COMP 3721
Introduction to Data Communications

11a - Week 11 - Part 1

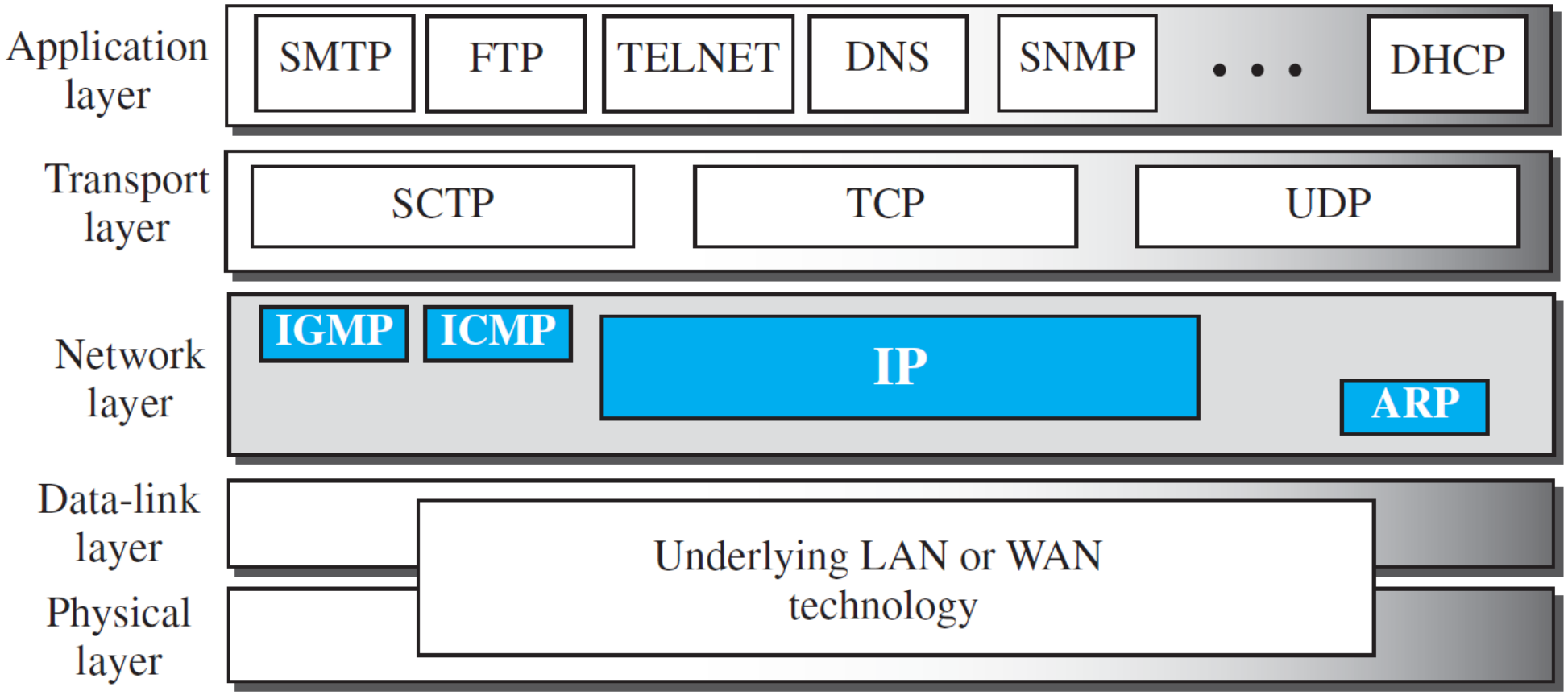
Learning Outcomes

- By the end of this lecture, you will be able to
  - Explain how the IPv4 protocol works.
  - Explain the behavior and purpose of ICMP protocol.

Introduction – Network-Layer Protocols

- **IPv4** (main protocol)
  - The main network-layer protocol — IPv4 — is responsible for **packetizing**, **forwarding**, and **delivery of a packet** at the network layer.
- Three auxiliary protocols:
  1. **Address Resolution Protocol** (**ARP**)
     - Mapping network-layer addresses (IP addresses) to link-layer addresses
  2. **Internet Control Message Protocol version 4** (**ICMPv4**)
     - Helps IPv4 to handle some errors that may occur in the network-layer delivery
  3. **Internet Group Management Protocol** (**IGMP**)
     - Helps IPv4 in multicasting

Introduction – Network-Layer Protocols



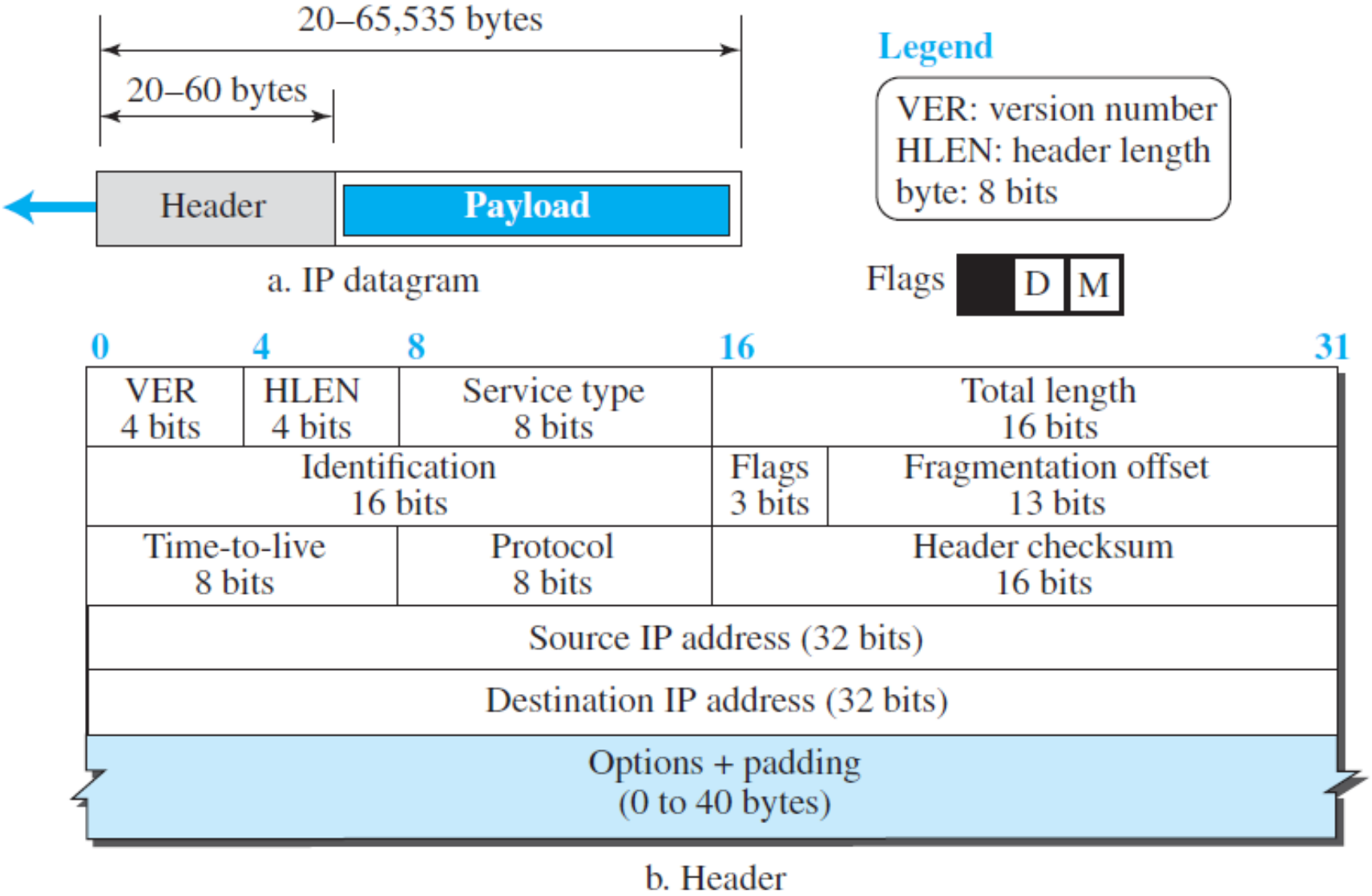
IPv4

- An **unreliable** datagram protocol which provides **best-effort delivery service**.
  - IPv4 packets can be corrupted, be lost, arrive out of order, or be delayed, and may create congestion for the network
- If reliability is important, IPv4 must be paired with a reliable transport-layer protocol such as **TCP**.
- A **connectionless** protocol that uses the **datagram approach**.
- Packets used by the IP are called **datagrams**.
- **Datagram**
  - A **variable-length packet** consisting of two parts: **header** and **payload** (**data**)
    - Header includes information essential for routing and delivery.
    - Payload is the packet coming from other protocols that use the service of IP.

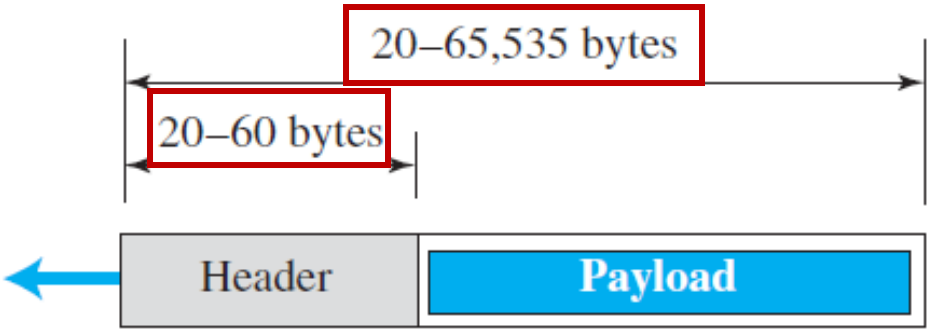
What do best-effort and connection-less mean?

- Imagine a post office service:
  - The post office does its **best** to deliver the regular mail but does not always succeed.
    - If an unregistered letter is lost or damaged, it is up to the sender or would-be recipient to discover this.
    - The post office itself does not keep track of every letter and cannot notify a sender of loss or damage of one.
- Connection-less protocol:
  - Each datagram is handled **independently**.
  - Each datagram can follow a **different route** to the **destination** → datagrams could arrive out of order.
- To take care of these problems, it relies on **higher-level protocol** (like TCP).

IPv4 – Datagram Format



IPv4 – Datagram Format

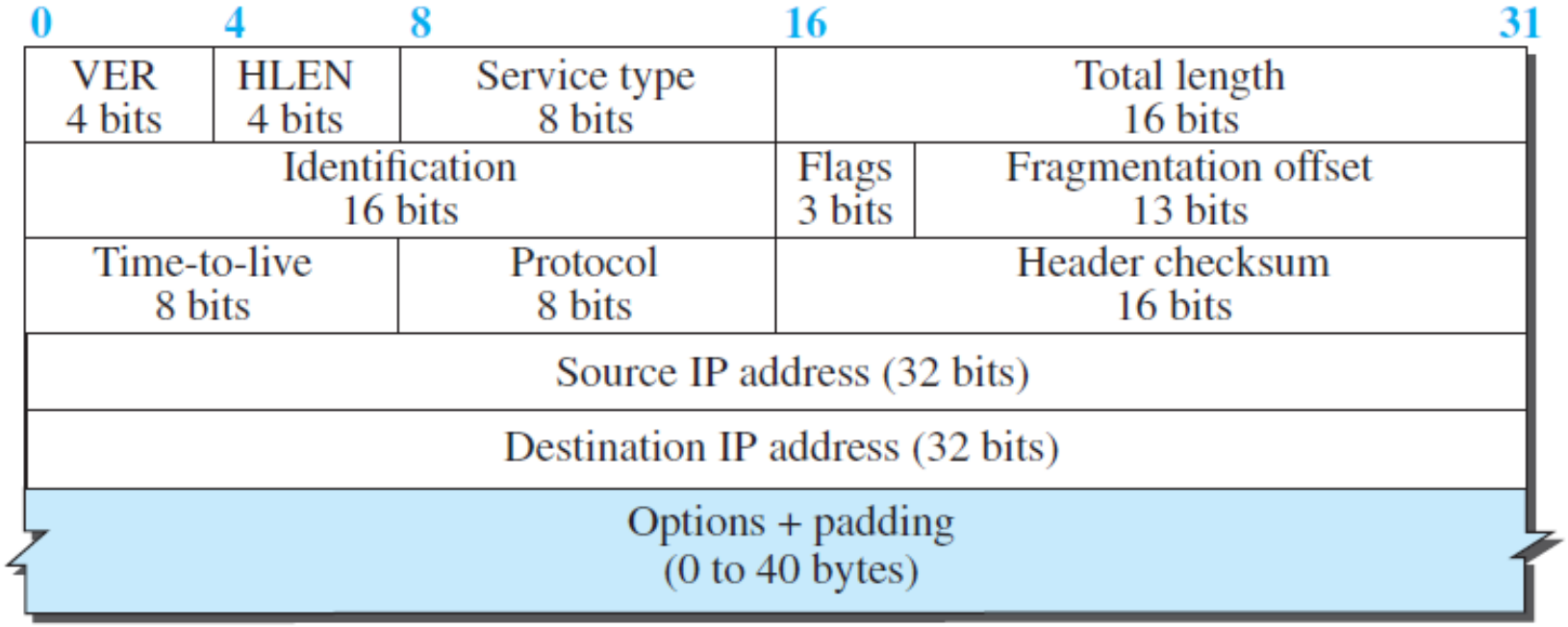


a. IP datagram

Legend

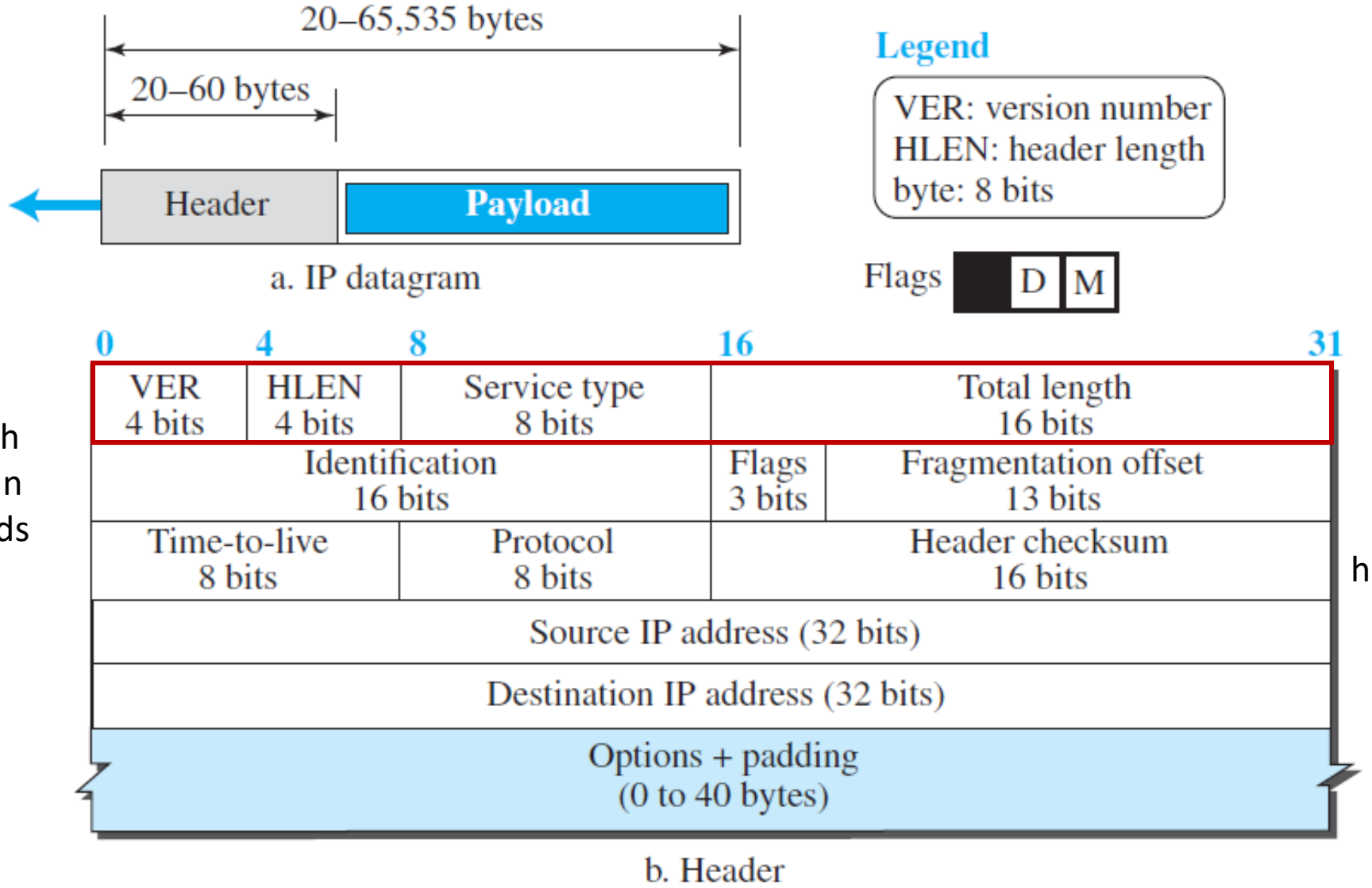
VER: version number
HLEN: header length
byte: 8 bits

Flags D M



b. Header

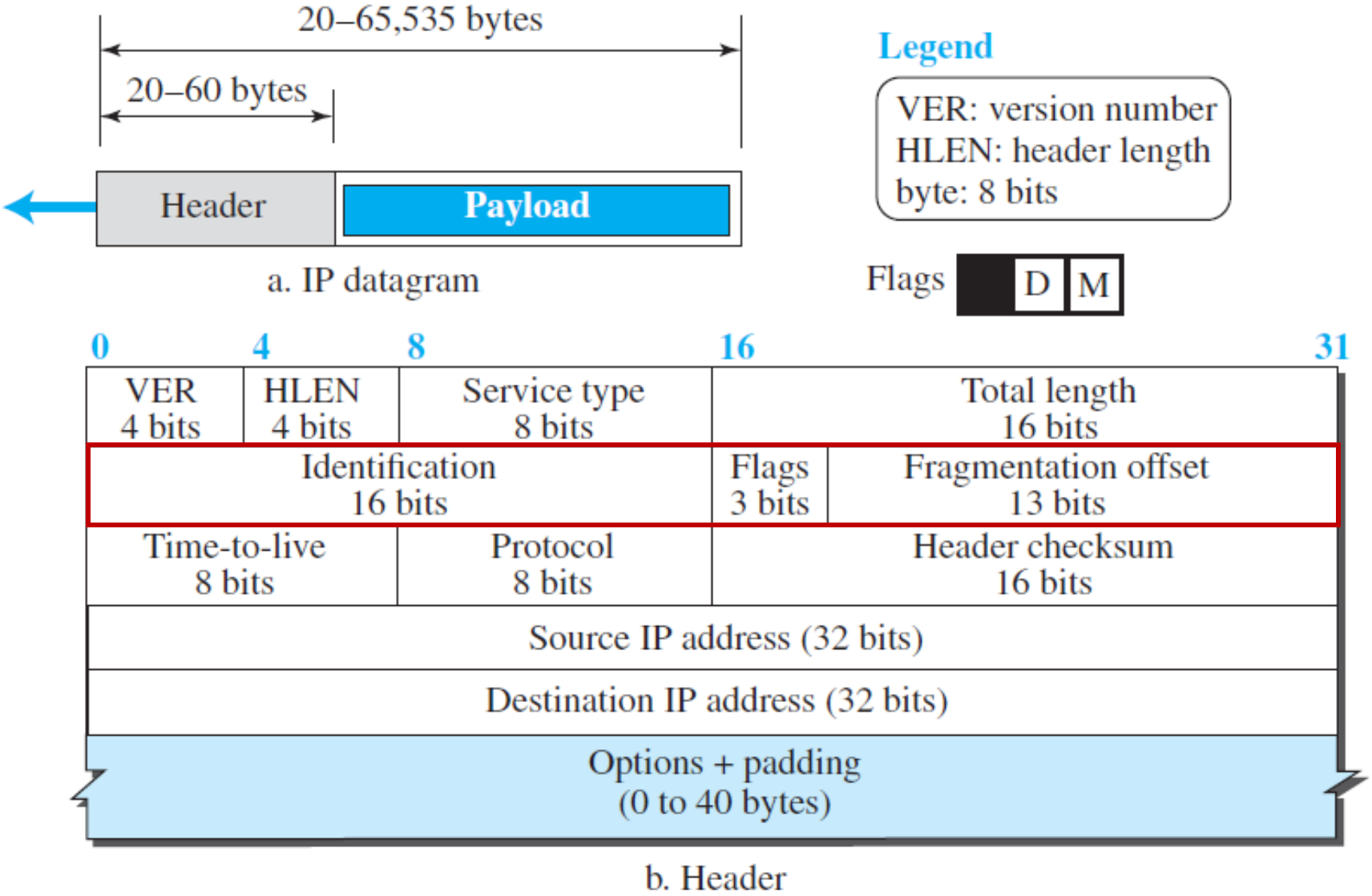
IPv4 – Datagram Format



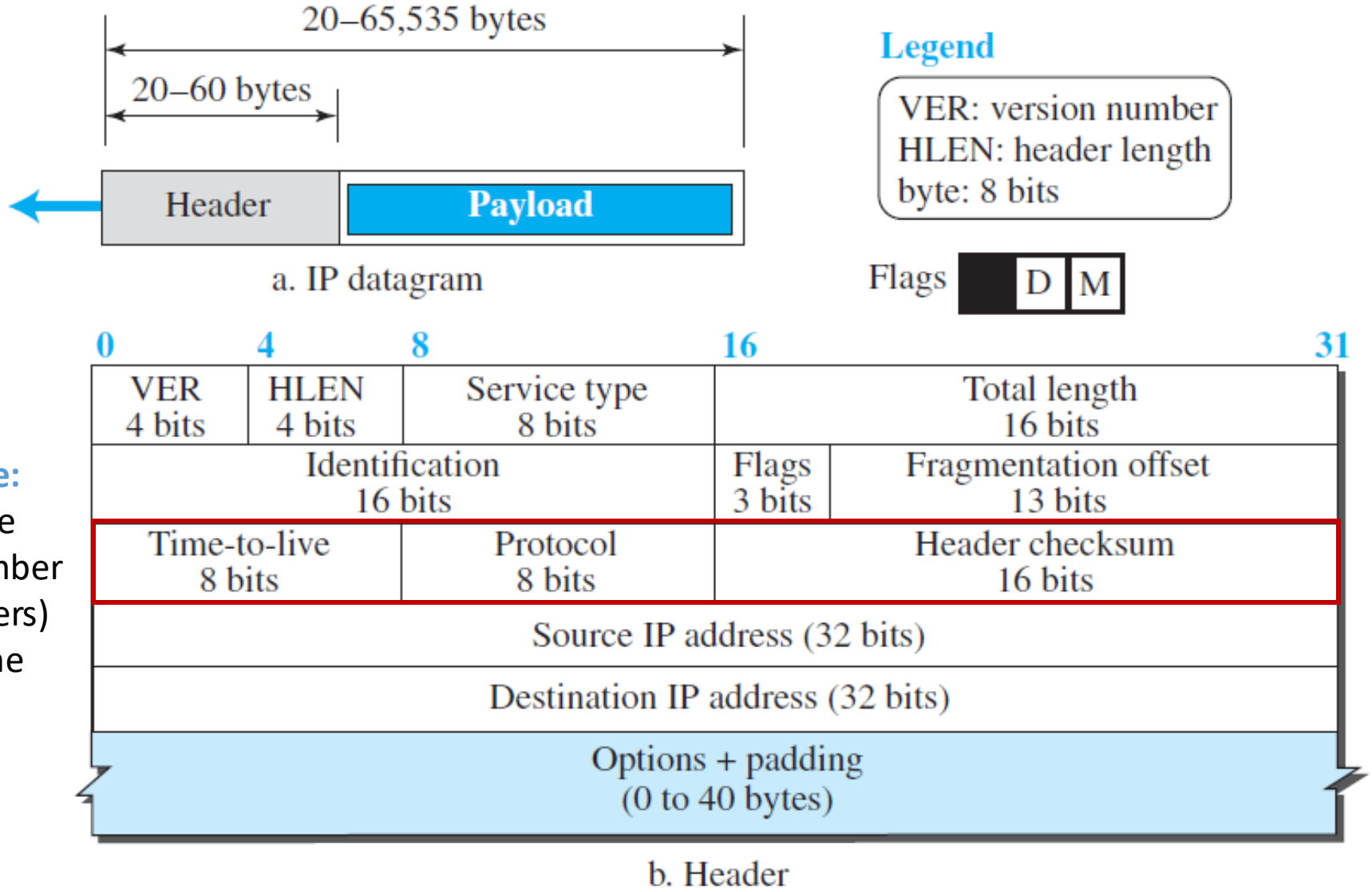
HLEN: Total length of header in 4-byte words

Total length: Total length of IP datagram in bytes (including both header and payload).

IPv4 – Datagram Format

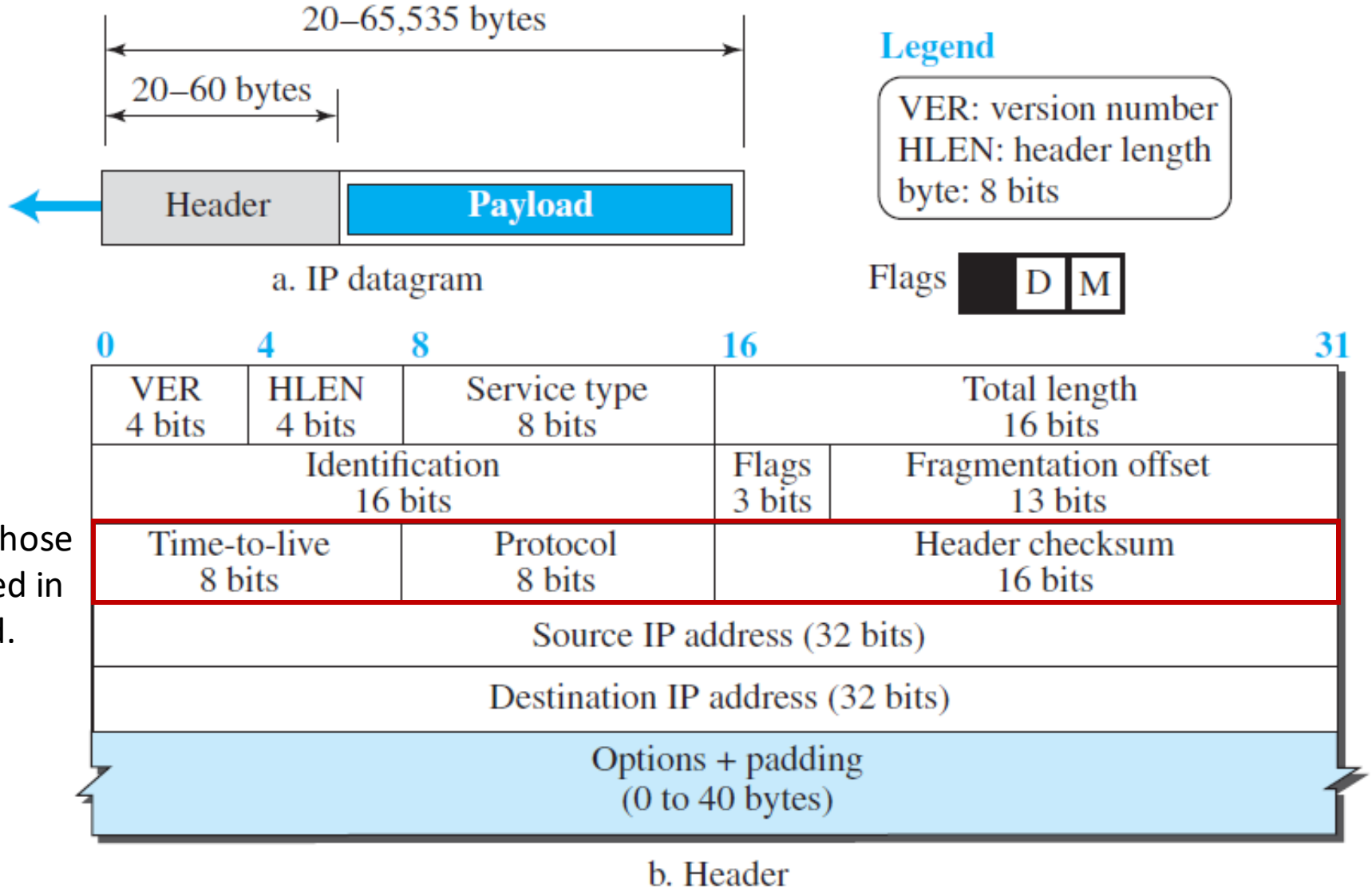


IPv4 – Datagram Format



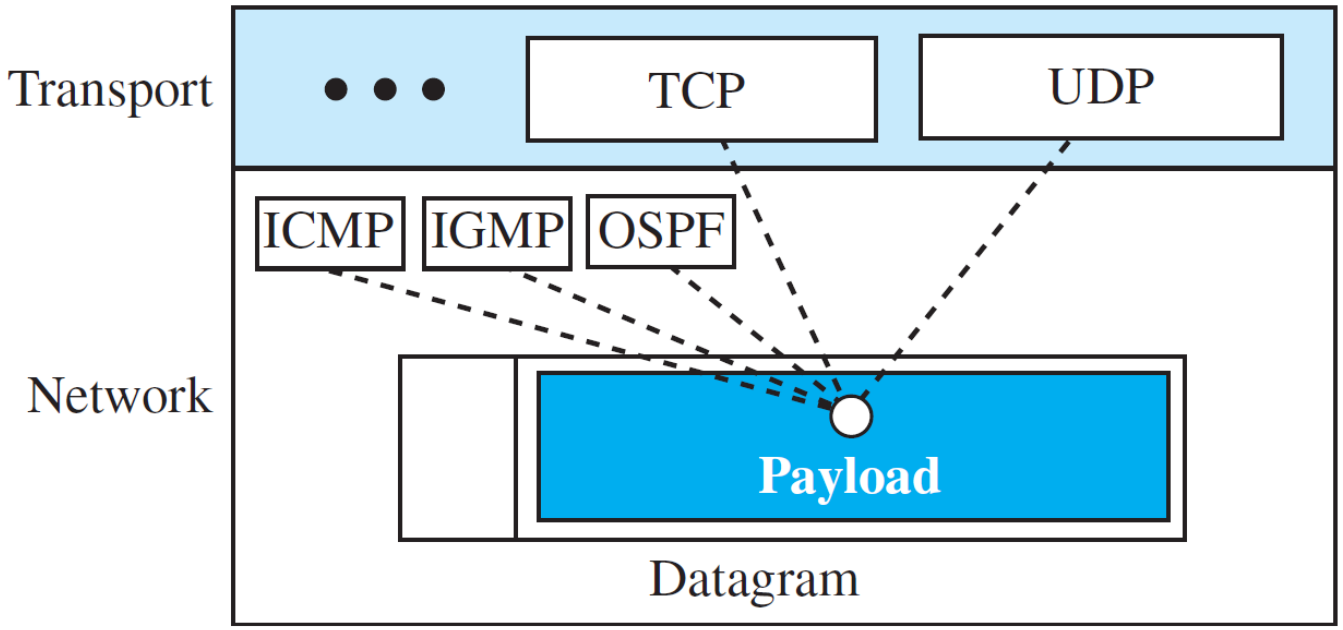
Time-to-live: Controls the maximum number of hops (routers) visited by the datagram.

IPv4 – Datagram Format



Protocol:
The protocol whose packet is carried in the payload.

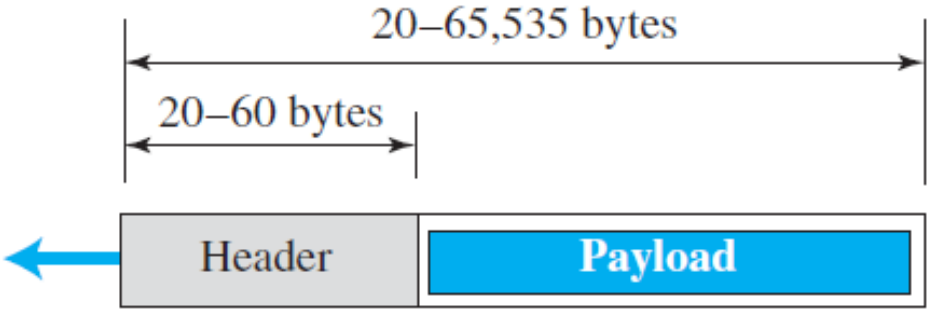
IPv4 – Datagram Protocol Field



Some protocol values

ICMP	01
IGMP	02
TCP	06
UDP	17
OSPF	89

IPv4 – Datagram Format

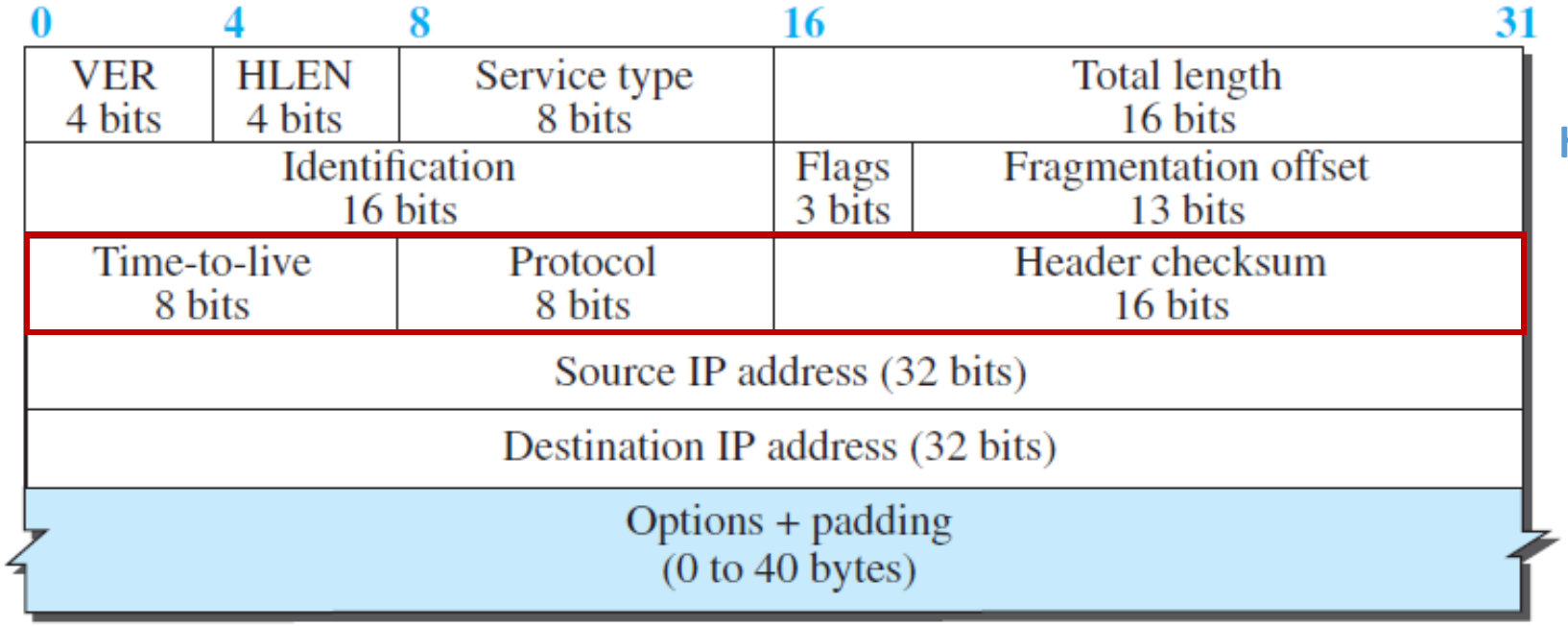


a. IP datagram

Legend

VER: version number
HLEN: header length
byte: 8 bits

Flags D M

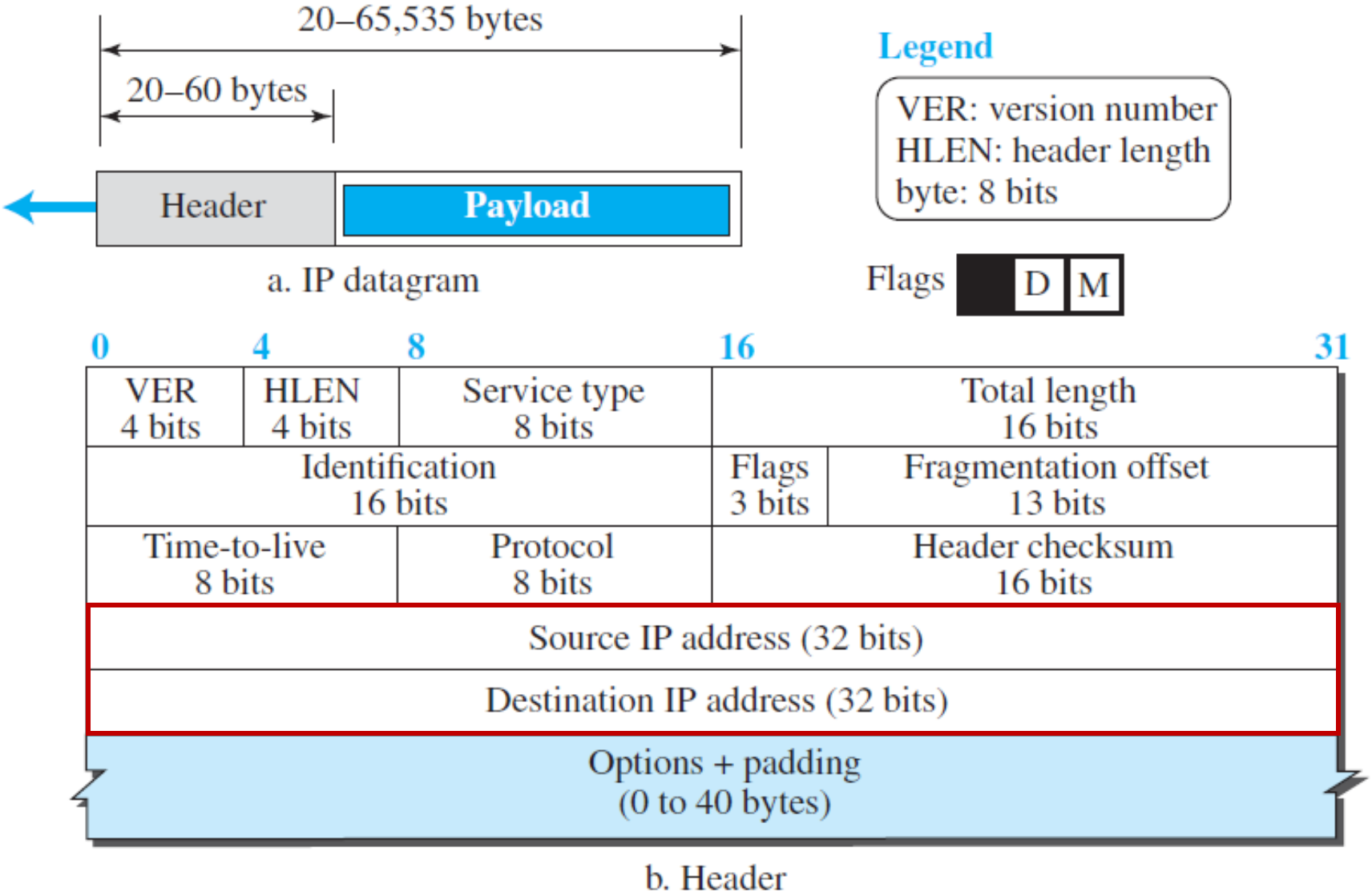


b. Header

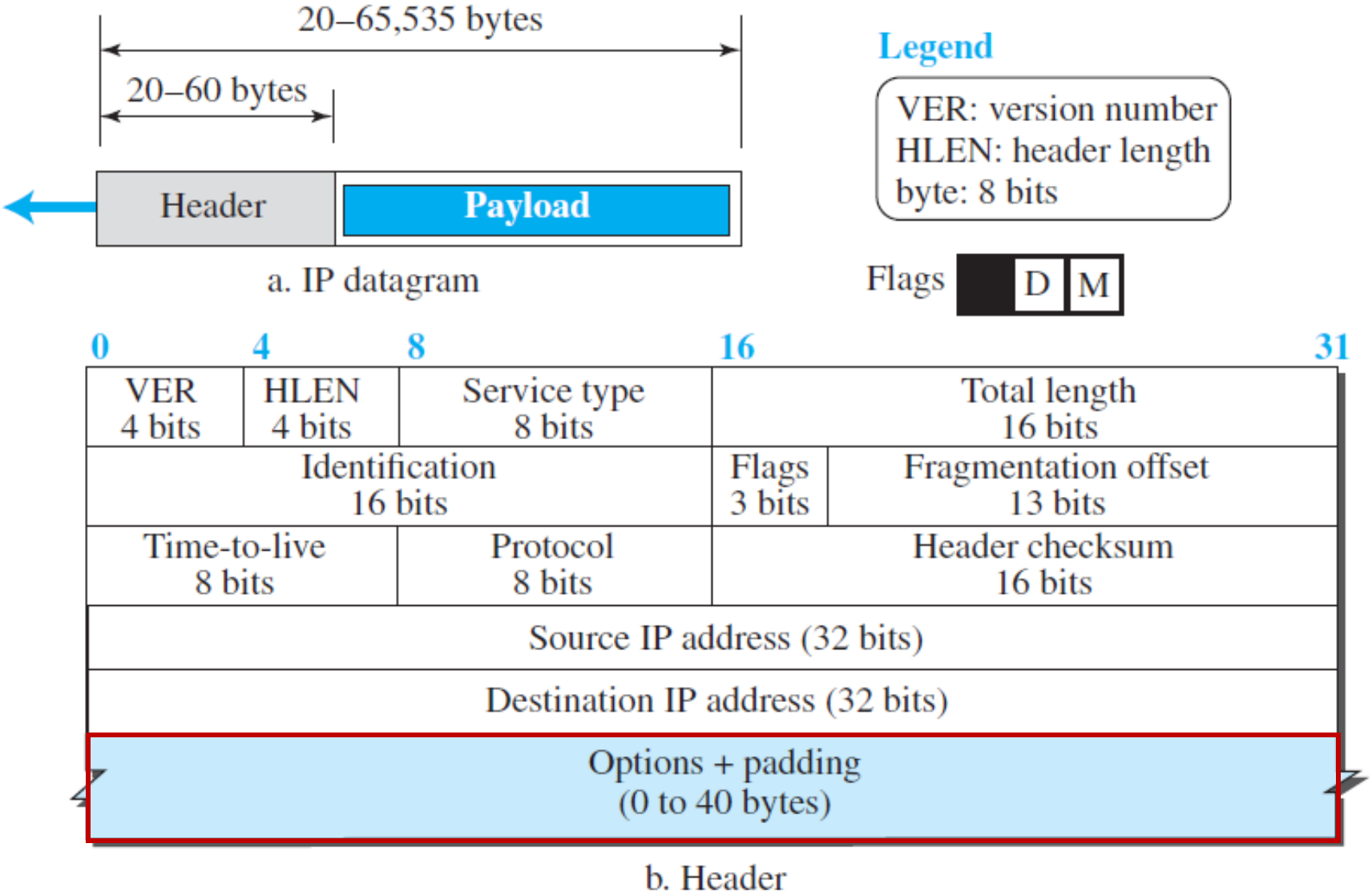
Why is error handing needed for IP header?

Why is the checksum recalculated at each router?

IPv4 – Datagram Format



IPv4 – Datagram Format



Example 1

- An IPv4 packet has arrived with the first 8 bits as $(01000010)_2$. The receiver discards the packet. Why?

Example 1

- An IPv4 packet has arrived with the first 8 bits as $(01000010)_2$. The receiver discards the packet. Why?

- **Answer**:
- There is an error in this packet. The 4 leftmost bits $(0100)_2$ show the version, which is **correct**.
- The next 4 bits $(0010)_2$ show an invalid header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been **corrupted** in transmission.

Example 2 – Header Checksum Calculation in IPv4

- We calculate a 16-bit checksum to be included in the checksum field of the IP header. We should consider each 16-bits (2-byte) section of the header. For simplicity of representation, we use hexadecimal format. Each hexadecimal letter/number represents 4 bits. Assume the following represents the IP header fields of an IP packet. Calculate the checksum.

4	5	0	28
49.153		0	0
4	17		0
10.12.14.5			
12.6.7.9			

4, 5, and 0	→	4	5	0	0
28	→	0	0	1	C
49.153	→	C	0	0	1
0 and 0	→	0	0	0	0
4 and 17	→	0	4	1	1
0	→	0	0	0	0
10.12	→	0	A	0	C
14.5	→	0	E	0	5
12.6	→	0	C	0	6
7.9	→	0	7	0	9

Example 2 – Answer

		2	
4	5	0	0
0	0	1	C
C	0	0	1
0	0	0	0
0	4	1	1
0	0	0	0
0	A	0	C
0	E	0	5
0	C	0	6
0	7	0	9
			E

0 + 12 + 1 + 0 + 1 + 0 + 12 + 5 + 6 + 9
= 46 = 0x2E

Example 2 – Answer

		2	
4	5	0	0
0	0	1	C
C	0	0	1
0	0	0	0
0	4	1	1
0	0	0	0
0	A	0	C
0	E	0	5
0	C	0	6
0	7	0	9
		4	E

2 + 0 + 1 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0
= 4 = 0x4

Example 2 – Answer

3			
4	5	0	0
0	0	1	C
C	0	0	1
0	0	0	0
0	4	1	1
0	0	0	0
0	A	0	C
0	E	0	5
0	C	0	6
0	7	0	9
3	4	4	E

5 + 0 + 0 + 0 + 4 + 0 + 10 + 14 + 12 + 7
= 52 = 0x34

Example 2 – Answer

	3			
	4	5	0	0
	0	0	1	C
	C	0	0	1
	0	0	0	0
	0	4	1	1
	0	0	0	0
	0	A	0	C
	0	E	0	5
	0	C	0	6
	0	7	0	9
1	3	4	4	E

3 + 4 + 0 + 12 + 0 + 0 + 0 + 0 + 0 + 0 + 0
= 19 = 0x13

Example 2 – Answer

	4	5	0	0
	0	0	1	C
	C	0	0	1
	0	0	0	0
	0	4	1	1
	0	0	0	0
	0	A	0	C
	0	E	0	5
	0	C	0	6
	0	7	0	9
Sum →	1	3	4	4	E

Example 2 – Answer

	4	5	0	0
	0	0	1	C
	C	0	0	1
	0	0	0	0
	0	4	1	1
	0	0	0	0
	0	A	0	C
	0	E	0	5
	0	C	0	6
	0	7	0	9
Sum →	1	3	4	4	E
Wrapped sum →		3	4	4	F

0x1 + 0xE = 0xF

Example 2 – Answer

	4	5	0	0
	0	0	1	C
	C	0	0	1
	0	0	0	0
	0	4	1	1
	0	0	0	0
	0	A	0	C
	0	E	0	5
	0	C	0	6
	0	7	0	9
Sum →	1	3	4	4	E
Wrapped sum →		3	4	4	F
Checksum →		**C**	**B**	**B**	**0**

Wrapped Sum = Sum mod FFFF

Checksum = FFFF − Wrapped Sum

Datagram and Frame

- 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.
- For example, if a router connects a **LAN** to a **WAN**, it receives a frame in the LAN format and sends a frame in the WAN format.

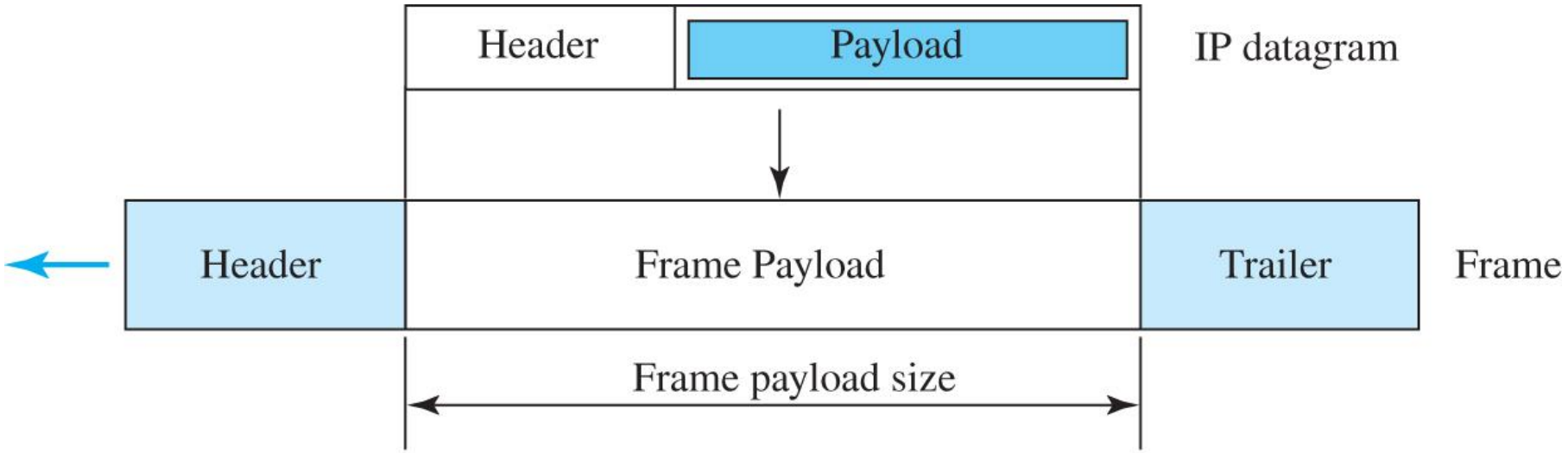
Frame Formats

- Each **link-layer protocol** has its own **frame format**.
  - One of the features of each format is the **maximum size** of the payload that can be encapsulated.
- When a datagram is encapsulated in a frame, the total size of the datagram must be **less than** this maximum size.

MTU

- **Maximum Transfer/Transmission Unit** (**MTU**).
  - The **maximum amount** of data that a **link-layer frame** can carry.
  - Its value for LAN is usually 1500 bytes.
- **Problem**: each of the links along the route between sender and destination can use different link-layer protocols, and each of these protocols can have different MTUs.
- To make the IP protocol **independent** of the physical network, the designers decided to make the maximum length of the **IP datagram** equal to 65,535 bytes.

MTU



MTU: Maximum size of frame payload

Fragmentation

- Dividing the datagram to make it possible for it to pass through different networks with different MTUs.
- When a datagram is fragmented, **each fragment has its own header** with most of the fields repeated, but **some have been changed**.
  - A datagram may be fragmented **several times** before it reaches the final destination (if passes through a network with an even smaller MTU).
- A datagram can be fragmented by the source host or any router in the path.
  - The **reassembly of the datagram** is done only by the destination host.
- The **payload** of a datagram is **fragmented**.
  - Most parts of the header, with the exception of some options, must be copied by all fragments.

Fragmentation

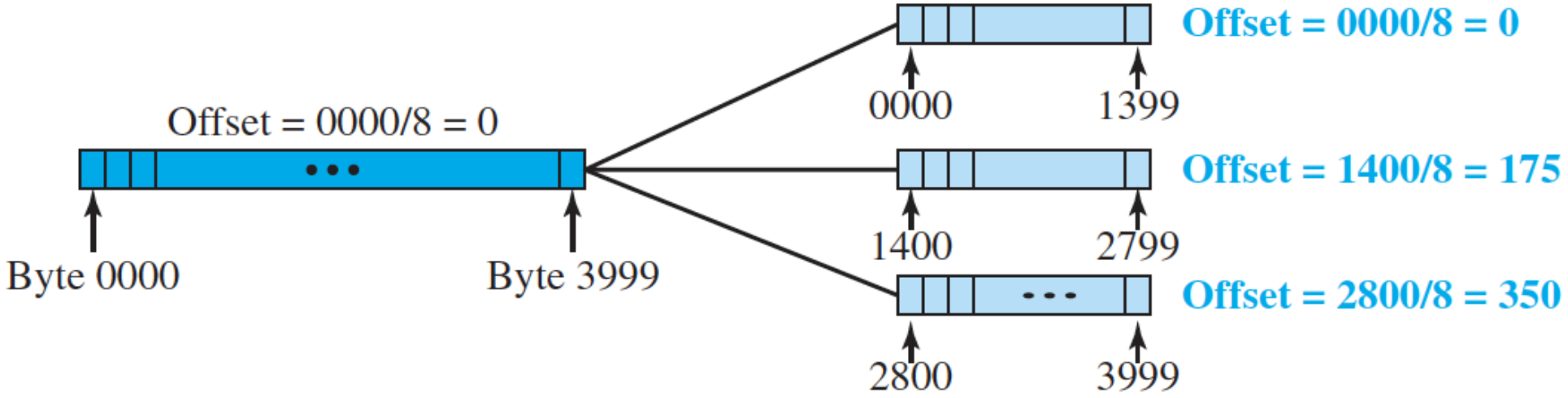
- The host or router that fragments a datagram must change the values of three fields: **flags**, **fragmentation offset**, and **total length**.
  - The value of the checksum must be recalculated regardless of fragmentation.

Fragmentation

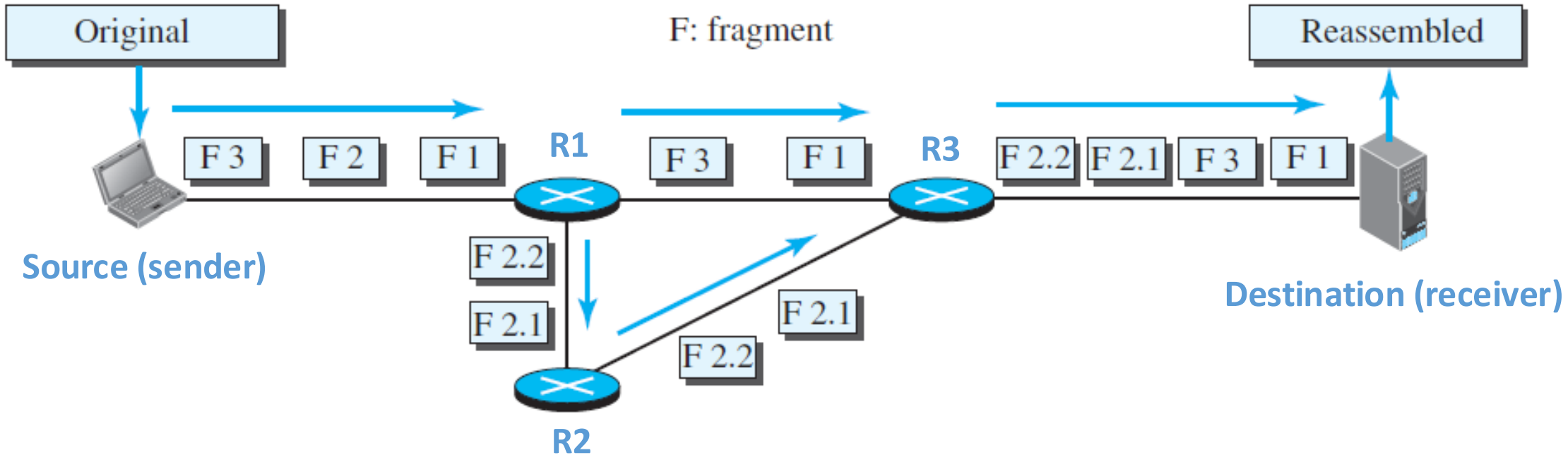
- To help the destination host to **perform the reassembly task**, three fields are included in the IP datagram header.
  - **Identification number** (16-bit field) → Same for the original and all fragments of a datagram
  - **Flags** (3-bit field)
    - Leftmost bit is not used
    - Second bit (**do not fragment** bit, **D**) → if 1, the datagram **must not be fragmented**, even if it cannot pass it (discards, ICMP error to the source).
    - Third bit (**more fragment** bit, **M**) → helps in determining the last fragment (0 for the last (or only) fragment)
  - **Fragmentation offset** → to specify where the fragment fits (position) within the original IP datagram (**offset** of the data in the original datagram measured in **units of 8 bytes**).

Fragmentation – Example

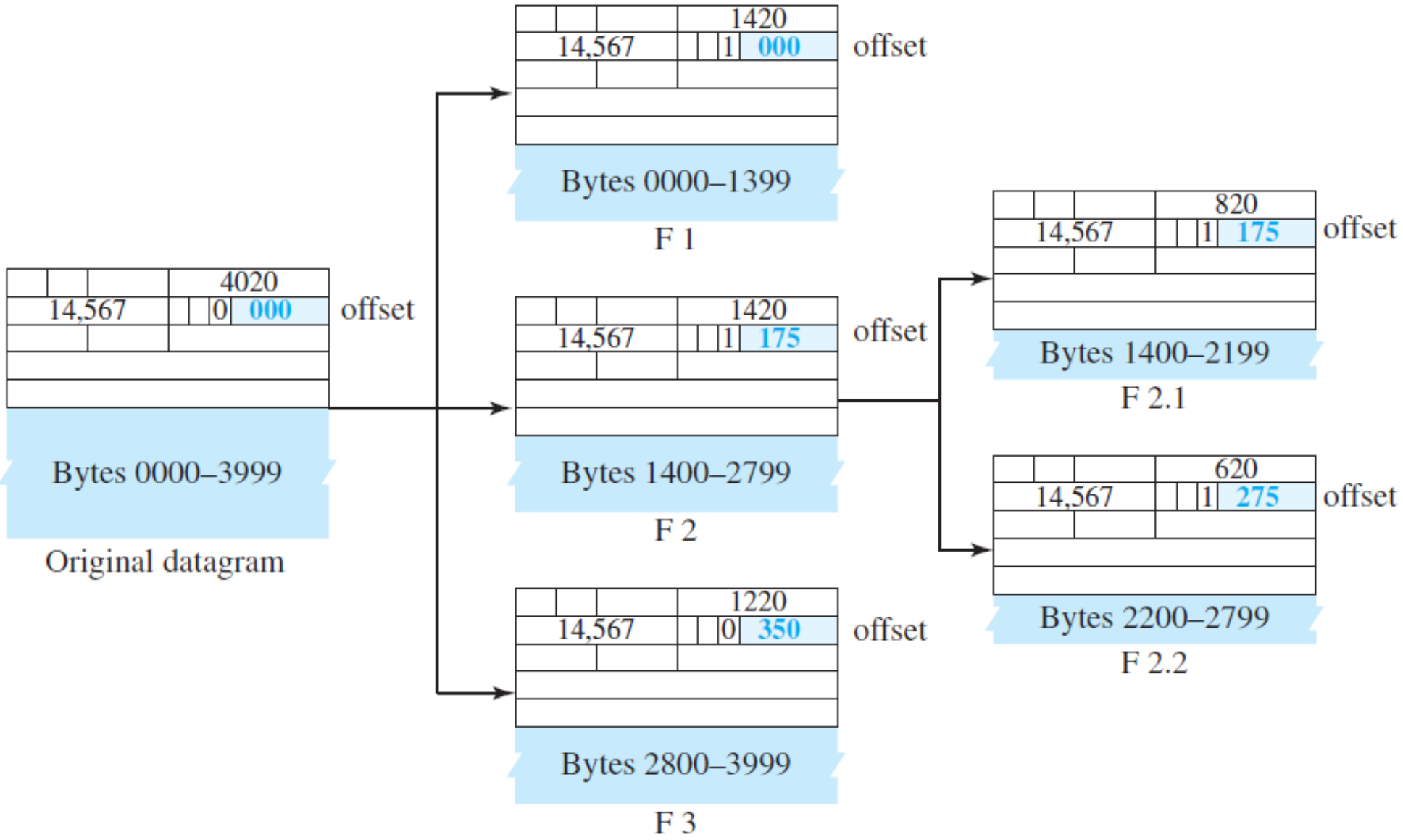
- The value of the offset is measured in **units of 8 bytes** because the length of the offset field is only 13 bits long and cannot represent a sequence of bytes greater than 8191 (2^{13}). This forces hosts or routers that fragment datagrams to choose the size of each fragment so that the **first byte number** is **divisible by 8**.



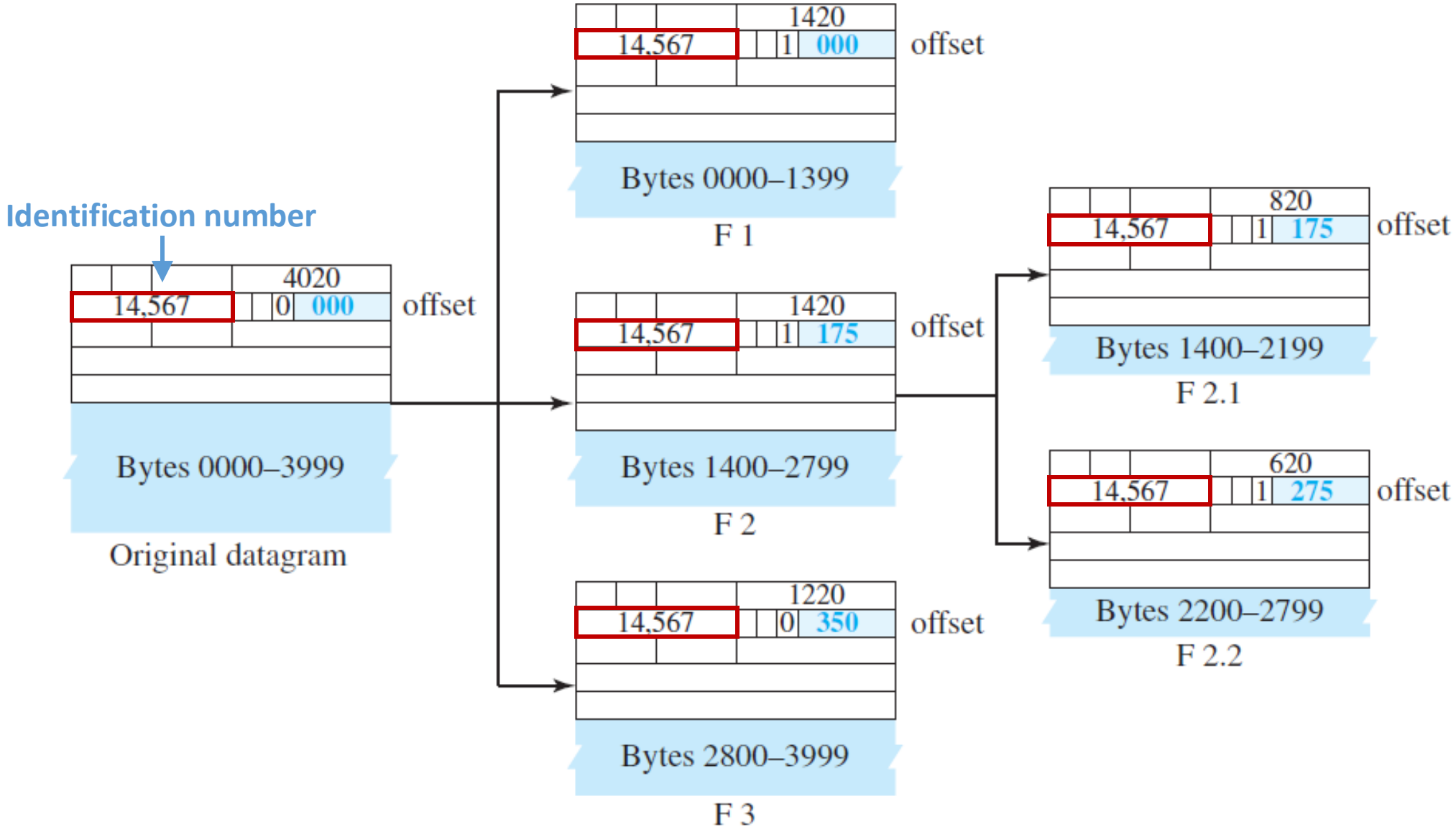
Fragmentation – Example



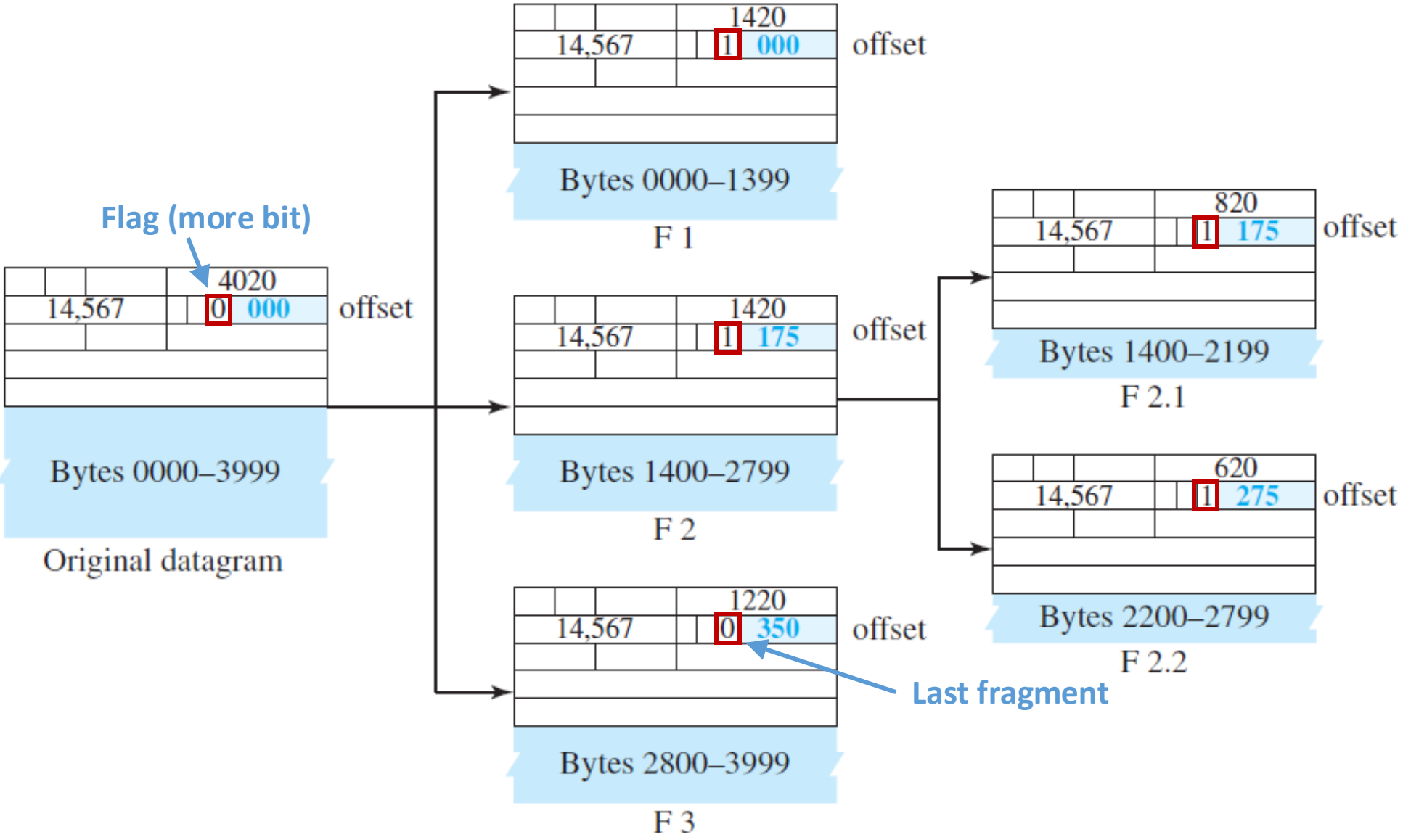
Fragmentation – Example



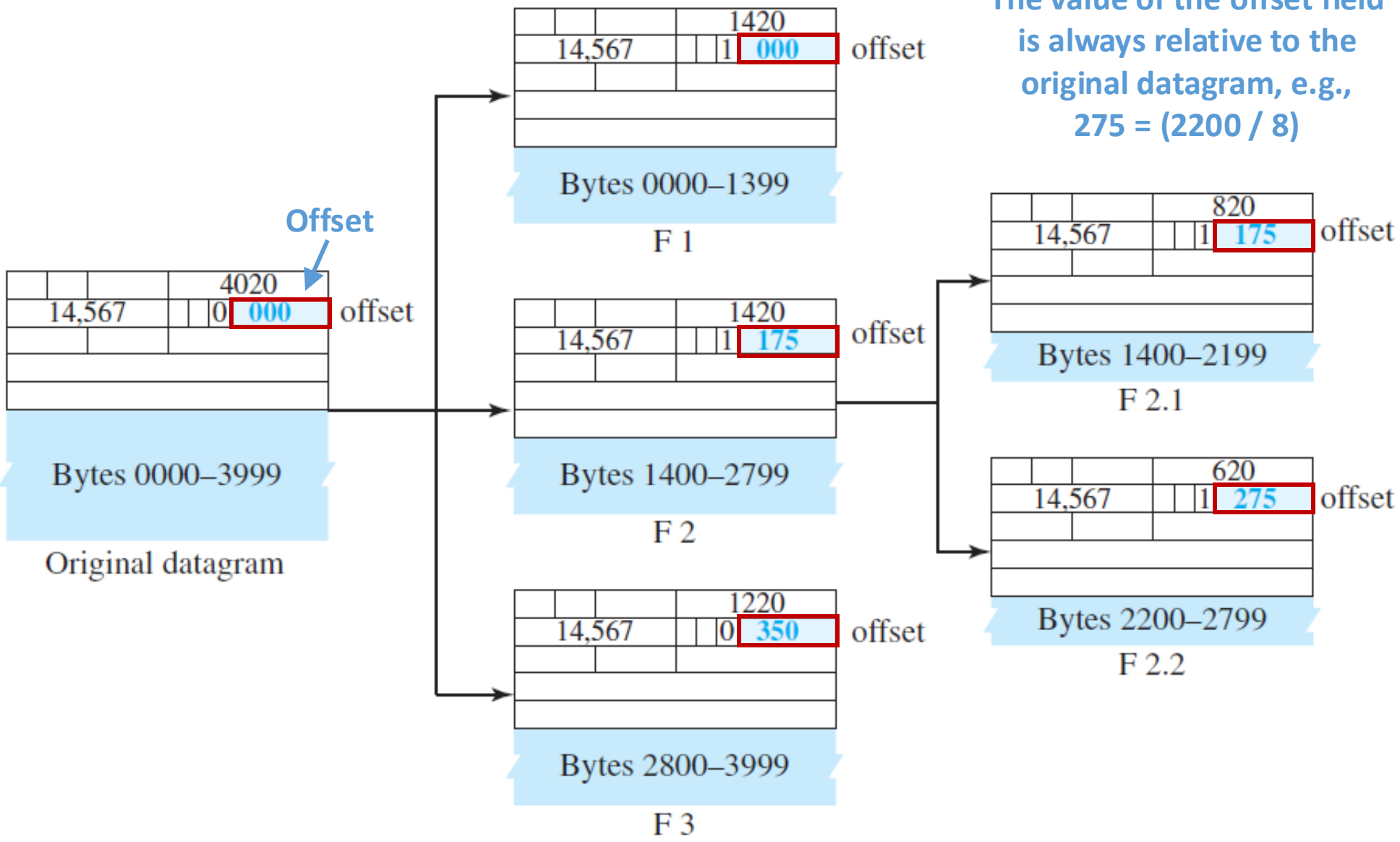
Fragmentation – Example



Fragmentation – Example



Fragmentation – Example



The value of the offset field is always relative to the original datagram, e.g., 275 = (2200 / 8)

ICMPv4

- The IPv4 has **no error-reporting** or **error-correcting** mechanism.

- Some errors that might happen during transmission?
  - Router discards a datagram because it cannot find a route.
  - Time-to-live field has a zero value.
  - Final destination discards the received fragments of a datagram because it has not received all fragments.

- These are examples of situations where an error has occurred, and the IP protocol has **no built-in mechanism** to **notify the original host**.

ICMPv4

- The IP protocol also lacks a mechanism for host and management queries.
  - A host sometimes needs to determine if a router or another host is alive.
  - Sometimes a network manager needs information from another host or router.

- **Internet Control Message Protocol version 4** (**ICMPv4**) has been designed to compensate for the mentioned deficiencies.

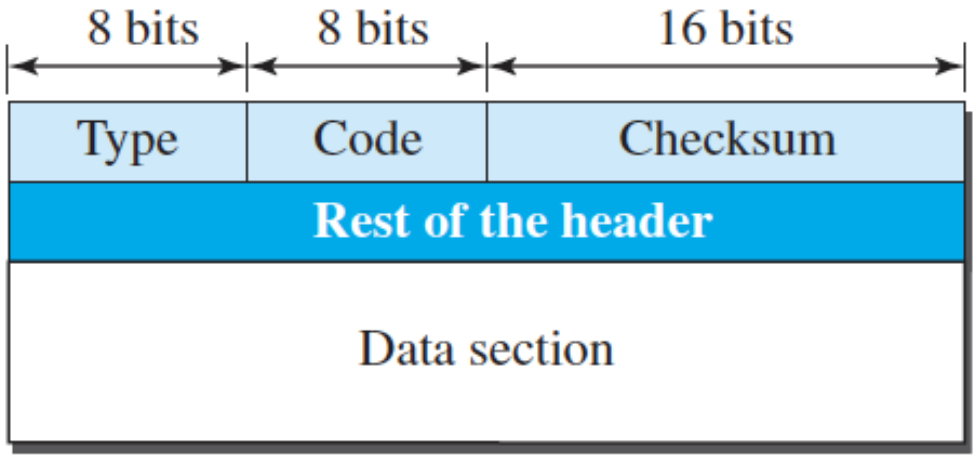
ICMPv4

- A **network-layer protocol** (RFC792).
- ICMP is used by **hosts** and **routers** to communicate network-layer information to each other.
- ICMP messages are first encapsulated inside IP datagrams before going to the lower layer.
  - The value of the **protocol** field in the IP datagram is **set to 1** to indicate that the **IP payload** is an **ICMP message**.

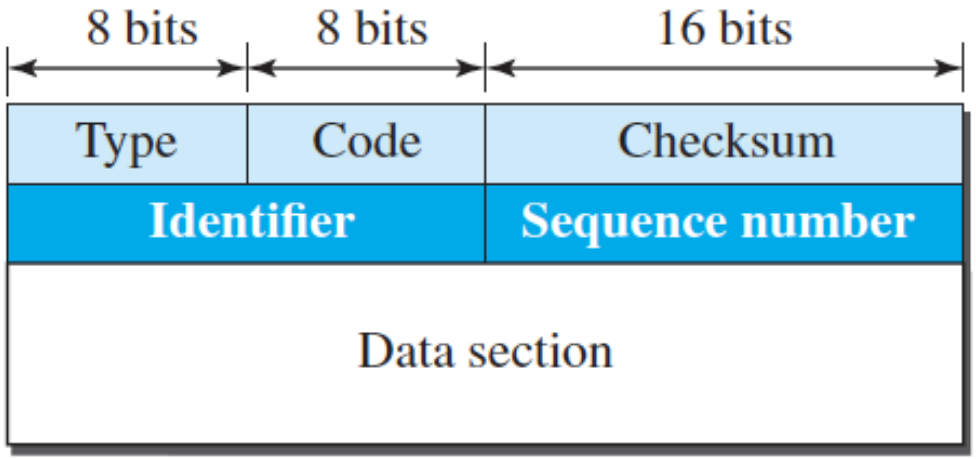
ICMPv4 – Messages

- Two categories of messages
  1. **Error-reporting messages**
     - Report problems that a **router** or a **host** (destination) may encounter when it processes an IP packet.
     - ICMP uses the source IP address to send the error message to the source (originator) of the datagram.
  2. **Query messages**
     - These messages occur in pairs (request and reply) and they help a host or a network manager get **specific information** from a **router** or **another host**.
     - Used to probe or test the **liveliness** of hosts or routers on the Internet, find the **one-way** or the **round-trip time** for an IP datagram between two devices, or even find out whether the clocks in two devices are synchronized.
- An ICMP message has an 8-byte header and a variable-size data section
  - Two important fields in the header: **Type** and **Code**

ICMPv4 – Messages



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)

ICMPv4 – Destination Unreachable

- The most widely used **error message** is the destination unreachable (Type 3).
  - Example: **Type 3**, **Code 0**: when we use the HTTP protocol to access a web page, but the server is down, the message "**destination host is not reachable**" is created and sent back to the source.

ICMPv4 – Time Exceeded Message with Code 0

- An error message.
- The **purpose** of the **time-to-live** (**TTL**) field in the IP datagram is to prevent a datagram from being aimlessly circulated on the Internet.
- When the TTL value becomes 0, the datagram is dropped by the visiting router and a **time exceeded message**, **Type 11** with **Code 0**, is sent to the source to inform it about the situation.

ICMPv4 – Echo Request and Reply

- A query message.
- The echo request (**Type 8**) and the echo reply (**Type 0**) pair of messages are used by a host or a router to test the **liveliness** of another host or router.
- A host or router sends an echo request message to another host or router; if the latter is alive, it responds with an echo reply message.
- Application → debugging tools like **ping** and **traceroute**.

ICMP and Debugging Tools on the Internet

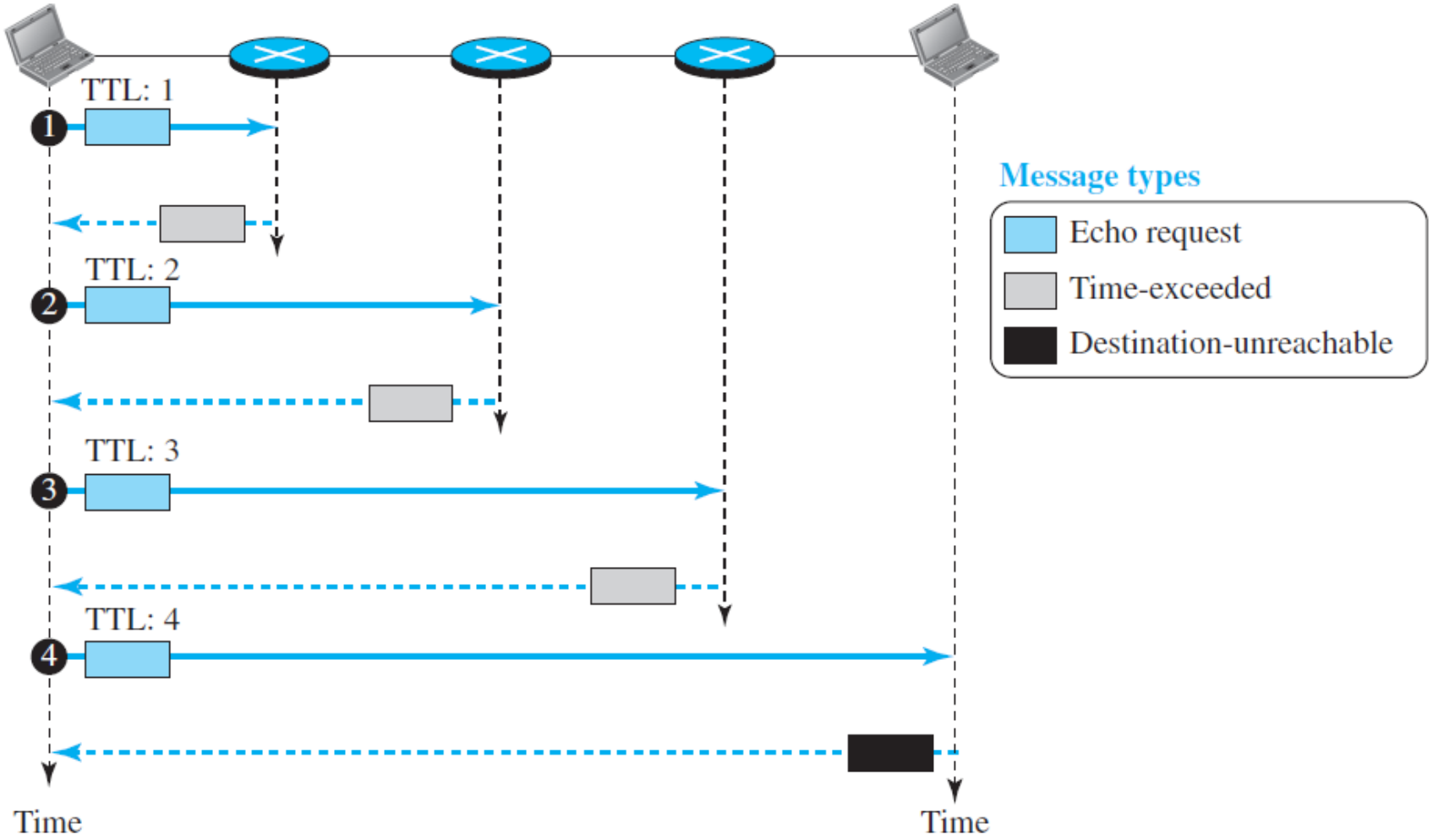
- **ping**
  - To find if a host is alive and responding.
  - Based on two **query messages** (echo request/reply).

```
C:\Users>ping bcit.ca

Pinging bcit.ca [142.232.230.10] with 32 bytes of data:
Reply from 142.232.230.10: bytes=32 time=24ms TTL=245
Reply from 142.232.230.10: bytes=32 time=22ms TTL=245
Reply from 142.232.230.10: bytes=32 time=22ms TTL=245
Reply from 142.232.230.10: bytes=32 time=21ms TTL=245

Ping statistics for 142.232.230.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 21ms, Maximum = 24ms, Average = 22ms
```

ICMP and Debugging Tools on the Internet

- **traceroute**/**tracert**
  - The **traceroute** program in UNIX/Linux or **tracert** in Windows.
  - Based on two **error-reporting messages**: **time-exceeded** and **destination-unreachable** as well as **echo-request query message**.
  - To **trace** the **path** of a packet from a source to the destination.
  - It can find the IP addresses of all the routers that are visited along the path.
  - Usually set to check for the maximum of 30 hops (routers) to be visited.

Traceroute


TTL: 1
1
TTL: 2
2
TTL: 3
3
TTL: 4
4
Time
Time
Message types
Echo request
Time-exceeded
Destination-unreachable

Traceroute

```
C:\Users>tracert bcit.ca

Tracing route to bcit.ca [142.232.230.10]
over a maximum of 30 hops:

  1     2 ms     1 ms     1 ms  192.168.1.254
  2     5 ms     6 ms     5 ms  10.31.234.1
  3     6 ms     5 ms     5 ms  154.11.10.159
  4     6 ms     8 ms     5 ms  as6939.vanix.ca [206.41.104.27]
  5     7 ms     6 ms     6 ms  bcnet.10gigabitethernet1-4.core1.yvr1.he.net [184.105.148.150]
  6    21 ms    21 ms    21 ms  cr1-100g-bb3927ae8.srry1.bc.net [207.23.253.121]
  7    21 ms    21 ms    21 ms  427-bcit-tx-cr1.vncv1.bc.net [207.23.240.25]
  8    22 ms    22 ms    21 ms  ip-142-232-230-10.ptr.bcit.ca [142.232.230.10]

Trace complete.
```

Summary

- **IPv4** is an **unreliable connectionless protocol** responsible for source-to-destination delivery.
- Packets in IP layer (network layer) are called **datagrams** (each datagram includes a header and a payload).
- The IPv4 datagrams can be **fragmented** in the path to the destination.
- **Checksum** is calculated only for the **header of datagram**.
- **ICMP** messages are encapsulated in IP datagrams.
- Two types of ICMP messages: **error-reporting** and **query**.

References

[1] Behrouz A.Forouzan, Data Communications & Networking with TCP/IP Protocol Suite, 6th Ed, 2022, McGraw-Hill companies.

[2] J.F. Kurose, K.W. Ross, Computer Networking: A Top-Down Approach, 7th Ed, 2017, Pearson Education, Inc.

Reading

- Chapter 7 of the textbook, sections 7.4.2 – 7.4.4.
- Chapter 7 of the textbook, section 7.8 (Practice Test)