IPv4 Header

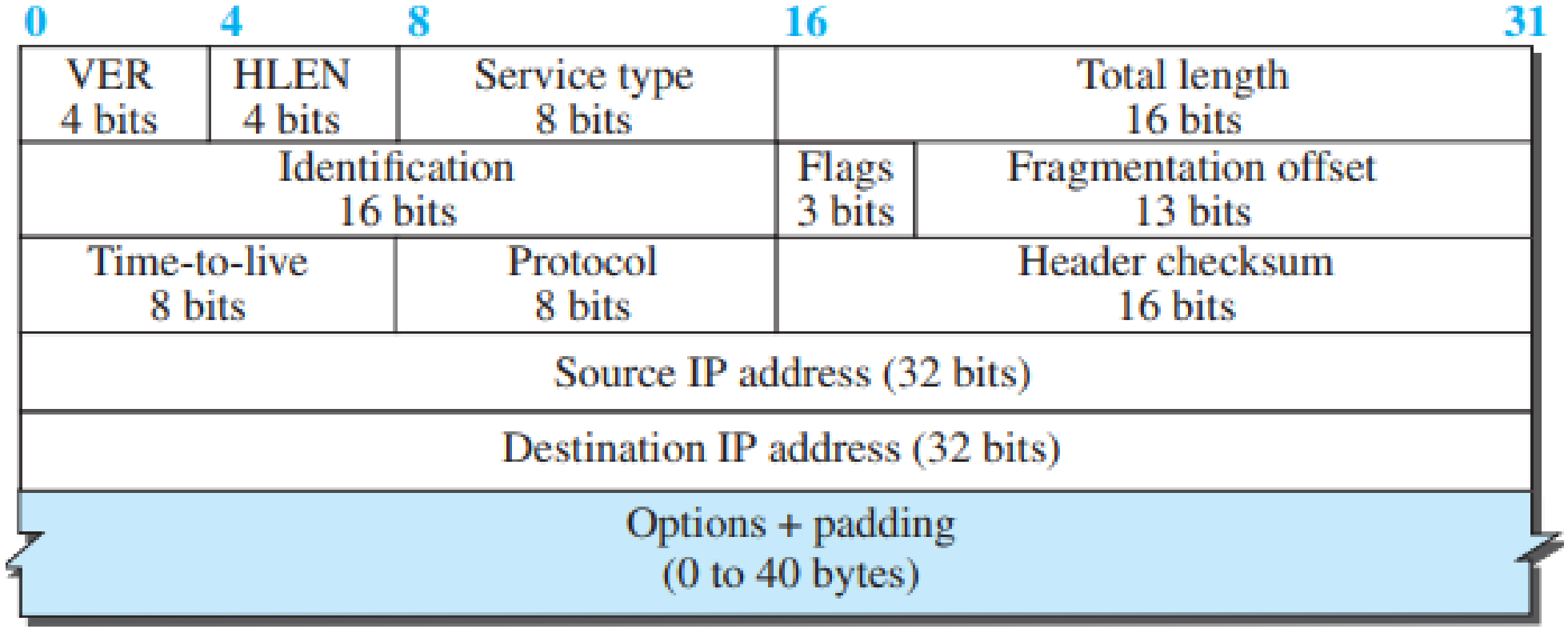


Fig 1(a). IPv4 Header

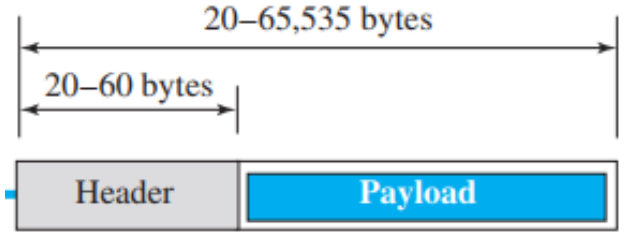


Fig 1(b). IPv4 Datagram

IPv4 Header

- **Version Number (4 bits)**
  - Defines version of IP protocol

- **Header Length (4 bits)**
  - Defines total length of datagram header
  - Multiply by 4 to get the length in bytes
  - Device needs to know when the header stops, and actual data started

- **Service Type (8 bits)**
  - Defines type of service
  - How datagram should be handled
  - Also called ***Differentiated Services Code Point (DSCP)***

- **Total Length (16 bits)**
  - Defines total length (header + data) of IP datagram
  - Receiving device knows when the packet arrived completely
  - Length of data = Total length – (Header length * 4)

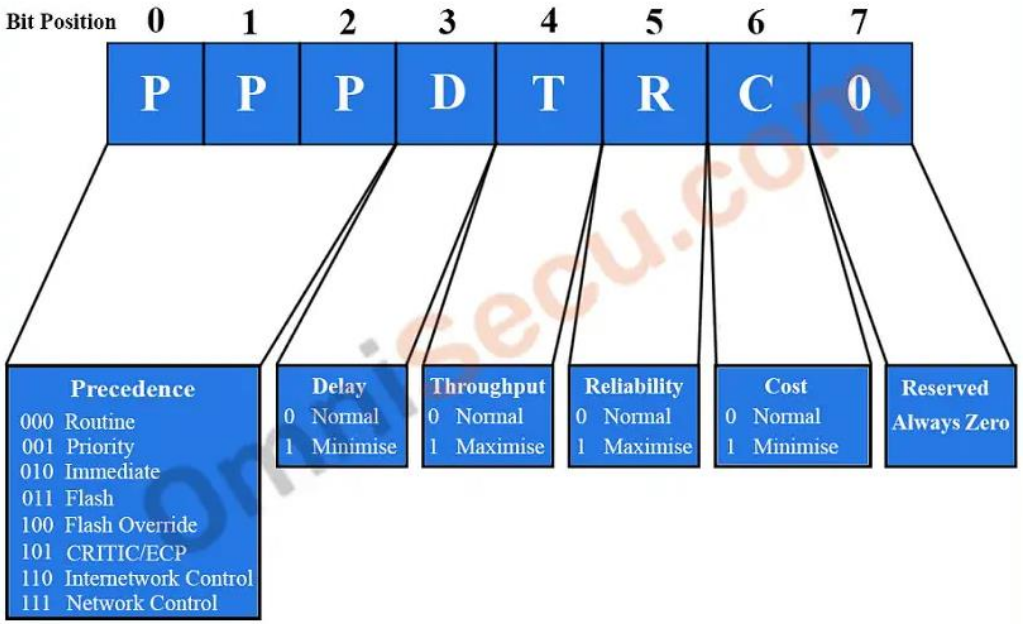


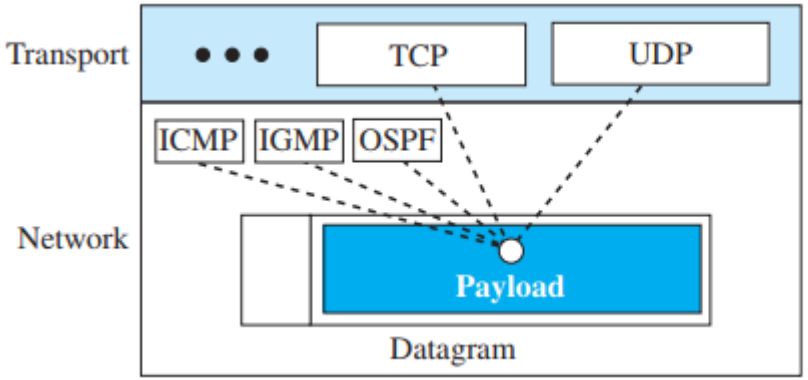
Fig 2. Service Type

- **Identification, Flags, and Fragmentation Offset (16 bits +3 bits +13 bits)**

IPv4 Header

- **Time-to-live (8 bits)**
  - Datagram may be circulating from one node to another
  - Create extra traffic on the internet
  - Use to control maximum number of hops to be travelled
  - Each router decrements this number by 1
  - When the value reaches 0, router discards the datagram

- **Protocol (8 bits)**
  - A packet (payload) from upper layer belongs to different protocol (TCP/UDP)
  - A packet can be from IP layer
  - Destination knows which protocol packet belongs to
  - Multiplexing at source, demultiplexing at destination



Some protocol values

ICMP	01
IGMP	02
TCP	06
UDP	17
OSPF	89

Fig 3. Protocol

IPv4 Header

LNMIIT
The LNM Institute of Information Technology

- **Header checksum (16 bits)**
  - Error checking of payload is done by Transport layer
  - Datagram header added by IP
  - Any error in the header is disastrous
  - Needs to be recalculated and checked at each router
- **Source IP address (32 bits)**
- **Destination IP address (32 bits)**
- **Options (40 bytes)**
  - Used for network testing, management, and debugging purpose
  - Optional field
  - Generally used by network administrator
- **Padding**
  - IP header length should be multiple of 32-bits
  - 0 bits are padded

Fragmentation

- **Maximum Transfer Unit (MTU)**
  - Maximum size of IP datagram that can be encapsulated in a Frame
  - Size of frame payload <= MTU

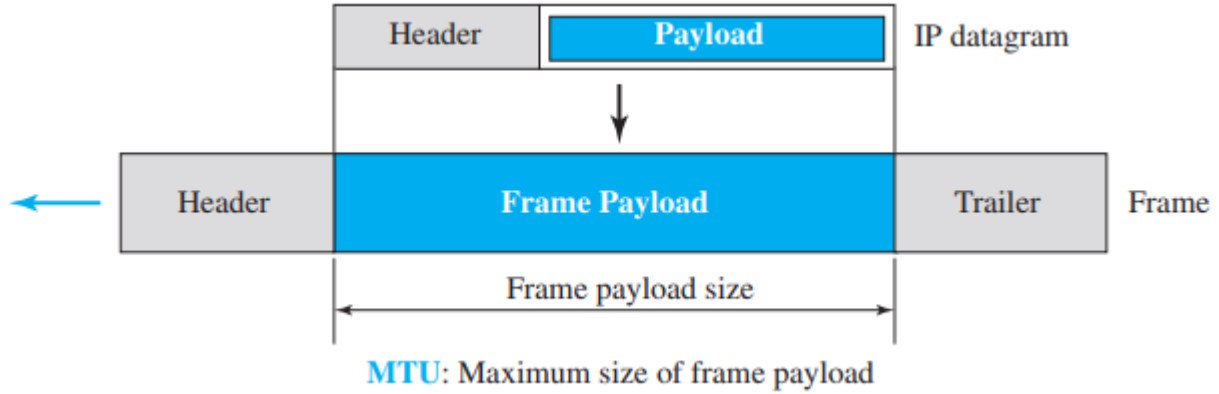


Fig 4. Maximum Transfer Unit

- **Fragmentation**
  - Payload of IP datagram is fragmented
  - When payload of IP datagram is fragmented, each fragment has its own header
  - Fragmented datagram can be further fragmented if it encounters a network with smaller MTU
  - Datagram may be fragmented several times before it reaches the destination

- **Reassembly**
  - Done at the Destination
  - Each fragment is an independent entity

Fragmentation Fields

- **Identification (16 bits)**
  - Provide uniqueness to each datagram
  - Positive number called counter is used
  - Fragmented datagram uses the same identification field
  - It helps the destination for reassembly of the datagram

- **Flags (3 bits)**
  - Leftmost bit is reserved
  - Second bit (D bit) is *do not fragment* bit (0: fragment, 1: don't fragment)
  - Third bit (M bit) is *more fragment* bit (0: no more fragments, 1: more fragments to come)

- **Fragmentation Offset (13 bits)**
  - Relative position of the fragment w.r.t the datagram
  - Measured in unit of 8 bytes

Fragmentation Example

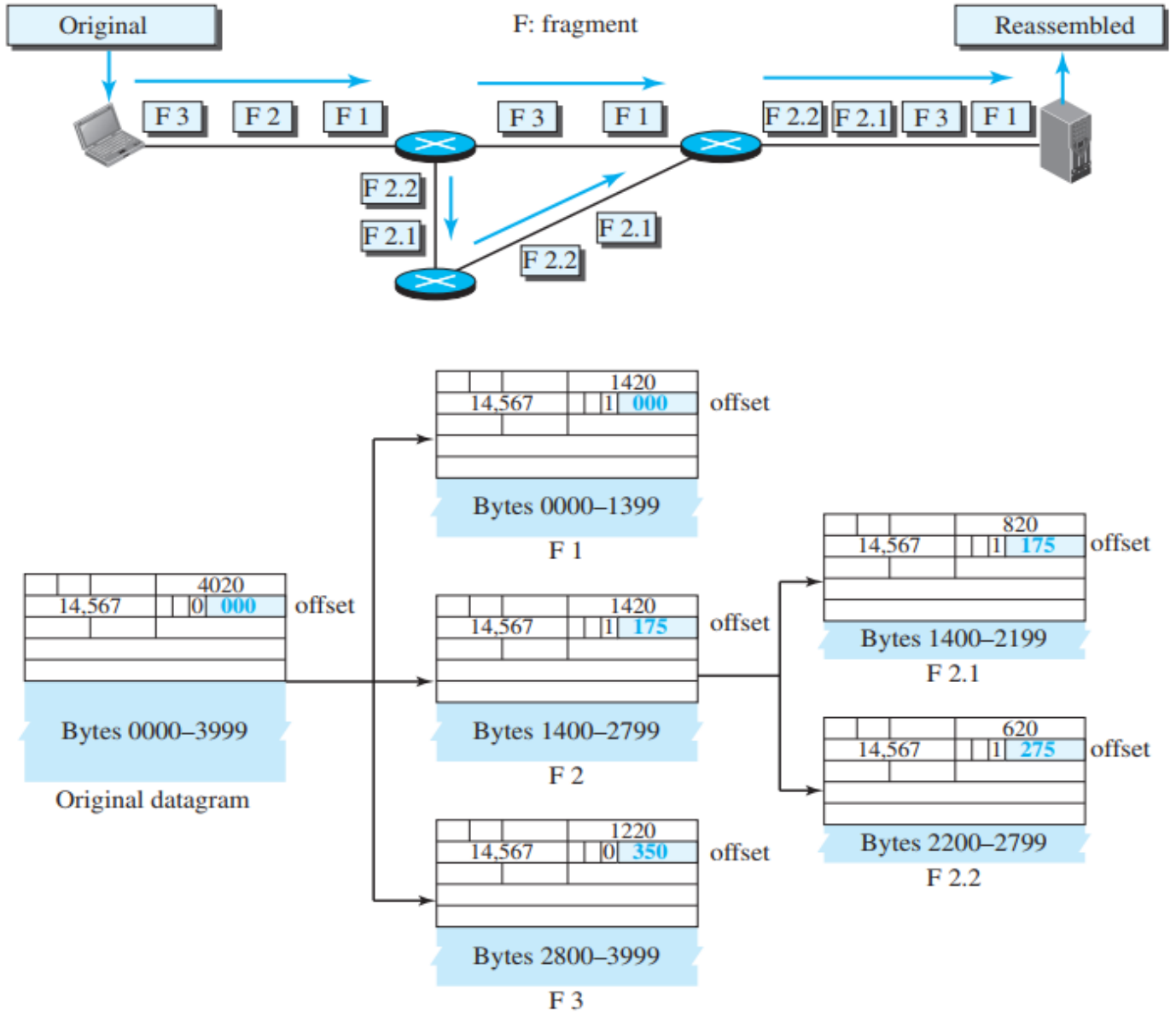


Fig 5. Example

IPv6 Addressing

- **IPv6 address is 128 bits (16 bytes) long**
- **Representation**

Binary (128 bits)	**1111111011110110**	**...**	**1111111100000000**
Colon Hexadecimal	**FEF6:BA98:7654:3210:ADEF:BBFF:2922:FF00**		

Fig 6(a). IPv6 Representation

FDEC:0:0:0:0:BBFF:0:FFFF ⟶ **FDEC::BBFF:0:FFFF**

Fig 6(b). IPv6 Representation

- **Transition from IPv4 to IPv6?**

Transition from IPv4 to IPv6

- **Dual Stack**
  - Source queries the DNS
  - DNS reply with IPv4/IPv6 address

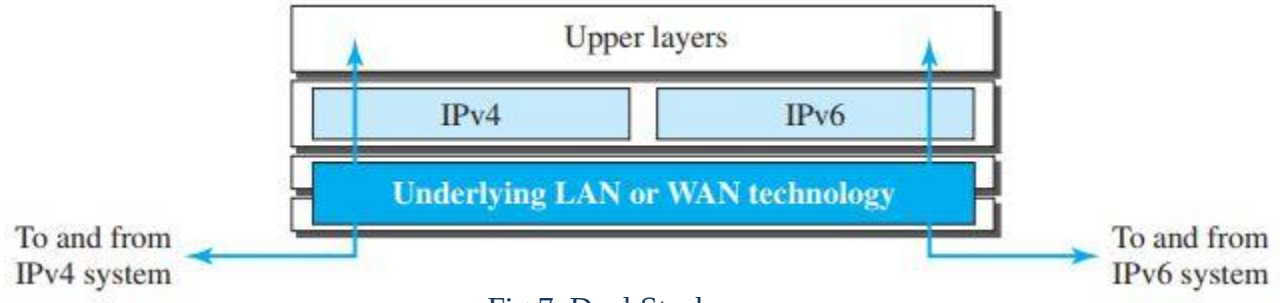


Fig 7. Dual Stack

- **Tunneling**
  - Source and Destination uses IPv6
  - Packet pass through a route which uses IPv4
  - IPv6 address encapsulated in IPv4 address
  - IPv4 packet carries IPv6 packet as data
  - Protocol value as 41

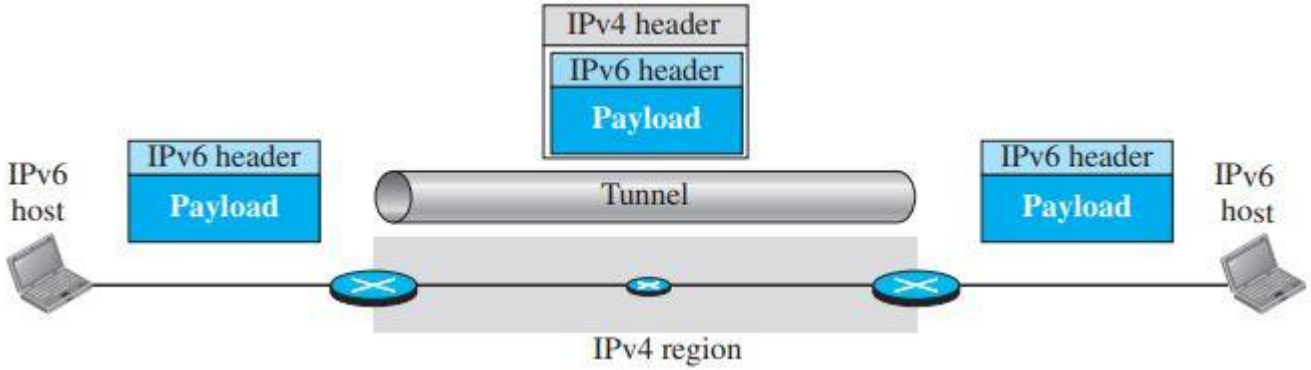


Fig 8. Tunneling

Transition from IPv4 to IPv6

- **Header Translation**
  - Source uses IPv6, but destination doesn't
  - Header of IPv6 packet gets converted to IPv4

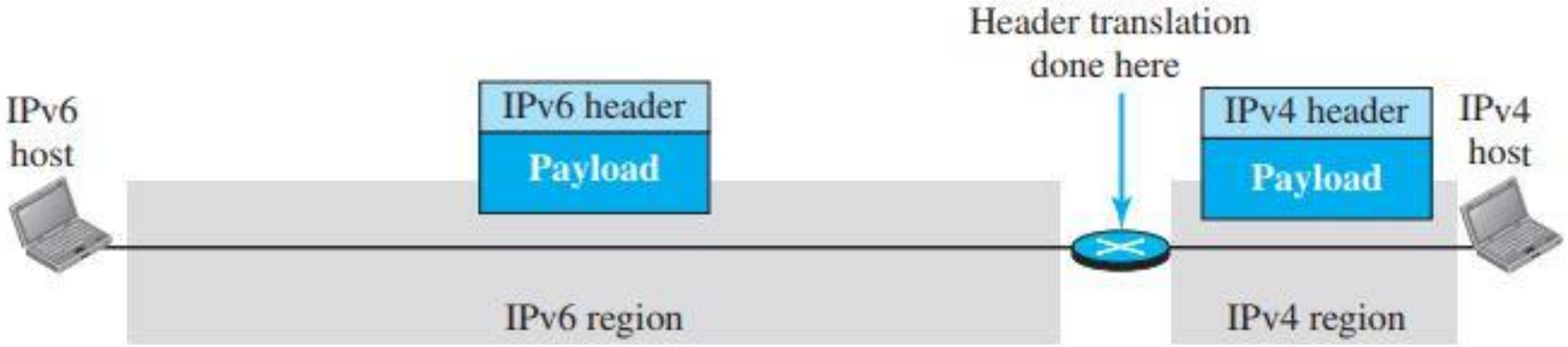


Fig 9. Header Translation