

# 7장. IPv4 (Internet Protocol Version4)

2025년 1학기  
단국대학교 컴퓨터공학과  
박태근

# Contents

---

- 7.1 소개 (Introduction)
- 7.2 데이터그램 (Datagrams)
- 7.3 단편화 (Fragmentation)
- 7.8 IP 패키지 (IP Package)

# 7.1 소개

---

✓ 인터넷 프로토콜 (IP: Internet Protocol)은

- TCP/IP 프로토콜이 사용하는 전송 메커니즘 (transmission mechanism)이며,
- 네트워크 계층 (network layer)에 위치함

# 7.1 소개 - Topics

---

- 1) TCP/IP 프로토콜 모음의 나머지 부분과 IP의 관계

## 7.1 소개 – TCP/IP 프로토콜 모음의 나머지 부분과 IP의 관계

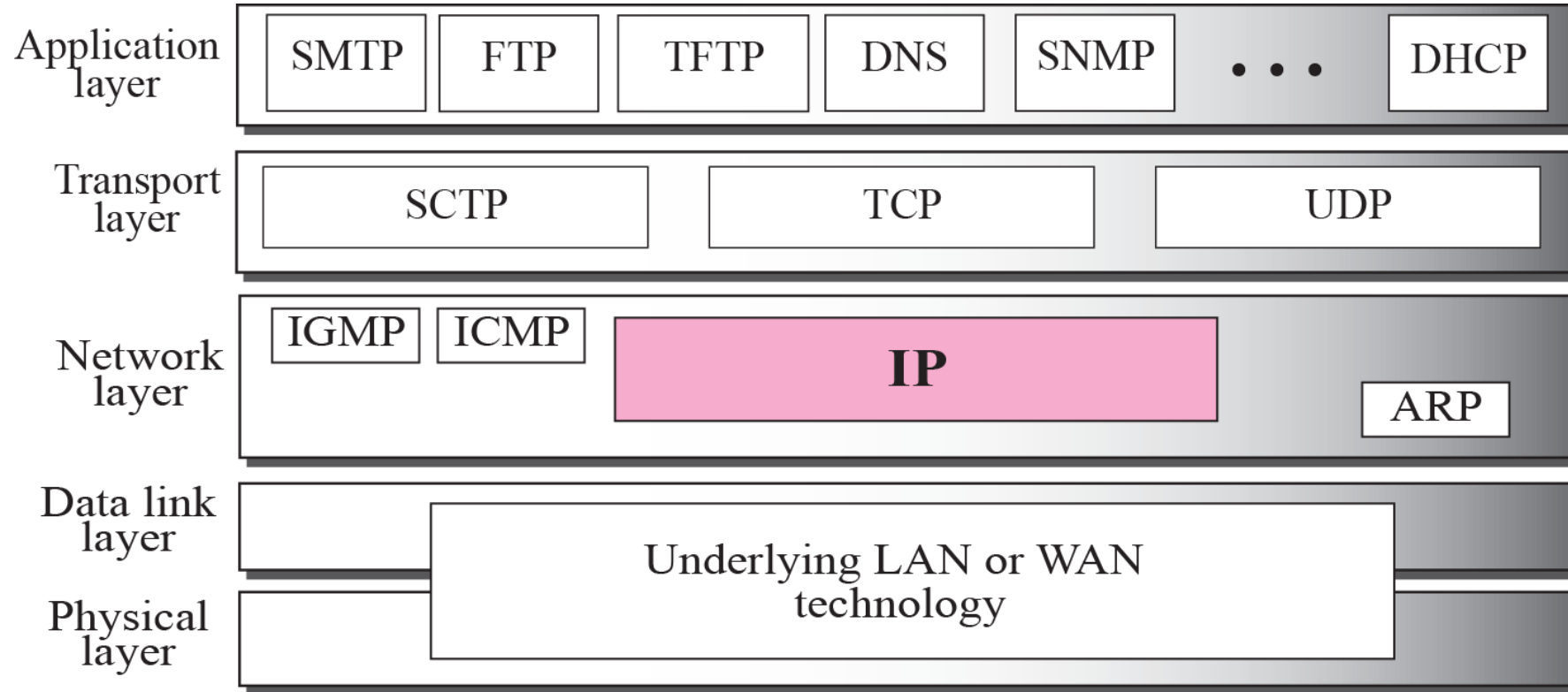


그림 7.1 TCP/IP 프로토콜 모음 내에서 IP의 위치  
(Position of IP in TCP/IP protocol suite)

## 7.2 데이터그램

---

✓ **IP** = Internet Protocol

- 신뢰성없고 (unreliable) 비연결형 (connectionless) 데이터그램 (datagram) 프로토콜
- IP는 **최선의 노력 전달 (best-effort delivery) 서비스**를 제공
  - **"최선의 노력 (best-effort)"**이라는 용어는 다음을 의미함
    - IP 패킷은
      - 손상/훼손 (corrupted)**되거나, **분실 (lost)**되거나, **틀린 순서로 도착 (arrive out of order)**하거나,
      - 지연될 (delayed)** 수 있으며, 네트워크 내에서 **혼잡 문제를 발생 (create congestion)**시킬 수 있음
- 보편적으로 이해할 수 있는 **최선의 노력 전달 서비스 (best-effort delivery service)의 예:**
  - **우체국 (post office)**

✓ **Datagrams** = **Packets** in the network layer (IP)

## 7.2 데이터그램 - Topics

---

- 1) 데이터그램의 형식 (Format of the datagram)
- 2) 예제 (Some examples)

## 7.2 데이터그램 - 데이터그램의 형식

- ✓ IP 데이터그램의 형식을 공부해야 하는 이유 - Example (Wireshark) (1)

The image shows the Wireshark network protocol analyzer interface. The main window displays a list of captured packets. Packet 11 is selected, and its details are shown in the lower pane. The packet list shows alternating FTP-DATA and TCP packets between 192.168.0.100 and 192.168.0.21. Packet 11 is a TCP ACK packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
2	0.000001	192.168.0.21	192.168.0.100	TCP	66	43546 → 20 [ACK] Seq=1 Ack=4294965849 W...
3	0.000001	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
4	0.000001	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
5	0.000002	192.168.0.21	192.168.0.100	TCP	66	43546 → 20 [ACK] Seq=1 Ack=1449 Win=552...
6	0.000002	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
7	0.000002	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
8	0.000003	192.168.0.21	192.168.0.100	TCP	66	43546 → 20 [ACK] Seq=1 Ack=4345 Win=552...
9	0.000003	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
10	0.000760	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
11	0.000760	192.168.0.21	192.168.0.100	TCP	66	43546 → 20 [ACK] Seq=1 Ack=7241 Win=552...
12	0.000761	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
13	0.000761	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
14	0.000761	192.168.0.21	192.168.0.100	TCP	66	43546 → 20 [ACK] Seq=1 Ack=10137 Win=55...

Details for Frame 11:

- Frame 11: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
- Ethernet II, Src: HewlettP\_a2:ef:e9 (18:a9:05:a2:ef:e9), Dst: SamsungE\_cd:51:47 (00:13:77:cd:51:47)
- Internet Protocol Version 4, Src: 192.168.0.21, Dst: 192.168.0.100
- Transmission Control Protocol, Src Port: 43546, Dst Port: 20, Seq: 1, Ack: 7241, Len: 0

Packet bytes (hex and ASCII):

```
0000  00 13 77 cd 51 47 18 a9 05 a2 ef e9 08 00 45 08  ..w.QG.. .....E.
0010  00 34 9f 93 40 00 40 06 19 5f c0 a8 00 15 c0 a8  .4..@.@. _.....
0020  00 64 aa 1a 00 14 c8 30 71 4d df 97 f2 f7 80 10  .d.....0 qM.....
0030  15 91 99 5d 00 00 01 01 08 0a 00 80 fd 6d 00 7f  ...].... .....m..
0040  91 5b  .[
```



## 7.2 데이터그램 - 데이터그램의 형식

- ✓ IP 데이터그램의 형식을 공부해야 하는 이유 - Example (Wireshark) (2)

The image shows a screenshot of the Wireshark network traffic analysis tool. The main window displays a list of captured packets. Packet 11 is selected, and its details are shown in the lower pane. The details pane shows the structure of the IP datagram, including the version, header length, differentiated services field, total length, identification, and flags. The flags field is expanded, showing the 'Don't Fragment' (DF) bit is set. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

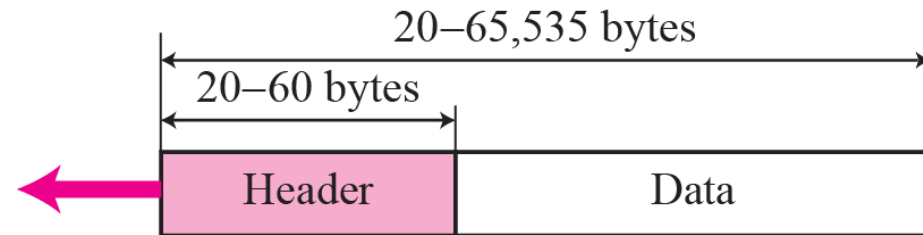
No.	Time	Source	Destination	Protocol	Length	Info
10	0.000760	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
11	0.000760	192.168.0.21	192.168.0.100	TCP	66	43546 → 20 [ACK] Seq=1 Ack=7241 Win=552
12	0.000761	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
13	0.000761	192.168.0.100	192.168.0.21	FTP-DATA	1514	FTP Data: 1448 bytes
14	0.000761	192.168.0.21	192.168.0.100	TCP	66	43546 → 20 [ACK] Seq=1 Ack=10137 Win=552

Frame 11: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0  
Ethernet II, Src: HewlettP\_a2:ef:e9 (18:a9:05:a2:ef:e9), Dst: SamsungE\_cd:51:47 (00:13:77:cd:51:47)  
Internet Protocol Version 4, Src: 192.168.0.21, Dst: 192.168.0.100  
0100 .... = Version: 4  
.... 0101 = Header Length: 20 bytes (5)  
Differentiated Services Field: 0x08 (DSCP: Unknown, ECN: Not-ECT)  
Total Length: 52  
Identification: 0x9f93 (40851)  
Flags: 0x02 (Don't Fragment)  
0... .... = Reserved bit: Not set  
.1.. .... = Don't fragment: Set  
..0. .... = More fragments: Not set  
Fragment offset: 0  
Time to live: 64  
Protocol: TCP (6)  
0000 00 13 77 cd 51 47 18 a9 05 a2 ef e9 08 00 45 08 ..w.QG.. .....E.  
0010 00 34 9f 93 40 00 40 06 19 5f c0 a8 00 15 c0 a8 .4..@. \_.....  
0020 00 64 aa 1a 00 14 c8 30 71 4d df 97 f2 f7 80 10 .d.....0 qM.....  
0030 15 91 99 5d 00 00 01 01 08 0a 00 80 fd 6d 00 7f ...].... .....m..  
0040 91 5b .[

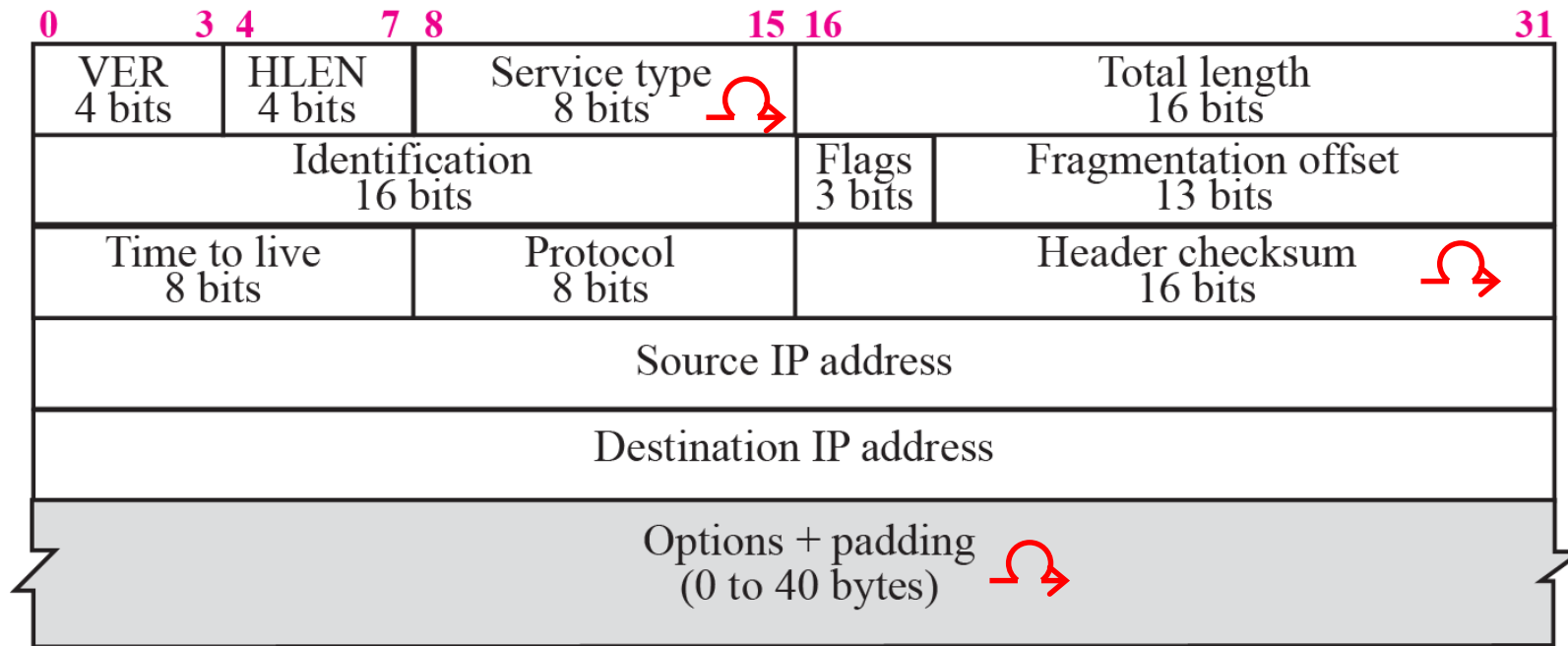
Flags (3 bits) (ip.flags), 1 byte

Packets: 199032 · Displayed: 199032 (100,0%) · Load time: 0:1,458 | Profile: Default

## 7.2 데이터그램 - 데이터그램의 형식



a. IP datagram



b. Header format

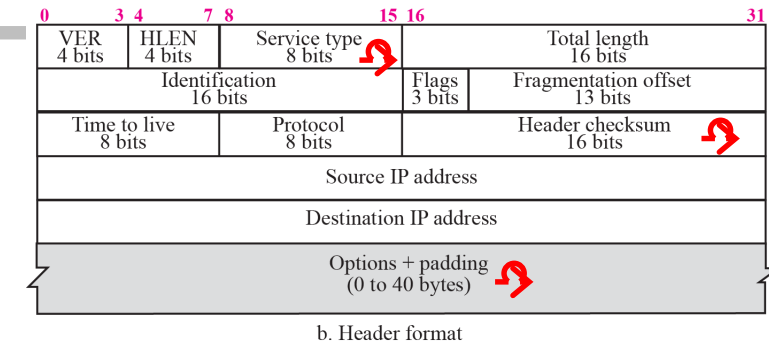
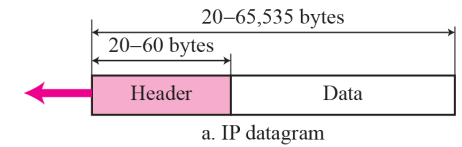
그림 7.2 IP 데이터그램

## 7.2 데이터그램 - 데이터그램의 형식

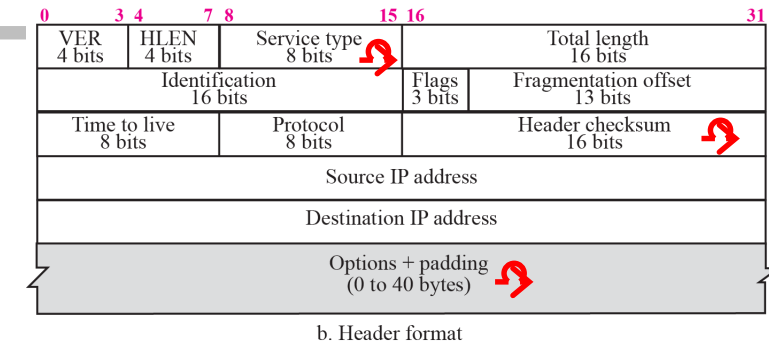
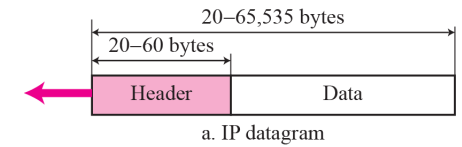
### ✓ IP 데이터그램 헤더 (header)의 필드 (fields)

#### ➤ 버전 (version (VER))

- IP 프로토콜의 **버전 (version)**을 나타냄
- 현재 버전은 4 (IPv4)임: 0100.
- 다음 버전은 6 (IPv6)임: 0110.
- 이 필드는 시스템 내에서 **수행되고 있는 IP 소프트웨어에게 데이터그램이 버전 4 또는 버전 6 형식으로 되어 있다는 것을 알려줌**
- 만약 시스템이 다른 버전의 IP 소프트웨어를 사용하고 있다면, **잘못 해석되지 않도록 이 데이터그램은 폐기 (discarded rather than interpreted incorrectly)됨**



## 7.2 데이터그램 - 데이터그램의 형식



✓ IP 데이터그램 헤더 (header)의 필드 (fields) - (계속)

### ➤ 헤더 길이 (header length (HLEN))

- 데이터그램 헤더 (header)의 전체 길이 (total length)를 4 바이트 단위 (4-byte words)로 나타냄
- $(2^4 - 1) \times 4 = 60 \rightarrow$  최대 길이는 60임
- 만약 선택사항이 없다면 (no options), 헤더의 길이는 20바이트이고, 이 필드의 값은 5가 됨
  - ❖  $HLEN = 20 / 4 = 5.$
- 만약 선택사항 필드가 최대 길이라면, 이 필드의 값은 15가 됨
  - ❖  $HLEN = 60 / 4 = 15.$

## 7.2 데이터그램 - 데이터그램의 형식

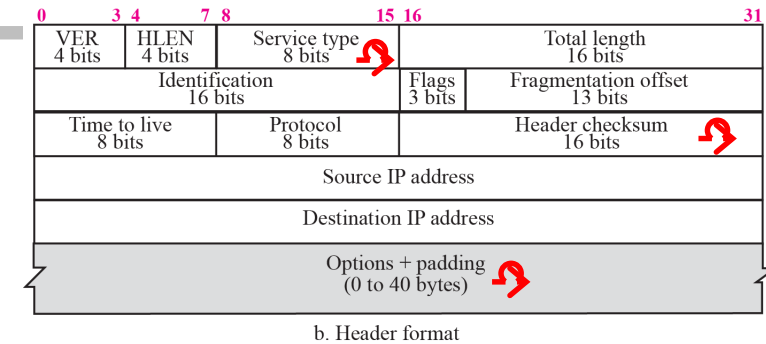
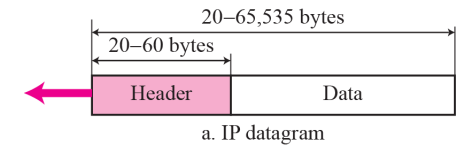
✓ IP 데이터그램 헤더 (header)의 필드 (fields) - (계속)

➤ 서비스 유형 (service type)

- 초기에 TOS (Type of Service)라고 불리었음
- 지금은 차별화 서비스 (differentiated services)의 집합을 정의함
- 생략 (SKIP)...

➤ 전체 길이 (total length)

- 헤더와 데이터를 포함 (header plus data)하는 IP 데이터그램의 전체 길이 (total length)를 바이트 단위 (in bytes)로 나타냄
- $2^{16} - 1 = 65,535 \rightarrow$  최대 길이는 65,535임
- 데이터의 길이 (length of data) = 전체 길이 (total length) - 헤더 길이 (header length)
  - ❖ 헤더 길이는 20바이트에서 60바이트 사이임 (HLEN 참고)

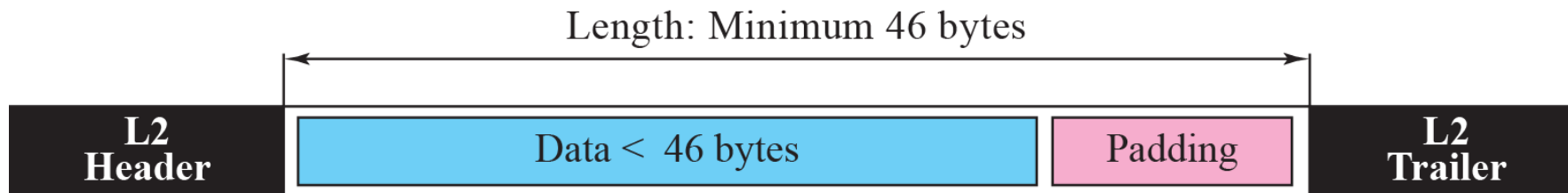
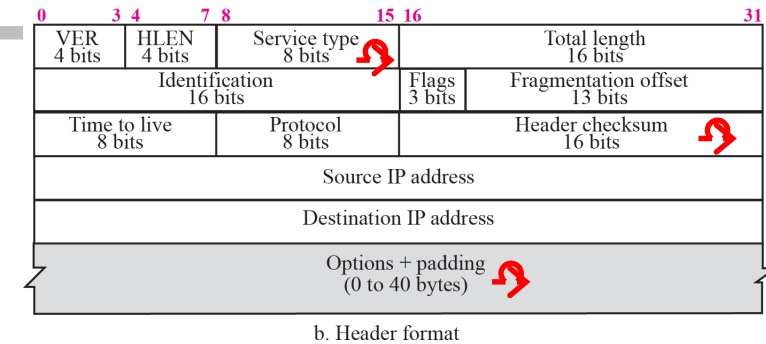
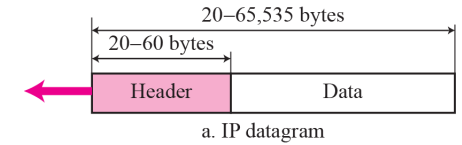


## 7.2 데이터그램 - 데이터그램의 형식

✓ IP 데이터그램 헤더 (header)의 **필드 (fields)** - (계속)

➤ 전체 길이 (total length) - (계속)

- 길이가 65,535 바이트라는 것은 크게 느껴질 수 있음
  - ❖ MTU 크기 참고 (in 7.3 fragmentation)
  - ❖ 그러나, 가까운 미래에 IP 데이터그램의 길이는 증가할 수 있을 것
- 왜 이러한 필드가 필요한가?
  - ❖ 라우터나 호스트와 같은 시스템이 프레임 (frame)을 받으면, 헤더 (header)와 트레일러 (trailer)는 없애고 데이터그램 (datagram)만 남기는데,
    - 왜 이러한 추가적인 필드 (extra field)를 필요로 하는가? (그림 7.4 참고)



**그림 7.4 이더넷 프레임에서 작은 데이터그램의 캡슐화  
(Encapsulation of a small datagram in an Ethernet frame)**

## 7.2 데이터그램 - 데이터그램의 형식

✓ IP 데이터그램 헤더 (header)의 **필드 (fields)** - (계속)

➤ 식별 (identification)

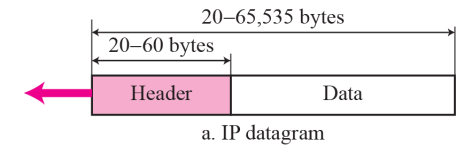
- 단편화 (fragmentation)와 관련된 필드 (7.3 단편화 참고)

➤ 플래그 (flag)

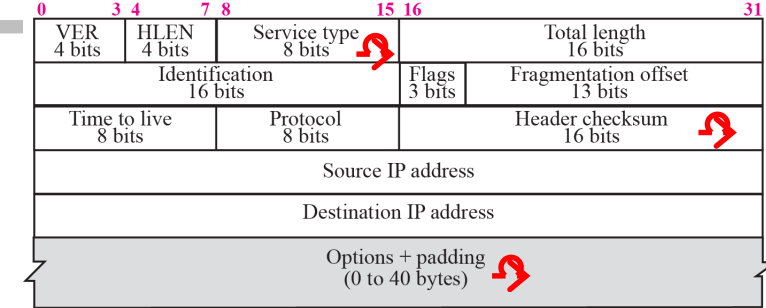
- 단편화 (fragmentation)와 관련된 필드 (7.3 단편화 참고)

➤ 단편화 오프셋 (fragmentation offset)

- 단편화 (fragmentation)와 관련된 필드 (7.3 단편화 참고)



a. IP datagram



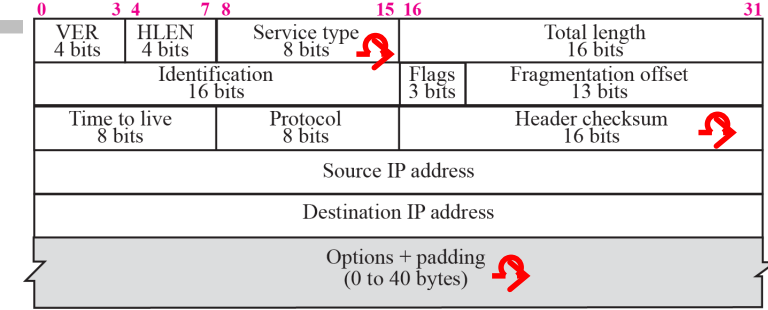
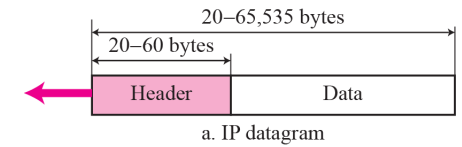
b. Header format

## 7.2 데이터그램 - 데이터그램의 형식

✓ IP 데이터그램 헤더 (header)의 필드 (fields) - (계속)

➤ 수명 (time to live)

- 이 필드가 왜 필요한가?
  - ❖ 데이터그램은, 목적지 호스트에 도달하지 못한 상태로, 오랜 기간 (long time) 동안 두개 이상의 라우터 (two or more routers)들 사이를 떠돌아 다닐 (travel) 수 있음
  - ❖ 이 필드는 데이터그램의 수명 (lifetime)을 제한
- 이 필드는 원래 (originally) 타임스탬프 (timestamp) 기능을 수행하기 위하여 설계됨
- 오늘날, 이 필드는 데이터그램에 의해 방문되는 홉 (라우터)의 최대 수 (maximum number of hops)를 제어 (control)하기 위하여 사용됨
  - ❖ 데이터그램을 처리하는 각 라우터는 이 필드의 값을 1씩 감소시킴
  - ❖ 감소시킨 다음, 이 필드 값이 0이면, 라우터는 데이터그램을 폐기 (discard)함
  - ❖ 이 필드의 초기 값은 대략 두 호스트 사이에 위치한 라우터 수의 두 배 (two times the maximum number of routers)임
- 발신지 (source)는 패킷이 전달되는 범위를 의도적으로 제한 (intentionally limit)할 수 있음
  - ❖ 이 값이 1이면, 데이터그램은 첫 번째 라우터에 도착한 후, 폐기 (discard)됨



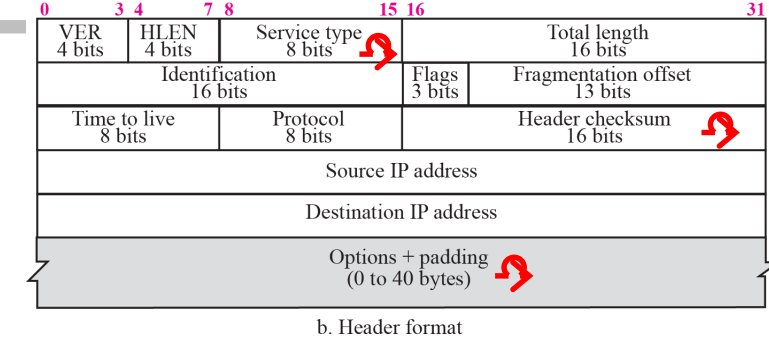
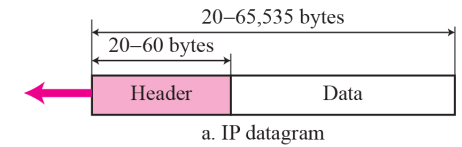


## 7.2 데이터그램 - 데이터그램의 형식

✓ IP 데이터그램 헤더 (header)의 **필드 (fields)** - (계속)

➤ **프로토콜 (protocol)**

- 이 필드는 IP 계층 (IP layer)의 서비스를 사용하는 상위 계층 프로토콜 (higher-level protocol)을 정의 (define)함
  - 이 필드는 IP 데이터그램이 **전달 (delivery)되어야 하는 최종 목적지 프로토콜 (final destination protocol)**을 나타냄
  - 이 필드는 왜 필요한가?
    - ❖ IP 프로토콜은 다른 종류의 상위 계층 프로토콜로부터 오는 데이터를 다중화 (multiplexing) 및 역다중화 (demultiplexing)하므로,
      - 최종 목적지에 도달한 경우, 이 필드의 값은 **역다중화 (demultiplexing) 과정을 도움**
- (다음 페이지 그림 7.5 참고)



## 7.2 데이터그램 - 데이터그램의 형식

✓ IP 데이터그램 헤더 (header)의 필드 (fields) - (계속)

➤ 프로토콜 (protocol) - (계속)

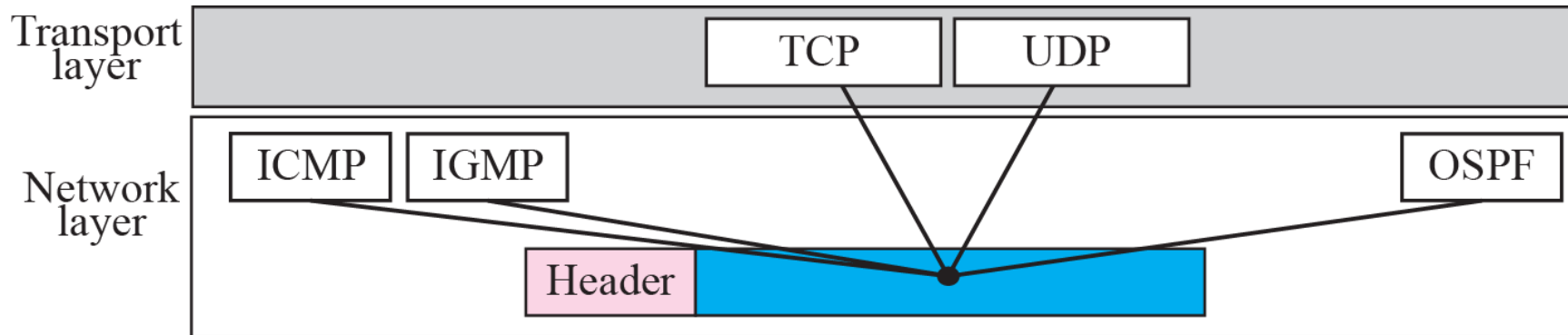
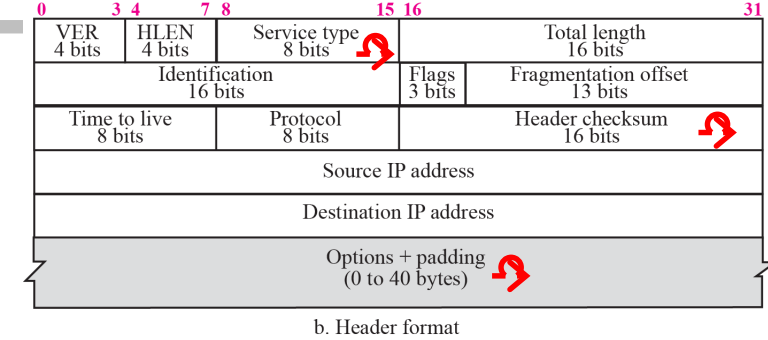
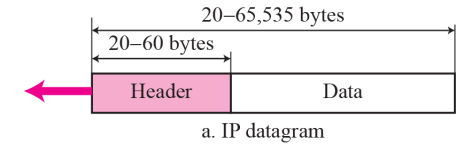


그림 7.5 다중화 (Multiplexing)

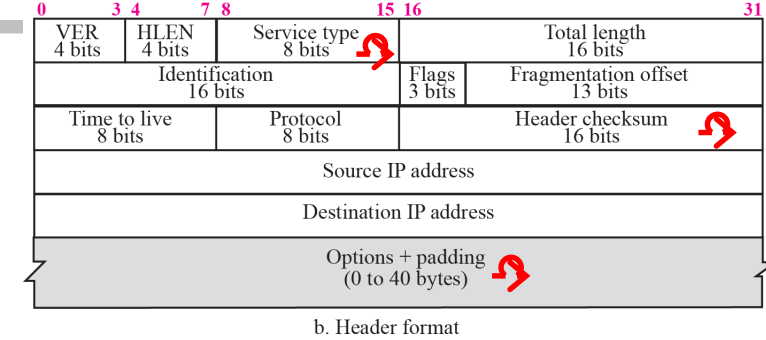
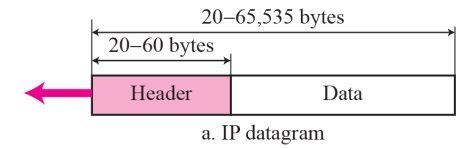
**Table 7.2** *Protocols*

<i>Value</i>	<i>Protocol</i>	<i>Value</i>	<i>Protocol</i>
1	ICMP	17	UDP
2	IGMP	89	OSPF
6	TCP		

## 7.2 데이터그램 - 데이터그램의 형식

✓ IP 데이터그램 헤더 (header)의 **필드 (fields)** - (계속)

- 검사합 (checksum)
  - 생략 (SKIP)...
- 발신지 주소 (source address)
  - 발신지 (source)의 IP 주소를 정의함
- 목적지 주소 (destination address)
  - 목적지 (destination)의 IP 주소를 정의함



## 7.2 데이터그램 – Example 7.1

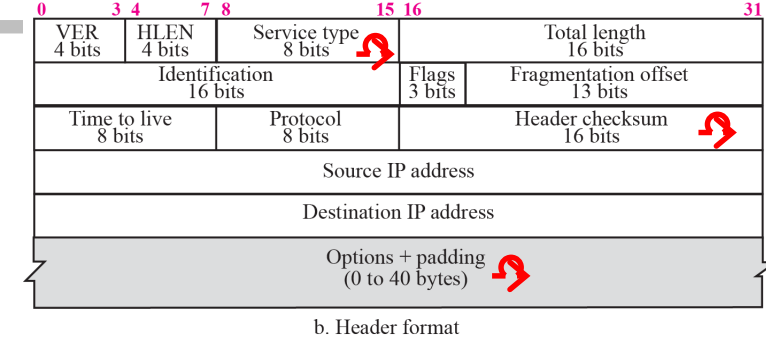
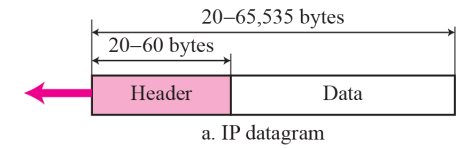
- ✓ 처음 8비트가 다음과 같은 IP 패킷이 도착하였다.

01000010

- ✓ 수신자 (receiver)는 이 패킷을 폐기 (discard)한다. 이유를 설명하라.

### 해답 (Solution)

- ✓ 패킷에 오류가 있다.
- ✓ 왼쪽 4비트 (4 left-most bits) (0100)는 버전 (version)으로, 이 값은 올바르다 (correct).
- ✓ 다음 4비트 (next 4 bits) (0010)는 헤더의 길이인데,  $2 \times 4 = 8$ 이 되므로 틀리다 (wrong).
- ✓ 헤더의 최소 바이트 수는 20이다.
- ✓ 패킷은 전송 (transmission) 도중에 훼손 (corrupted)되었다.

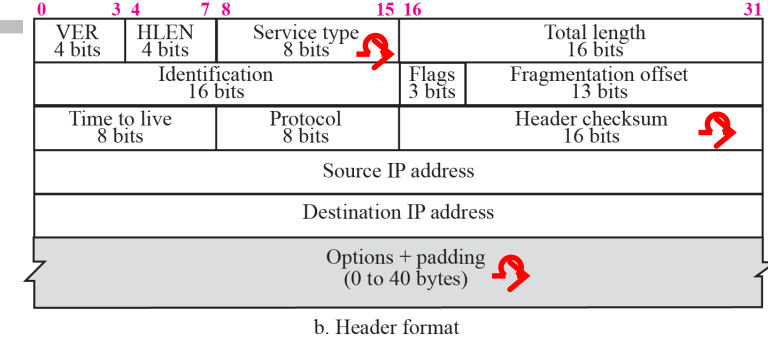
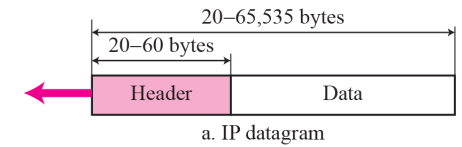


## 7.2 데이터그램 – Example 7.2

- ✓ IP 패킷에서, HLEN 값이 이진수로 **1000**이다.
- ✓ 이 패킷에는 옵션 (**options**)이 몇 바이트 있는가?

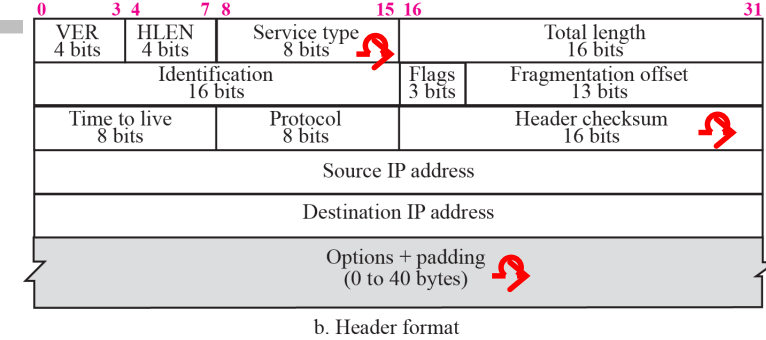
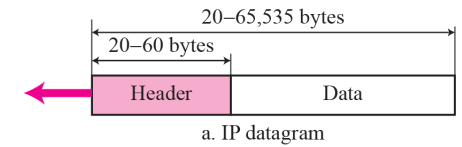
### 해답 (Solution)

- ✓ HLEN의 값은 **8**이고, 이것은 헤더 내 바이트의 총 수는  **$8 \times 4$  or 32** 바이트임을 알려준다.
- ✓ 처음 **20바이트**는 주 헤더이므로, 다음 **12바이트**가 옵션이 된다.



## 7.2 데이터그램 – Example 7.3

- ✓ IP 패킷에서 HLEN의 값이  $5_{16}$ 이고,  
전체 길이 필드의 값이  $0028_{16}$ 이다.
- ✓ 이 패킷은 몇 바이트의 데이터를 전송하고 있는가?



### 해답 (Solution)

- ✓ HLEN 값이 5 이므로, 헤더는  $5 \times 4$  or 20바이트이고, 옵션은 없음 (no options)을 알 수 있다.
- ✓ 전체 길이는 40바이트이므로,  $40 - 20 = 20$ 바이트의 데이터가 전송되고 있다.

## 7.2 데이터그램 – Example 7.4

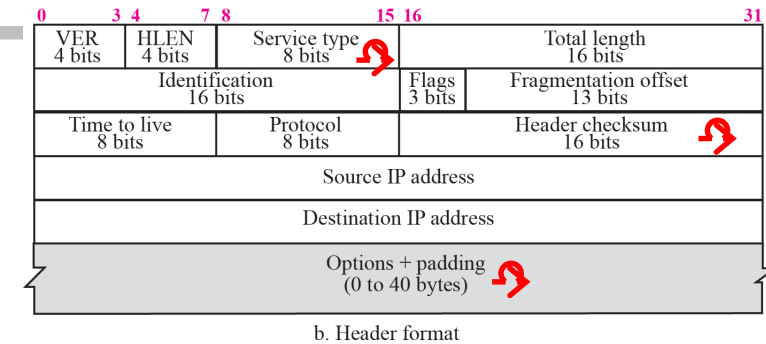
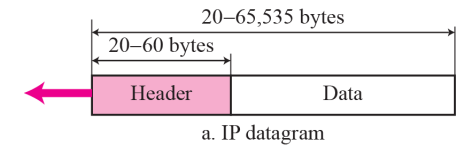
- ✓ 처음 16진수 숫자들이 다음과 같은 패킷이 도착하였다.

45000028000100000102 ...

- ✓ 이 패킷은 폐기 (drop)되기 전에 몇 홉 (hop)을 지나갈 수 있는가?
- ✓ 이 패킷은 어떤 상위 계층 프로토콜 (upper layer protocol)의 데이터를 전송하고 있는가?

### 해답 (Solution)

- ✓ 수명 (time-to-live) 필드를 찾기 위해, 처음 8바이트 (16개의 16진법 숫자)를 건너뛰다 (skip).
- ✓ 수명 (time-to-live) 필드는 9번째 바이트이고, 값은 01이다.
- ✓ 이것은 패킷이 한 홉만 (only one hop) 갈 수 있다는 것을 의미한다.
- ✓ 프로토콜 (protocol) 필드는 다음 바이트이며, 값은 02이고, 이것은 상위 계층 프로토콜이 IGMP라는 것을 의미한다 (Table 7.2 참고)



## 7.3 단편화

---

- ✓ 데이터그램은 **다른 네트워크를 지나갈 (travel through different networks)** 수 있음
- ✓ 각 라우터는 수신한 프레임에서 **IP 데이터그램을 역캡슐화 (decapsulate)**한 후, **처리 (process)**하고, 다른 프레임 (frame) 속에 **캡슐화 (encapsulate)**함
- ✓ 수신한 프레임 (received frame)의 **형식 (format)**과 **크기 (size)**는 프레임이 방금 **지나온 (travel) 네트워크**가 사용하는 프로토콜에 의존함
- ✓ 송신할 프레임 (sent frame)의 **형식 (format)**과 **크기 (size)**는 프레임이 바로 다음에 **지나갈 (travel) 네트워크**가 사용하는 프로토콜에 의존함



## 7.3 단편화 - Topics

---

- 1) 최대 전달 단위 (MTU: Maximum Transfer Unit)
- 2) 단편화와 관련된 필드 (Fields Related to Fragmentation)

## 7.3 단편화 – MTU

### ✓ 최대 전달 단위 (MTU: Maximum Transfer Unit)

- 각 데이터링크 계층 프로토콜은 **자신의 프레임 형식 (its own frame format)**을 가짐
- **MTU**는 프레임 (frame) 내 **데이터 필드 (data field)**의 **최대 크기 (maximum size)**를 의미함
- 데이터그램이 프레임에 캡슐화 (encapsulate)될 때, 데이터그램의 전체 크기 (total size)는 **이 최대 크기보다 작아야 함**



**Header**

**MTU**  
Maximum length of data that can be encapsulated in a frame

**Trailer**

Frame

**그림 7.6 MTU**

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

## 7.3 단편화 – MTU

---

### ✓ 최대 전달 단위 (MTU: Maximum Transfer Unit) – (계속)

- 데이터그램의 길이가 물리 네트워크 (physical network)의 MTU보다 큰 경우, 전송을 위해 데이터그램을 나누어야 (divide) 함
  - ➔ 단편화 (Fragmentation)
- 누가 단편화 (fragment)하는가?
  - 데이터그램은 **발신지 호스트 (source host)**나 경로 상 (in the path)의 임의의 **라우터 (router)**에 의해서도 단편화될 수 있음
- 누가 재조립 (reassemble)하는가?
  - 각 단편 (fragment)이 독립된 데이터그램이므로, **데이터그램의 재조립 (reassembly of the datagram)**은 오직 **목적지 호스트 (destination host)**에 의해서만 행해짐

## 7.3 단편화 - 단편화와 관련된 필드

### ✓ 단편화와 관련된 필드 (Fields related to fragmentation)

➤ 데이터그램이 단편화 (fragmented)될 때,

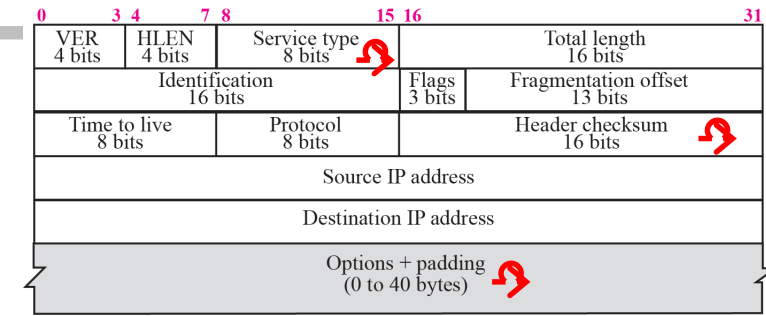
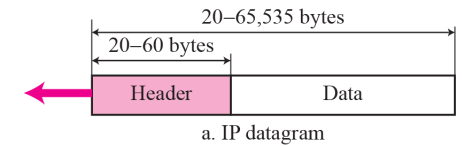
헤더의 특정 부분은 모든 단편 (fragments)에 복사 (copy)되어야 함

➤ 데이터그램을 단편화하는 호스트나 라우터에 의하여 변경 (change)되어야 하는 세 개의 필드 (field)

- 플래그 (flag)
- 단편화 오프셋 (fragmentation offset)
- 전체 길이 (total length)

➤ 단편화와 관련은 있지만, 변경되지 않는 하나의 필드 (field)

- 식별 (Identification)

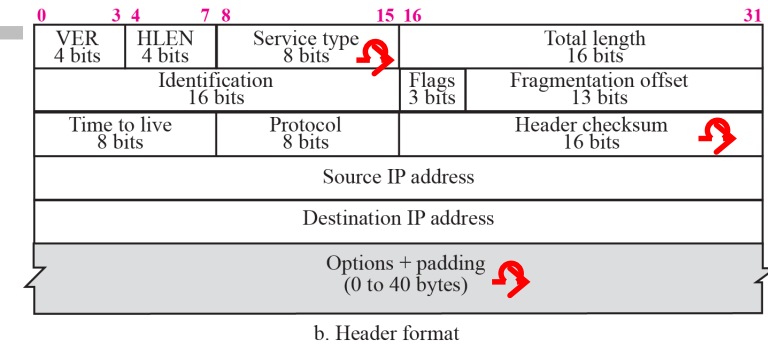
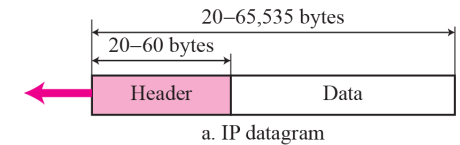


## 7.3 단편화 - 단편화와 관련된 필드

### ✓ 단편화와 관련된 필드 (Fields related to fragmentation) - (계속)

#### ➤ 식별 (Identification)

- 이 필드는 발신지 호스트 (source host)로부터 전송된 데이터그램을 유일하게 식별 (identify)함
- 식별 (identification)와 발신지 IP 주소 (source IP address)의 조합 (combination)은 데이터그램이 발신지 호스트 (source host)를 떠날 때, 유일하게 정의 (uniquely define)되어야 함
- 이러한 유일성을 보장 (guarantee uniqueness)하기 위하여, IP 프로토콜은 카운터 (counter)를 사용하여 데이터그램에 레이블을 붙임
  - ❖ IP 프로토콜이 데이터그램을 보낼 때, 카운터의 현재 값을 식별 필드에 복사한 뒤, 카운터 값을 1 증가시킴
- 데이터그램이 단편화될 때, 식별 필드의 값은 모든 단편에 복사됨
- 식별 필드는 목적지에서 데이터그램을 재조립하는 것에 도움이 됨
  - (같은 식별 값을 가지는 모든 단편은 하나의 데이터그램으로 재조립됨)



## 7.3 단편화 - 단편화와 관련된 필드

✓ 단편화와 관련된 필드 (Fields related to fragmentation) - (계속)

✓ 플래그 (Flag)

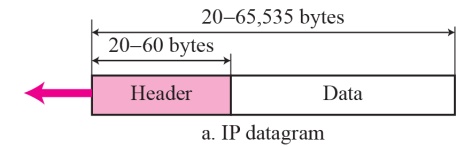
✓ 3비트의 필드

D: Do not fragment  
M: More fragments

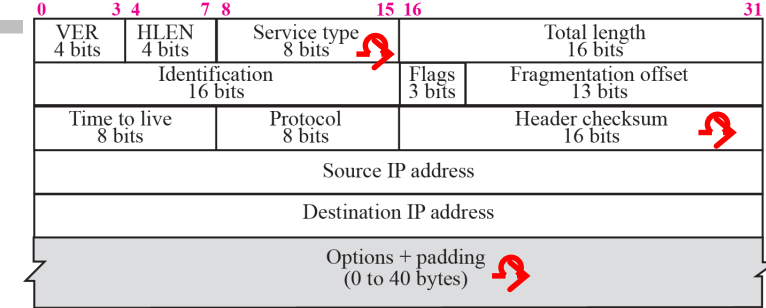


✓ 데이터그램의 길이가 MTU보다 크지만, 플래그 값이 "do not fragment"이면,

→ IP 프로토콜은 데이터그램을 폐기 (discard)하고, 발신지 호스트 (source host)에게 ICMP 오류 메시지 (ICMP error message)를 전송 (send)함 (9장 참조)



a. IP datagram



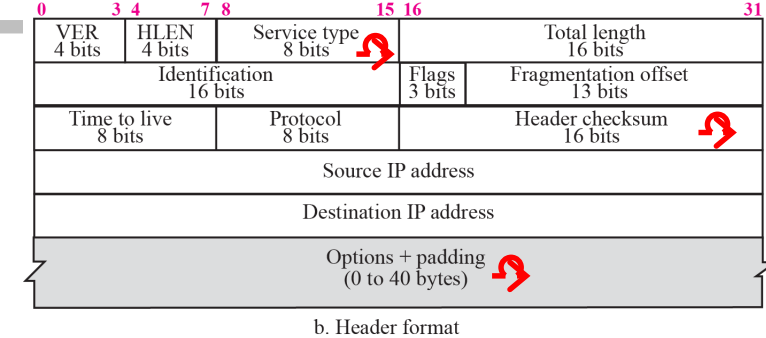
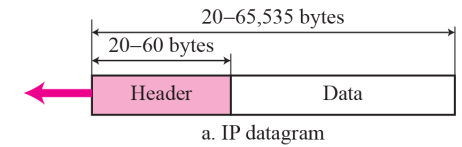
b. Header format

## 7.3 단편화 – 단편화와 관련된 필드

### ✓ 단편화와 관련된 필드 (Fields related to fragmentation) - (계속)

#### ➤ 단편화 오프셋 (Fragmentation Offset)

- 이 필드는 원 데이터그램 (original datagram) 내에서 데이터의 오프셋 (offset)을 8 바이트 단위 (units of 8 bytes)로 나타낸 것임
  - 첫 번째 단편 (first fragment)의 오프셋 (offset)은  $0/8 = 0$ 임.
  - 두 번째 단편 (second fragment)의 오프셋 (offset)은?
    - ❖ 첫 번째 단편에 0부터 1399번까지의 바이트가 있으면,
    - ❖  $1400/8 = 175$
    - 두 번째 단편의 오프셋 = 175.
  - $i$ 번째 단편 ( $i^{th}$  fragment)의 오프셋 (offset)은?
- (다음 페이지)



## 7.3 단편화 - 단편화와 관련된 필드

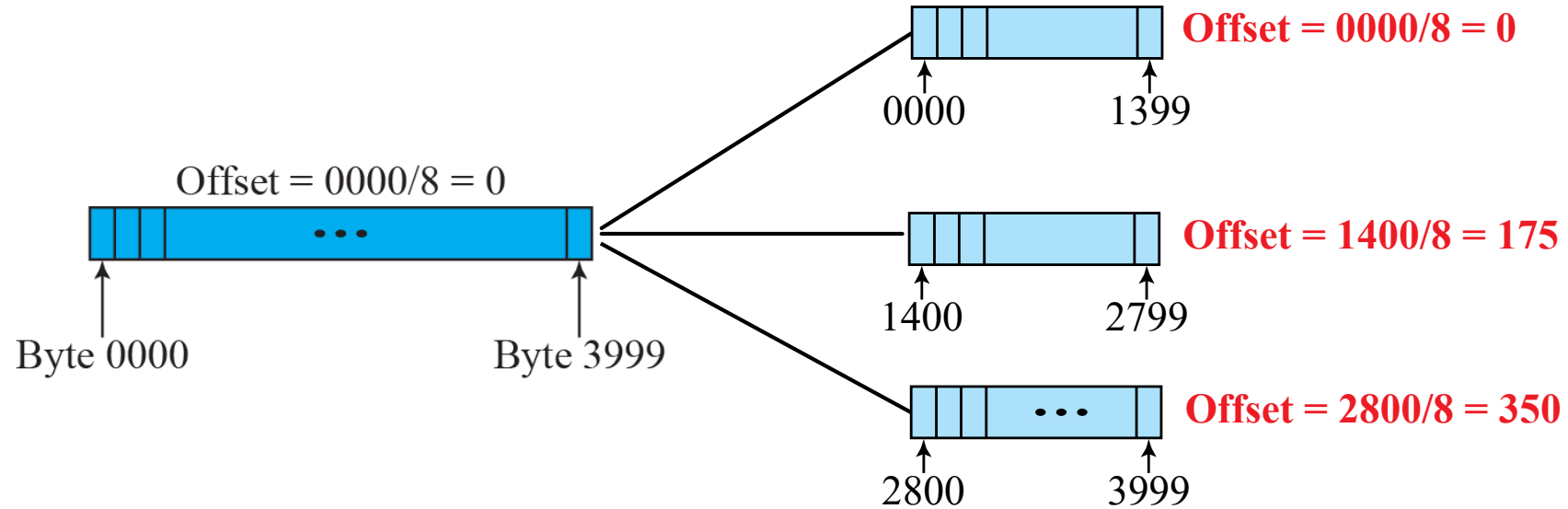
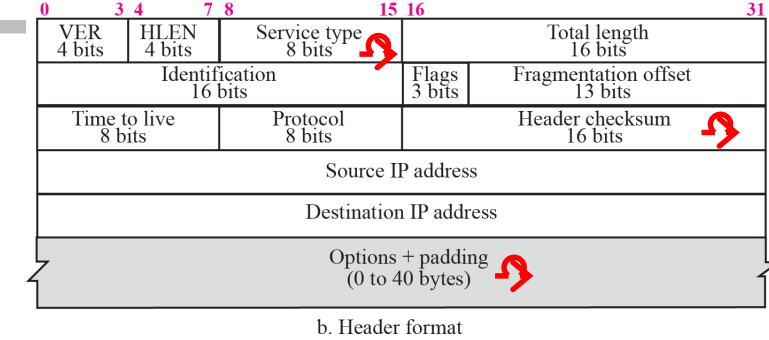
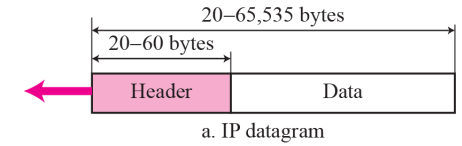


그림 7.8 단편화 예 (Fragmentation example)



## 7.3 단편화 - 단편화와 관련된 필드

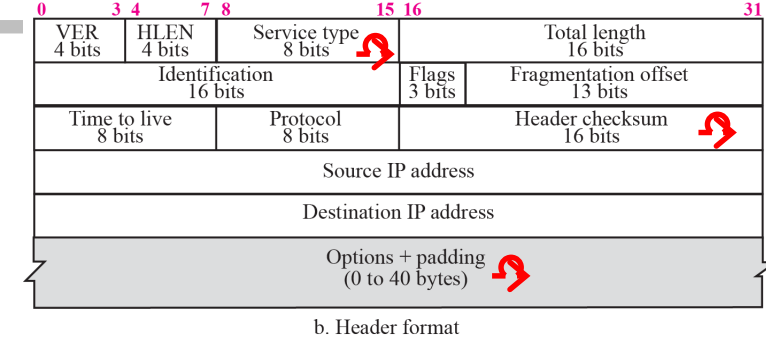
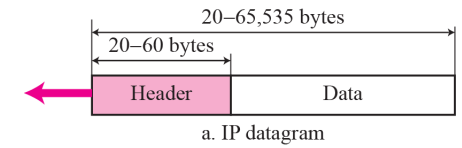
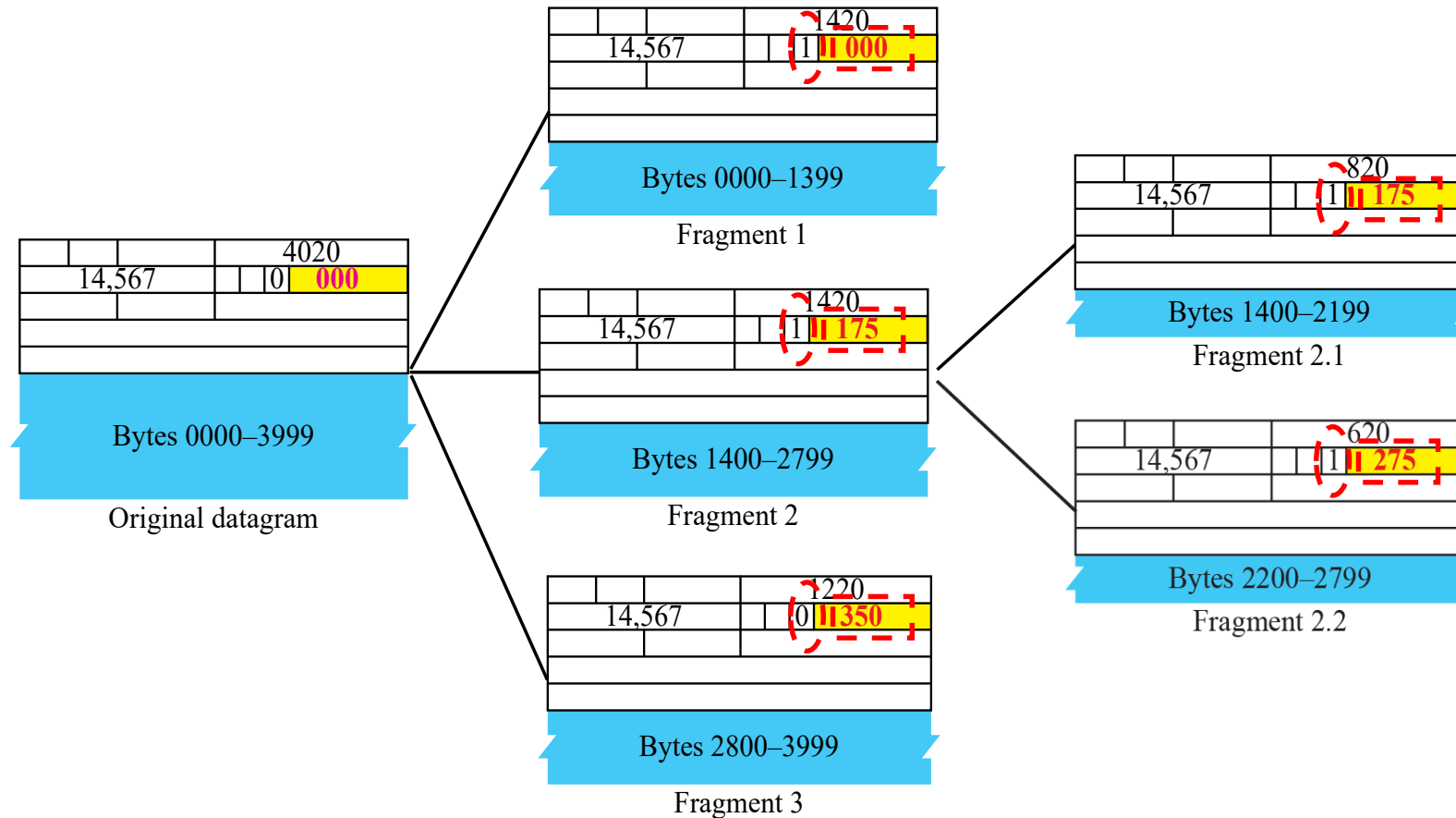


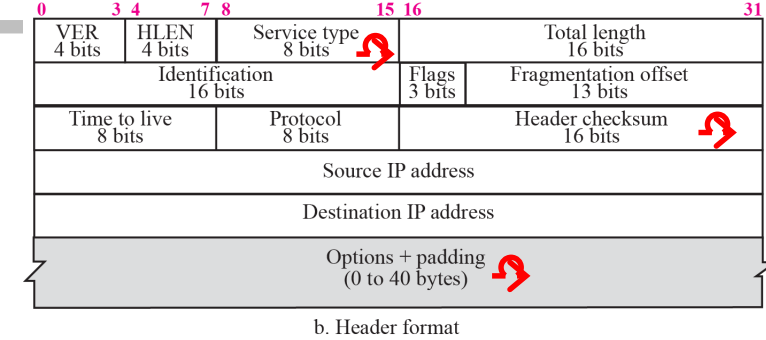
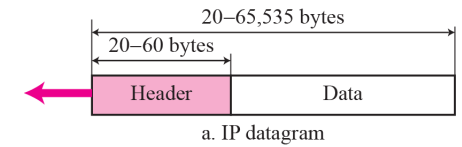
그림 7.9 상세한 단편화 예 (Detailed fragmentation example)

## 7.3 단편화 – Example 7.5

- ✓ M비트 값이 0인 패킷이 도착 (arrive)하였다.
- ✓ 이것은 첫 번째 (first) 단편인가? 마지막 (last) 단편인가?  
또는 중간 (middle) 단편인가?
- ✓ 패킷이 단편화되었는지 알 수 있는가?

### 해답 (Solution)

- ✓ M 비트가 0이면 단편이 더 이상 없다는 것이다. 따라서, 단편은 **마지막 (last)** 단편이다.
- ✓ 그러나, 원래의 패킷이 **단편화**되었는지는 **알 수 없다**.
- ✓ 단편화되지 않은 패킷 (non-fragmented packet)은 **마지막 (last)** 단편으로 고려된다.



D: Do not fragment  
M: More fragments

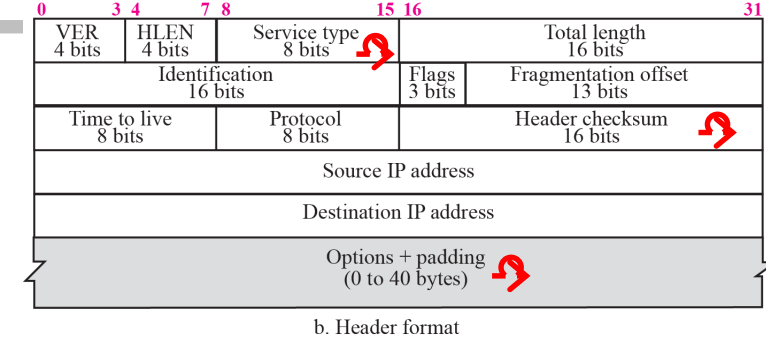
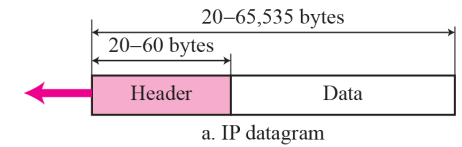


## 7.3 단편화 – Example 7.6

- ✓ M비트 값이 1인 패킷이 도착 (arrive)하였다.
- ✓ 이것은 첫 번째 (first) 단편인가? 마지막 (last) 단편인가?  
또는 중간 (middle) 단편인가?
- ✓ 패킷이 단편화되었는지 알 수 있는가?

### 해답 (Solution)

- ✓ M 비트가 1이면, 적어도 한 개의 단편 (at least one more fragment)이 더 있다는 것을 의미한다.
- ✓ 이 패킷은 첫 번째 (first) 또는 중간 (middle) 단편이지만, 마지막 (last) 단편은 아니다.
- ✓ 첫 번째 (first)인지 또는 중간 (middle) 단편인지는 알 수 없고, 이를 알기 위해서는, 단편화 오프셋 (fragmentation offset)과 같은 추가 정보가 필요하다.
- ✓ 다음 예제를 참조하라.



D: Do not fragment  
M: More fragments

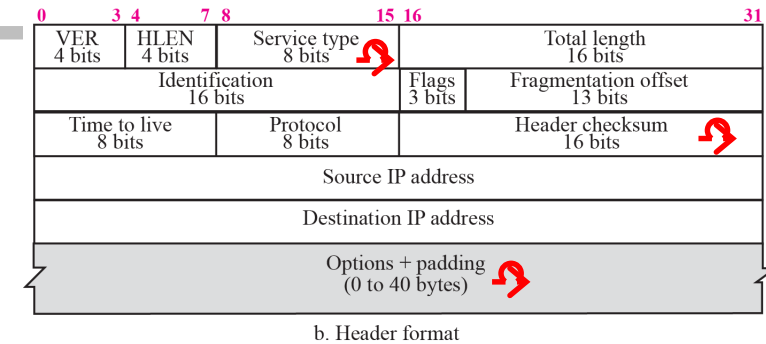
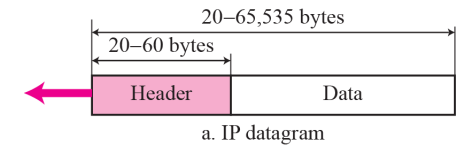


## 7.3 단편화 – Example 7.7

- ✓ M 비트가 1이고 단편화 옵션이 0인 패킷이 도착 (arrive) 하였다.
- ✓ 이것은 첫 번째 (first) 단편인가? 마지막 (last) 단편인가?  
또는 중간 (middle) 단편인가?

### 해답 (Solution)

- ✓ M 비트가 1이므로, 첫 번째 (first) 또는 중간 (middle) 단편이지만,
- ✓ 옵션이 0이므로, 첫 번째 (first)이다.



D: Do not fragment  
M: More fragments

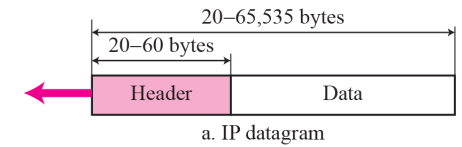


## 7.3 단편화 – Example 7.8

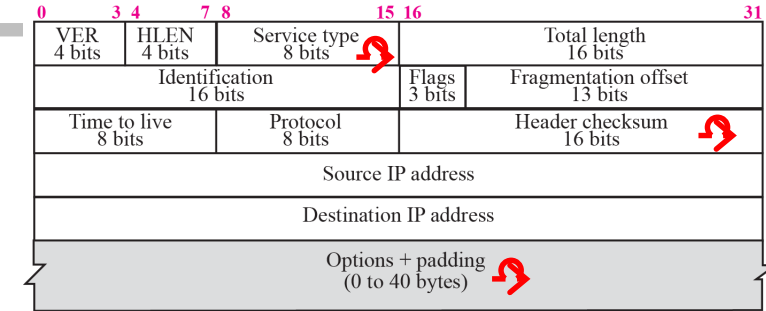
- ✓ 오프셋 값 (offset value)이 100인 패킷이 도착 (arrive) 하였다.
- ✓ 첫 번째 바이트 (first byte)의 번호는 무엇인가?
- ✓ 마지막 바이트 (last byte)의 번호를 알 수 있는가?

### 해답 (Solution)

- ✓ 첫 번째 바이트의 번호를 구하기 위해, 오프셋 값에 8을 곱한다.
- ✓ 그 결과, 첫 번째 바이트의 번호는 **800**임을 알 수 있다.
- ✓ 데이터의 길이를 모르면 마지막 바이트의 번호를 **알 수 없다**.



a. IP datagram



b. Header format

D: Do not fragment  
M: More fragments

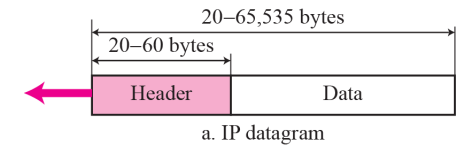


## 7.3 단편화 – Example 7.9

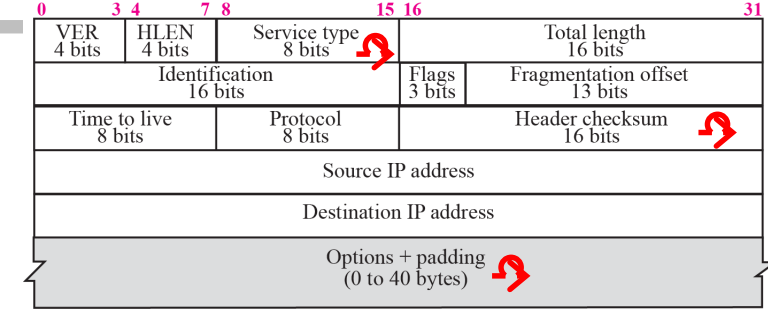
- ✓ **오프셋 값 (offset value)이 100**이고, **HLEN이 5**이며,  
**전체 길이 (total length)가 100인 패킷이 도착 (arrive) 하였다.**
- ✓ What is the number of the first byte and the last byte?
- ✓ **첫 번째 바이트 (first byte)와 마지막 바이트 (last byte)의 번호를 구하라.**

### 해답 (Solution)

- ✓ 첫 번째 바이트 번호는  $100 \times 8 = 800$ 이다.
- ✓ 전체 길이는 **100 바이트**이고, 헤더 길이는 **20 바이트 ( $5 \times 4$ )**이므로, 이 **데이터그램 내에는 80바이트의 데이터**가 있다.
- ✓ 첫 번째 바이트 번호는 **800**이고, 마지막 바이트 번호는 **879**이다.



a. IP datagram



b. Header format

D: Do not fragment  
M: More fragments



## 7.8 IP 패키지

---

- ✓ 가상 IP 패키지 (hypothetical IP package)의 단순화된 예 (simplified example)를 살펴봄
- ✓ 목적은 본 장에서 설명한 다른 개념들 (different concepts) 사이의 관계 (relationships)를 살펴보기 위한 것임

## 7.8 IP 패키지

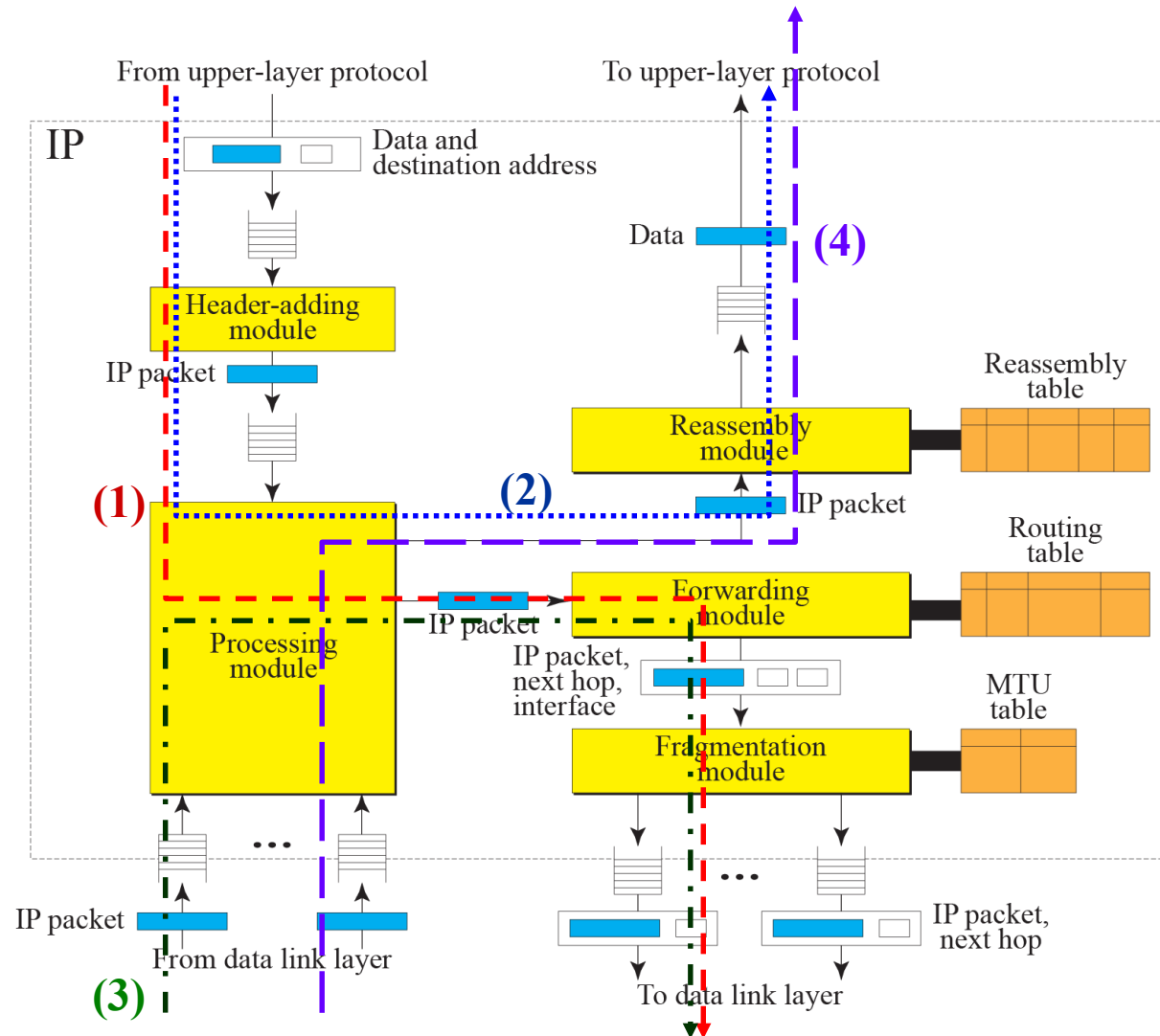


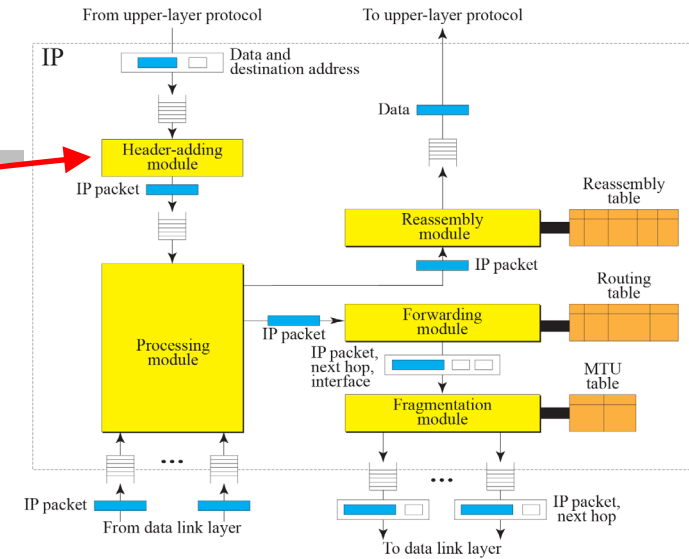
그림 7.29 IP의 구성요소 (IP components)



## 7.8 IP 패키지

### ✓ 헤더 추가 모듈 (Header-Adding Module)

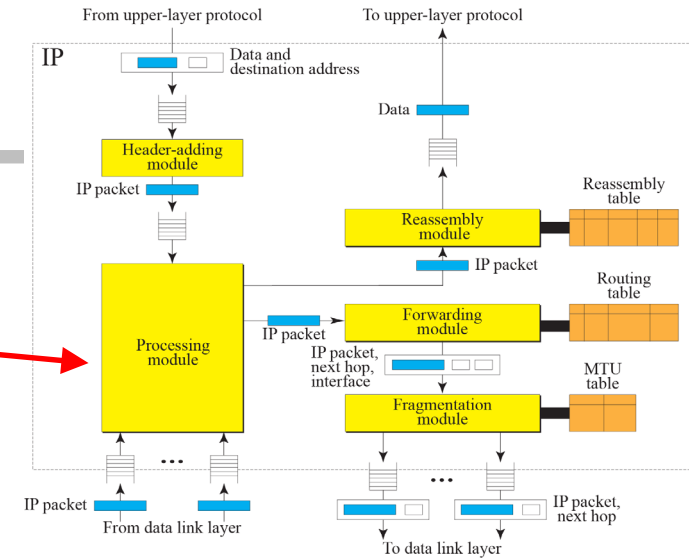
- 상위 계층 프로토콜 (upper-layer protocol)로부터,  
데이터 (data)와 목적지 IP 주소 (destination IP address)를 수신 (receive)함
- 그 후, IP 헤더 (IP header)를 추가함 (adding)으로써,  
데이터를 IP 데이터그램 내에 캡슐화 (encapsulate)함



## 7.8 IP 패키지

### ✓ 처리 모듈 (Processing Module)

- IP 패키지의 **핵심 (heart)**임
- 처리 모듈은 인터페이스나 헤더 추가 모듈 (header-adding module)로부터 **데이터그램을 수신 (receive)**함
- 데이터그램이 최종 목적지 (final destination)에 도착하였다면, 패킷은 **재조립 모듈 (reassemble module)**로 보내짐
- 만약 노드라 라우터라면, 수명 (TTL: time-to-live) 필드의 값을 **1 감소**시킴
- 만일 TTL 값이 **0 이하**라면, 데이터그램은 **폐기 (discard)**되고 이 패킷의 원래 송신자에게 ICMP 오류 메시지가 보내짐 (9장 참조)
- 감소한 후에도 TTL 값이 **0보다 크면**, 처리 모듈은 데이터그램을 **포워딩 모듈 (forwarding module)**에 **전송 (send)**함



## 7.8 IP 패키지

### ✓ 라우팅 테이블 (Routing Table)

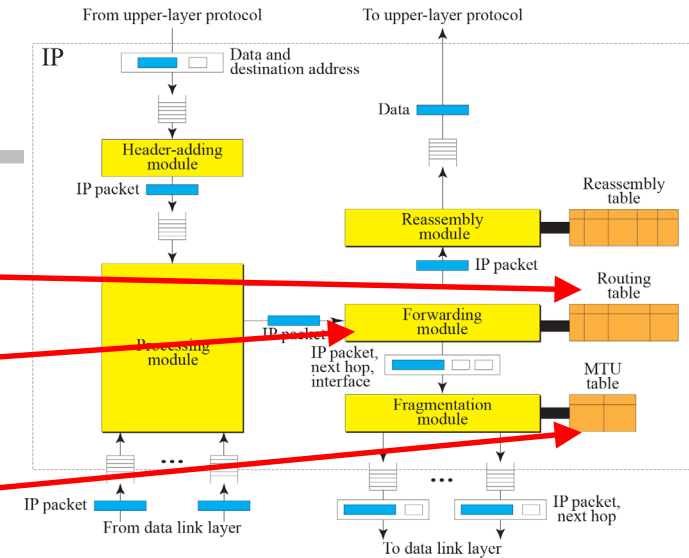
- 이미 배운 내용

### ✓ 포워딩 모듈 (Forwarding Module)

- 이미 배운 내용

### ✓ MTU 테이블 (MTU Table)

- 이 테이블은 단편화 모듈이 특정 인터페이스의 MTU를 찾기 위해 사용됨

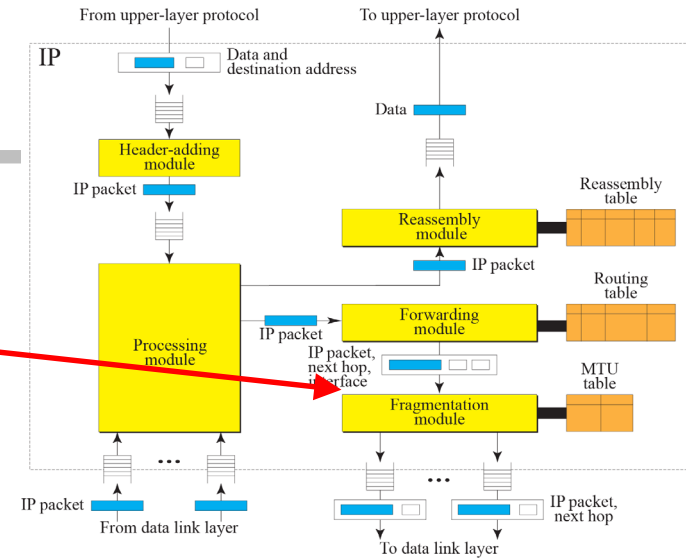


Interface Number	MTU
.....	.....
.....	.....
.....	.....

## 7.8 IP 패키지

### ✓ 단편화 모듈 (Fragmentation Module)

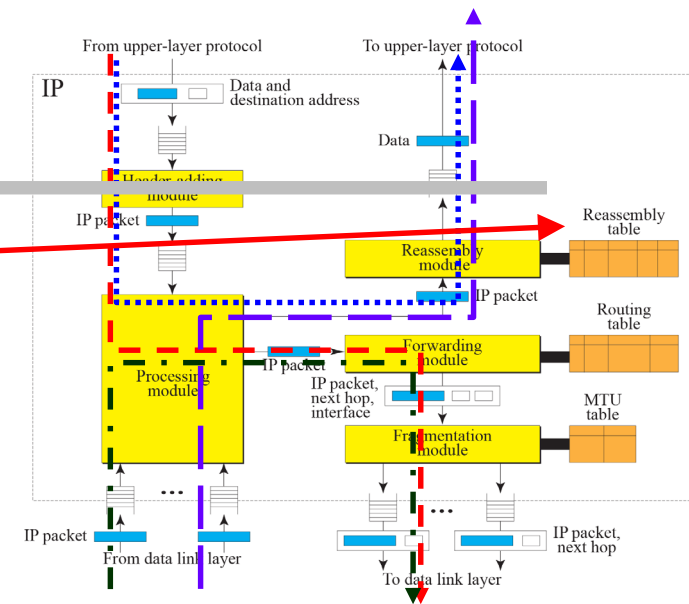
- 포워딩 모듈로부터 다음의 것들을 수신 (receive)함
  - IP 데이터그램 (IP datagram)
  - 다음 노드의 IP 주소 (IP address of the next station)
    - ❖ 직접 전달 (direct delivery)의 경우, 최종 목적지 (final destination)의 IP 주소
    - ❖ 간접 전달 (indirect delivery)의 경우, 다음 라우터 (next router)의 IP 주소
  - 데이터그램이 **내보내져야 (sent out)** 하는 인터페이스 번호 (interface number)
- 만약 데이터그램의 길이가 **MTU보다 더 크다면**, 단편화 모듈은 데이터그램을 **단편화 (fragment)**함
- 그 다음, 주소 해석 (address resolution)과 전달 (delivery)을 위하여 **ARP 패키지 (8장 참조)에게 데이터그램 (또는 여러 단편들)을 전송 (send)**



## 7.8 IP 패키지

### ✓ 재조립 테이블 (Reassembly Table)

- 재조립 모듈 (reassembly module)에 의하여 사용됨
- 미리 결정된 시간 (predetermined amount of time) 내에 도착한 (arrive) 모든 단편 (all fragments)들을 관리함



St.: State  
 S. A.: Source address  
 D. I.: Datagram ID  
 T. O.: Time-out  
 F.: Fragments

St.	S. A.	D. I.	T. O.	F.
				→ [blue box] [pink box] → [blue box] [pink box] → [blue box] [pink box]
		...		
				→ [blue box] [pink box] → [blue box] [pink box] → [blue box] [pink box]

## 7.8 IP 패키지

### ✓ 재조립 모듈 (Reassembly Module)

- 처리 모듈 (processing module)로부터  
최종 목적지 (final destination)에 도착한 (arrive)  
데이터그램 단편들 (fragments)을 수신 (receive)함
- 단편화되지 않은 데이터그램 (unfragmented datagram)도  
하나의 단편만을 가진 데이터그램의 단편으로 취급함
- 단편들이 **순서대로 도착한다는 보장이 없기 (no guarantee)** 때문에, 재조립 테이블에서는 **연결 리스트 (linked lists)**가 사용됨
- 만약 정의된 **타임아웃 (time-out)**이 발생하였는데도 아직 도착하지 않은 단편이 있으면, 재조립 모듈은 **모든 단편들을 폐기 (discard)**함

