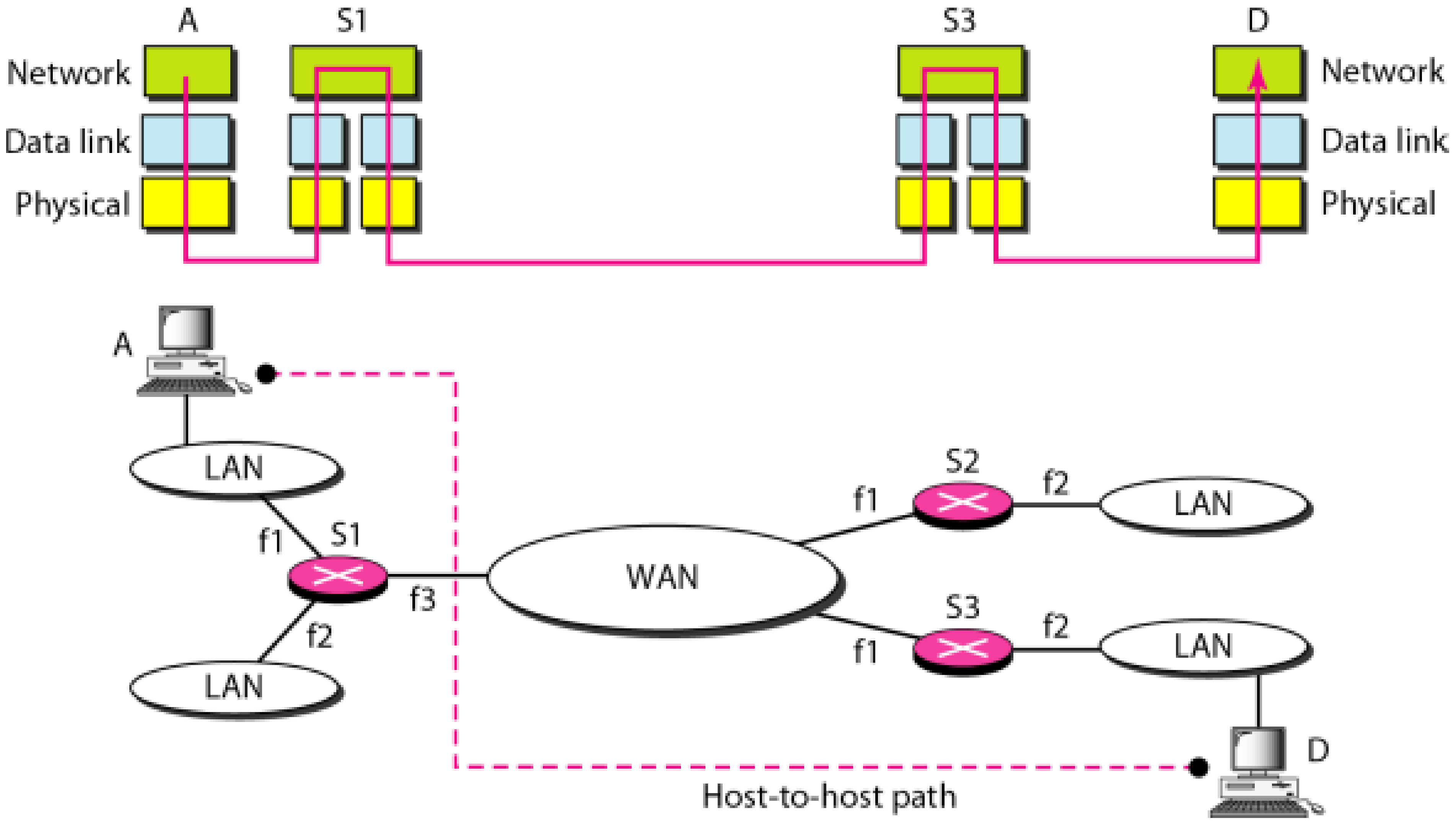
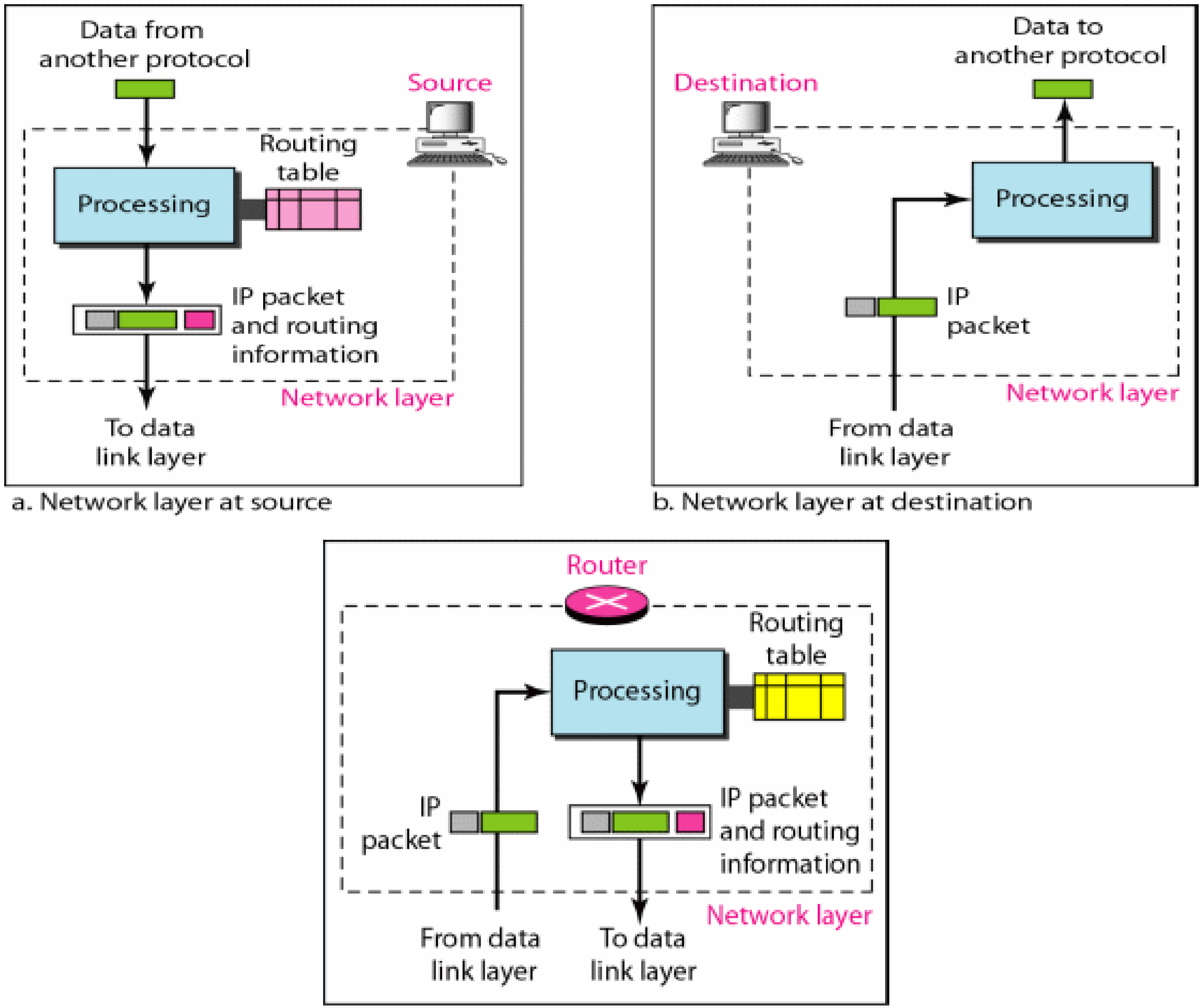
Network Layer:
Internet Protocol
And Helper Protocols

Internetworking



Internetworking (2)



a. Network layer at source

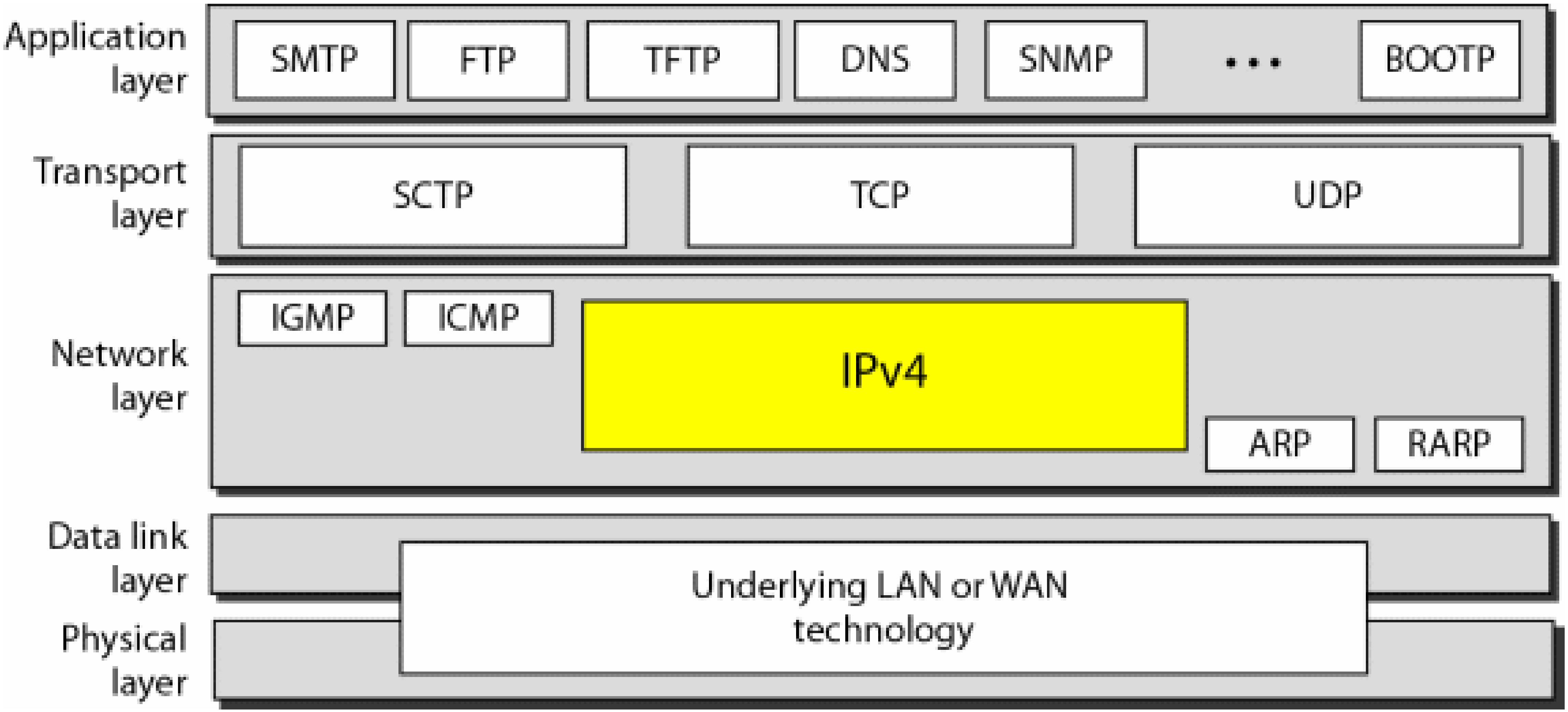
b. Network layer at destination

c. Network layer at a router

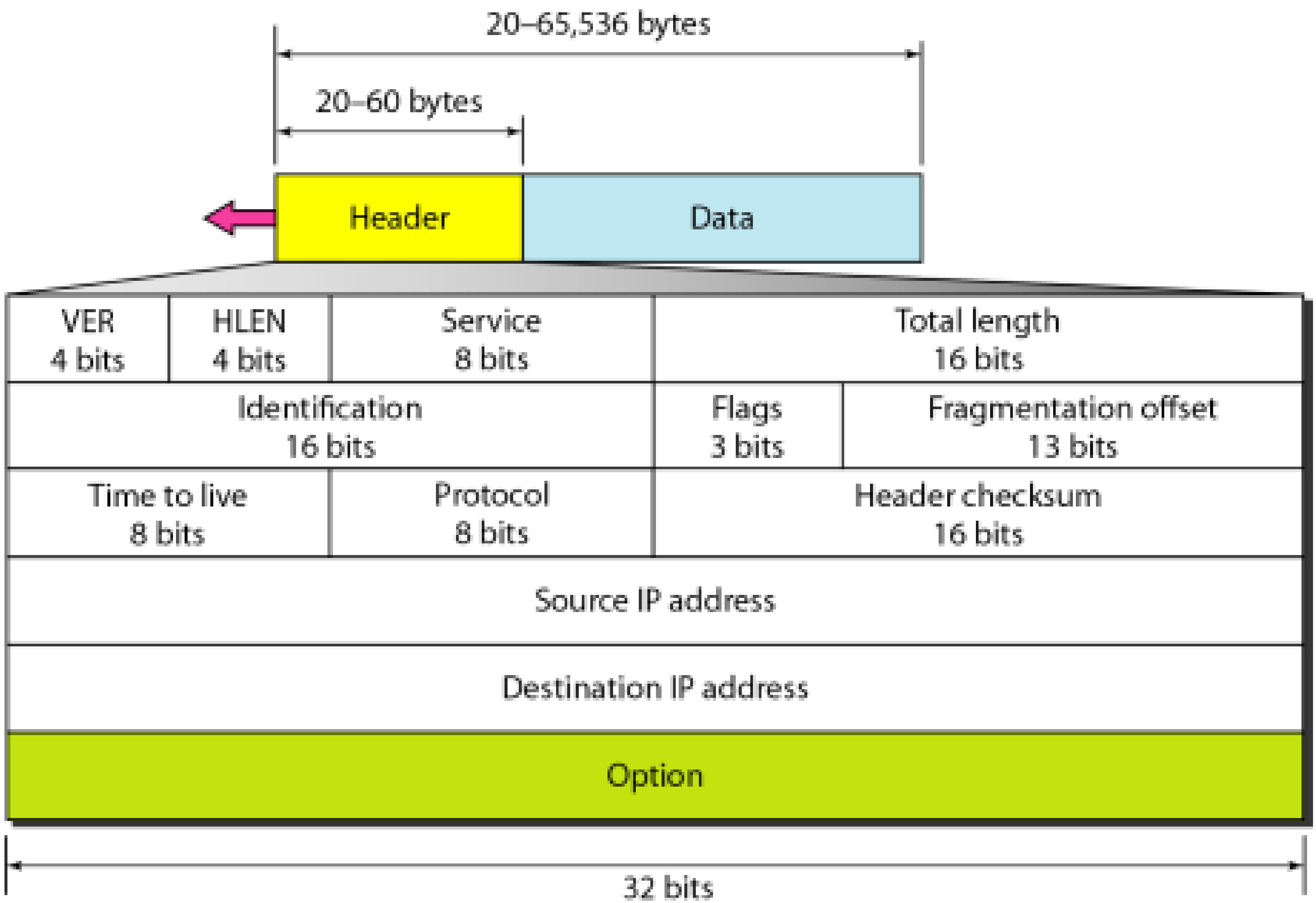
The Internet at the Network Layer

- Uses the datagram approach to packet switching
  - Uses the universal address defined in the network layer to route packets from the source to the destination
- Communication is connectionless
  - Treats each packet independently

IPv4 at the TCP/IP Protocol Suite

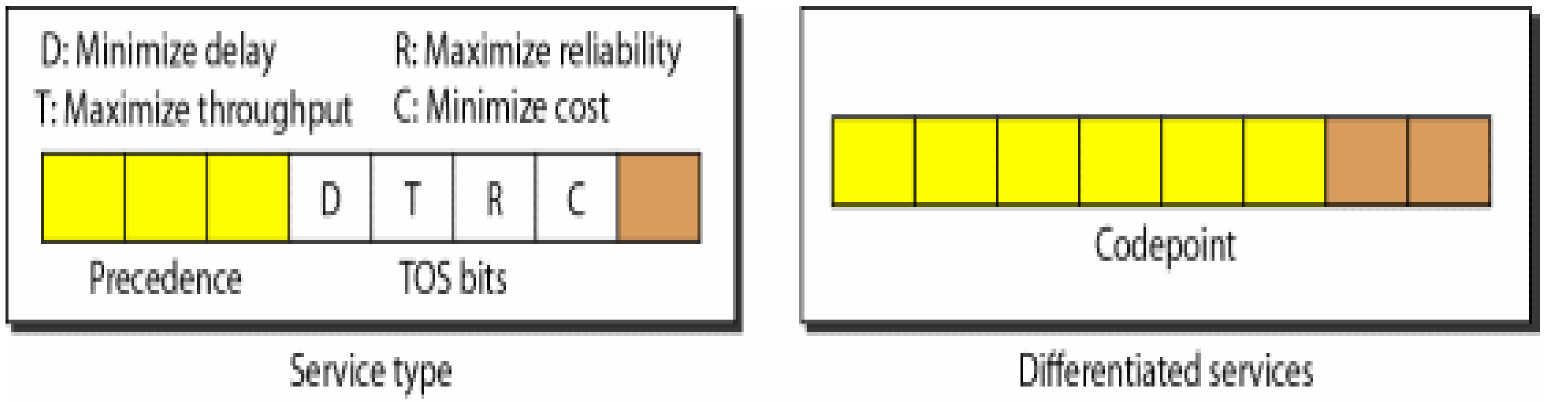


IPv4 Datagram Format



IPv4

- **Version** (VER) – 4-bit field that defines the version of the IPv4 protocol
- **Header length** (HLEN) – 4-bit field that defines the total length of the datagram header in 4-byte words
- **Service** – previously called service type, now called differentiated service



Service Type

- **Precedence** - 3-bit subfield that defines the priority of datagram in issues such as congestion
- **TOS** (Type of Service) bits - 4-bit subfield with each bit having a special meaning

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

Service Type of Some Applications

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Differentiated Service

- First 6 bits make up the codepoint subfield, last 2 bits are not used
  - When the 3 rightmost bits are 0s, the 3 leftmost bits are interpreted the same as the precedence bits
  - When the 3 rightmost bits are not all 0s, the 6 bit define 64 services based on the priority assignment by the Internet or local authorities

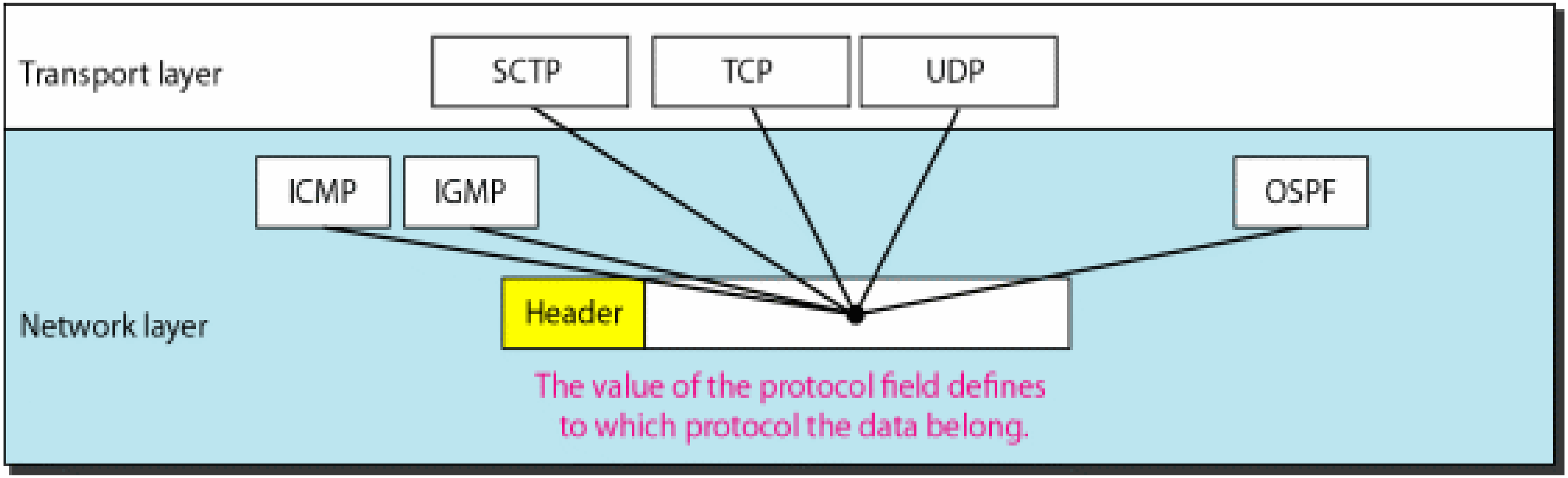
Category	*Codepoint*	*Assigning Authority*
1	XXXXX0	Internet
2	XXXX11	Local
3	XXXX01	Temporary or experimental

IPv4 (2)

- **Total length** – 16-bit field that defines the total length of the IPv4 datagram in bytes
- **Identification** – used in fragmentation
- **Flags** – used in fragmentation
- **Fragmentation offset** – used in fragmentation
- **Time to live** – used to control the maximum number of hops (routers) visited by the datagram

IPv4 (3)

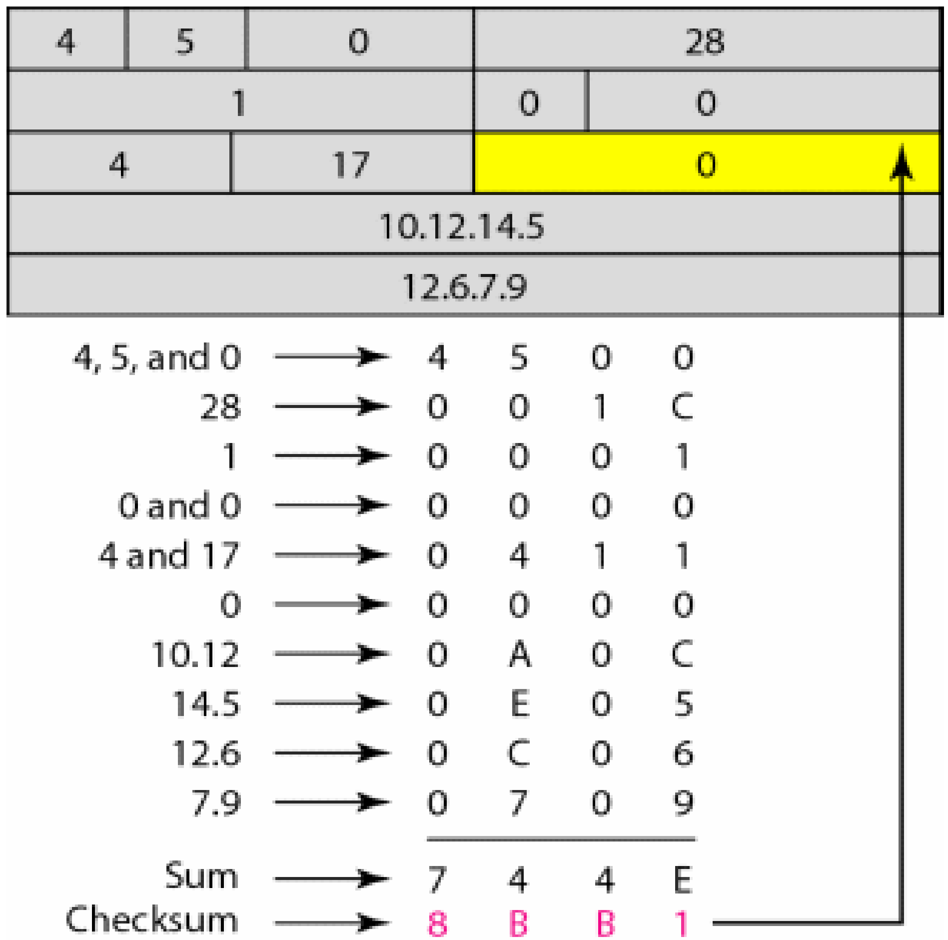
- **Protocol** – 8-bit field that defines the higher-level protocol that uses the services of the IPv4 layer



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

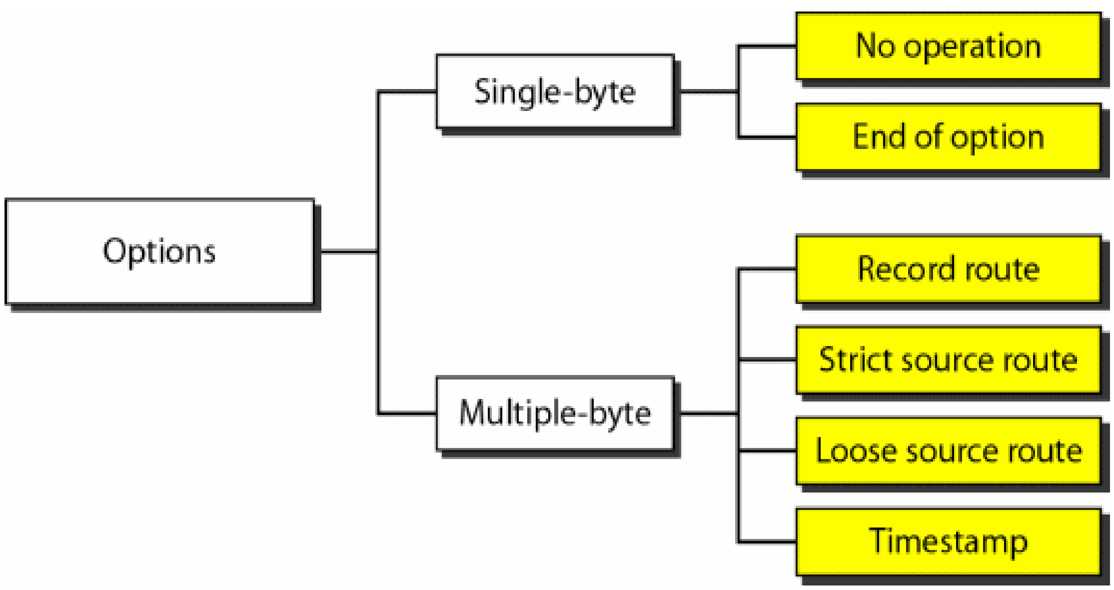
IPv4 (4)

- **Checksum** – for error checking, covers only the header



IPv4 (5)

- **Source Address** – 32-bit address that defines the IPv4 address of the source
- **Destination Address** – 32-bit address that defines the IPv4 address of the destination
- **Options** – maximum of 40 bytes, used for network testing and debugging



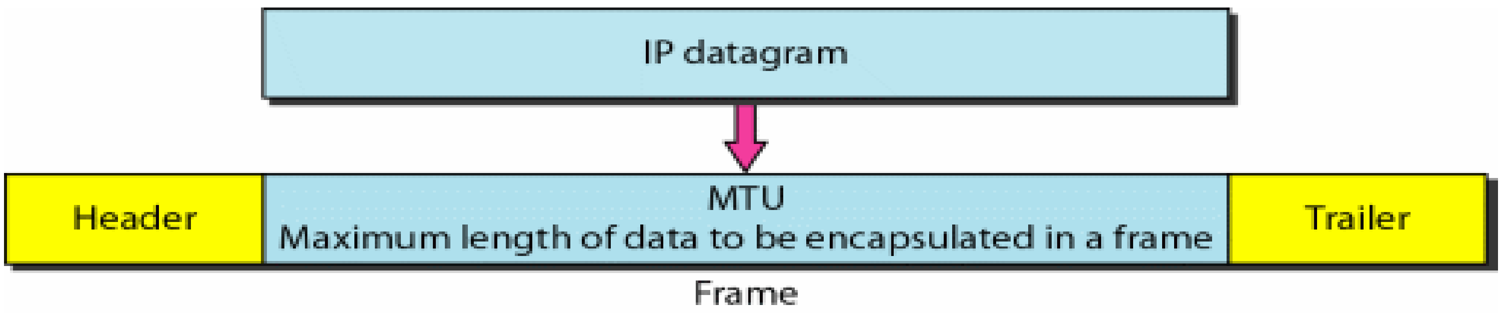
Options

- **No operation** – 1-byte option used as a filler between options
- **End of option** – 1-byte option used for padding at the end of the option field
- **Record route** – used to record the Internet routers that handle the datagram (*up to 9 router addresses*)
- **Strict source route** – used by the source to predetermine a route from the datagram as it travels through the Internet
- **Loose source route** – similar to strict source route in which each router in the list must be visited, but the datagram can visit other routers as well
- **Timestamp** – used to record the time of datagram processing by a router

Fragmentation

- Each data link protocol has its own frame format in most protocol, where one of the fields defined in the format is the maximum size of the data field, the **Maximum Transfer Unit** or **Maximum Transmission Unit** (MTU)
- When a datagram is encapsulated in a frame, the total size of the frame must be less than the MTU, which is defined by the restrictions imposed by the hardware and software used

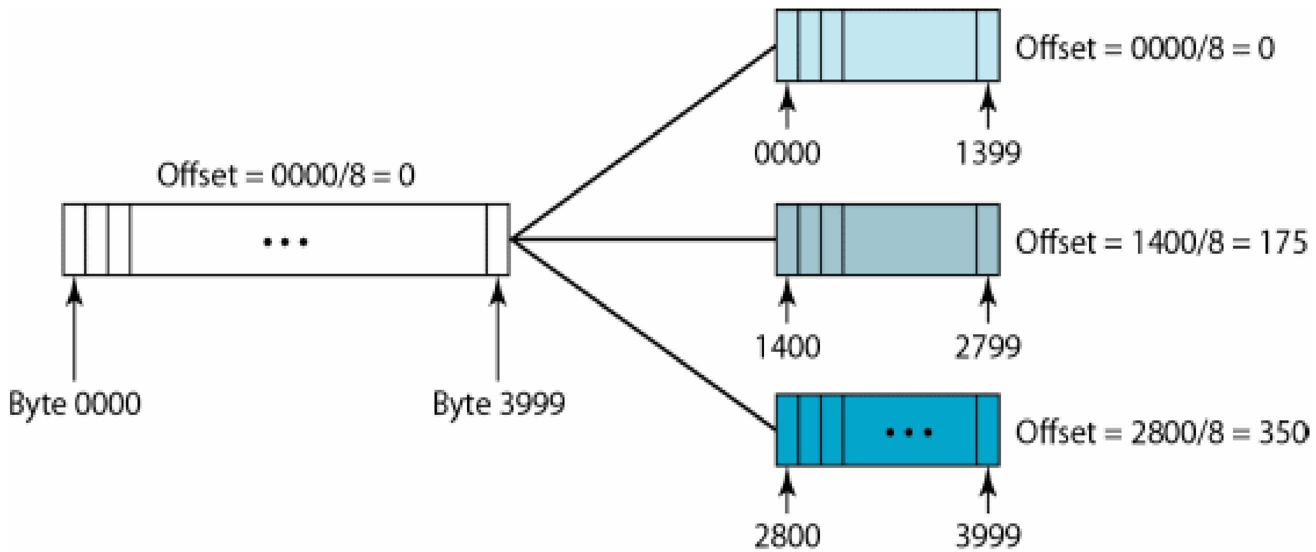
Fragmentation (2)



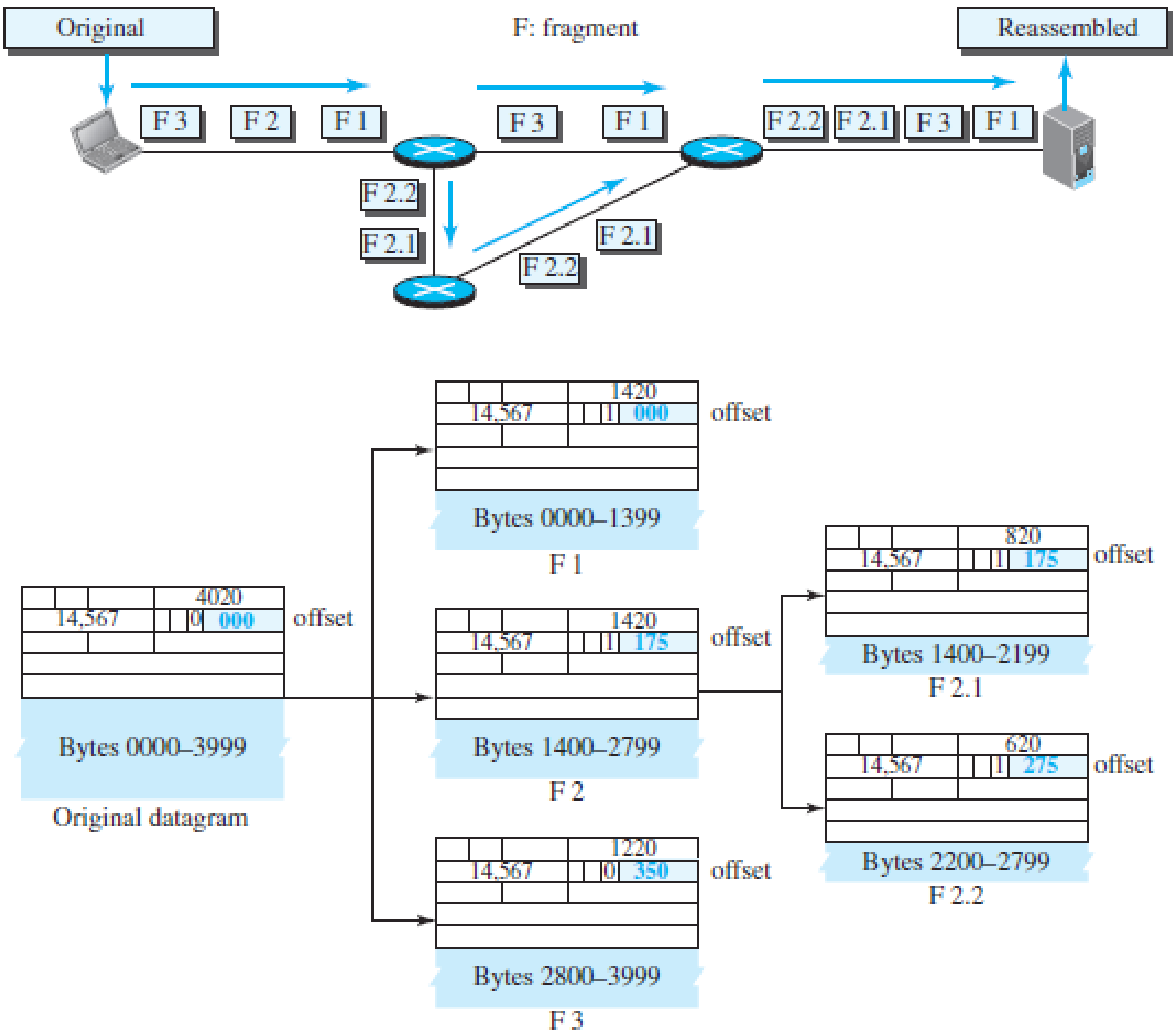
Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

Fragmentation (3)

- **Identification** – 16-bit field that identifies the datagram originating from the source host
- **Flags** – 3-bit field; 1st bit is reserved, 2nd bit is called *do not fragment* bit, 3rd bit is *more fragment* bit
- **Fragmentation offset** – 13-bit field that shows the relative position of the fragment with respect to the whole datagram



Fragmentation (4)



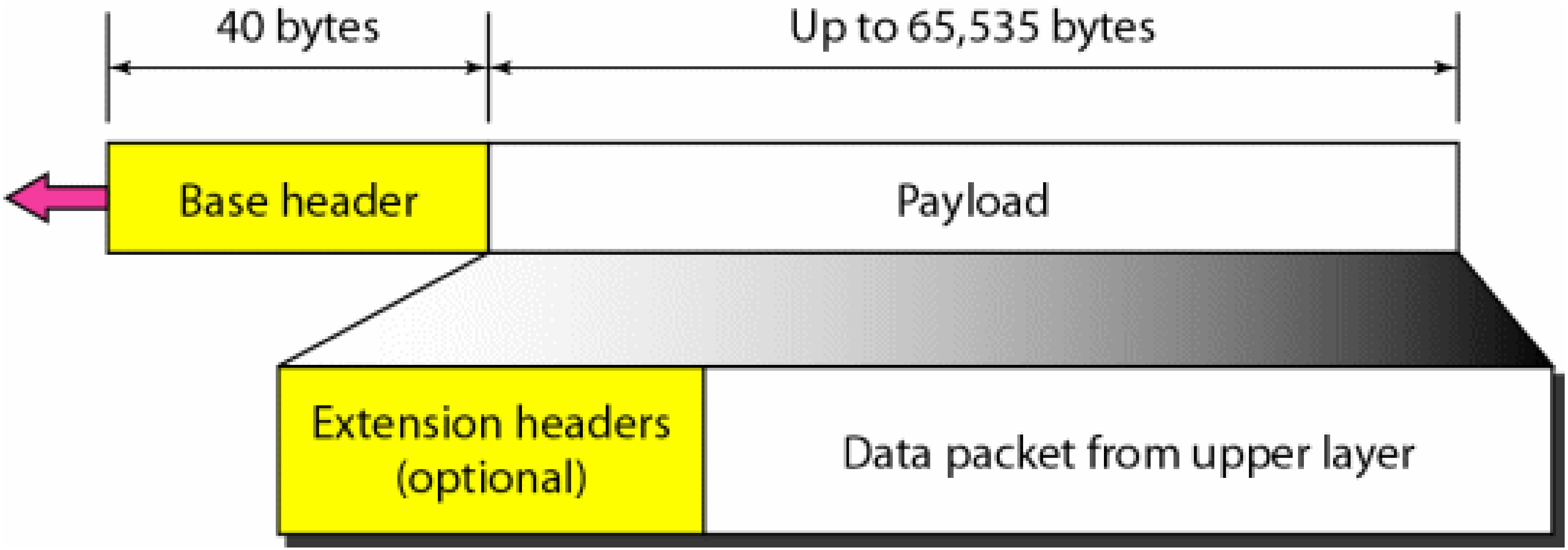
IPv6

- Internetworking Protocol, version 6, also known as **IPng** (Internetworking Protocol, next generation)
- To overcome deficiencies of IPv4
  - Address depletion is a long-term problem
  - Internet must accommodate real-time audio and video transmission
  - Internet must accommodate encryption and authentication of data

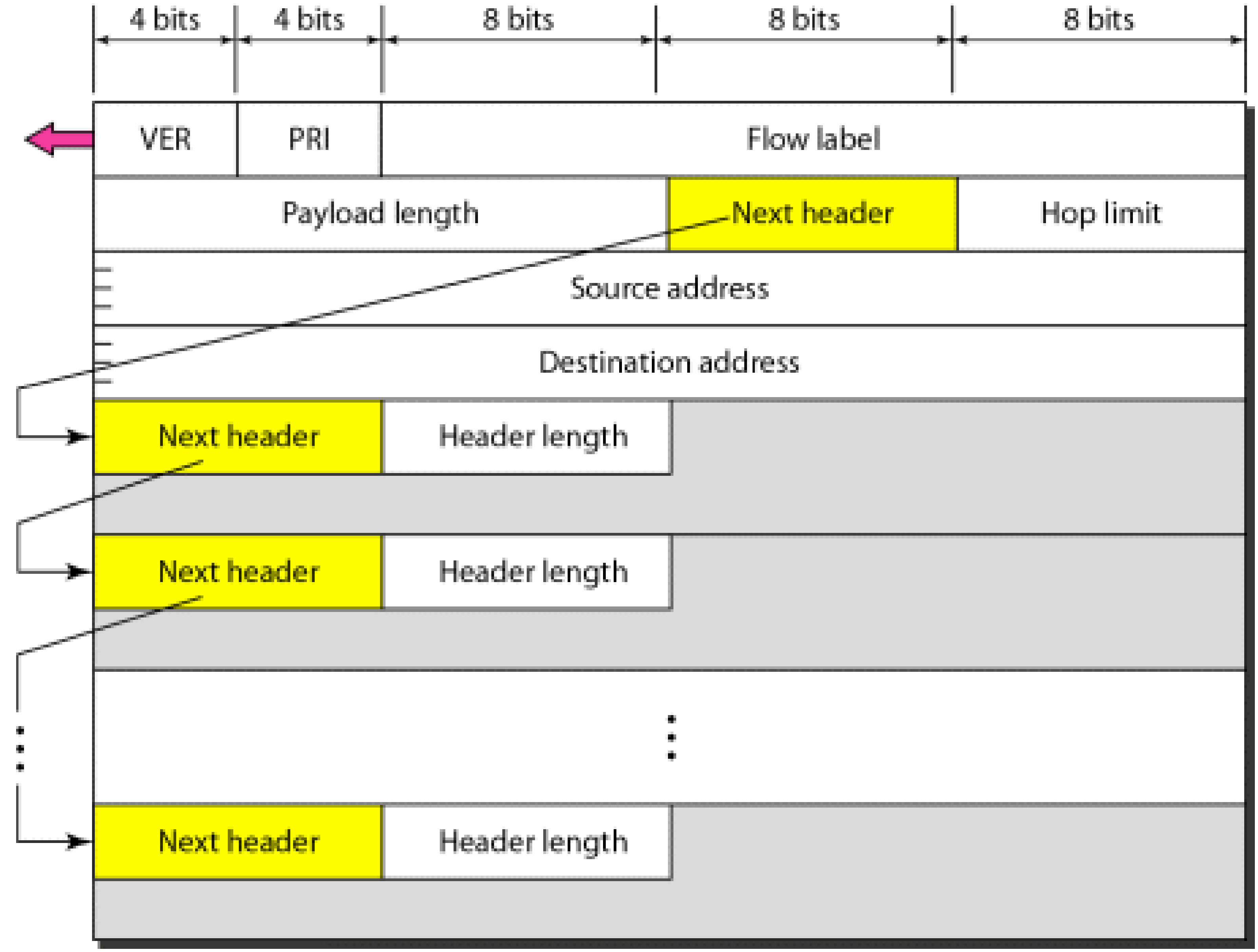
IPv6 (2)

- Advantages
  - Larger address space
  - Better header format
  - New options
  - Allowance for extension
  - Support for resource allocation
  - Support for more security

IPv6 Header and Payload



IPv6 Format



4 bits
4 bits
8 bits
8 bits
8 bits
VER
PRI
Flow label
Payload length
Next header
Hop limit
Source address
Destination address
Next header
Header length
Next header
Header length
Next header
Header length

IPv6 Base Header

- **Version** – 4-bit field that defines the version number of the IP
- **Priority** – 4-bit field that defines the priority of the packet with respect to traffic congestion
- **Flow label** – 3-byte field that is designed to provide special handling for a particular flow of data
  - A flow is a sequence of packets that share the same characteristics
- **Payload length** – 2-byte field that defines the length of the IP datagram excluding the base header

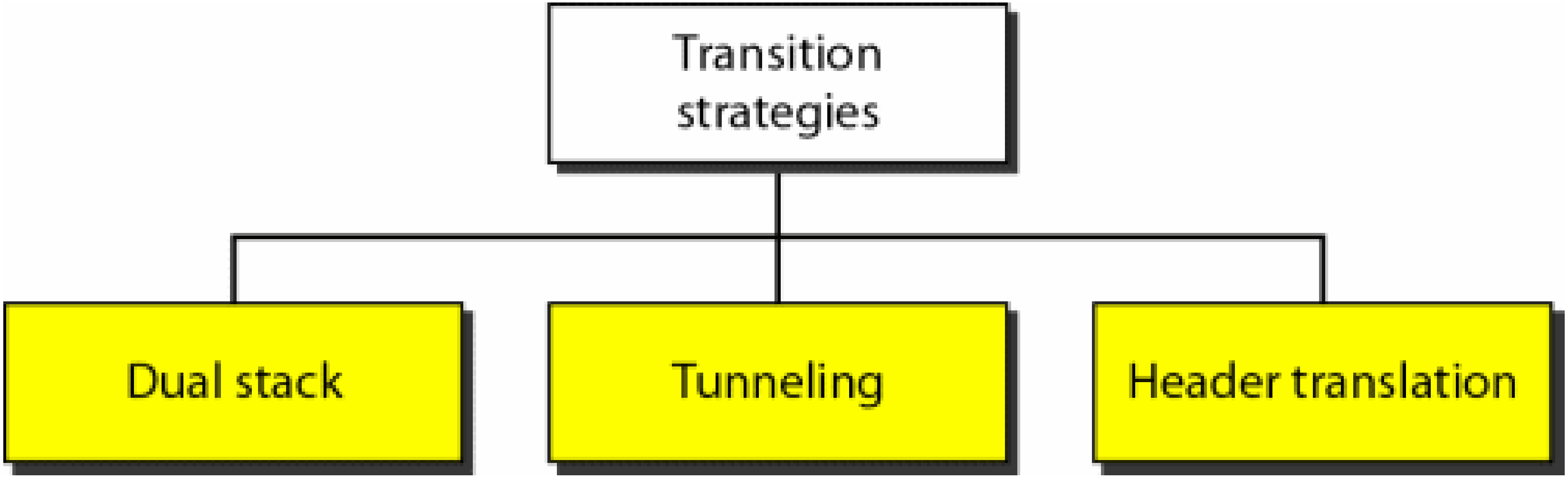
IPv6 Base Header (2)

- **Next header** – 8-bit field defining the type of the first extension (if present) or the type of data that follows the base header
- **Hop limit** – 8-bit field that serves the same purpose of TTL field in IPv4
- **Source address** – 16-byte IP address that identifies the original source of the datagram
- **Destination address** – 16-byte IP address that identifies the original source of the datagram

IPv4 vs IPv6 Header

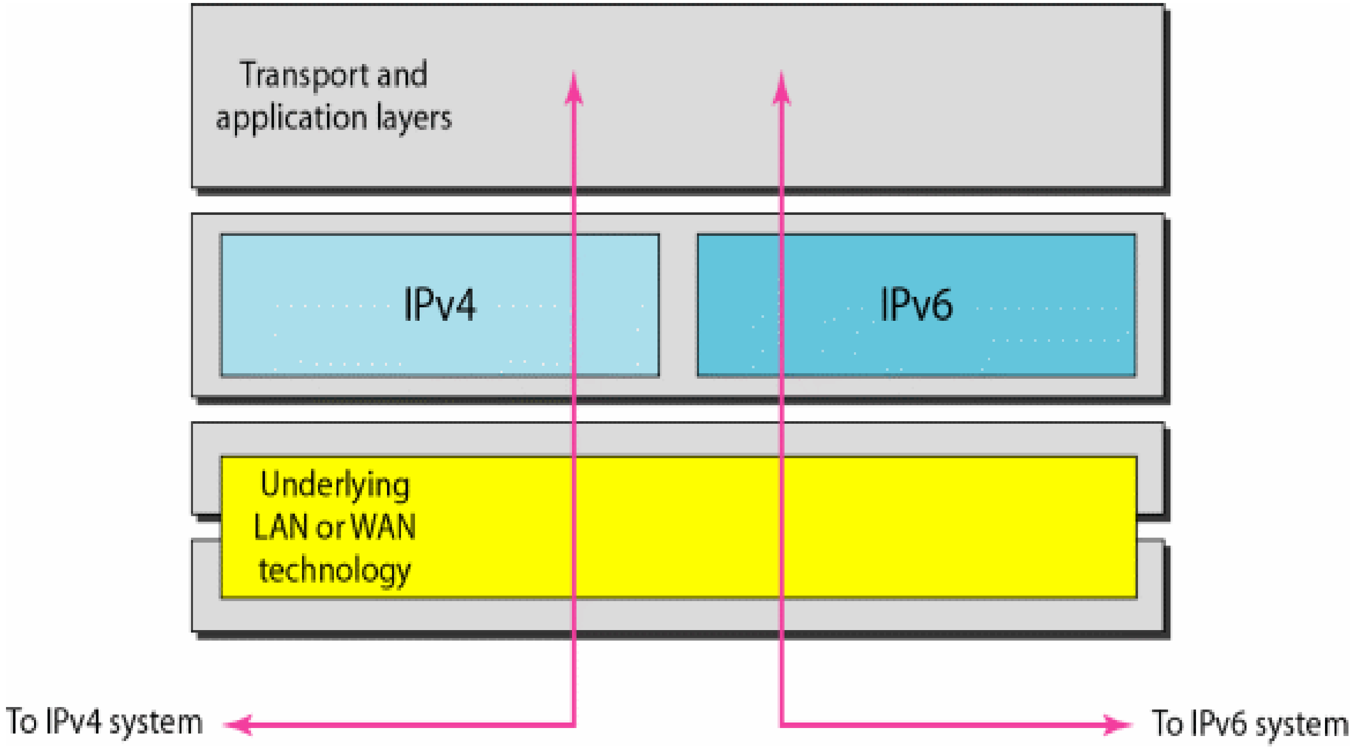
Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. 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.
7. 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.

Transition from IPv4 to IPv6

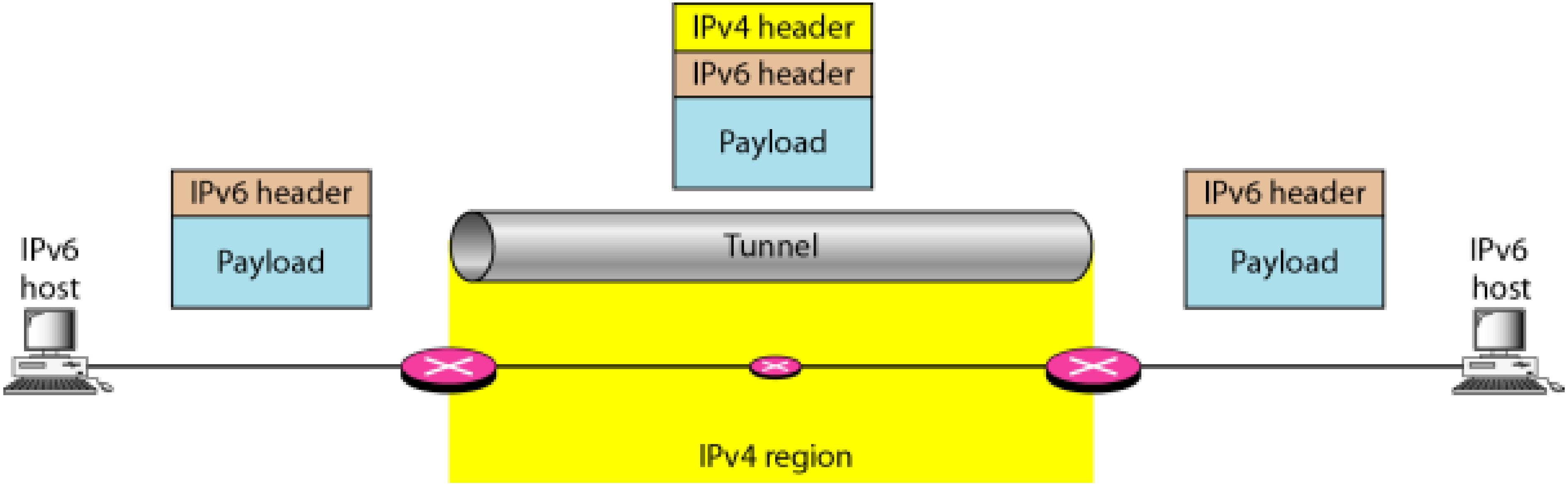


Dual Stack

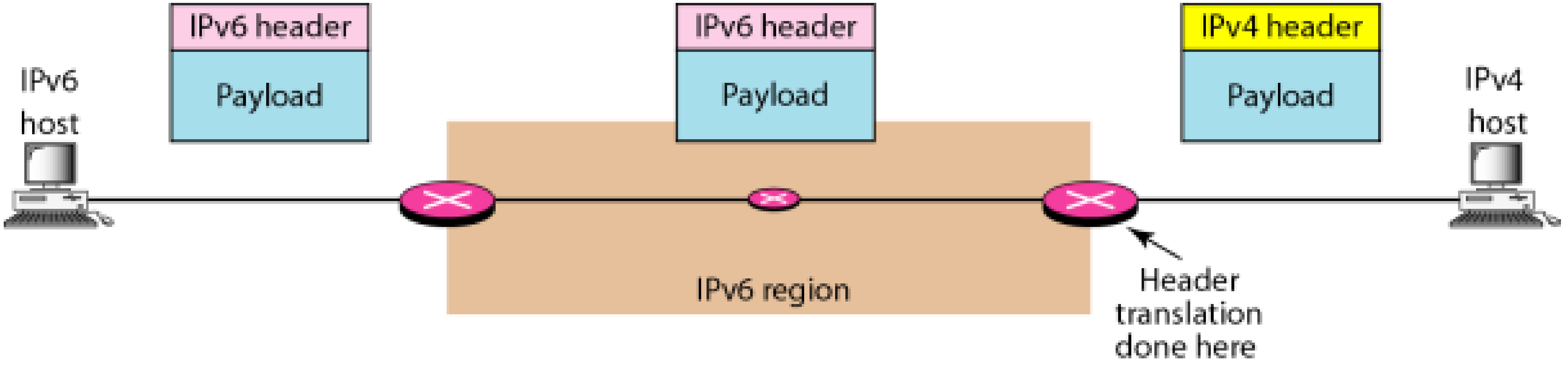
- Station must run IPv4 and IPv6 simultaneously



Tunneling Strategy



Header Translation



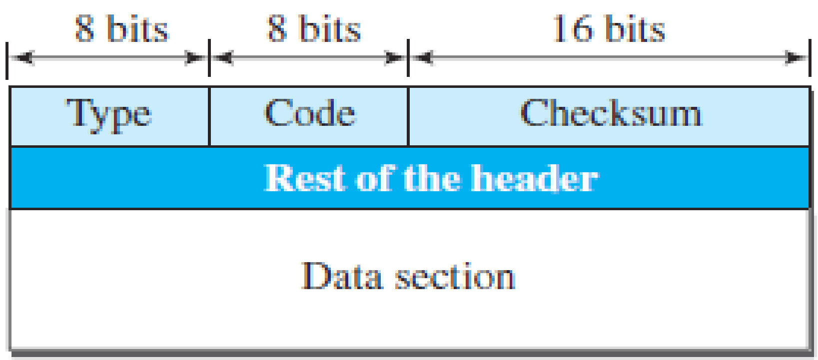
Header Translation (2)

Header Translation Procedure
1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.

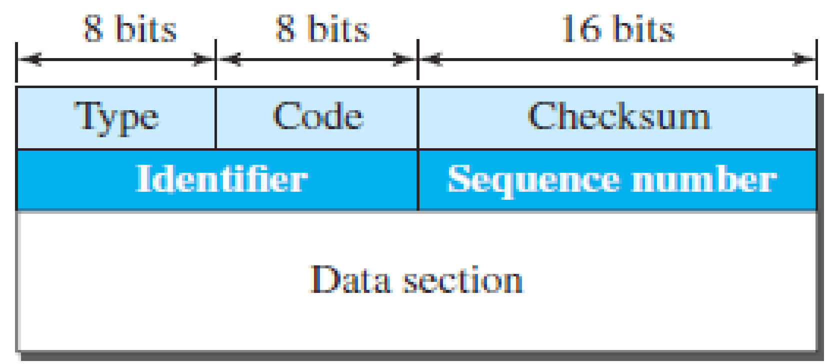
ICMPv4

- IPv4 has deficiencies
  - No error-reporting or error correcting mechanism
  - Lacks mechanism for host and management queries
- **Internet Control Message Protocol version 4** (ICMPv4) compensate for the deficiencies, companion protocol to Ipv4
  - Error-reporting messages – report problems that a router or a host may encounter when it process an IP packet
  - Query messages – helps a host get specific information from a router or another host

ICMPv4 Format



Error-reporting messages



Query messages

Type and code values

Error-reporting messages
03: Destination unreachable (codes 0 to 15)
04: Source quench (only code 0)
05: Redirection (codes 0 to 3)
11: Time exceeded (codes 0 and 1)
12: Parameter problem (codes 0 and 1)

Query messages
08 and 00: Echo request and reply (only code 0)
13 and 14: Timestamp request and reply (only code 0)

Debugging Tools

- ping
  - To find if a host is alive and responding
  - Can calculate round-trip time

```
D:\>ping www.google.com -t

Pinging www.google.com [172.217.24.36] with 32 bytes of data:
Reply from 172.217.24.36: bytes=32 time=20ms TTL=53
Reply from 172.217.24.36: bytes=32 time=21ms TTL=53
Reply from 172.217.24.36: bytes=32 time=21ms TTL=53
Reply from 172.217.24.36: bytes=32 time=21ms TTL=53
Reply from 172.217.24.36: bytes=32 time=21ms TTL=53
Reply from 172.217.24.36: bytes=32 time=20ms TTL=53
Reply from 172.217.24.36: bytes=32 time=21ms TTL=53

Ping statistics for 172.217.24.36:
    Packets: Sent = 7, Received = 7, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 20ms, Maximum = 21ms, Average = 20ms
```

Debugging Tools (2)

- traceroute (UNIX) / tracert (Windows)
  - Used to trace path of a packet from a source to the destination
  - Can find all the IP addresses of all routers that are visited along the path (usually set to 30 hops max)

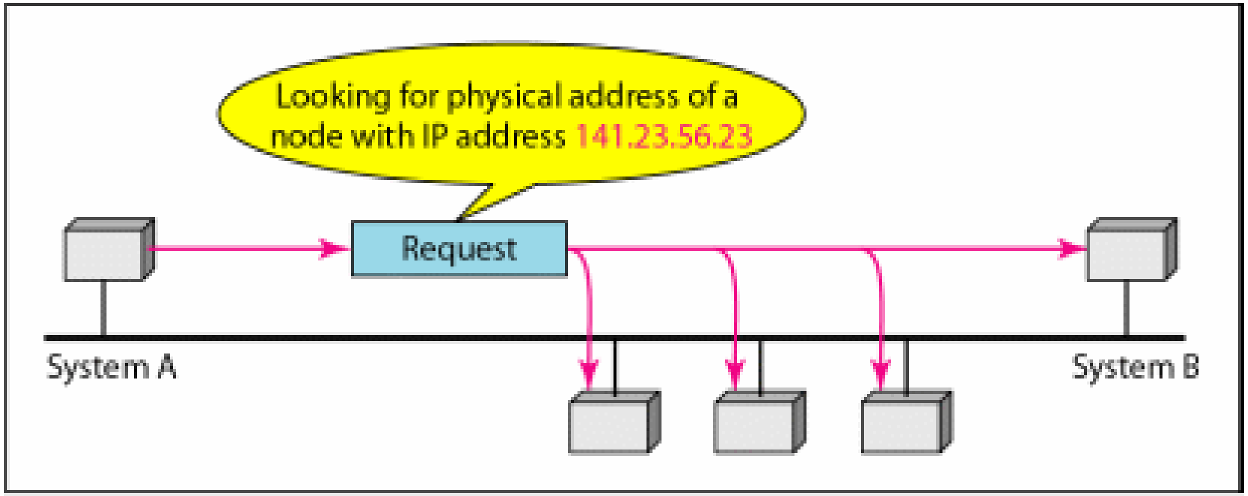
```
D:\>tracert www.google.com

Tracing route to www.google.com [172.217.24.36]
over a maximum of 30 hops:

  1    <1 ms     1 ms     1 ms  10.0.100.254
  2     1 ms     1 ms     1 ms  10.255.255.254
  3     1 ms     1 ms     1 ms  202.92.144.254
  4     1 ms     1 ms     1 ms  ge-0.626-802.1q-vlan-subif.core-7304-irri.pregi.net [202.90.129.237]
  5     6 ms     5 ms     5 ms  ge-1-3.border-asti.pregi.net [202.90.129.253]
  6     6 ms     6 ms     6 ms  tengige0-0-2-0.border-asti-asr.pregi.net [202.90.132.229]
  7    21 ms    20 ms    21 ms  tengige0-0-2-0.border-hk-asr.pregi.net [202.90.129.50]
  8    20 ms    21 ms    20 ms  72.14.212.20
  9    21 ms    21 ms    21 ms  108.170.241.1
 10    21 ms    20 ms    20 ms  108.170.238.131
 11    20 ms    20 ms    21 ms  hkg07s23-in-f36.1e100.net [172.217.24.36]

Trace complete.
```

Address Mapping

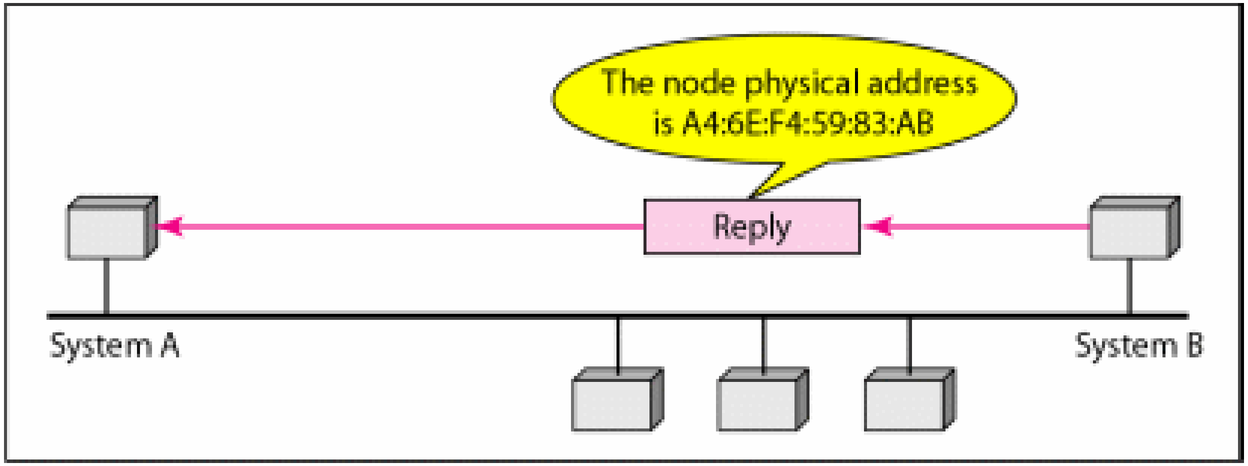
- Physical address and logical addresses are two different identifiers and are needed both:
  - A physical network such as Ethernet can have 2 different protocols at the network layer such as IP and IPX (Novell) at the same time
  - A packet at a network layer such as IP may pass through different physical networks such as Ethernet and LocalTalk (Apple)

Mapping Logical to Physical Address

- **Address Resolution Protocol** (ARP)

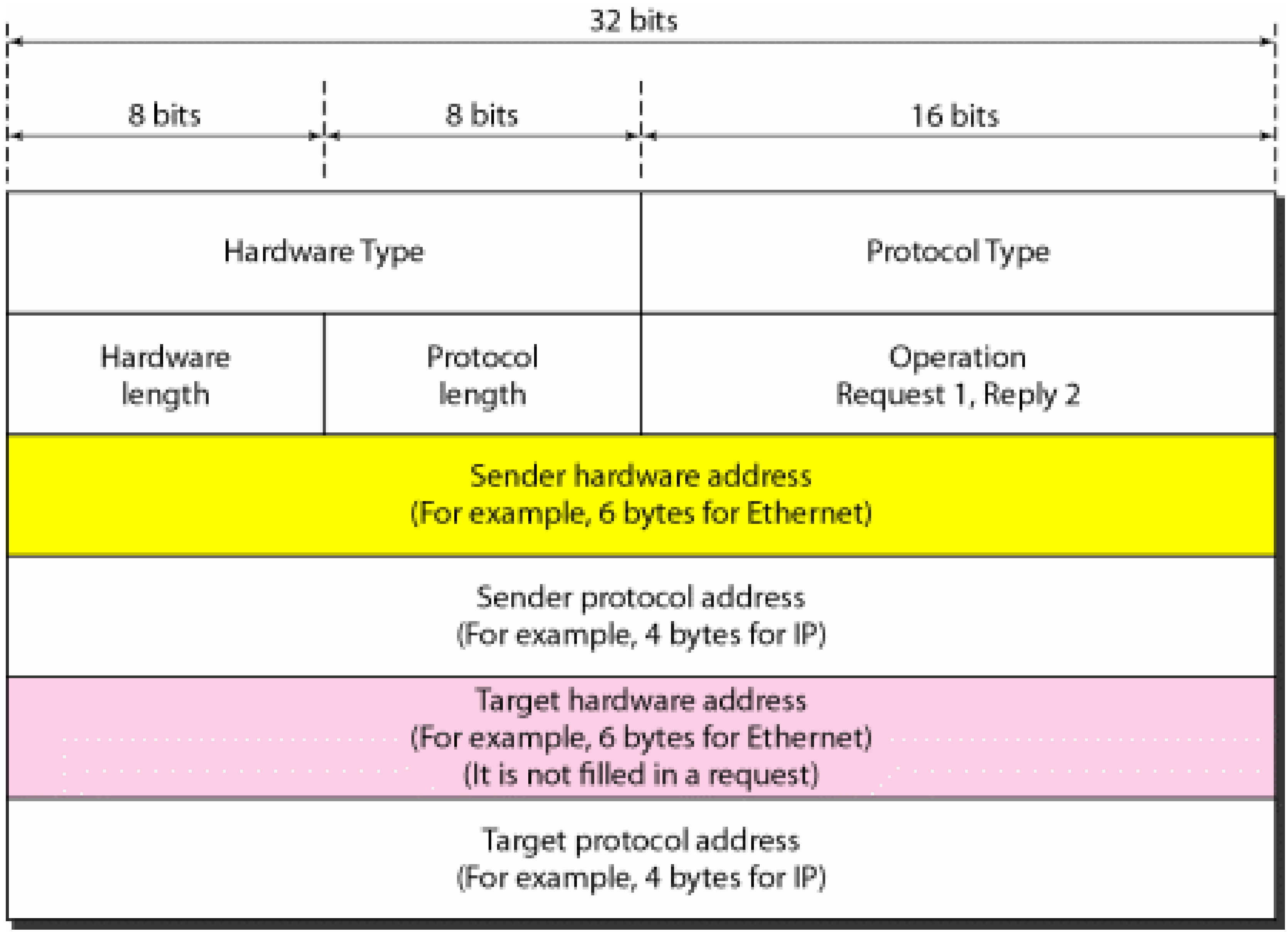


a. ARP request is broadcast

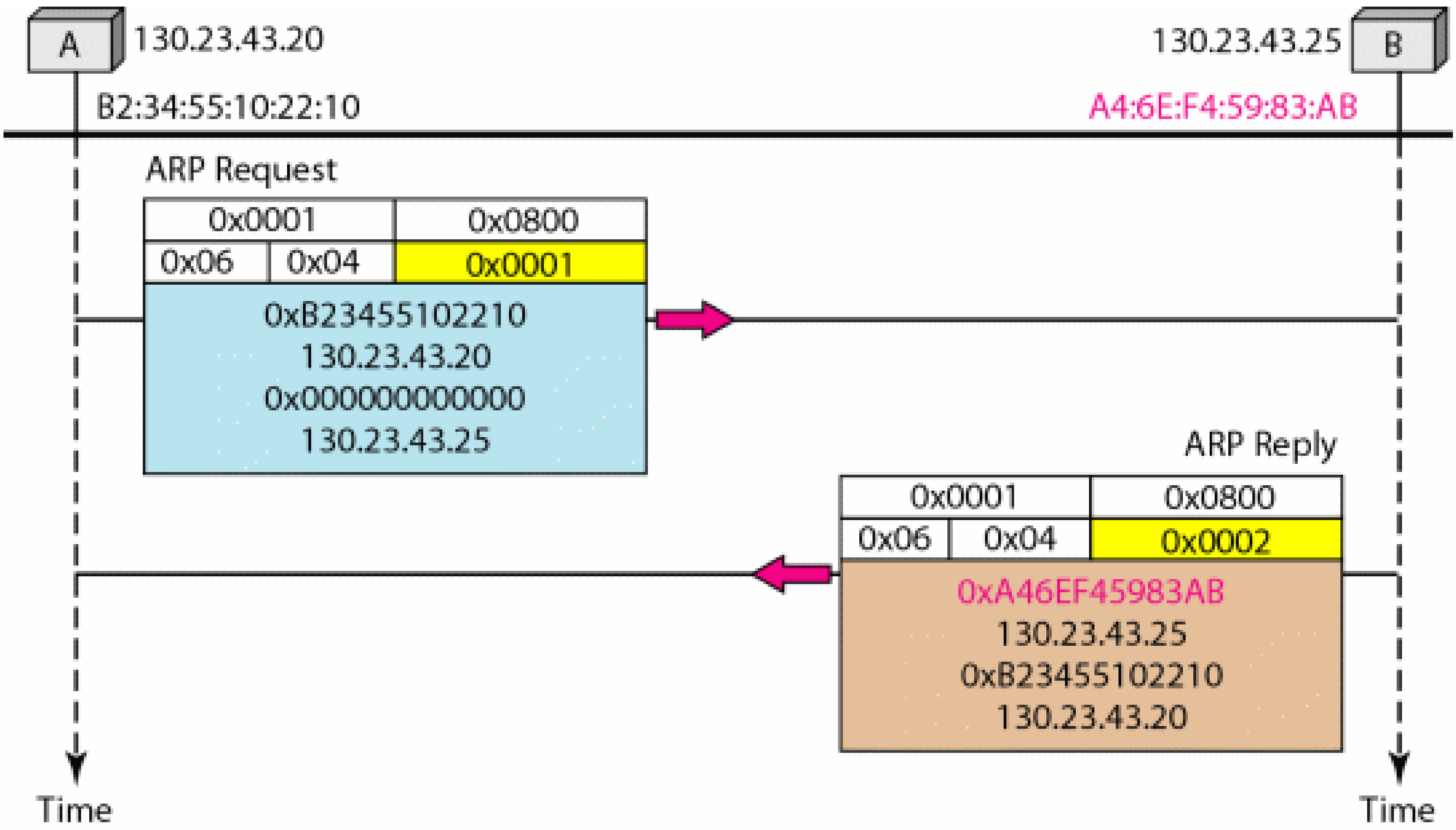


b. ARP reply is unicast

ARP Packet


32 bits
8 bits
8 bits
16 bits
Hardware Type
Protocol Type
Hardware length
Protocol length
Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)
Sender protocol address (For example, 4 bytes for IP)
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)
Target protocol address (For example, 4 bytes for IP)

ARP



A
130.23.43.20
B2:34:55:10:22:10
130.23.43.25
B
A4:6E:F4:59:83:AB
ARP Request
0x0001
0x0800
0x06
0x04
0x0001
0xB23455102210
130.23.43.20
0x000000000000
130.23.43.25
ARP Reply
0x0001
0x0800
0x06
0x04
0x0002
0xA46EF45983AB
130.23.43.25
0xB23455102210
130.23.43.20
Time
Time

Mapping Physical to Logical Address

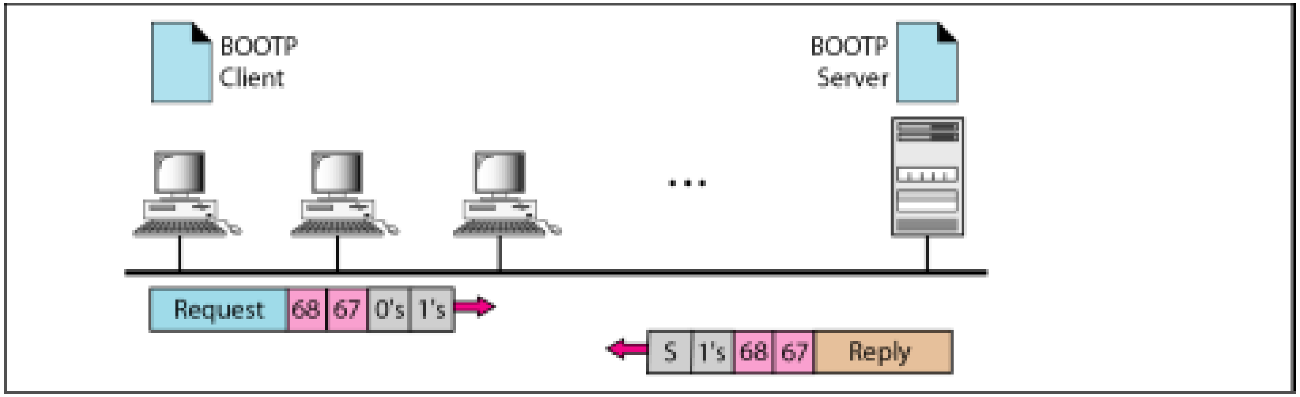
- Occasions in which host know its physical address but needs to know its logical address
  - A diskless station is booted
  - An organization does not have enough IP addresses to assign to each station
- RARP, BOOTP and DHCP

RARP

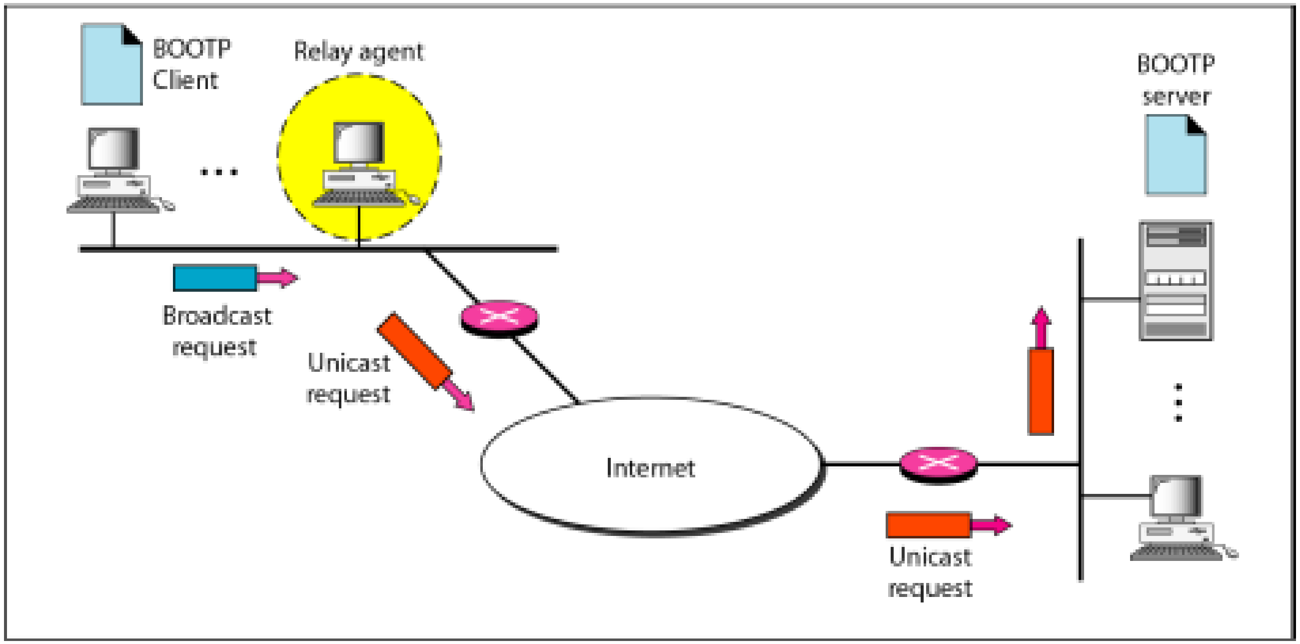
- **Reverse Address Resolution Protocol**
  - A RARP request is created and broadcast on the local network
  - Another machine on the local network that knows all the IP address will respond to the RARP reply
  - Problem: broadcasting is done at the data link layer, it does not pass the boundaries of a network

BOOTP

- **Bootstrap Protocol** – application layer protocol



a. Client and server on the same network

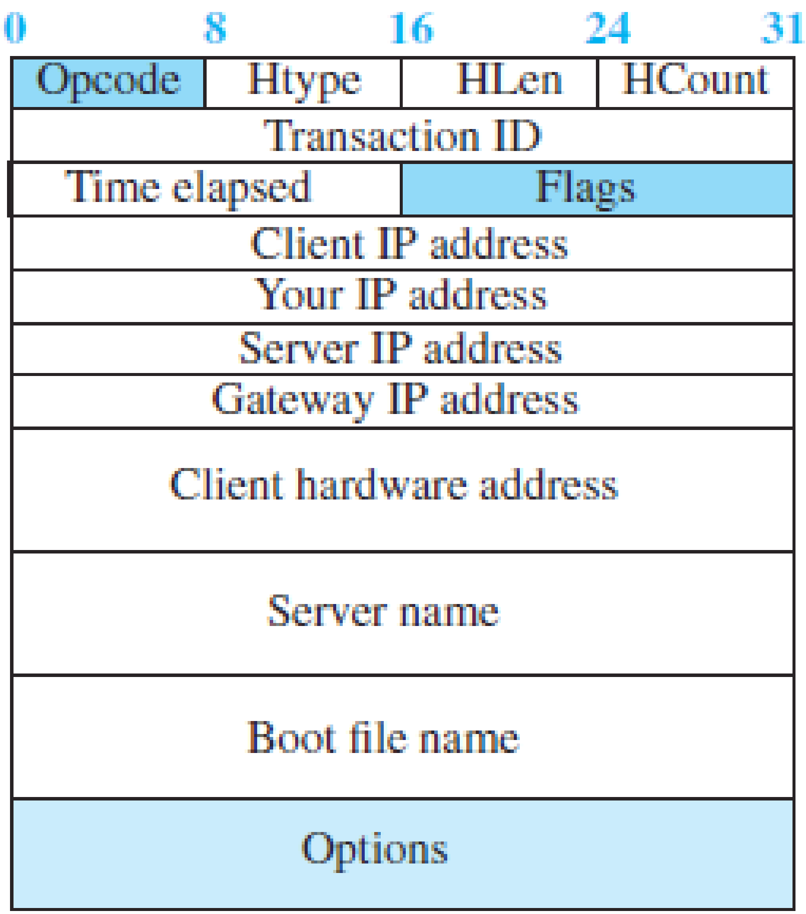


b. Client and server on different networks

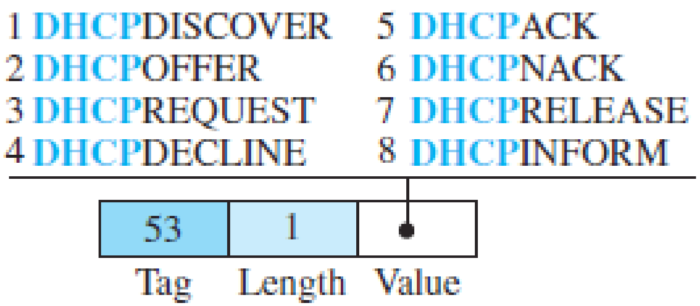
DHCP

- **Dynamic Host Configuration Protocol** – provides static and dynamic addresses allocation that can be manual or automatic
- Static address allocation – acts as what BOOTP does. DHCP server has a databases that statically binds physical addresses to IP addresses
- Dynamic address allocation – DHCP has a second database with pool of available IP addresses

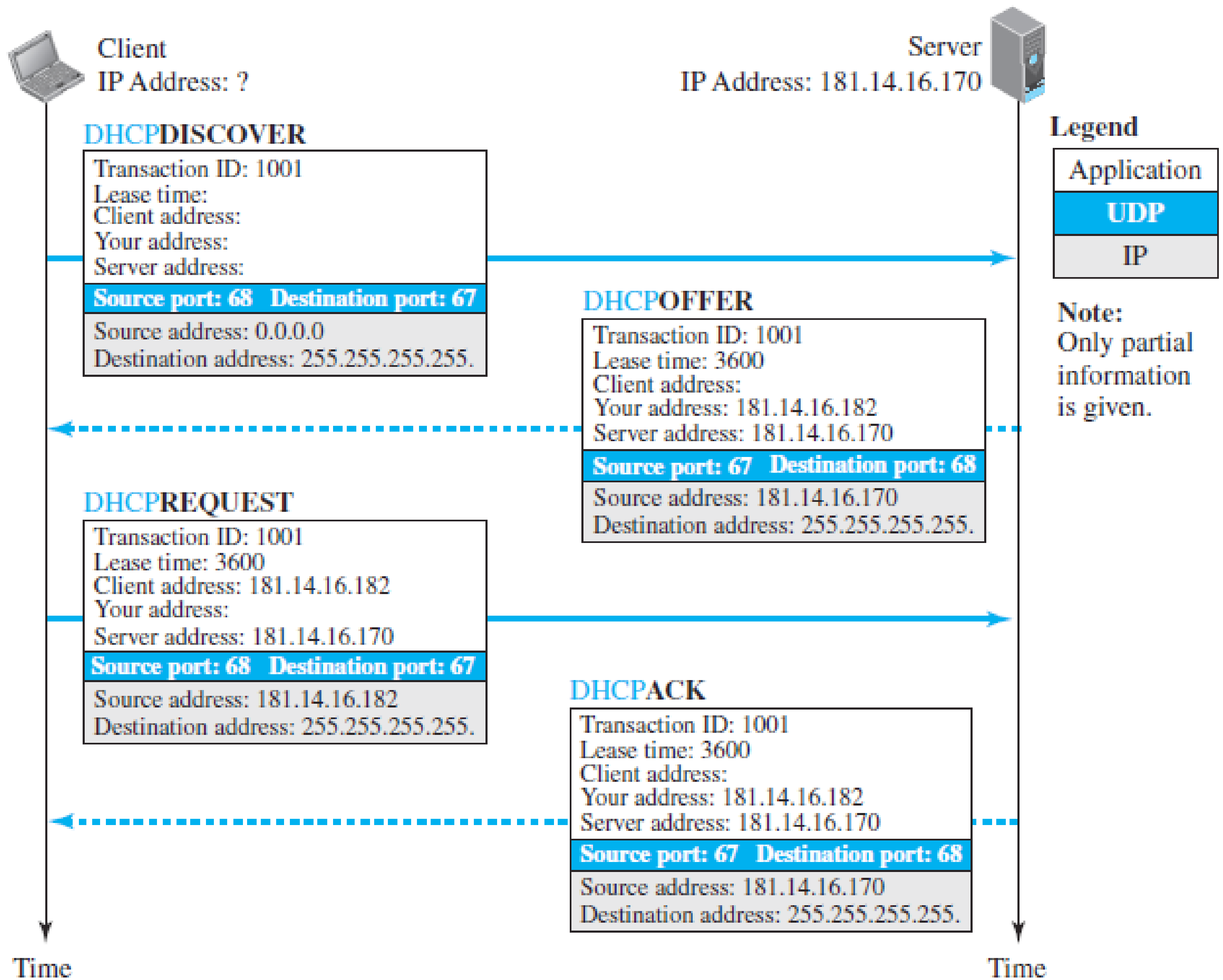
DHCP Message Format



Option format



DHCP Operation

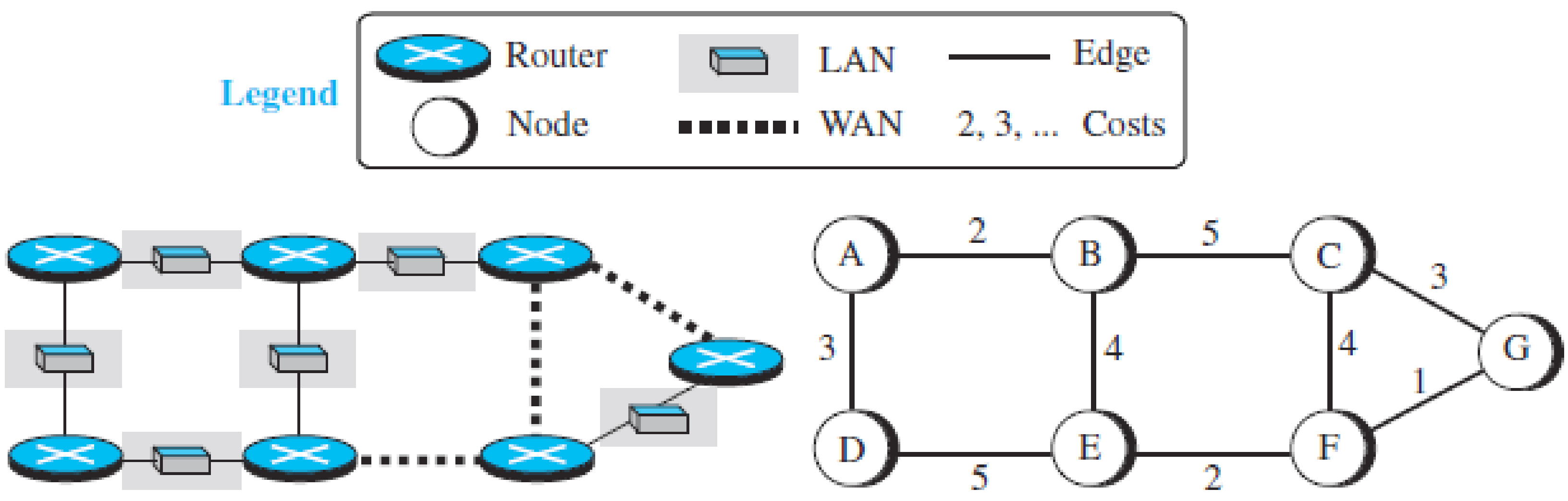


Network Layer Terms

- **Delivery** – the way a packet is handled by the underlying networks under the control of the network layer
- **Forwarding** – the way a packet is delivered to the next station
- **Routing** – the way routing tables are created to help in forwarding
  - Unicast Routing – if a datagram is destined for only one destination
  - Multicast Routing – if a datagram is destined for several destinations

Unicast Routing

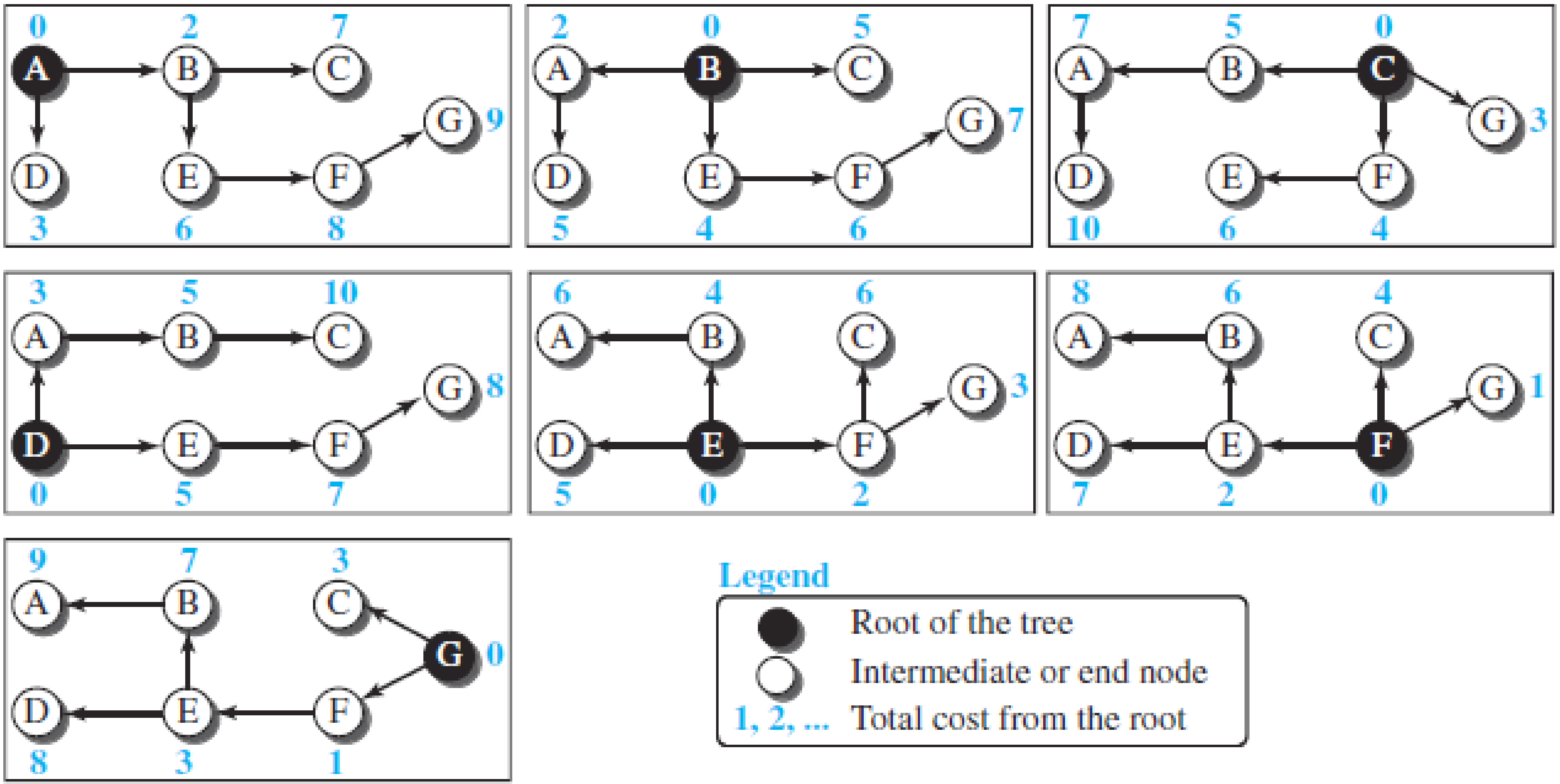
- A packet is routed, hop by hop, from its source to its destination by the help of forwarding tables
- To find the best route, an internet can be modeled as a graph



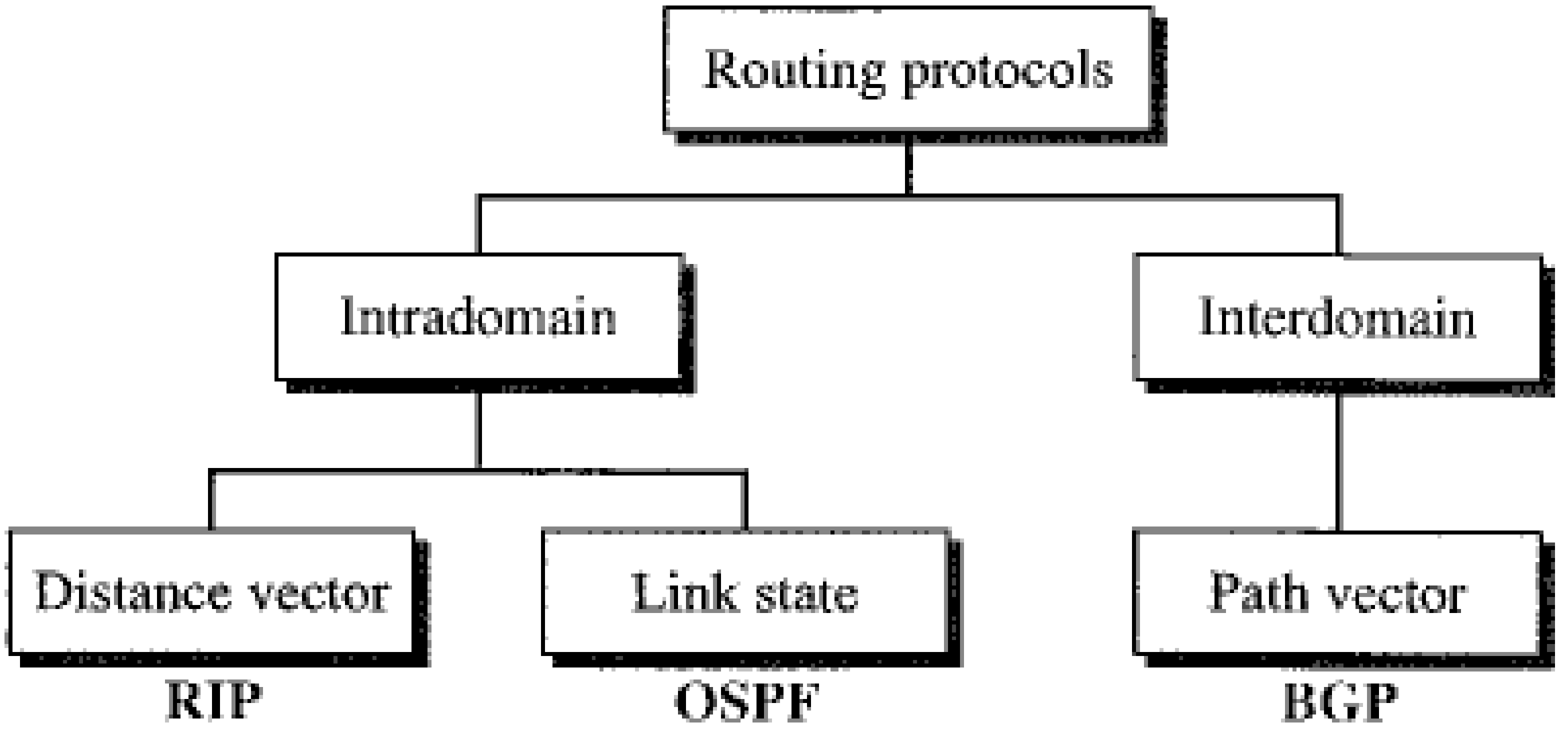
a. An internet

b. The weighted graph

Least-Cost Routing

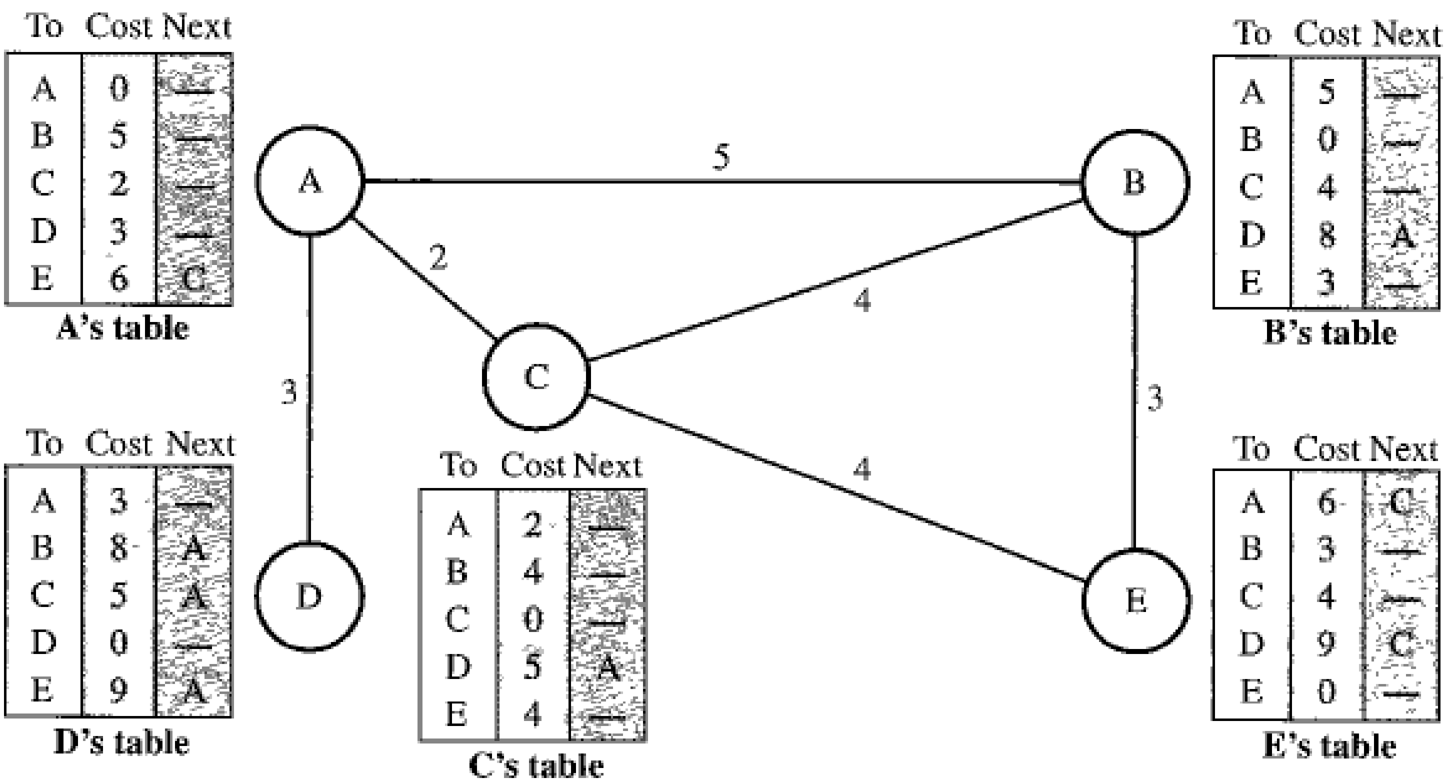


Routing Algorithms



Distance-Vector Routing

- Each node maintains a vector (table) of minimum distances to every node
- The table at each node also guides the packets to the desired node by showing the next stop in the route



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	8	A
E	3	—

B's table

To	Cost	Next
A	3	—
B	8	A
C	5	A
D	0	—
E	9	A

D's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	5	A
E	4	—

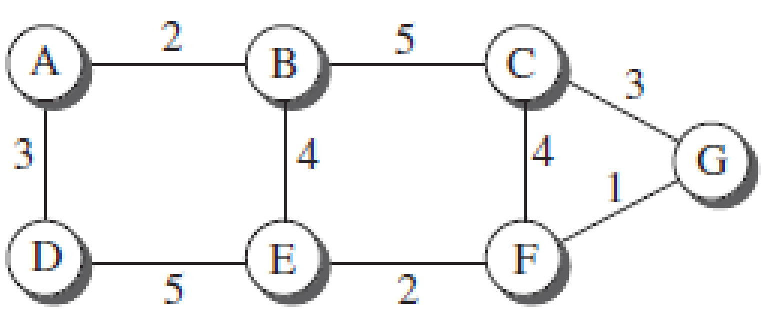
C's table

To	Cost	Next
A	6	C
B	3	—
C	4	—
D	9	C
E	0	—

E's table

Link-State Routing

- This method uses the term *link-state* to define the characteristic of a link (edge) that represents a network in the internet
- The collection of states for all links is called the link-state database (LSDB)
- To create a least-cost tree, each node needs a complete map of the network (using Dijkstra's Algorithm)



a. The weighted graph

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

b. Link state database

Path-Vector Routing

- The best path is determined by the source using the policy it imposes on the route, i.e. the source controls the path
- The path is also determined by the best spanning tree. It is not the least-cost tree; it is the tree determined by the source when it imposes its own policy