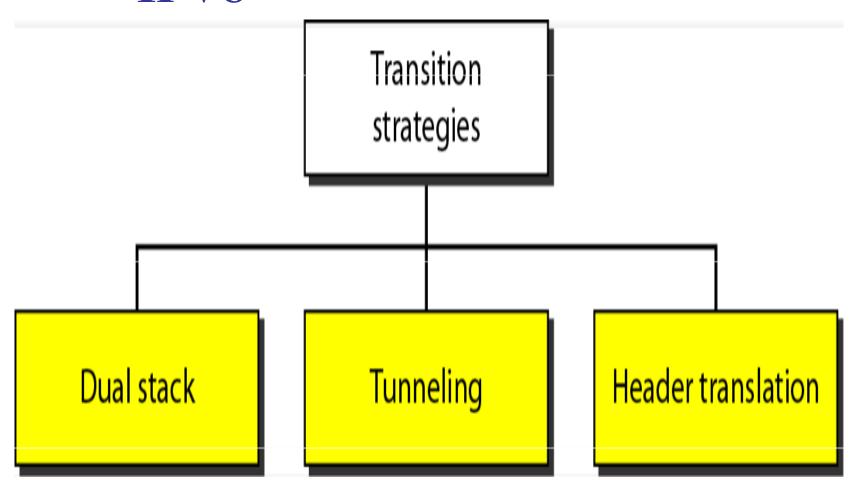


Figure 20.19 Dual

stack

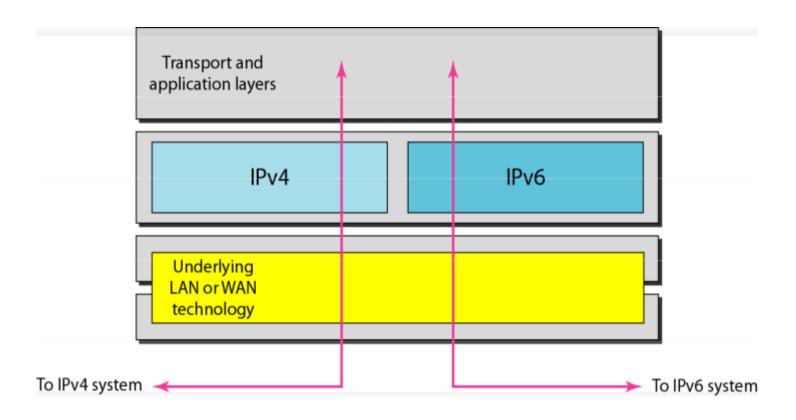


Figure 20.20 Tunneling

strategy

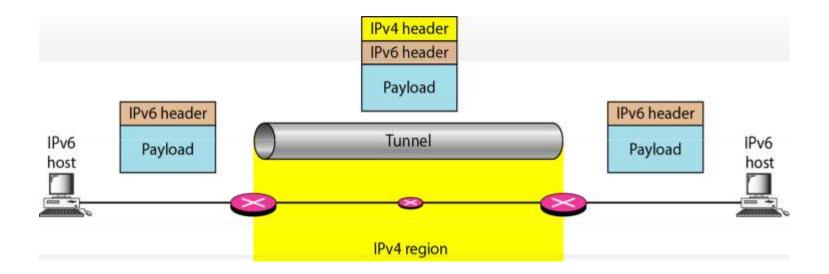


Figure 20.21 Header translation strategy

