

제 7 장 IPv4

- 7.1 소개 (Introduction)
- 7.2 데이터그램 (Datagrams)
- 7.3 단편화 (Fragmentation)
- 7.4 옵션 (Options)
- 7.5 검사합 (Checksum)
- 7.6 ATM 상의 IP (IP over ATM)
- 7.7 보안 (Security)
- 7.8 IP Package

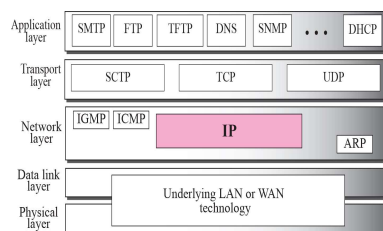
TCP/IP

7-1

7.1 소개 (Introduction)

■ IP 프로토콜

- ➔ TCP/IP에서 사용하는 전송 메커니즘
- ➔ _____ 없는 _____ 프로토콜
- ➔ 최선노력(_____) 전달 서비스
- ➔ 오류 검사 및 추적 기능을 수행하지 않음
- ➔ 각 데이터그램은 독립적으로 처리
- ➔ 각 데이터그램은 서로 다른 경로로 전달될 수 있음
- ➔ 각 데이터그램은 순서가 바뀌어 전달될 수 있음

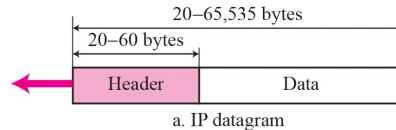


TCP/IP

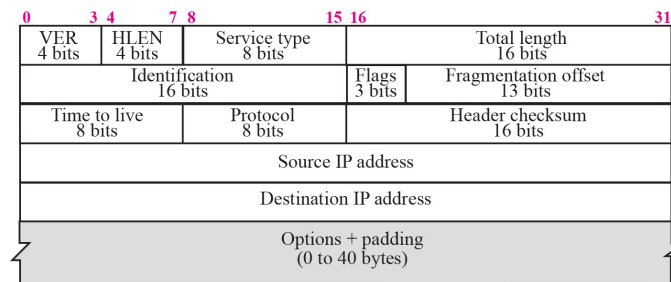
7-2

7.2 데이터그램 (Datagram)

■ IP 데이터그램 : 네트워크 계층의 패킷



a. IP datagram



b. Header format

TCP/IP

7-3

7.2 데이터그램 (Datagram)

■ IP 데이터그램

- ➡ 가변 길이 패킷
- ➡ 헤더와 데이터 부분으로 구성
- ➡ 헤더 : 20 ~ 60 바이트 길이, 라우팅과 전달에 필요한 정보
- ➡ 헤더 내의 필드
 - ◆ 버전(Ver) - IP 프로토콜 버전(4비트)
 - ◆ 헤더길이(HLEN) - 헤더의 전체 길이를 4 바이트 단위로 표시(4 비트)

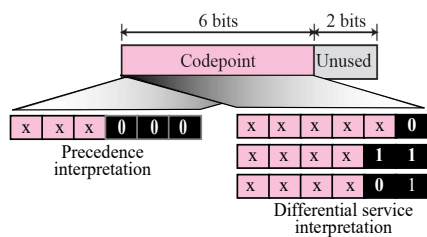
TCP/IP

7-4

IP 데이터그램(계속)

■ 서비스 유형 (Service type) : 8 bits

- ➡ DS (Differentiated Services) cf. 과거 : TOS (Type of Services)
- ➡ 라우터에 의해 처리하는 방법
- ➡ Codepoint : 우선순위 (Precedence) 또는 차별화서비스(DS)



Category	Codepoint	Assigning Authority	
1	XXXXX0	Internet (IETF)	(0, 2, 4, ..., 62)
2	XXXX11	Local	(3, 7, 11, ..., 63)
3	XXXX01	Temporary or experimental	(1, 5, 9, ..., 61)

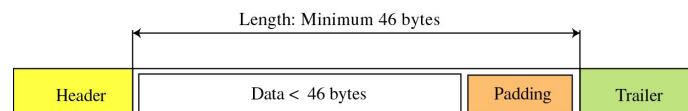
TCP/IP

7-5

IP 데이터그램(계속)

■ 전체 길이(total length)

- ➡ 16 bit 필드(65,535 까지 표현)
- ➡ IP 데이터그램의 전체 길이를 바이트 단위로 표시
- ➡ 데이터 길이 = 전체 길이 - 헤더 길이
- ➡ 이더넷 프레임에서 작은 데이터그램의 캡슐화
 - ◆ Data와 Padding 구분위해 필요



TCP/IP

7-6

IP 데이터그램(계속)

- ID (Identification) – 단편화에 사용
- 플래그(flag) – 단편화에 사용
- 단편 오프셋(fragmentation offset) – 단편화에 사용
- 수명(time to live) – 데이터그램의 수명
- 프로토콜(protocol) – IP 계층의 서비스를 사용하는 상위 계층 프로토콜 구분 cf. 그림 7.5, 표 7.2 참조
- 검사합(checksum) – 오류 확인
- 발신지 주소(source address)
- 목적지 주소(destination address)

TCP/IP

7-7

7.3 단편화 (Fragmentation)

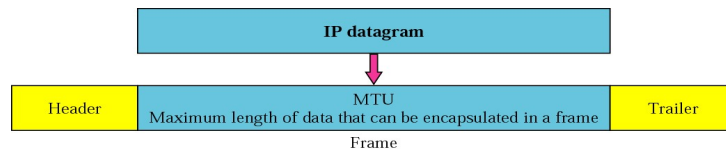
- ➡ 물리적인 네트워크가 사용하는 프로토콜에 따라 프레임 형식과 크기가 서로 다르다
- ➡ 각 링크에서 전달될 수 있는 최대 전송 길이를 _____ (Maximum Transfer Unit) 라고 함
 - ◆ MTU는 SDU (encapsulated data, ex. IP datagram)
- ➡ MTU 길이에 따라 나누어 보내는 것을 단편화 (fragmentation) 라고 함
- ➡ 단편들의 재조립은 목적지 Host에서만 이루어짐

TCP/IP

7-8

단편화(계속)

■ MTU (Maximum Transfer Unit)



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

TCP/IP

7-9

단편화(계속)

■ 단편화와 관련된 필드

➡ 식별자(Identification) – 단편들은 같은 식별자 값을 가짐

➡ 플래그(flag) – 3 비트 필드

◆ 1st bit : not used

◆ 2nd bit : _____

- '1' : 단편화 불가 -> 전송 불가시 폐기 (ICMP 에러 메시지 전송)
- '0' : 단편화 가능

◆ 3rd bit : _____

- '1' : 마지막 단편이 아님
- '0' : 마지막 단편이거나, 유일한 단편

D: Do not fragment

M: More fragments



TCP/IP

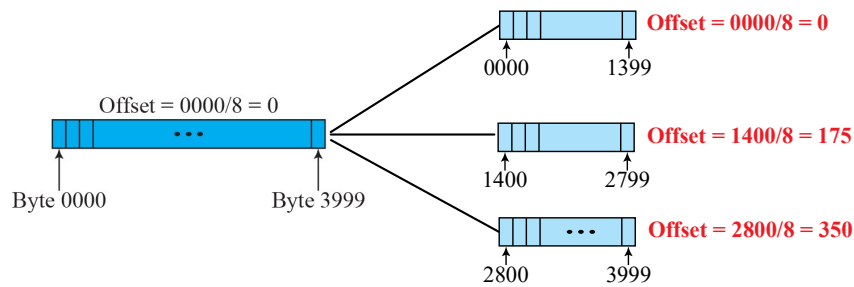
7-10

단편화(계속)

➡ 단편화 오프셋 (Fragmentation offset) : 13 bits 필드

- ◆ 전체 데이터그램에서 단편의 상대적인 위치(8 바이트 단위)

➡ 단편화 예

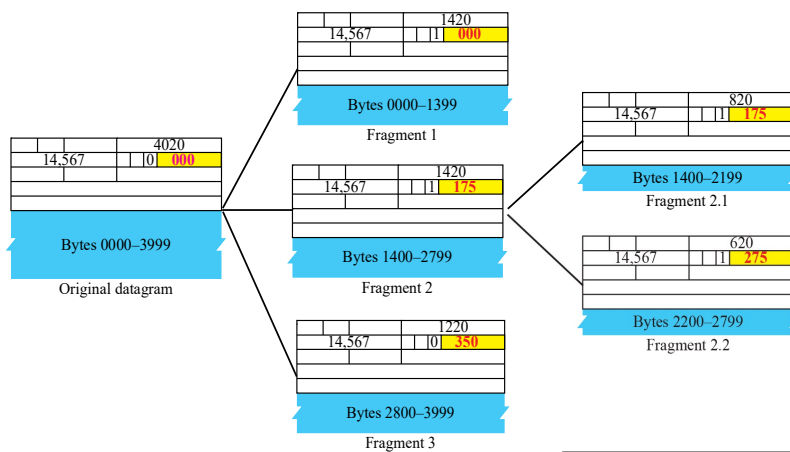


TCP/IP

7-11

단편화(계속)

■ 상세한 단편화 예



※ 재조립은 어떻게 할까?

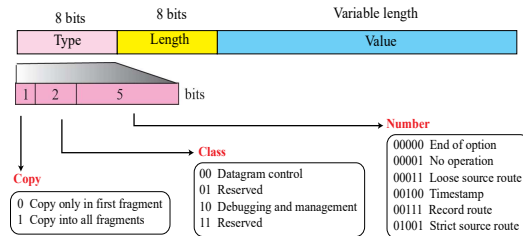
TCP/IP

7-12

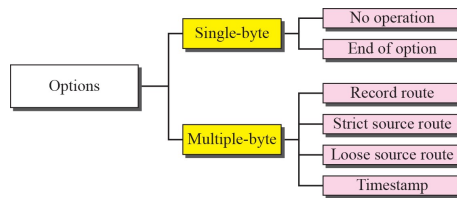
7.4 옵션

➔ IP 데이터그램 헤더 내의 가변 부분

➔ 옵션 형식



➔ 옵션 부류



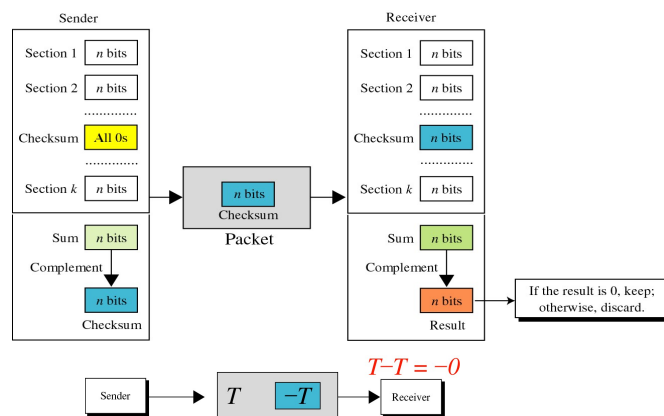
TCP/IP

7-13

7.5 검사합

➔ TCP/IP 프로토콜에 의해 사용되는 오류 검출 방법

➔ 헤더만 대상으로 하고 데이터는 검사하지 않는다.



TCP/IP

7-14

7.7 보안 (Security)

■ 보안 문제점들

- ➡ 패킷 엿보기 (Packet sniffing) → 패킷 _____
- ➡ 패킷 수정 (Packet modification) → 데이터 _____ 메커니즘
- ➡ IP 위장하기 (IP spoofing) → 발신지 _____ (Origin authentication)

■ _____ (IP Security)

- ➡ 알고리즘과 키의 결정 : 양쪽 개체(Entity)는 보안 알고리즘 및 키에 대해 합의
- ➡ 패킷 암호화 : _____ 유지
- ➡ 데이터 무결성 (Data integrity) : 패킷 변경 유무 확인
- ➡ 발신지 인증

TCP/IP

7-15

검사합(계속)

■ IP 패킷의 검사합 예

5	0		
1		0	
	17		
10.12.14.5			
12.6.7.9			

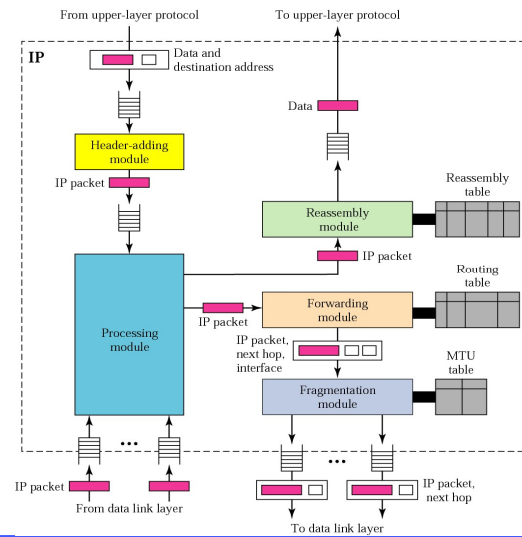
TCP/IP

7-16

7.8 IP 패키지

■ IP 구성 요소

- ➡ 헤더 추가 모듈
- ➡ 처리 모듈
- ➡ 큐
- ➡ 포워딩 모듈
- ➡ 라우팅 테이블
- ➡ 단편화 모듈
- ➡ MTU 테이블
- ➡ 재조립 모듈
- ➡ 재조립 테이블



TCP/IP

7-17

IP 패키지(계속)

■ 헤더 추가 모듈(Header-Adding Module)

Table 7.3 Adding module

```

1  IP_Adding_Module (data, destination_address)
2  {
3      Encapsulate data in an IP datagram
4      Calculate checksum and insert it in the checksum field
5      Send data to the corresponding queue
6      Return
7  }
    
```

TCP/IP

7-18

IP 패키지(계속)

Table 7.4 Processing module

■ 처리 모듈 (Processing Module)

```

1  IP_Processing_Module (Datagram)
2  {
3      Remove one datagram from one of the input queues.
4      If (destination address matches a local address)
5      {
6          Send the datagram to the reassembly module.
7          Return.
8      }
9      If (machine is a router)
10     {
11         Decrement TTL.
12     }
13     If (TTL less than or equal to zero)
14     {
15         Discard the datagram.
16         Send an ICMP error message.
17         Return.
18     }
19     Send the datagram to the forwarding module.
20     Return.
21 }
```

TCP/IP

7-19

IP 패키지(계속)

- 큐(Queue) – 입/출 큐
- 라우팅 테이블 : 패킷의 다음 홉 주소 결정하기 위해
라우팅 모듈이 사용
- 포워딩 모듈 : 6장 참조
- MTU 테이블 : 단편화 모듈이 특정 인터페이스의
MTU를 찾기 위해 사용

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

TCP/IP

7-20

IP 패키지(계속)

■ 단편화 모듈(Fragmentation Module)

Table 7.5 Fragmentation module

```

1 IP_Fragmentation_Module (datagram)
2 {
3     Extract the size of datagram
4     If (size > MTU of the corresponding network)
5     {
6         If (D bit is set)
7         {
8             Discard datagram
9             Send an ICMP error message
10            return
11        }
12        Else
13        {
14            Calculate maximum size
15            Divide the segment into fragments
16            Add header to each fragment
17            Add required options to each fragment

```

Table 7.5 Fragmentation module (continued)

```

18            Send fragment
19            return
20        }
21    }
22    Else
23    {
24        Send the datagram
25    }
26    Return.
27 }

```

TCP/IP

7-21

IP 패키지(계속)

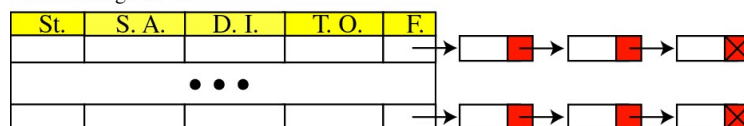
■ 재조립 테이블(Reassembly table)

➡ 재조립 모듈에 의해 사용

➡ 재조립 테이블

St.: State
S. A.: Source address
D. I.: Datagram ID

T. O.: Time-out
F.: Fragments



TCP/IP

7-22

IP 패키지(계속)

■ 재조립 모듈 (Reassembly Module)

Table 7.6 Reassembly module

```
1 IP_Reassembly_Module (datagram)
2 {
3     If (offset value = 0 AND M = 0)
4     {
5         Send datagram to the appropriate queue
6         Return
7     }
8     Search the reassembly table for the entry
9     If (entry not found)
10    {
11        Create a new entry
12    }
13    Insert datagram into the linked list
14    If (all fragments have arrived)
15    {
16        Reassemble the fragment
17        Deliver the fragment to upper-layer protocol
18        return
19    }
20    Else
21    {
22        If (time-out expired)
23        {
24            Discard all fragments
25            Send an ICMP error message
26        }
27    }
28    Return.
29 }
```