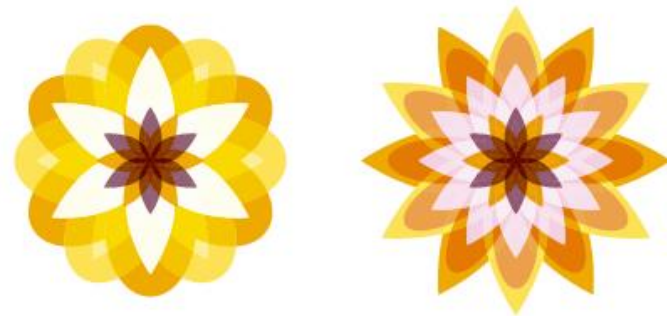


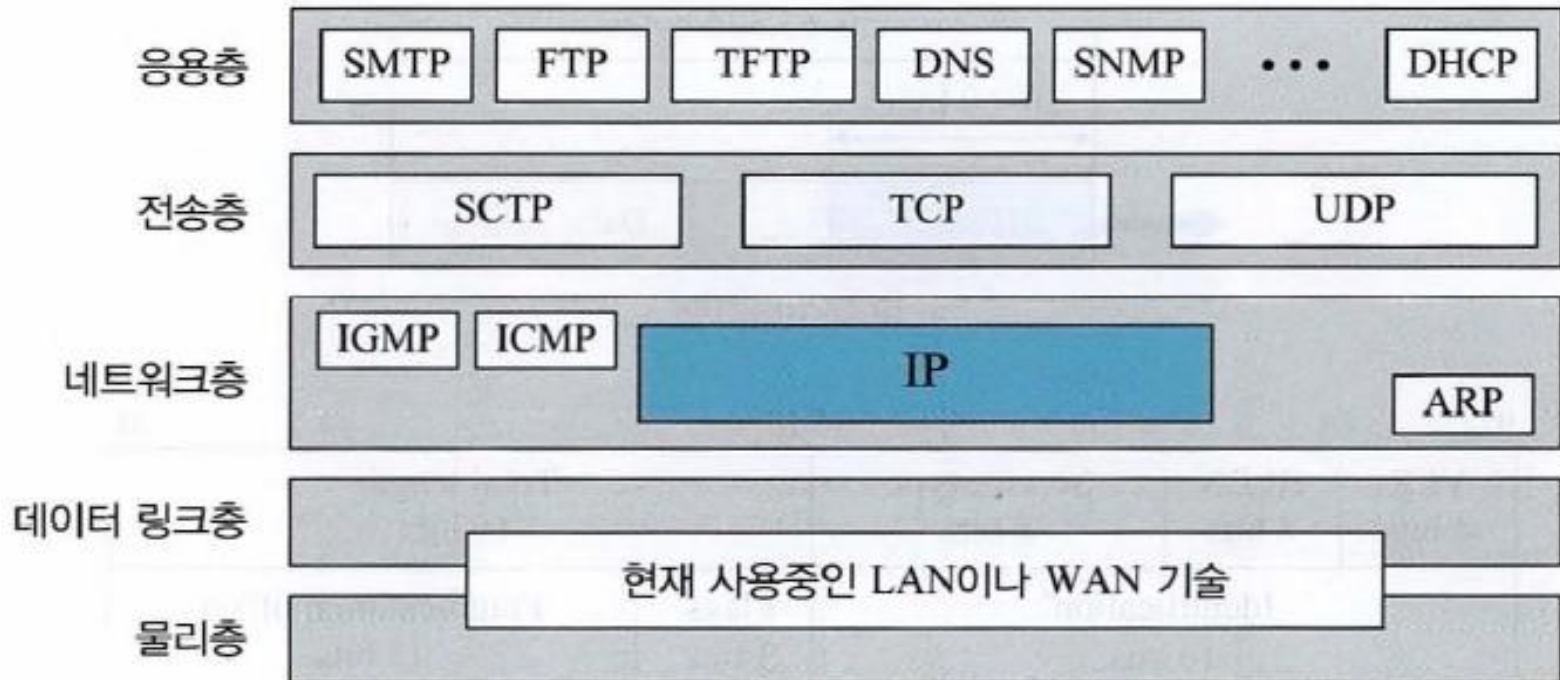
Chapter 08

인터넷 프로토콜(IP)



1. 개요

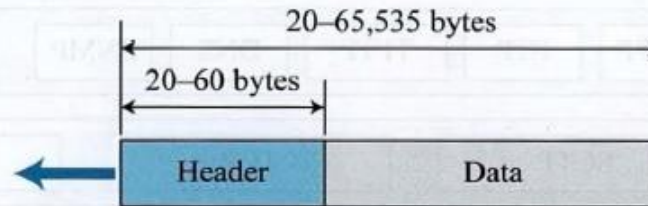
- 인터넷 프로토콜 (IP: Internet Protocol)은 네트워크층에서 TCP/IP 프로토콜이 사용하는 전송 메커니즘이다.
- 그림은 프로토콜 모음에서 IP의 위치를 보여주고 있다.



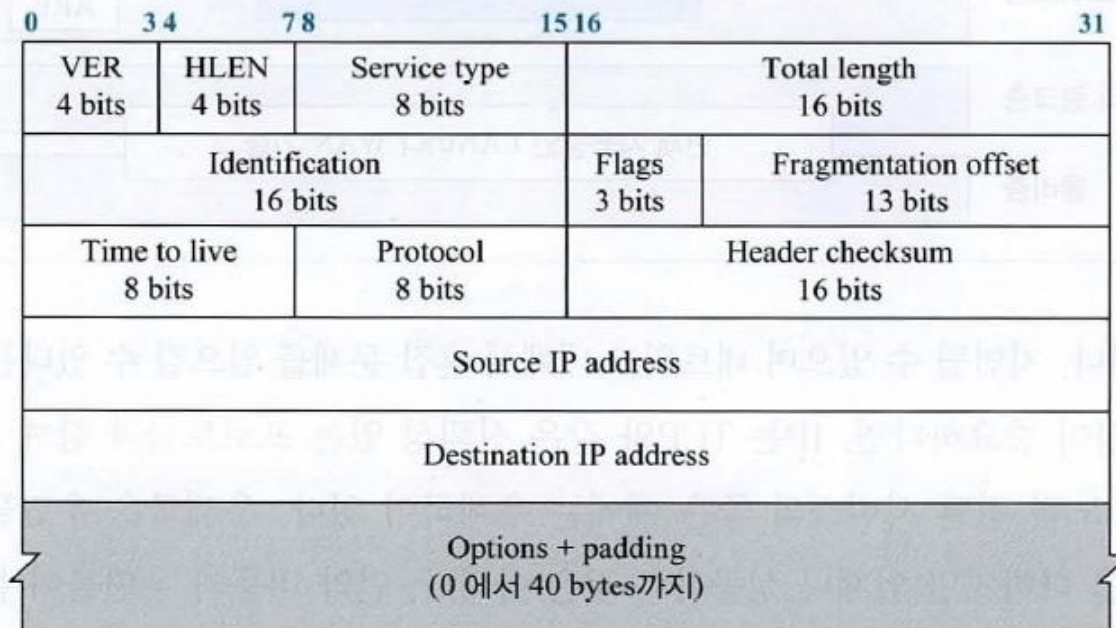
- IP는 신뢰성이 없고 비연결형 데이터그램 프로토콜로서 최선의 노력 (best effort) 전달 서비스를 제공한다.
- 최선의 노력이란 의미는 IP 패킷은 오류가 발생하거나, 분실되거나, 틀린 순서로 도착하거나, 지연될 수 있으며 네트워크 내에서 혼잡 문제를 일으킬 수 있다는 것이다.
- 만약 신뢰성이 중요하다면, IP는 TCP와 같은 신뢰성 있는 프로토콜과 함께 사용되어야 한다.
- IP는 데이터그램 방법을 사용하는 패킷 교환망을 위해 설계된 비연결형 프로토콜이다.
- 이것은 각 데이터그램이 독립적으로 처리되고 목적지까지 서로 다른 경로를 통하여 전달될 수 있다는 것이다.
- 이 중의 일부는 분실될 수 있고 일부는 전달 도중 훼손될 수도 있다.
- 인터넷 프로토콜(IP)은 이 모든 문제를 해결하기 위하여 상위 계층의 프로토콜에 의존한다.

2. 데이터그램

- 네트워크(인터넷)층의 패킷을 데이터그램 (datagram) 이라고 한다.
- 그림은 IPv4 데이터그램의 형식을 보여준다.



a. IP 데이터그램



b. 헤더 형식

2. 데이터그램

- 데이터그램은 가변 길이 패킷으로 헤더와 데이터 부분으로 구성된다.
- 헤더는 20바이트에서 60바이트이고 라우팅과 전달에 필요한 정보를 포함하고 있다.
- TCP/IP에서는 보통 헤더를 4바이트 단위로 보여준다.
- 다음은 헤더 내의 필드에 대한 간단한 설명이다.
 - 버전(VER, version)
 - 이 4bit 필드는 IP 프로토콜의 버전을 나타낸다.
 - 현재 버전은 4이다.
 - 그러나 앞으로 버전 6(또는 IPv6)으로 대체될 것이다.
 - 이 필드는 시스템에서 수행되고 있는 IP 소프트웨어에 데이터그램이 버전 4 형식으로 되어 있다는 것을 알려준다.
 - 모든 필드는 버전 4 프로토콜에 지정된 바와 같이 해석되어야 한다.

2. 데이터그램

■ 다음은 헤더 내의 필드에 대한 간단한 설명이다.

■ 헤더 길이(HLEN, header length)

- 이 4bit 필드는 데이터그램 헤더의 전체 길이를 4바이트 단위로 나타낸다.
- 데이터그램의 헤더 길이는 20바이트에서 60바이트까지이다.
- 옵션이 없으면 헤더의 길이는 20바이트이고, 이 필드의 값은 5가 된다($5 \times 4 = 20$).
- 만약 옵션 필드가 최대 길이라면 이 필드의 값은 15가 된다($15 \times 4 = 60$).

■ 서비스 유형 (TOS)

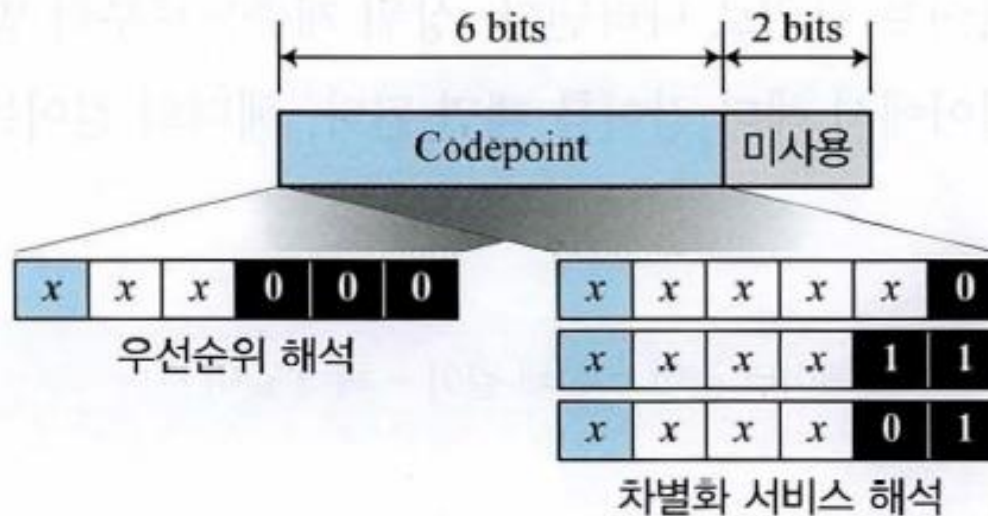
- IP 헤더의 초기 설계에서 이 필드는 TOS(Type of Service)라 불리었고 데이터그램이 어떻게 처리되는가를 정의하였다.
- 이 필드의 일부는 데이터그램의 우선순위 (precedence)를 지정하고 나머지 부분은 (저지연, 고처리율 등과 같은) 서비스 유형을 정의하였다.
- IETF는 이 8비트 필드의 의미를 수정하였다.
- 이 필드는 현재 차별화 서비스(differentiated services) 집합이라고 불린다.
- 그림은 이 필드의 새로운 의미를 보여주고 있다.

2. 데이터그램

■ 다음은 헤더 내의 필드에 대한 간단한 설명이다.

▪ 서비스 유형 (TOS)

- 그림은 이 필드의 새로운 의미를 보여주고 있다.



2. 데이터그램

■ 다음은 헤더 내의 필드에 대한 간단한 설명이다.

■ 서비스 유형 (TOS)

- 새로운 정의에서 처음 6비트는 코드포인트(codepoint) 부필드이고 마지막 2 비트는 사용되지 않는다.
- 코드포인트 부필드는 다음과 같이 두 방법으로 사용된다.
 - 오른쪽 세 비트가 0이면 왼쪽 세 비트는 서비스 유형에서 우선순위와 같은 의미로 해석된다.
 - 오른쪽 세 비트가 모두 0이 아니면 6 비트는 표와 같이 인터넷 지정 기관에 의하여 부여된 우선순위에 따라 56(64-8)가지 서비스를 정의한다.

범주	코드포인트	지정기관
1	XXXXX0	Internet
2	XXXX11	Local
3	XXXX01	임시 또는 실험용

■ 다음은 헤더 내의 필드에 대한 간단한 설명이다.

■ 전체 길이 (Total Length)

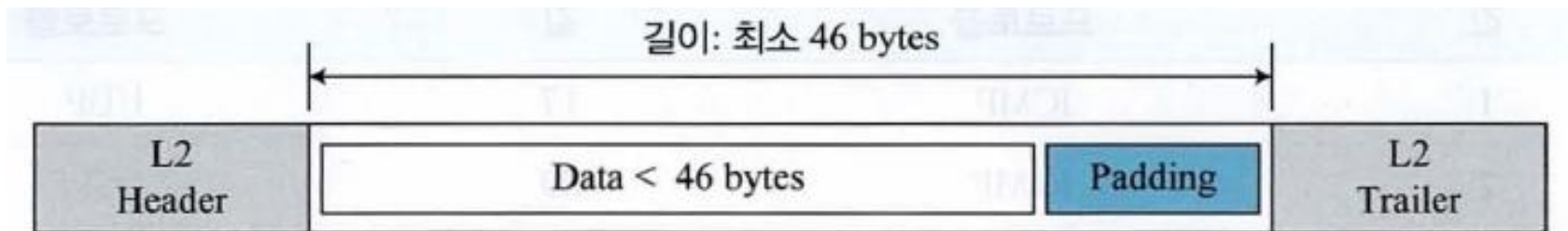
- 이 16 비트 필드는 헤더와 데이터를 포함하는 IP 데이터그램의 전체 길이를 바이트 단위로 나타낸다.
- 상위 계층으로부터 받은 데이터 길이를 알아내기 위해 전체 길이에서 헤더 길이를 빼면 된다.
- 헤더의 길이는 HLEN 필드의 값에 4를 곱하면 된다.
- 필드의 길이가 16 비트이므로 IP 데이터그램의 전체 길이는 $65,535(2^{16} - 1)$ 바이트로 제한되고, 이 중 20바이트에서 60바이트는 헤더이고 나머지가 상위 계층으로부터 받는 데이터이다.
- 65,535 바이트 길이는 현재 기술에서 매우 크게 느껴진다.
- 그러나 하위 계층의 기술이 발달하여 더 높은 처리율(더 큰 대역폭)을 갖는 네트워크를 사용할 수 있게 된다면 가까운 미래에 IP 데이터그램의 길이는 증가할 수 있을 것이다.
- 왜 이러한 여분의 필드가 필요한가?
- 물론 대부분은 이 필드의 값이 필요없다.
- 그러나 어떤 경우에는 프레임에 포함된 것이 데이터그램이 아니라 추가된 패딩이 있을 수도 있다.

2. 데이터그램

■ 다음은 헤더 내의 필드에 대한 간단한 설명이다.

▪ 전체 길이 (Total Length)

- 예를 들어 이더넷 프로토콜은 프레임에 포함될 수 있는 데이터 길이에 최소값(46바이트)과 최대값(1,500바이트)이 있다.
- 만약 IP 데이터그램의 길이가 46바이트보다 작으면 이 요구 사항을 맞추기 위해 패딩이 필요하다.
- 이러한 경우 시스템이 데이터그램을 프레임에서 추출한 후 어디까지가 데이터이고 어디까지가 패딩인지 알기 위해 전체 길이를 확인해야 한다.



2. 데이터그램

■ 다음은 헤더 내의 필드에 대한 간단한 설명이다.

■ 식별 (identification)

- 이 필드는 단편화를 위해 사용되고 다음 절에서 설명된다.

■ 플래그(flags)

- 이 필드도 단편화를 위해 사용되며 다음 절에서 설명된다.

■ 단편화 오프셋 (fragmentation offset)

- 이 필드도 단편화를 위해 사용되고 다음 절에서 설명된다.

■ 수명 (time to live)

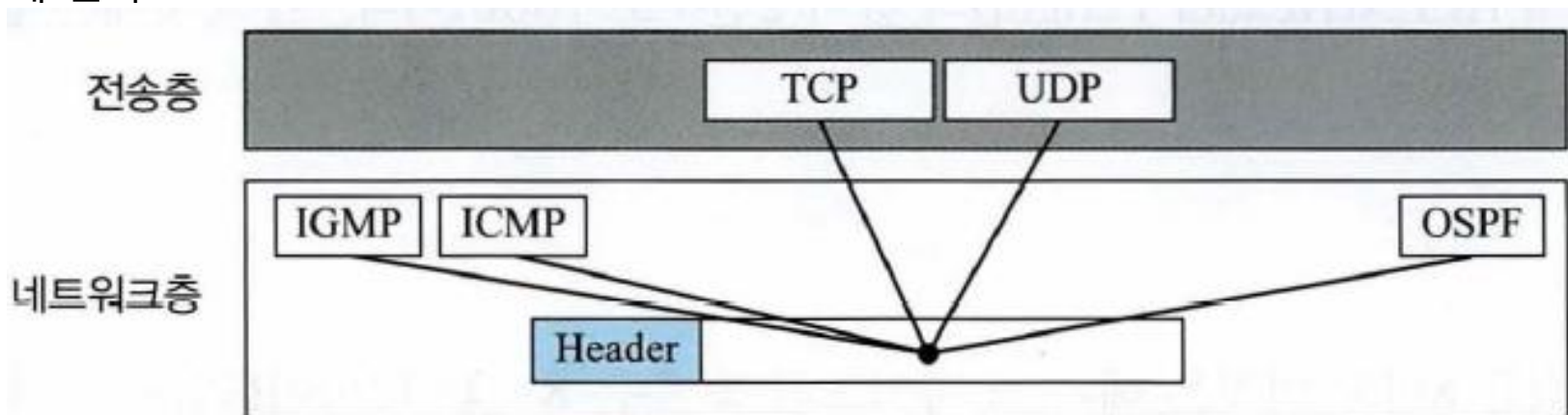
- 데이터그램은 인터넷을 통해 전달되는 동안 제한된 수명을 갖는다.
- 이 필드는 데이터그램에 의해 방문하는 최대 홉(라우터) 수를 제어하기 위해 사용된다.
- 발신지 호스트가 데이터그램을 보낼 때 이 필드에 숫자를 저장한다.
- 이 값은 대략 두 호스트 사이에 있는 라우터 수의 두 배이다.
- 데이터그램을 처리하는 각 라우터는 이 필드의 값을 1 씩 감소시킨다.
- 만약 감소 후 값이 0 이 되면 라우터는 데이터그램을 폐기한다.

2. 데이터그램

■ 다음은 헤더 내의 필드에 대한 간단한 설명이다.

■ 프로토콜(protocol)

- 이 8bit 필드는 IP 계층의 서비스를 사용하는 상위 계층 프로토콜을 정의한다.
- IP 데이터그램은 TCP, UDP, ICMP, IGMP와 같은 여러 종류의 상위 계층 프로토콜을 캡슐화할 수 있다.
- 이 필드는 IP 데이터그램이 전달되어야 하는 최종 프로토콜을 나타낸다.
- 다시 말하면 IP 프로토콜은 상위 계층 프로토콜로부터 오는 데이터를 다중화 및 역다중화하므로 데이터그램이 최종 목적지에 도달한 경우 이 필드의 값은 역다중화 과정을 돕게 된다.



2. 데이터그램

■ 다음은 헤더 내의 필드에 대한 간단한 설명이다.

■ 프로토콜(protocol)

- 여러 상위 계층에 해당하는 이 필드의 일부 값이 표에 나와 있다.

값	프로토콜	값	프로토콜
1	ICMP	17	UDP
2	ICMP	89	OSPF
6	TCP		

■ 검사합(checksum)

- 검사합의 개념과 계산법은 뒤에 설명되어 있다.

■ 발신지 주소(source address)

- 이 32bit 필드는 발신지의 IP 주소를 정의한다.
- IP 데이터그램이 발신지에서 목적지까지 전달되는 동안 이 값은 변해서는 안 된다.

■ 목적지 주소(destination address)

- 이 32bit 필드는 목적지의 IP 주소를 정의한다.
- IP 데이터그램이 발신지에서 목적지까지 전달되는 동안 이 값은 변해서는 안 된다.

2. 데이터그램

- 다음은 헤더 내의 필드에 대한 간단한 설명이다.
 - 예제. 처음 8bit가 다음과 같은 IP 패킷이 도착하였다.

01000010

- 수신지는 이 패킷을 폐기한다.
- 이유를 설명하라.

2. 데이터그램

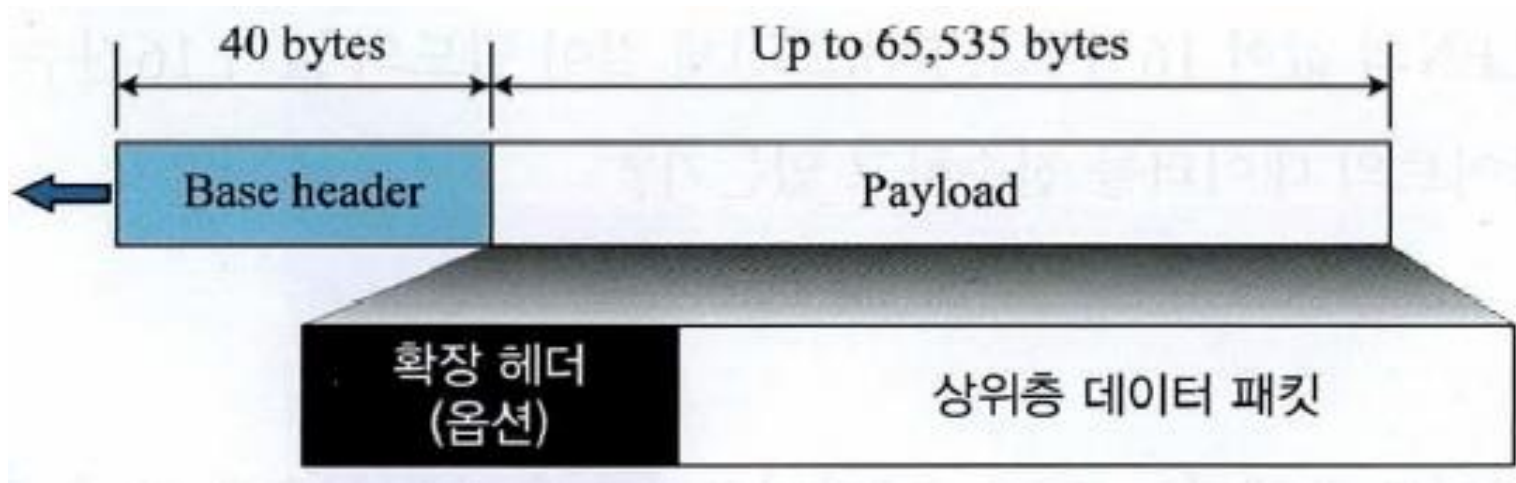
- 다음은 헤더 내의 필드에 대한 간단한 설명이다.
 - 예제. IP 패킷에서 HLEN(Header Length)의 값이 2진수로 1000_2 이다.
 - 이 패킷에는 옵션이 몇 바이트있는가?

2. 데이터그램

- 다음은 헤더 내의 필드에 대한 간단한 설명이다.
 - 예제. IP 패킷에서 HLEN의 값이 16진수로 5 이고 전체 길이 필드의 값이 16진수로 0028₁₆이다.
 - 이 패킷은 몇 바이트의 데이터를 전송하고 있는가?

2. 데이터그램

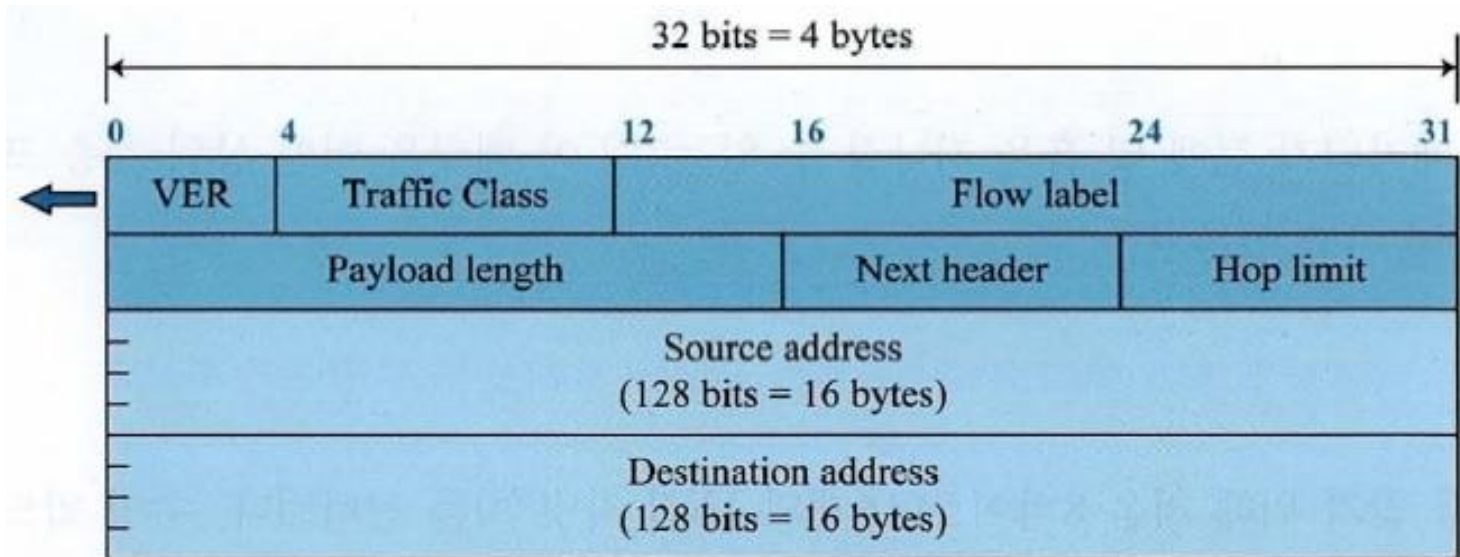
- 최근에 IPv4에서 사용되는 주소가 다 소진되어 새로이 설계된 IPv6 패킷 형식은 그림에 나타나 있다.
- 각 패킷은 필수적인 기본 헤더와 페이로드(payload) 로 구성된다.
- 페이로드는 두 개의 필드인 선택적인 확장 헤더와 상위 계층에서 온 데이터로 구성된다.
- 기본 헤더는 40바이트를 차지하며 반면에 확장 헤더와 상위 계층에서 온 데이터가 최대 65,535 바이트까지 차지한다.



2. 데이터그램

■ IPv6 데이터그램 기본 헤더

- 그림은 8개의 필드를 가진 IPv6 데이터그램의 기본 헤더(base header)를 보여준다.



■ 버전 (version)

- 이 4bit 필드는 IP의 버전을 정의한다.
- IPv6의 경우 그 값은 6이다.

■ IPv6 데이터그램 기본 헤더

■ 트래픽 클래스(traffic class)

- 이 8bit 필드는 서로 다른 전달 요구 사항을 갖는 페이로드를 구분하는 데 쓰인다.
- 이 필드는 IPv4의 서비스 클래스 필드를 대신한다.

■ 흐름 레이블(flow label)

- 흐름 레이블은 20bit로 구성되며 특정 데이터의 흐름을 특별하게 제어하기 위해 설계되었다.

■ 페이로드 길이(payload length)

- 이 2바이트 페이로드 길이 필드는 기본 헤더를 제외한 IP 데이터그램의 전체 길이를 정의한다.

■ 다음 헤더(next header)

- 다음 헤더는 데이터그램에서 기본 헤더의 다음 헤더를 정의하는 8bit 필드이다.
- 다음 헤더는 IP에 의해 이용되는 선택적인 확장 헤더 중의 하나이거나 UDP 또는 TCP와 같은 상위 계층 프로토콜을 위한 헤더이다.

2. 데이터그램

■ IPv6 데이터그램 기본 헤더

■ 다음 헤더(next header)

- 다음 표는 다음 헤더 값을 나타낸다.
- 버전 4에서 이 필드는 프로토콜이라고 부르고 있다는 것에 주목해야 한다.

코드	다음 헤더	코드	다음 헤더
0	Hop-by-hop option	44	Fragmentation
1	ICMP	50	Encrypted security payload
6	TCP	51	Authentication
17	UDP	59	Null (No next header)
43	Source routing	60	Destination option

2. 데이터그램

■ IPv6 데이터그램 기본 헤더

▪ 홉 제한(hop limit)

- 이 8bit의 홉 제한 필드는 IPv4의 TTL 필드와 같은 목적으로 사용된다.

▪ 발신지 주소(source address)

- 발신지 주소 필드는 보통 데이터그램의 원래 발신지 주소를 식별하는 16바이트(128bit) 인터넷 주소이다.

▪ 목적지 주소(destination address)

- 목적지 주소 필드는 데이터그램의 최종 목적지를 식별하는 16바이트(128bit) 인터넷 주소이다.
- 그러나 발신지 라우팅이 사용된다면 이 필드는 다음 라우터의 주소를 갖는다.

■ 흐름 테이블

- IP 프로토콜은 원래 비연결형 프로토콜로서 설계되었다.
- 그러나 최근에는 IP 프로토콜을 연결형 프로토콜로 사용하는 경향이 있다.
- MPLS(Multi-Protocol Label Switching) 기술은 레이블 필드를 사용하여 MPLS 헤더 내에 IPv4 패킷을 캡슐화할 수 있게 해준다.
- IPv6에서는 흐름 레이블(flow label)이 IPv6 데이터그램의 형식에 추가되어 IPv6를 연결형 프로토콜로 사용할 수 있게 해 준다.
- 라우터에서 흐름은 같은 경로를 통과하고 같은 자원을 사용하고 같은 수준의 보안을 갖는 등 동일 특성을 공유하는 패킷의 연속이다.
- 흐름 레이블을 제어할 수 있는 라우터는 흐름 레이블 테이블을 가진다.
- 그 테이블은 각 능동적 흐름 레이블을 위한 항목을 갖는다.
- 각 항목은 해당 흐름 레이블에 의해 요구되는 서비스를 정의한다.
- 패킷을 수신할 때 라우터는 패킷에서 정의된 흐름 레이블 값에 해당하는 항목을 찾기 위해 흐름 레이블 테이블을 참조한다.
- 그리고 나서 라우터는 항목에서 언급된 서비스들과 함께 패킷을 제공한다.

■ 흐름 테이블

- 가장 간단한 형태로, 흐름 레이블은 라우터에 의한 패킷의 처리를 빠르게 할 수 있다.
- 라우터가 패킷을 수신할 때 다음 홉의 주소를 결정하기 위해 라우팅 테이블을 참조하고 라우팅 알고리즘을 통과하는 대신에 흐름 레이블 테이블에서 다음 홉을 쉽게 찾을 수 있게 된다.
- 더 복잡한 형태에서 흐름 레이블은 실시간 오디오와 비디오 전송을 제공할 수 있다.
- 하나의 프로세스는 실시간 데이터가 자원의 부족으로 인하여 지연되지 않도록 미리 이러한 자원을 예약할 수 있다.
- 실시간 데이터의 시용과 이러한 자원의 예약은 IPv6에 추가하여 실시간 프로토콜 (RTP: Real Time Protocol)과 자원 예약 프로토콜(RSVP: Resource Reservation Protocol)과 같은 프로토콜을 요구한다.

■ 흐름 테이블

- 흐름 레이블의 효과적인 사용을 위해서 다음 3가지 법칙이 정의된다.
 - 흐름 레이블은 발신지 호스트에 의해 패킷에 할당된다. 레이블은 1과 $2^{24}-1$ 사이의 임의의 수이다. 발신지는 현재 흐름이 아직 살아있는 동안에는 새로운 흐름을 위해서 흐름 레이블을 재사용해서는 안 된다.
 - 호스트가 흐름 레이블을 제공하지 않으면 이 필드를 0으로 설정한다. 만약 라우터가 흐름 레이블을 제공하지 않으면 흐름 레이블을 단순히 무시하면 된다.
 - 동일 흐름에 속하는 모든 패킷은 동일 자원과 동일 목적지, 동일 우선권, 동일 옵션을 가져야 한다.

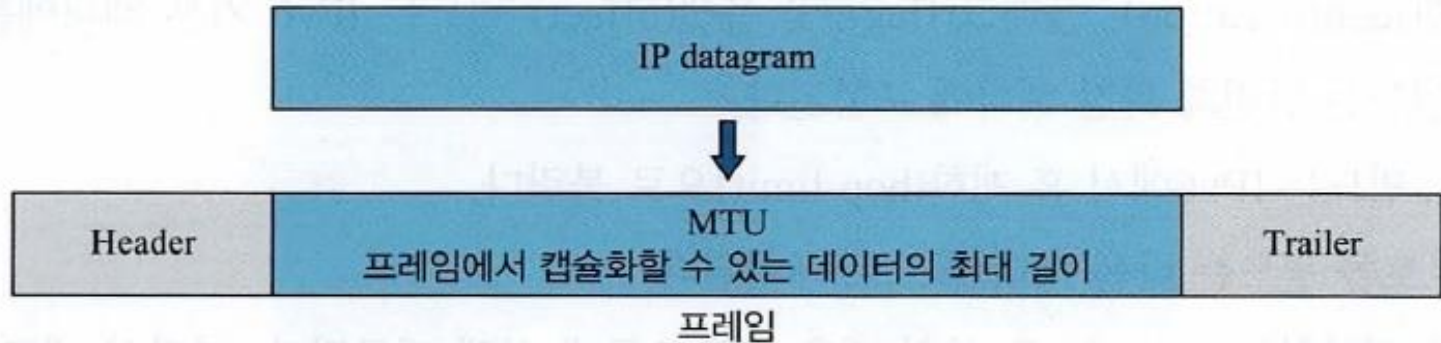
■ IPv4와 IPv6 헤더 비교

- IPv6에서 헤더의 길이는 고정되어 있기 때문에 헤더 길이 필드가 없다.
- IPv6에서 서비스 유형 필드는 없다. 트래픽 클래스와 흐름 레이블 필드는 서비스 유형 필드의 기능을 대신한다.
- 총 길이 필드는 IPv6 에는 없고 페이로드 길이 필드로 대체된다.
- 식별(identification), 플래그(flag) 및 오프셋(offset) 필드는 IPv6 기본 헤더에는 없다. 이 필드는 단편화 확장 헤더에 포함된다.
- TTL 필드는 IPv6에서 홉 제한(hop limit)으로 불린다.
- 프로토콜 필드는 다음 헤더 필드로 대체된다.
- 헤더 검사합(checksum)은 상위 계층 프로토콜에 의해 제공된다. 따라서 네트워크 계층에서는 필요없다.
- IPv4에서 옵션 필드(option field)는 IPv6 에서는 확장 헤더로 구현된다.

3. 단편화

■ 최대 전달 단위(MTU)

- 대부분의 프로토콜에서 각 데이터 링크층은 자신의 프레임 형식을 가지고 있다.
- 프레임의 형식에 정의된 필드 중 하나는 데이터 필드의 최대 크기인 최대 전달 단위 (MTU: Maximum Transfer Unit) 이다.
- 이 최대 크기는 네트워크 내에서 사용되는 하드웨어와 소프트웨어에 의해 주어지는 제한 조건에 의해 결정된다.



■ 최대 전달 단위(MTU)

- IP 프로토콜을 물리 네트워크와 독립적으로 만들기 위해 설계자들은 IP 데이터그램의 최대 길이를 65,535바이트로 결정하였다.
- 만약 MTU에 맞는 프로토콜을 사용한다면 패킷의 전달은 효율적으로 될 수 있을 것이다.
- 그러나 MTU가 작은 다른 네트워크에서는 데이터그램을 나누어서 보내야 한다.
- 이것을 단편화(Fragmentation) 라고 한다.
- 보통 발신자는 IP 패킷을 단편화하지 않는다.
- 발신자 노드에서 사용되고 있는 IP와 데이터 링크층에서 수용하기 수월하도록 발신자 전송층이 데이터를 적절한 크기의 세그먼트로 나눈다.
- 데이터그램이 단편화될 때 각 단편은 자신의 헤더를 가지는데 이 헤더 내의 대부분의 필드는 본래의 값과 같으나 일부 필드의 값은 변경된다.
- 단편화된 데이터그램이 더 작은 MTU 값을 갖는 네트워크를 지나게 되면 다시 단편화된다.
- 각 단편은 독립된 데이터그램이므로 데이터그램의 재조립은 최종 목적지 호스트에 의해서만 행해진다.

■ 최대 전달 단위(MTU)

- 데이터그램이 단편화될 때 헤더에서 요구되는 부분은 모든 단편에 복사되어야 한다.
- 데이터그램을 단편화하는 호스트나 라우터는 다음 세 필드의 값을 변경하여야 한다.
- 이는 플래그, 단편화 오프셋, 전체 길이이다.
- 나머지 필드들은 반드시 복사되어야 한다.
- 물론, 검사합 값은 단편화와 관계없이 다시 계산되어야 한다.

■ 단편화와 관련된 필드

■ 식별자(identification)

- 이 16bit 필드는 발신지 호스트가 보낸 데이터그램을 유일하게 식별한다.
- 식별지와 발신지 IP 주소의 조합은 데이터그램이 발신지 호스트를 떠날 때 유일하게 정의되어야 한다.
- 이러한 유일성을 보장하기 위해 IP 프로토콜은 카운터를 사용하여 데이터그램에 레이블을 붙인다.
- 이 카운터는 양의 정수 값으로 초기화된다.
- IP 프로토콜이 데이터그램을 보낼 때 카운터의 현재 값을 식별자 필드에 복사하고 카운터 값을 1 증가시킨다.
- 카운터 값이 주기억장치 내에 유지되는 한 유일성은 보장된다.
- 데이터그램이 단편화될 때 식별자 필드의 값은 모든 단편에 복사된다.
- 다시 말해서 모든 단편은 같은 식별자 값을 가지게 되며 이 값은 원 데이터그램의 값과 같다.
- 식별자는 목적지에서 데이터그램을 재조립하는 데 도움이 된다.
- 같은 식별자 값을 가지는 모든 단편은 하나의 데이터그램으로 재조립되어야 한다.

3. 단편화

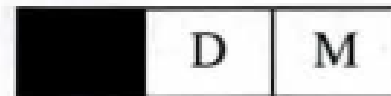
■ 단편화와 관련된 필드

■ 플래그(flag)

- 이는 3bit 필드이다.
- 처음 bit는 사용되지 않으며, 두 번째 bit는 "do not fragment" bit이다.
- 이 플래그의 값이 1 이면 데이터그램을 단편화하면 안된다.
- 단편화를 수행해야 하는데 이 bit가 설정되어 있어 네트워크를 통하여 데이터그램을 전달할 수 없다면 데이터그램을 폐기하고 ICMP 오류 메시지를 발신지 호스트에 보낸다.
- 만약 이 값이 0이면 필요한 경우 데이터그램은 단편화될 수 있다.
- 세 번째 비트의 이름은 "more fragment" bit이다.
- 이 값이 1이면 데이터그램은 마지막 단편이 아니라는 것을 알려준다.
- 만약 0 이면 마지막 단편이거나 유일한 단편이다.

D: Do not fragment

M: More fragments

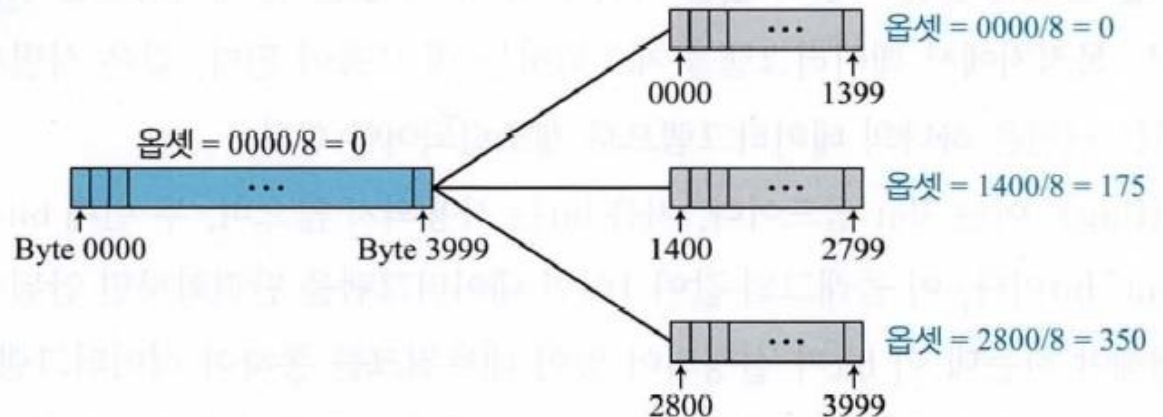


3. 단편화

■ 단편화와 관련된 필드

▪ 단편화 오프셋 (fragmentation offset)

- 이 13bit 필드는 전체 데이터그램 내에서 단편의 상대적 위치를 나타낸다.
- 이 필드는 원 데이터그램 내에서 데이터의 오프셋을 8바이트 단위로 나타낸 것이다.
- 그림은 데이터 크기가 4,000바이트인 데이터그램이 세 개의 단편으로 나누어진 경우를 보여준다.
- 원 데이터그램의 바이트는 0부터 3,999의 번호를 가진다.
- 첫 번째 단편에는 0 에서 1,399 번까지의 바이트가 있으며 $0/8=0$ 의 오프셋 값을 가진다.
- 두 번째 단편은 1,400 번에서 2,799 번까지의 바이트를 가지고 있으며 오프셋 값은 $1,400/8 = 175$ 이다.
- 마지막 세 번째 단편은 2,800 번에서 3,999 번까지의 바이트를 가지고 있으며 오프셋 값은 $2,800/8 = 350$ 이다.

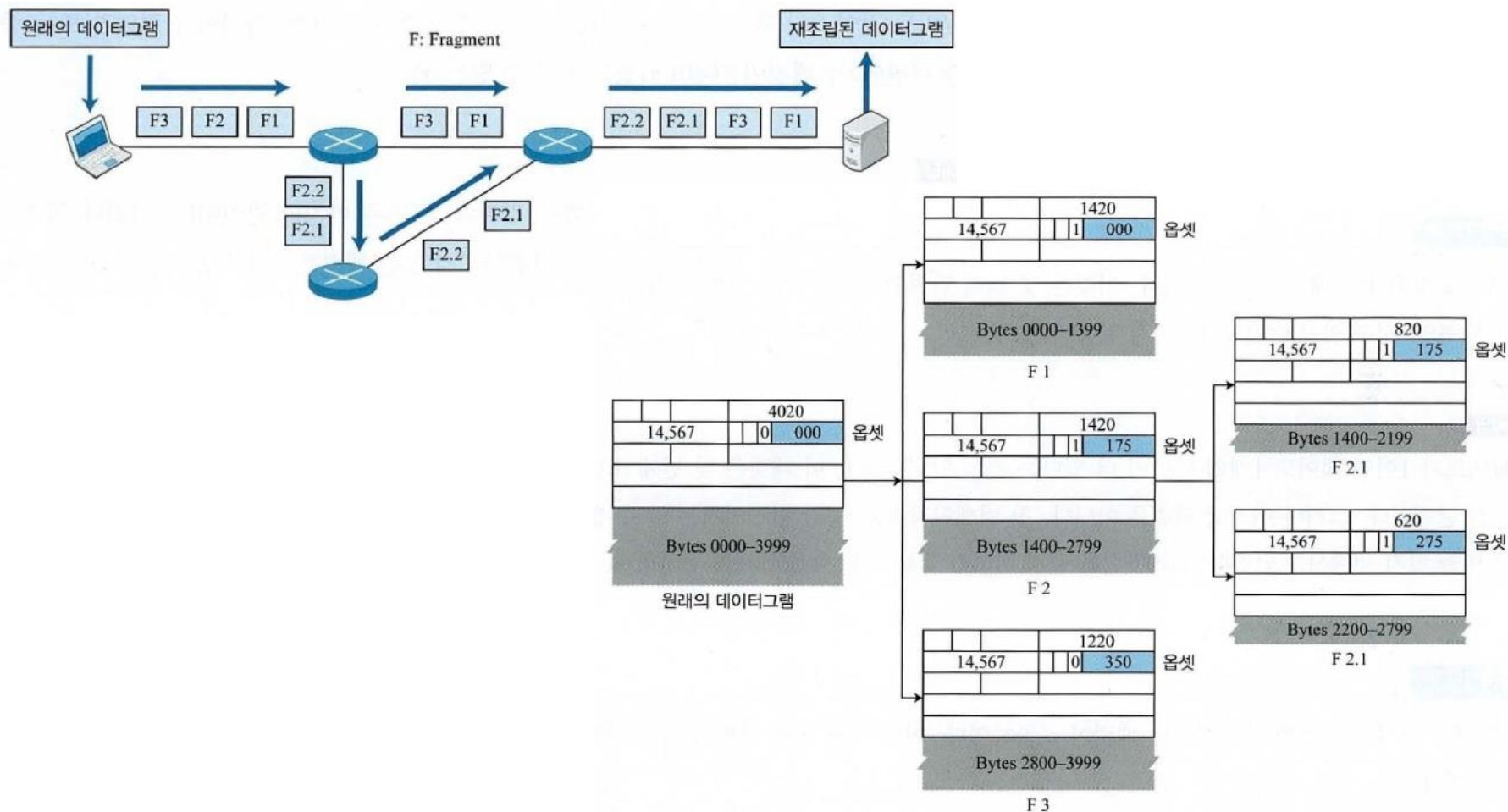


3. 단편화

■ 단편화와 관련된 필드

▪ 단편화 오프셋 (fragmentation offset)

- 그림은 앞 그림 단편을 확대하여 보여준다.



■ 단편화와 관련된 필드

■ 단편화 오프셋 (fragmentation offset)

- 비록 각 단편이 다른 경로를 따라 전달되어 순서가 뒤바뀌어 도착하더라도 최종 목적지 호스트는 분실된 단편이 없다면 다음 방법에 따라 수신한 단편들로부터 원래의 데이터그램을 재조립할 수 있다.
 - 첫 번째 단편의 오프셋 값은 0 이다.
 - 첫 번째 단편의 길이를 8로 나눈다. 두 번째 단편의 오프셋 값은 이 결과와 같아야 한다.
 - 첫 번째와 두 번째 단편의 길이의 합을 8로 나눈다. 세 번째 단편의 오프셋 값은 이 결과와 같아야 한다.
 - 이러한 과정을 반복한다. 마지막 오프셋은 more fragment 값이 0 이다.

3. 단편화

■ 단편화와 관련된 필드

■ 단편화 오프셋 (fragmentation offset)

- 예제. M비트 값이 0 인 패킷이 도착하였다. 이것은 첫 번째 단편인가, 마지막 단편인가 또는 중간 단편인가? 패킷이 단편화되었는지 알 수 있는가?

3. 단편화

■ 단편화와 관련된 필드

■ 단편화 오프셋 (fragmentation offset)

- 예제. M비트 값이 1 인 패킷이 도착하였다. 이것은 첫 번째 단편인가, 마지막 단편인가 또는 중간 단편인가? 패킷이 단편화되었는지 알 수 있는가?

3. 단편화

■ 단편화와 관련된 필드

■ 단편화 오프셋 (fragmentation offset)

- 예제. M비트가 1 이고 단편화 오프셋이 0 인 패킷이 도착하였다. 이것은 첫 번째 단편인가, 마지막 단편인가 또는 중간 단편인가?

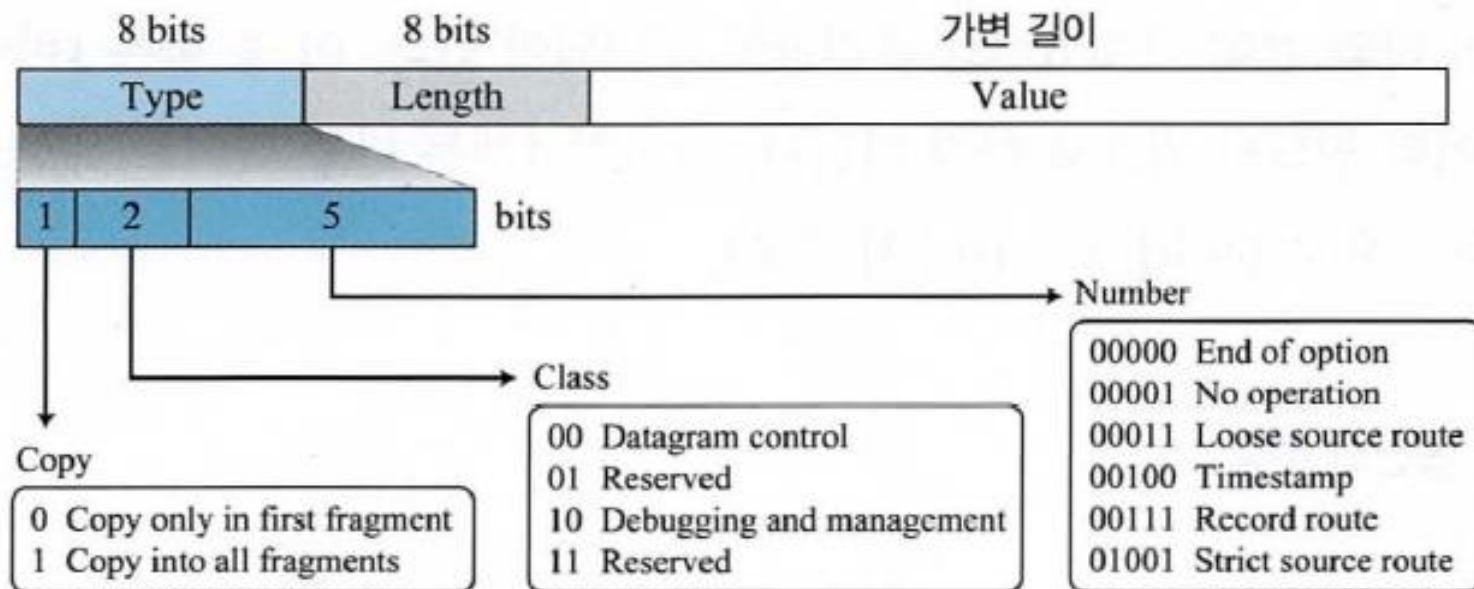
4. 옵션

- IP 데이터그램의 헤더는 고정 부분과 가변 부분으로 구성되어 있다.
- 고정 부분의 길이는 20바이트이고, 가변 부분은 옵션으로 구성되고 최대 길이는 40바이트이다.
- 옵션은 그 이름이 암시하듯이 데이터그램 내에서 반드시 필요한 것은 아니다.
- 옵션이 IP 헤더의 필수적인 부분은 아니지만, 옵션 처리 기능은 IP 소프트웨어의 필수 부분이다.

4. 옵션

■ 형식

- 그림은 옵션의 형식을 보여준다.
- 한 바이트의 유형 필드와 한 바이트의 길이 필드, 그리고 가변 길이의 값 필드로 구성된다.
- 이 세 개의 필드는 종종 TLV(Type-Length-Value) 라고도 불린다.



■ 형식

■ 유형(Type) 필드

- 복사(copy)
 - 이 한비트 부필드는 단편화에 옵션을 포함시킬 것인지를 제어한다.
 - 0 인 경우에 옵션은 첫 번째 단편에만 복사되어야 한다.
 - 1 인 경우에는 옵션이 모든 단편에 복사되어야 한다.
- 클래스(class)
 - 이 두 비트 부필드는 옵션의 일반적인 목적을 정의한다.
 - 00이면 옵션이 데이터그램의 제어에 사용된다는 것을 의미한다.
 - 10 인 경우에는 옵션이 디버그나 관리 목적이라는 것을 의미한다.
 - 다른 두 값인 01과 11은 아직 정의되지 않았다.
- 번호(number)
 - 이 다섯 비트의 부필드는 옵션의 유형을 정의한다.
 - 5비트는 32개의 서로 다른 유형을 정의할 수 있지만 현재 여섯 유형만 정의되어 있다.

■ 형식

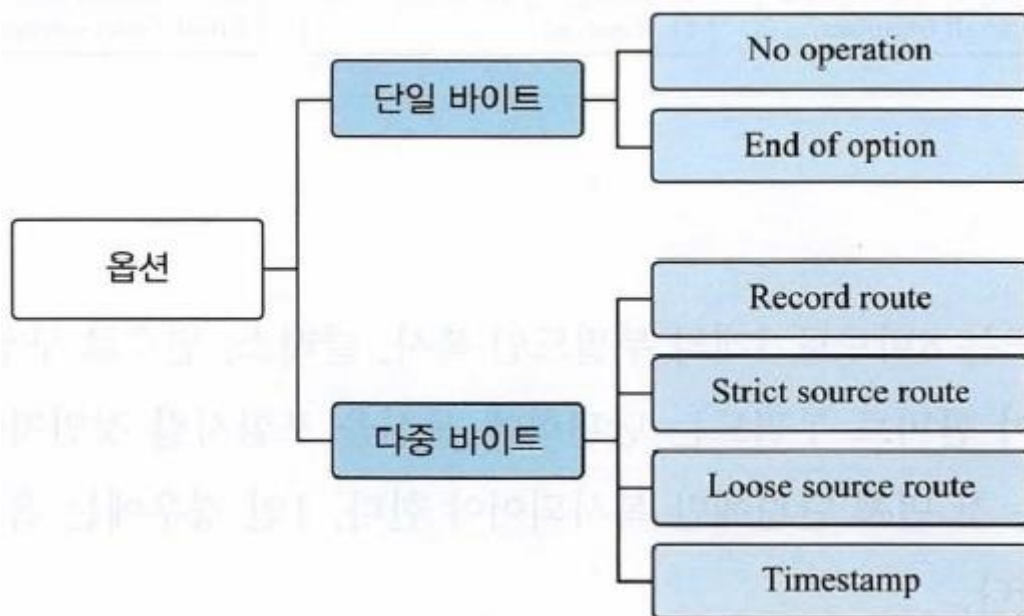
■ 길이(Length) 필드

- 복사(copy)
 - 길이(length) 필드는 유형 필드와 길이 필드를 포함한 옵션의 전체 길이를 정의한다.
 - 이 필드는 모든 옵션 유형에 있는 것은 아니다.
- 값(value)
 - 값(value) 필드는 특정 옵션이 필요로 하는 데이터 를 포함하고 있다.
 - 길이 필드와 마찬가지로 모든 옵션 유형에 있는 것은 아니다.

4. 옵션

■ 옵션 유형

- 이미 언급한 바와 같이 현재 6개의 옵션만이 정의되어 있다.
- 이 중 둘은 1 바이트 옵션이고 길이나 데이터 필드를 필요로 하지 않는다.
- 나머지 4개는 다중 바이트 옵션으로 길이와 데이터 필드를 필요로 한다.

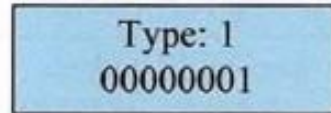


4. 옵션

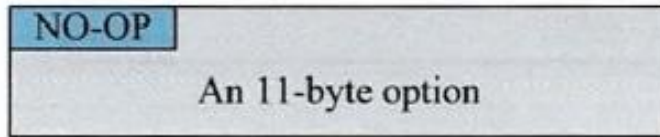
■ 옵션 유형

▪ 무연산 옵션(No operation)

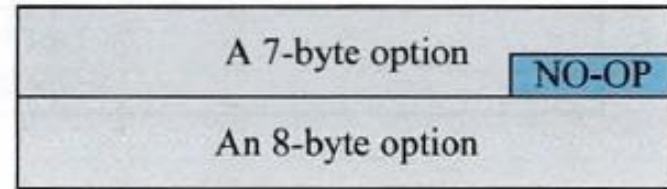
- 무연산(no operation) 옵션은 한 바이트 옵션으로 옵션들 사이의 여백을 채워준다.
- 예를 들어 다음 옵션을 16bit나 32bit 경계에 위치시키기 위해 사용될 수 있다.



a. 무연산 옵션



b. 옵션의 정렬 시작에 사용



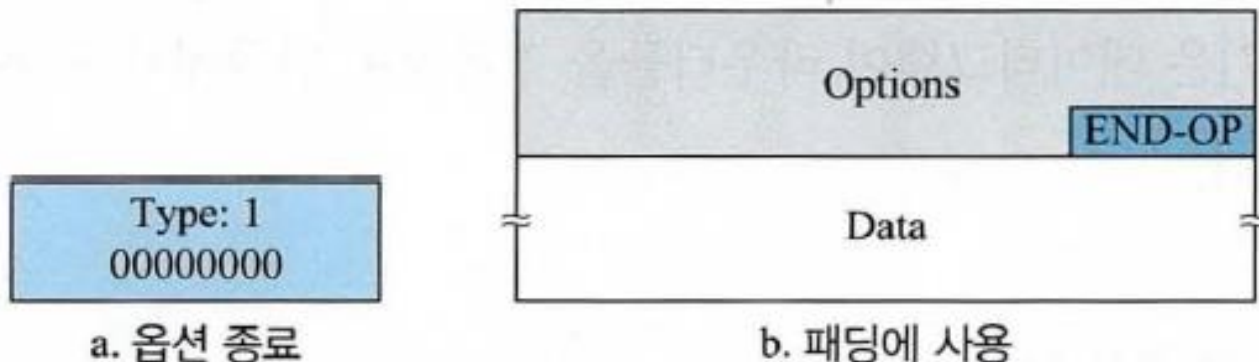
c. 다음 옵션 정렬에 사용

4. 옵션

■ 옵션 유형

■ 옵션 종료(end of option) 옵션

- 옵션 종료(end of option) 옵션은 한 바이트이고 옵션의 필드 끝의 패딩 목적으로 사용된다.
- 그러나 마지막 옵션으로만 사용될 수 있다.
- 오직 하나의 옵션 종료만 사용될 수 있다.
- 오직 하나의 옵션 종료만 사용될 수 있다.
- 이 옵션 이후에는 페이로드 데이터가 있다.
- 즉, 옵션 필드의 경계를 맞추기 위해 한 바이트 이상이 필요하다면 무연산 옵션이 사용된 후 마지막에 옵션 종료 옵션이 사용되어야 한다.

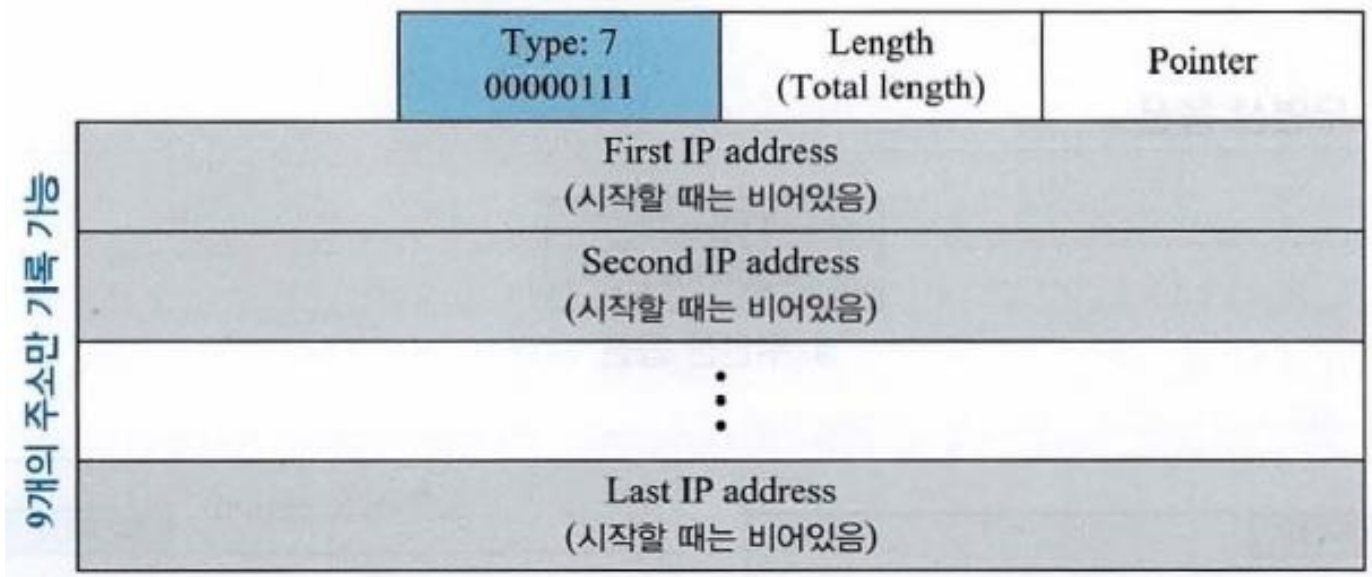


4. 옵션

■ 옵션 유형

▪ 경로 기록(record route) 옵션

- 경로 기록 옵션은 데이터그램을 처리한 인터넷 라우터들을 기록하기 위해서 사용된다.
- IP 데이터그램의 헤더 최대 길이가 60바이트이고, 이 중 20바이트는 기본 헤더이므로 최대 9개의 IP 주소까지 기록할 수 있다.
- 발신지는 방문되는 라우터에 의해 채워질 수 있는 공간을 미리 준비한다.
- 그림은 경로 기록 옵션의 형식을 보여준다.

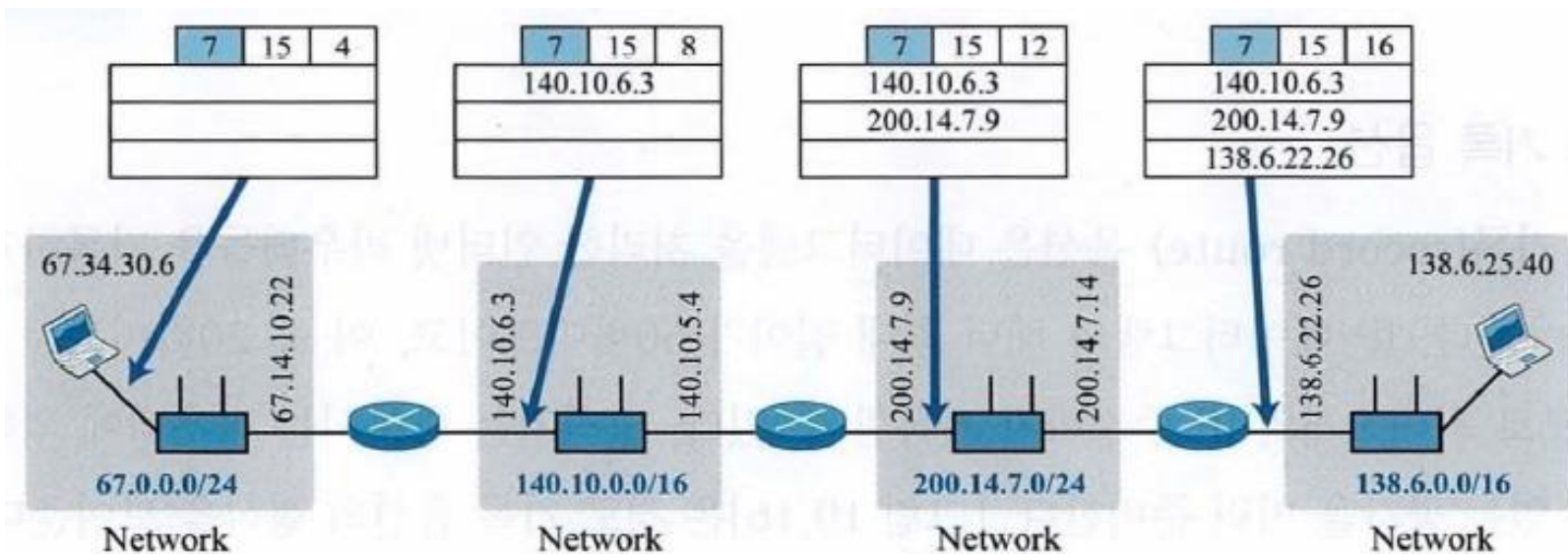


4. 옵션

■ 옵션 유형

▪ 경로 기록(record route) 옵션

- 포인터 필드는 첫 번째 빈 엔트리의 바이트 번호를 포함하는 옵션 정수 필드이다.
- 즉, 첫 번째 사용 가능한 엔트리를 가리키고 있다.
- 그림은 데이터그램이 라우터들을 경유하여 왼쪽에서 오른쪽으로 이동하는 과정을 설명하고 있다.



■ 옵션 유형

■ 엄격한 발신지 경로 (strict source route) 옵션

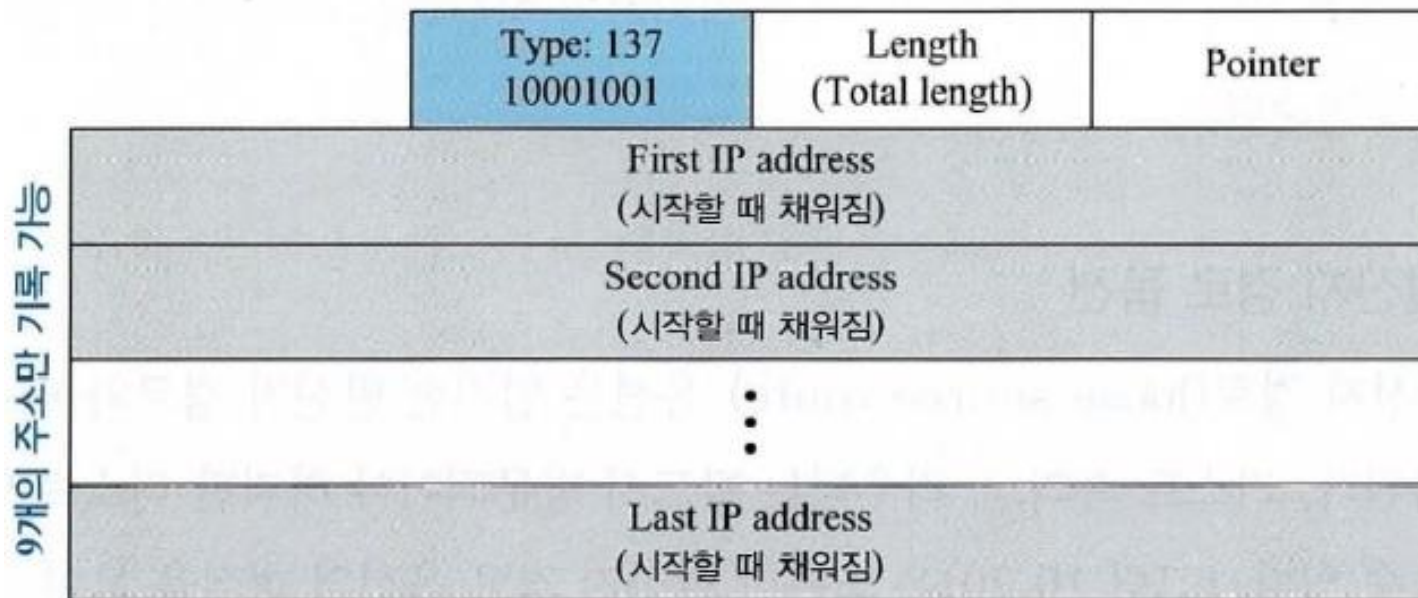
- 엄격한 발신지 경로(strict source route) 옵션은 데이터그램이 인터넷에서 거쳐야 할 경로를 미리 지정하기 위해 사용된다.
- 발신지에서 경로를 미리 지정하면 여러 경우에 유용할 수 있다.
- 발신지는 최소지연을 갖거나 최대 처리율을 제공하는 것과 같이 특별한 유형의 서비스를 제공하는 경로를 선택할 수 있다.
- 또는 더욱 안전하거나 신뢰성이 있는 경로를 선택할 수도 있다.
- 만약 데이터그램이 엄격한 발신지 경로를 지정하면 데이터그램은 옵션에 정의된 모든 라우터를 방문하여야 한다.
- 데이터그램에 주소가 없는 라우터는 방문해서는 안 된다.
- 만약 데이터그램이 리스트에 없는 라우터를 방문하면 데이터그램은 폐기되고 오류 메시지가 발송된다.
- 만약 데이터그램이 최종 목적지에 도착하였는데 리스트에 있는 라우터 중에 방문하지 않은 것이 있었다면 데이터그램은 폐기되고 오류 메시지가 보내진다.

4. 옵션

■ 옵션 유형

■ 엄격한 발신지 경로 (strict source route) 옵션

- 그림은 엄격한 발신지 경로 옵션의 형식을 보여주고 있다.
- 모든 IP 주소가 송신자에 의해 미리 지정되어 있다는 것을 제외하고는 경로 기록 옵션과 유사하다.

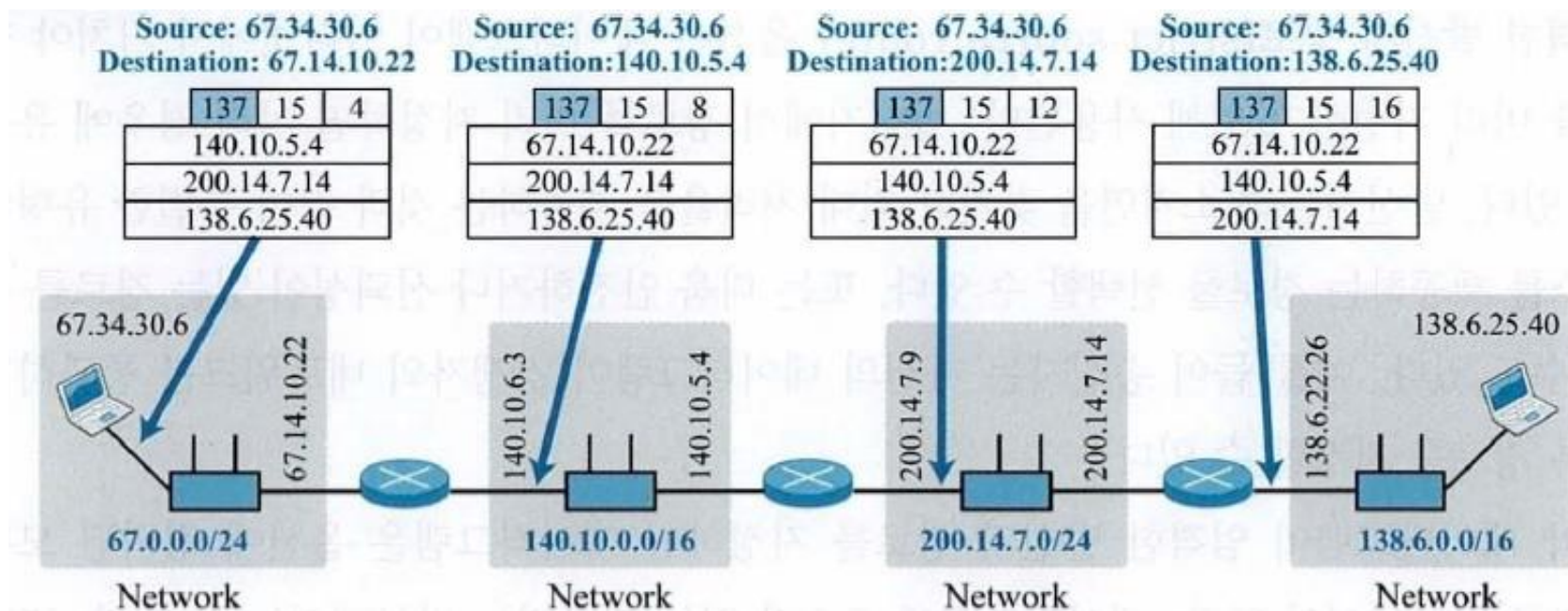


4. 옵션

■ 옵션 유형

■ 엄격한 발신지 경로 (strict source route) 옵션

- 모든 IP 주소가 송신자에 의해 미리 지정되어 있다는 것을 제외하고는 경로 기록 옵션과 유사하다.



■ 옵션 유형

■ 느슨한 발신지 경로 (loose source route) 옵션

- 느슨한 발신지 경로(loose source route) 옵션은 엄격한 발신지 경로와 비슷하나 제약이 조금 완화된다.
- 리스트 속의 각 라우터는 반드시 방문되어야 하지만 리스트에 없는 라우터도 방문할 수 있다.
- 그림은 느슨한 발신지 경로 옵션의 형식을 보여주고 있다.

9개의 주소만 기록 가능	Type: 131 10000011	Length (Total length)	Pointer
	First IP address (시작할 때 채워짐)		
	Second IP address (시작할 때 채워짐)		
	⋮		
	Last IP address (시작할 때 채워짐)		

4. 옵션

■ 옵션 유형

■ 타임 스탬프

- 타임스탬프(Time stamp) 옵션은 라우터가 데이터그램을 처리하는 시간을 기록하기 위해 사용된다.
- 시간은 세계 표준시 (Universal Time) 자정으로부터의 밀리초 단위로 표시된다.
- 데이터그램이 처리되는 시간을 알 수 있으면 사용자나 관리자가 인터넷상의 라우터들이 어떻게 동작하는지 추적하는 데 도움이 된다.
- 그림은 타임스탬프 옵션의 형식을 보여준다.

Code: 68 01000100	Length (Total length)	Pointer	O-Flow 4 bits	Flags 4 bits
First IP address				
Second IP address				
⋮				
Last IP address				

4. 옵션

■ 옵션 유형

■ 타임 스탬프

- 오버플로우(overflow) 필드는 필드가 더 이상 없기 때문에 타임스탬프를 기록하지 못한 라우터의 수를 기록한다.
- 플래그 필드는 방문된 라우터의 동작을 지정한다.
- 만약 0 이라면 각 라우터는 주어진 필드에 타임스탬프만 추가한다
- 1인 경우에는 라우터는 출력 인터페이스 IP 주소와 타임스탬프를 기록한다.
- 3 인 경우에는 라우터는 주어진 IP 주소와 입력 인터페이스 IP 주소를 비교한다.
- 만약 이 둘이 같으면 라우터는 IP 주소에 출력 인터페이스 IP 주소를 덮어쓰고 타임스탬프를 추가한다.

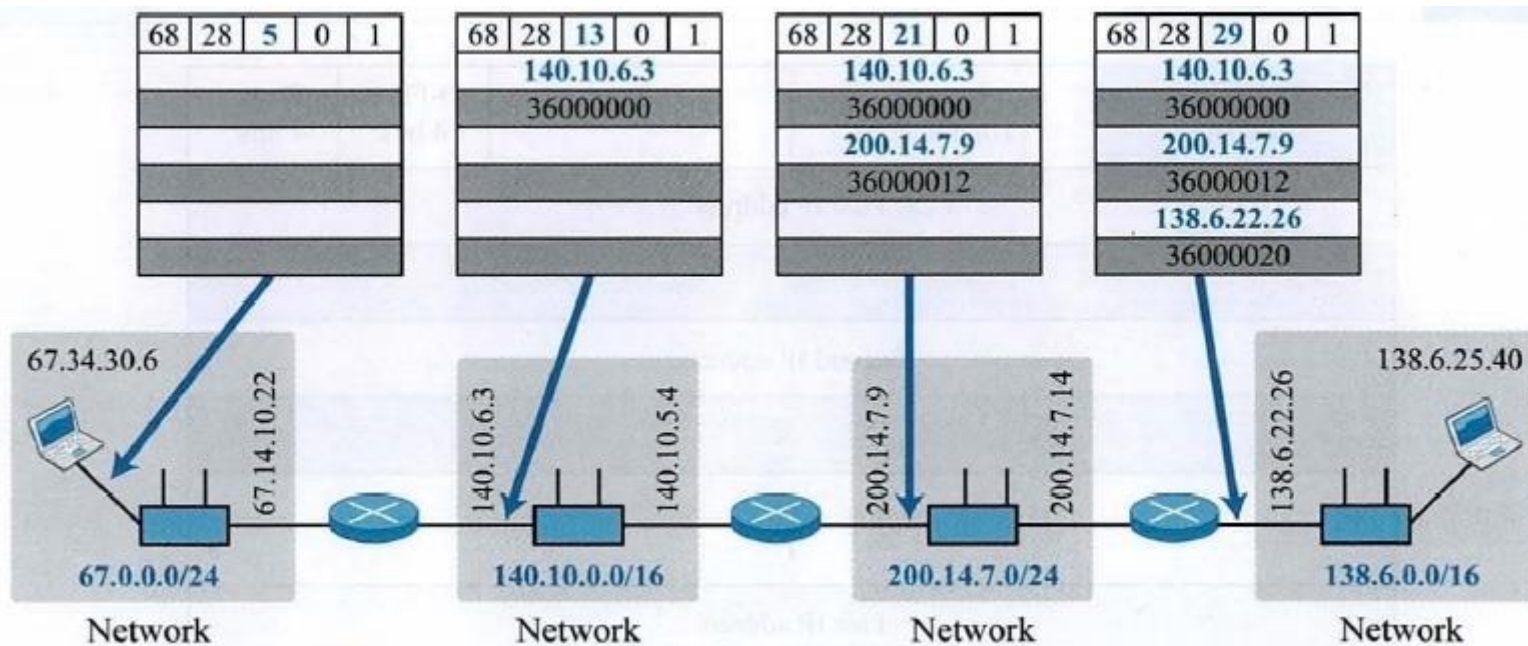


4. 옵션

■ 옵션 유형

■ 타임 스탬프

- 그림은 데이터그램이 발신지에서 목적지까지 이동하는 동안 각 라우터가 수행하는 동작을 보여주고 있다.
- 그림에서 플래그 값은 1 이라고 가정한다.



5. 검사합

- 대부분의 TCP/ IP 프로토콜에 의해 사용되는 오류 검출 방법은 검사합 (checksum) 이다.
- 이것은 패킷에 추가되는 중복된 정보이다.
- 검사합은 송신자에 의해 계산되고 패킷과 함께 전송된다.
- 수신자는 검사합을 포함하고 있는 전체 패킷에 대해 같은 계산을 반복한다.
- 결과가 만족되면 패킷은 받아들여지고 그렇지 않으면 폐기된다.

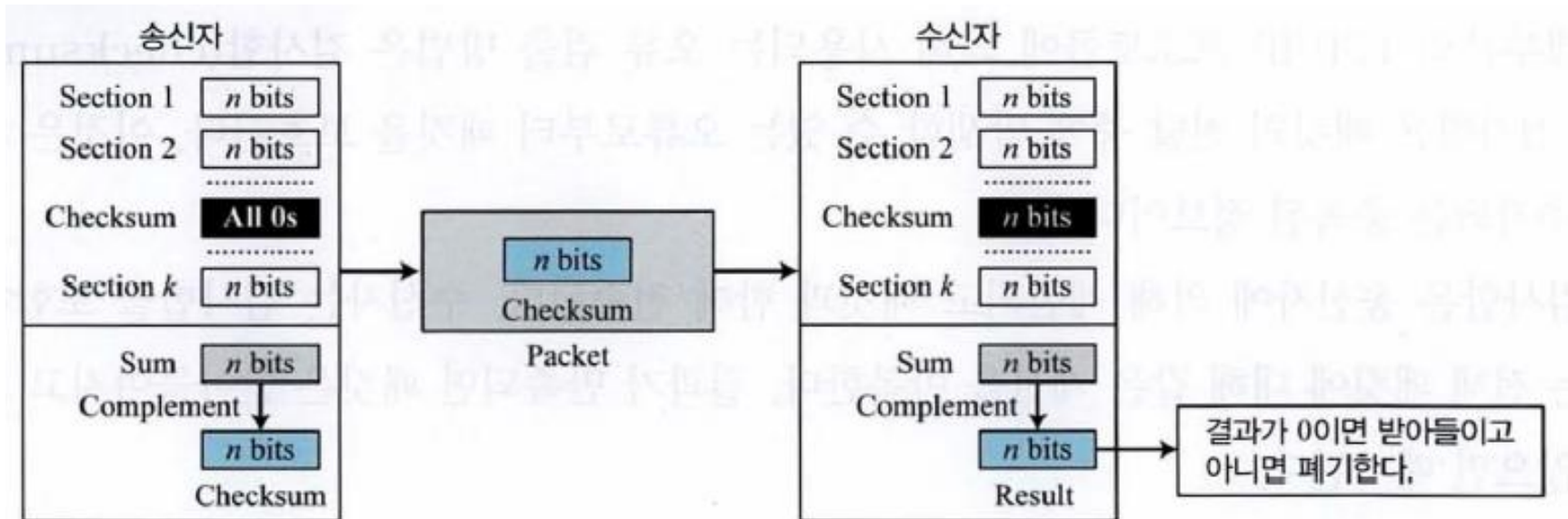
■ 송신자의 검사합 계산

- 송신자에서 패킷은 n 비트 단위로 나누어진다.
- 보통 n 은 16 이다.
- 이 조각들은 1 의 보수 연산을 사용하여 전부 더해져서 n 비트의 결과를 생성한다.
- 이 결과값의 0을 1 로, 1은 0으로 바꾸는 방법을 사용하여 이 결과에 대한 보수를 구하게 되는데 이 보수가 검사합이 된다.
- 검사합을 구하기 위하여 송신자는 다음을 행한다.
 - 패킷을 크기가 n 비트인 k 개의 조각으로 나눈다.
 - 모든 조각을 1 의 보수 연산을 사용하여 합한다.
 - 합에 대한 1 의 보수가 검사합이다.

5. 검사합

■ 수신자의 검사합 계산

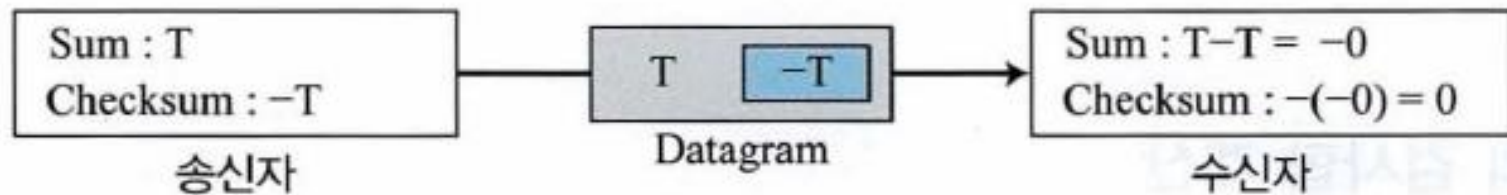
- 수신자는 수신된 패킷을 k 개의 n 비트 단위로 나눈 후 이들을 전부 합한다.
- 다음 이 합에 대한 1의 보수를 구한다.
- 만약 결과가 0이면 패킷을 받아들이고 그렇지 않으면 거부한다.
- 그림은 송신자와 수신자에서 발생하는 과정을 보여준다.



5. 검사합

■ 수신자의 검사합 계산

- 송신자에서 모든 조각을 더한 결과가 T라고 가정하자.
- 이 결과에 대한 1의 보수는 이 수에 대한 음의 수인 $-T$ 가 된다.
- 수신자는 패킷을 수신한 후 모든 조각을 합한다.
- 이것은 T와 $-T$ 를 합한 것과 같고 그 결과는 0이 된다.
- 다시 이에 대한 보수를 구하면 0이 된다.
- 그러므로 최종 결과가 0이면 받아들이고 그렇지 않으면 거부한다.



■ IP 패킷의 검사합

- IP 패킷 내의 검사합은 헤더만 포함하지 데이터는 포함하지 않는다.
- 여기에는 다음과 같은 두 개의 이유가 있다.
- 먼저 IP 데이터그램 내의 데이터를 캡슐화하는 모든 상위 계층 프로토콜은 전체 패킷을 포함하는 검사합을 가지고 있다.
- 그러므로 IP 데이터그램의 검사합은 캡슐화된 데이터를 점검할 필요가 없다.
- 두 번째로 IP 패킷의 헤더는 라우터를 방문할 때마다 변경될 수 있지만 데이터는 그렇지 않다.
- 그러므로 검사합은 변화되는 부분에 대해서만 구해야 한다.

5. 검사합

■ IP 패킷의 검사합

- 예제. 그림은 옵션이 없는 경우 발신지에서 IP 헤더에 검사합을 구하는 예를 보여주고 있다.
- 헤더는 16bit 단위로 나누어지고 이 단위들을 전부 합한 후 보수를 구한다.
- 그 결과는 검사합 필드에 삽입된다.

4, 5 그리고 0	→	01000101	00000000	
28	→	00000000	00011100	
1	→	00000000	00000001	
0 그리고 0	→	00000000	00000000	
4 그리고 17	→	00000100	00010001	
0	→	00000000	00000000	
10.12	→	00001010	00001100	
14.5	→	00001110	00000101	
12.6	→	00001100	00000110	
7.9	→	00000111	00001001	
Sum	→	01110100	01001110	
Checksum	→	10001011	10110001	

4	5	0	28
1		0	0
4	17	0	
10.12.14.5			
12.6.7.9			

35761

5. 검사합

■ IP 패킷의 검사합

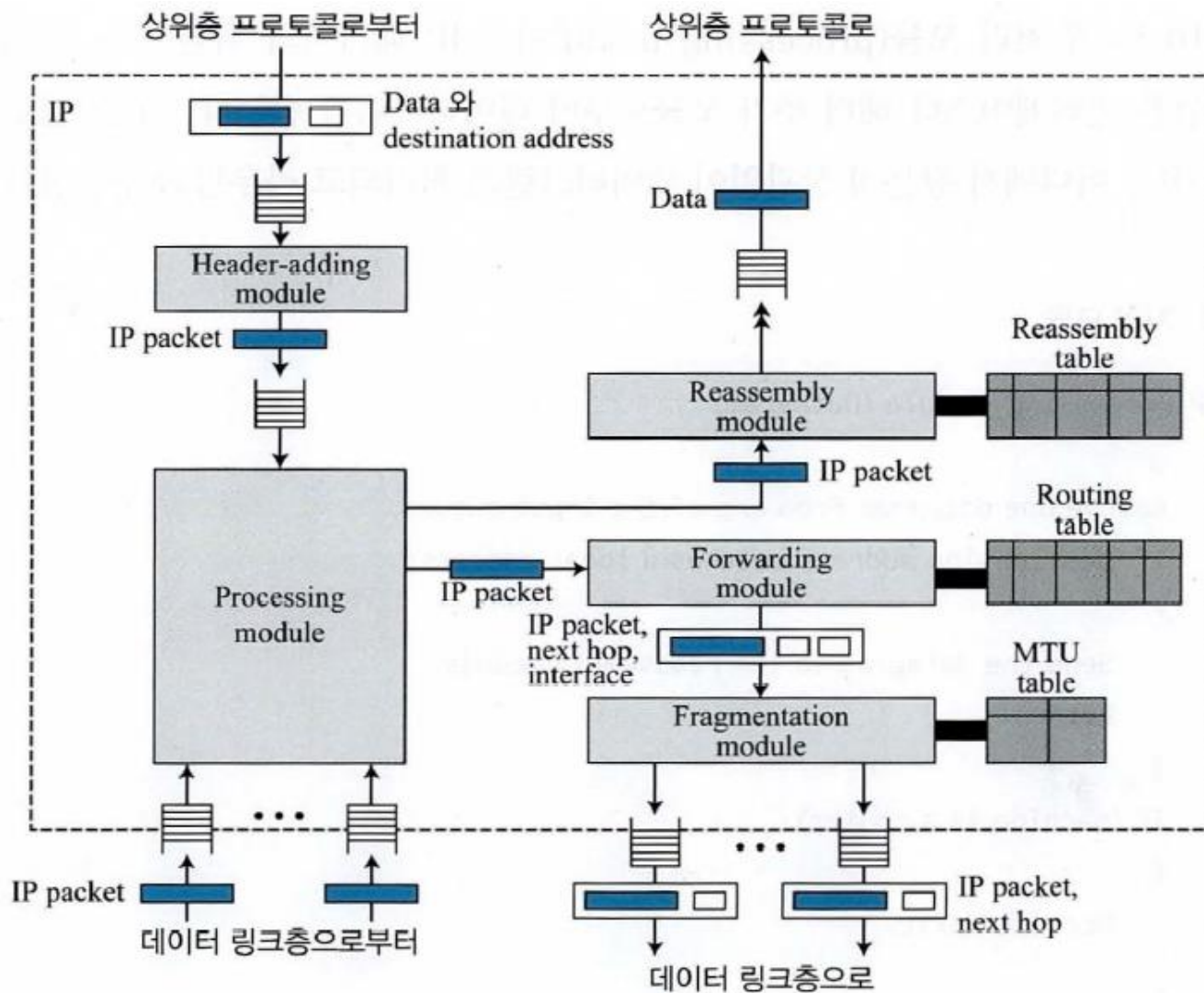
- 예제. 그림은 수신자 사이트(또는 중간 라우터)에서의 검사합의 점검 과정을 보여준다.
- 헤더는 16bit 단위로 나뉜다.
- 모든 단위를 전부 합한 후 보수를 구한다.
- 결과가 16개의 0이므로 패킷은 받아들여진다.

4, 5 그리고 0	→	01000101	00000000
28	→	00000000	00011100
1	→	00000000	00000001
0 그리고 0	→	00000000	00000000
4 그리고 17	→	00000100	00010001
35761	→	10001011	10110001
10.12	→	00001010	00001100
14.5	→	00001110	00000101
12.6	→	00001100	00000110
7.9	→	00000111	00001001
<hr/>			
Sum	→	1111 1111	1111 1111
Checksum	→	0000 0000	0000 0000

4	5	0	28	
1			0	0
4	17	35761		
10.12.14.5				
12.6.7.9				

6. IP 패키지

- 그림은 8개의 구성 요소와 이들 사이의 상호작용을 보여주고 있다.



6. IP 패키지

- IP 패키지에는 다음과 같은 8개의 구성 요소가 있다.
- 이들은 헤더 추가 모듈, 처리 모듈, 포워딩 모듈, 단편화모듈, 재조립 모듈, 라우팅 테이블, MTU 테이블, 재조립 테이블이다.
- 그리고 패키지는 입력 큐와 출력 큐도 포함한다.

■ 헤더 추가 모듈

- 표의 헤더 추가 모듈(header-adding module)은 상위 계층 프로토콜로부터 데이터와 목적지 IP 주소를 받은 뒤 IP 헤더를 더함으로써 데이터를 IP 데이터그램 내에 캡슐화한다.

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	}

6. IP 패키지

■ 처리 모듈

- 표의 처리 모듈(`processing module`)은 IP 패키지의 핵심이다.
- 이 설계에서 처리 모듈은 인터페이스나 헤더 추가 모듈로부터 데이터그램을 받는다.

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	}

6. IP 패키지

■ 처리 모듈

- 표의 처리 모듈(processing module)은 IP 패키지의 핵심이다.
- 이 설계에서 처리 모듈은 인터페이스나 헤더 추가 모듈로부터 데이터그램을 받는다.

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	}

■ 큐

- 설계에는 입력 큐(queue)와 출력 큐가 있다.
- 입력 큐는 데이터 링크층이나 상위 계층 프로토콜로부터 온 데이터그램을 저장하고 출력 큐는 데이터 링크층이나 상위 계층 프로토콜로 가는 데이터그램을 저장한다.
- 처리 모듈은 입력 큐로부터 데이터그램을 가져온다.
- 단편화 모듈과 재조립 모듈은 출력 큐에 데이터그램을 넣는다.

■ 라우팅 테이블

- 라우팅 테이블은 패킷의 다음 홉 주소를 결정하기 위하여 포워딩 모듈에 의해 사용된다.

■ 포워딩 모듈

- 포워딩 모듈은 처리 모듈로부터 IP 패킷을 받는다.
- 만약 패킷이 전달되어야 하면 패킷은 이 모듈에 보내져야 한다.
- 이 모듈은 보내져야 할 노드의 주소와 패킷이 보내져야 하는 인터페이스의 번호를 찾는다
- 그리고 이 정보와 함께 패킷을 단편화 모듈에 보낸다.

■ MTU 테이블

- MTU 테이블은 단편화 모듈이 특정 인터페이스의 MTU를 찾기 위해 사용된다.
- 이 테이블은 인터페이스와 MTU 열만을 가진다.

■ 단편화 모듈

- 표의 단편화 모듈(fragmentation module)은 포워딩 모듈로부터 IP 데이터그램을 받는다.

1	IP_Fragmentation_Module (datagram)
2	{
3	Extract the size of datagram
4	If (size > MTU of the corresponding network)
5	{
6	If (0 bit is set)
7	{
8	Discard datagram
9	Send an ICMP error message

■ 단편화 모듈

- 표의 단편화 모듈(fragmentation module)은 포워딩 모듈로부터 IP 데이터그램을 받는다.

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
18	Send fragment
19	return
20	}

■ 단편화 모듈

- 표의 단편화 모듈(fragmentation module)은 포워딩 모듈로부터 IP 데이터그램을 받는다.

21	Else
22	{
23	Send the datagram
24	}
25	Return.
26	}

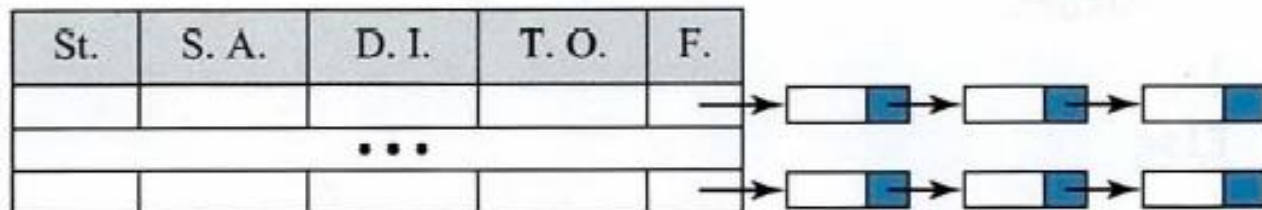
- 포워딩 모듈은 IP 데이터그램 다음 노드(직접 전달의 경우 최종목적지 그리고 간접 전달의 경우에는 다음 라우터)의 IP 주소와 데이터그램이 출력되어야 하는 인터페이스의 번호를 보낸다.
- 단편화 모듈은 MTU 테이블을 참조하여 해당하는 인터페이스 번호의 MTU를 찾는다.

6. IP 패키지

■ 재조립 테이블

- 재조립 테이블(reassembly table)은 재조립 모듈에 의하여 사용된다.
- 우리의 설계에서 재조립 테이블은 다음과 같은 다섯 개의 필드를 가지고 있다.
- 이들은 각각 상태(state), 발신지 IP 주소, 데이터그램 ID, 타임아웃(time-out), 단편(fragments) 필드이다.

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



- 상태 필드의 값은 FREE이거나 IN -USE일 수 있다.
- IP 주소 필드는 데이터그램의 발신지 IP 주소를 정의한다.
- 데이터그램 ID는 데이터그램과 이 데이터그램에 속하는 단편들을 유일하게 정의하는 번호이다.
- 타임아웃은 미리 결정된 시간으로 모든 단편이 이 시간 내에는 도착하여야 한다.
- 마지막으로 단편 필드는 단편들의 연결 리스트(linked list)에 대한 포인터이다.

■ 재조립 모듈

- 표의 재조립 모듈(reassembly module)은 처리 모듈로부터 최종목적지에 도착한 데이터그램 단편을 받는다.

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	}

■ 재조립 모듈

- 표의 재조립 모듈(reassembly module)은 처리 모듈로부터 최종목적지에 도착한 데이터그램 단편을 받는다.

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

■ 재조립 모듈

- 표의 재조립 모듈(reassembly module)은 처리 모듈로부터 최종목적지에 도착한 데이터그램 단편을 받는다.

25	Send an ICMP error message
26	}
27	}
28	Return.
29	}

6. IP 패키지

■ 재조립 모듈

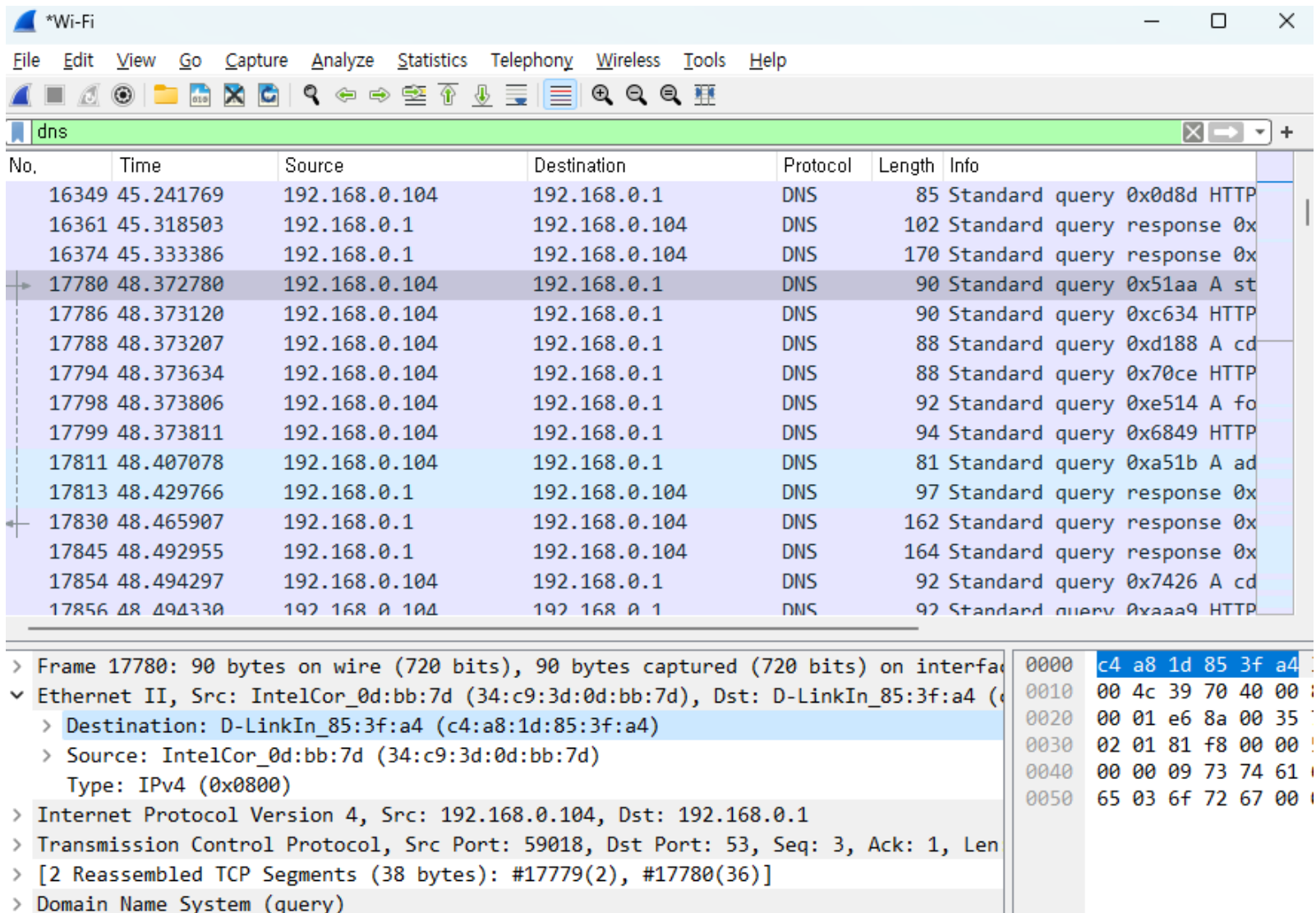
- 표의 재조립 모듈(reassembly module)은 처리 모듈로부터 최종목적지에 도착한 데이터그램 단편을 받는다.

25	Send an ICMP error message
26	}
27	}
28	Return.
29	}

- IP 프로토콜은 비연결형 프로토콜이므로 모든 단편이 순서대로 들어온다는 보장은 없다.
- 게다가 한 데이터그램의 단편은 다른 데이터그램의 단편과 섞일 수도 있다.
- 이 두 가지 문제점을 해결하기 위해 모듈은 앞에서 설명한 바와 같은 재조립 테이블 내의 연결 리스트를 사용한다.
- 재조립 모듈이 하는 일은 단편이 속한 데이터그램을 찾고 같은 데이터그램에 속한 단편의 순서를 맞추고 모든 단편이 다 도착한 후 한 데이터그램에 속한 모든 단편을 재조립하는 것이다.

7. 인터넷 프로토콜(IP) 덤프 분석

- 그림에 있는 17780 번 프레임의 IP를 덤프 분석해 보자.



*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns

No.	Time	Source	Destination	Protocol	Length	Info
16349	45.241769	192.168.0.104	192.168.0.1	DNS	85	Standard query 0x0d8d HTTP
16361	45.318503	192.168.0.1	192.168.0.104	DNS	102	Standard query response 0x
16374	45.333386	192.168.0.1	192.168.0.104	DNS	170	Standard query response 0x
17780	48.372780	192.168.0.104	192.168.0.1	DNS	90	Standard query 0x51aa A st
17786	48.373120	192.168.0.104	192.168.0.1	DNS	90	Standard query 0xc634 HTTP
17788	48.373207	192.168.0.104	192.168.0.1	DNS	88	Standard query 0xd188 A cd
17794	48.373634	192.168.0.104	192.168.0.1	DNS	88	Standard query 0x70ce HTTP
17798	48.373806	192.168.0.104	192.168.0.1	DNS	92	Standard query 0xe514 A fo
17799	48.373811	192.168.0.104	192.168.0.1	DNS	94	Standard query 0x6849 HTTP
17811	48.407078	192.168.0.104	192.168.0.1	DNS	81	Standard query 0xa51b A ad
17813	48.429766	192.168.0.1	192.168.0.104	DNS	97	Standard query response 0x
17830	48.465907	192.168.0.1	192.168.0.104	DNS	162	Standard query response 0x
17845	48.492955	192.168.0.1	192.168.0.104	DNS	164	Standard query response 0x
17854	48.494297	192.168.0.104	192.168.0.1	DNS	92	Standard query 0x7426 A cd
17856	48.494330	192.168.0.104	192.168.0.1	DNS	92	Standard query 0xaaa9 HTTP

> Frame 17780: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface

✓ Ethernet II, Src: IntelCor_0d:bb:7d (34:c9:3d:0d:bb:7d), Dst: D-LinkIn_85:3f:a4 (c4:a8:1d:85:3f:a4)

- > Destination: D-LinkIn_85:3f:a4 (c4:a8:1d:85:3f:a4)
- > Source: IntelCor_0d:bb:7d (34:c9:3d:0d:bb:7d)
Type: IPv4 (0x0800)
- > Internet Protocol Version 4, Src: 192.168.0.104, Dst: 192.168.0.1
- > Transmission Control Protocol, Src Port: 59018, Dst Port: 53, Seq: 3, Ack: 1, Len
- > [2 Reassembled TCP Segments (38 bytes): #17779(2), #17780(36)]
- > Domain Name System (query)

0000	c4 a8 1d 85 3f a4
0010	00 4c 39 70 40 00
0020	00 01 e6 8a 00 35
0030	02 01 81 f8 00 00
0040	00 00 09 73 74 61
0050	65 03 6f 72 67 00

7. 인터넷 프로토콜(IP) 덤프 분석

- 그럼 바로 Internet Protocol의 헤더 부분을 확인해 보자.
- 패킷 목록 정보에서 17780 번 프레임을 선택한 다음 패킷 상세 정보의 [Internet Protocol] 앞의 [>]를 클릭해보자.

```
> Frame 17780: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on inter-
> Ethernet II, Src: IntelCor_0d:bb:7d (34:c9:3d:0d:bb:7d), Dst: D-LinkIn_85:3f:a4
v Internet Protocol Version 4, Src: 192.168.0.104, Dst: 192.168.0.1
  0100 .... = Version: 4 ①
  .... 0101 = Header Length: 20 bytes (5) ②
  v Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) ③
    0000 00.. = Differentiated Services Codepoint: Default (0) ④
    ⑤ .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  ⑥ Total Length: 76
    Identification: 0x3970 (14704) ⑦
  v 010. .... = Flags: 0x2, Don't fragment ⑧
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0 ⑨
    Time to Live: 128 ⑩
  ⑪ Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled] ⑫
    [Header checksum status: Unverified] ⑬
    Source Address: 192.168.0.104 ⑭
  ⑮ Destination Address: 192.168.0.1
```

7. 인터넷 프로토콜(IP) 덤프 분석

- IP 헤더의 맨 앞에는 ① [Version] 필드가 있다.
- 이 필드는 4bit로서 IP 버전을 나타낸다.
- 이 패킷은 버전 4를 사용하고 있다.
- 다음의 ② [Header length] 필드는 IPv4의 헤더 길이를 나타낸다.
- 앞에서 IP 헤더는 가변 길이라고 설명했지만, 이 [Header length] 필드의 값을 이용하여 IP 헤더의 길이를 지정한다.
- 그러나 이 필드는 4bit로 헤더 길이를 나타내고 있다.
- 따라서 4 바이트(3 2bit) 단위로 헤더의 길이를 나타내게 되어 있다.

7. 인터넷 프로토콜(IP) 덤프 분석

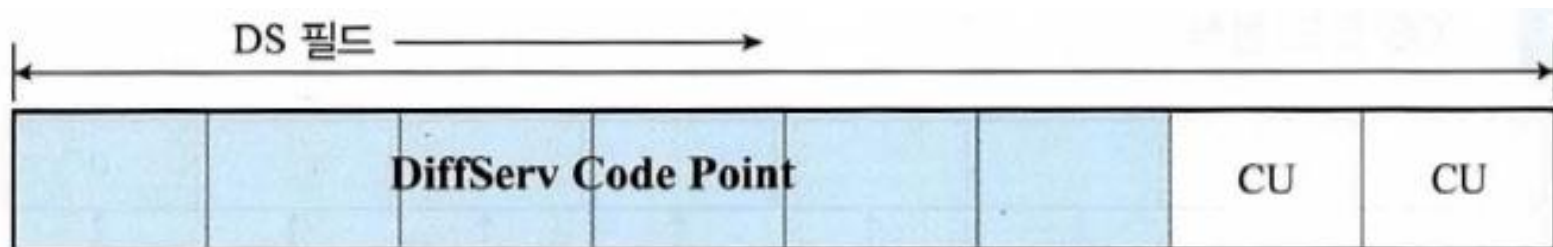
- 여기서 패킷 바이트 정보를 살펴보자.
- 화면 예에서 "45"란 값이 나타나 있다.
- "45"에서 "4"는 IP 버전 "5"는 헤더 길이를 나타낸다.
- 4바이트(32bit) x 5 = 20이 되므로 헤더 길이는 20바이트임을 알 수 있다.

```
> Frame 17780: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface
> Ethernet II, Src: IntelCor_0d:bb:7d (34:c9:3d:0d:bb:7d), Dst: D-LinkIn_85:3f:a4 (c
v Internet Protocol Version 4, Src: 192.168.0.104, Dst: 192.168.0.1
  0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  v Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 76
  Identification: 0x3970 (14704)
v 010. .... = Flags: 0x2, Don't fragment
  0... .... = Reserved bit: Not set
  .1.. .... = Don't fragment: Set
  ..0. .... = More fragments: Not set
```

0000	c4 a8 1d 85 3f a4 34 c9 3d 0d bb 7d 08 00 45 00?.4. =..
0010	00 4c 39 70 40 00 80 06 00 00 c0 a8 00 68 c0 a8	..L9p@... ..
0020	00 01 e6 8a 00 35 7c ec 28 84 5c 79 dd dd 50 185 . (..
0030	02 01 81 f8 00 00 51 aa 01 00 00 01 00 00 00 00Q.
0040	00 00 09 73 74 61 6e 64 61 72 64 73 04 69 65 65	...stand arc
0050	65 03 6f 72 67 00 00 01 00 01	e.org... ..

7. 인터넷 프로토콜(IP) 덤프 분석

- 이어서 ③[**Differentiated Services Field**]에는 1 바이트 값이 나타나며 [>]를 클릭하면 더 자세하게 전개되어 나타난다.
- 이 [**Differentiated Services Field**]는 일반적으로 [**DiffServ**]라고 하며 패킷의 대역을 제어하는 QoS(Quality of Service : 서비스 품질 제어)에 이용된다.
- IP의 QoS에 의해 예를 들어 서버용 패킷을 우선적으로 처리하거나 VoIP 통신은 최악의 경우 폐기 처리할 수 있다.
- 그런데 이 [**Differentiated Services Field**]는 1 바이트 길이이지만 앞쪽 6bit로 ④[**DSCP(DiffServ Code Point)**]라는 패킷을 식별하는 필드를 나타낸다.



(DiffServ Code Point : 6bit CU : 미사용 2bit)

7. 인터넷 프로토콜(IP) 덤프 분석

- DSCP는 패킷마다 전송 우선순위나 지연 정도를 [클래스] 단위로 분류한다.
- 패킷 단위에서 이와 같은 우선 제어나 지연 처리 등을 하는 것을 [PHB(Per Hop Behavior)]라고 하며 이 처리는 보통 라우터나 3 계층 스위치, 로드 밸런서(부하 분산 장치) 등에서 이루어진다.
- 또 DSCP에 값을 설정하는 것을 [마킹 (marking)]이라고 하는데, 이 마킹된 패킷의 DSCP값을 바탕으로 라우터는 대역 제어를 한다.
- DSCP 값은 DS(Differentiated Services) 필드의 앞쪽 6bit를 의미한다.
- 또한 PHB(Per Hop Behavior) 값은 DSCP 값에 의해 지정되는 패킷 단위의 우선 순위나 품질 제어(클래스)를 지정한다.

7. 인터넷 프로토콜(IP) 덤프 분석

■ DSCP 필드에서 우선순위 비트의 주요 값과 PHB

DSCP 값	PHB(Per Hop Behavior)
000000	표준(디폴트) PHB, 최선 노력 /일반
001000	클래스 선택자 PHB(우선순위 1 : 우선)
010000	클래스 선택자 PHB(우선순위 2 : 즉시)
011000	클래스 선택자 PHB(우선순위 3 : 고속)
100000	클래스 선택자 PHB(우선순위 4 : 초고속)
101000	클래스 선택자 PHB(우선순위 5 : 긴급)
110000	클래스 선택자 PHB(우선순위 6 : 네트워크간 연결 제어)
111000	클래스 선택자 PHB(우선순위 7 : 네트워크 제어)
101110	EF(Expedited Forwarding) PHB 최우선, 저손실 처리

7. 인터넷 프로토콜(IP) 덤프 분석

- DS(Differentiated Services) 필드에 대해 좀 더 살펴보자.
- 이 필드는 예전에는 [Type Of Service(TOS)]라고 불리었고 실제로 현재에도 TOS 필드로서 사용될 수가 있다.
- TOS 필드는 통신 품질을 지정하는 1 바이트 필드로서 맨 앞 3bit로 패킷의 우선 순위를 나타낸다.
- 또 다음 4bit로 서비스 종류를 나타낸다.
- 그리고 마지막 1bit로 예약이 끝났음을 의미하는 “0” 이 들어간다.



7. 인터넷 프로토콜(IP) 덤프 분석

- Differentiated Services Field나 TOS 필드는 계층3에서 통신 품질 제어 (QoS)를 하기 위해 사용된다.
- 예를 들어 하나의 회선에서 중요한 통신과 그렇지 않은 통신을 나눠 보낼 수 있다.
- TOS 필드의 우선순위 비트 값

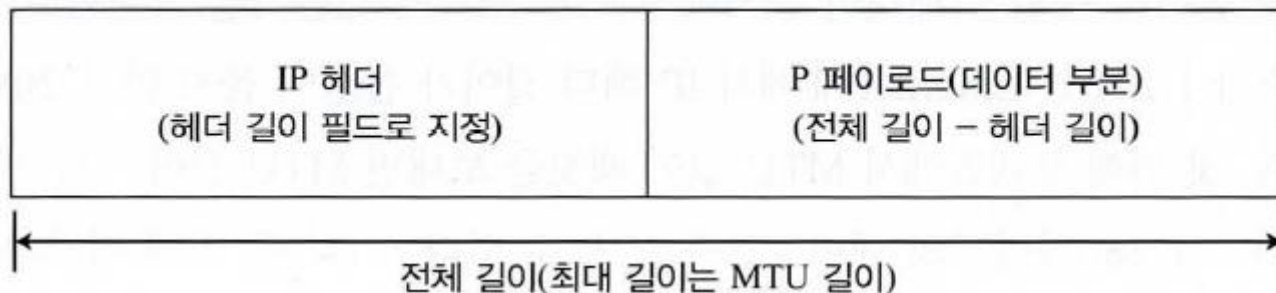
값	우선순위
000	표준
001	우선
010	즉시
011	고속
100	초고속
101	긴급
110	네트워크간 연결 제어
111	네트워크 제어

7. 인터넷 프로토콜(IP) 덤프 분석

- 기존의 TOS 필드에 의한 IntServ에서는 IP의 종단간 통신 단위로 우선순위 제어를 한다.
- 이에 비해 DSCP를 이용한 DiffServ에 의한 방법에서는 라우터나 접속 장치가 패킷마다 DSCP 헤더(라벨)를 붙여 우선순위 제어를 한다.
- 우선 통신을 하는 클라이언트와 서버 사이에서 IntServ용 신호방식 프로토콜인 RSVP를 사용하며 미리 필요한 네트워크 자원(대역)을 예약해 둔다.
- 그리고 이 예약을 바탕으로 라우터가 대역을 확보한다.
- IntServ는 일정한 대역을 보증하는 [보장형 서비스(GS: Guaranteed Service)]와 최선 노력으로 처리하는 [부하 제어형 서비스(CL : Controlled Load Service)]가 규정되어 있다.

7. 인터넷 프로토콜(IP) 덤프 분석

- DSCP 다음의 ⑤ [Explicit Congestion Notification]의 2bit는 IPv4에서 혼잡을 통지하는 ECN 기능 때문에 이용된다.
- 또한 ⑥ [Total Length] 필드에서는 IP 패킷의 전체 길이가 2바이트로 나타낸다.
- IP 패킷은 Ethernet II 뿐만이 아니라 브로드밴드에서 이용하는 PPPoE나 PPP 등과 같은 여러 가지 2계층 프로토콜을 사용할 수 있다.
- 따라서 IP 패킷은 데이터 링크층에 따라 가변 패킷 길이로 되어 있다.
- Total Length(전체 길이)에 따라 IP 패킷은 이론상 최대로 65,535 바이트(FF)까지 길이를 지정할 수 있다.
- 또한 Total Length에서 Header Length를 뺀으로써 IP 패킷의 내용(페이로드) 길이를 알 수 있다.



7. 인터넷 프로토콜(IP) 덤프 분석

- [Total Length] 필드에 이어 ⑦ [Identification]은 2바이트 필드로 송신하는 패킷을 식별하기 위한 값이다.
- 이 값은 송신하는 PC에 의해 설정되며 연속하여 IP 데이터그램을 보낼 경우에는 값을 늘린다.
- 이로써 발신지와 목적지가 같고 내용도 같은 패킷을 구별하도록 하고 있다.
- 다음으로 ⑧ [Flags] 필드가 이어진다.
- [Flags] 필드는 그 뒤에 이어지는 [Fragment Offset] 필드와 함께 단편화 · 재조립 기능에 이용된다.

7. 인터넷 프로토콜(IP) 덤프 분석

- 이제 [Flags] 필드를 확인해 보기로 하자.

```
✓ 010. .... = Flags: 0x2, Don't fragment
  0... .... = Reserved bit: Not set
  .1... .... = Don't fragment: Set
  ..0. .... = More fragments: Not set
```

- 처음 플래그는 Reserved bit(예약 비트)이다.
- 그 다음이 DF(Don't Fragment) 비트이다.
- DF 비트가 ON(="1") 이 되면 단편화는 금지되며 단편화가 발생하면 패킷은 폐기된다.
- 그 다음으로 MF(More Fragments) 비트가 이어진다.
- MF 비트는 단편화가 금지되어 있지 않은 경우(DF="0")에 후속 단편화가 있는지 여부를 구별하기 위해 이용된다.
- 분할된 패킷이 이 다음에 올 경우 MF 비트가 ON이 된다.

7. 인터넷 프로토콜(IP) 덤프 분석

- 이 플래그와 함께 이용되는 것이 ⑨ [단편화 오프셋(Fragment Offset)]이란 13bit 필드이다.
- 단편화 오프셋은 분할된 패킷이 전체에서 어느 위치에 있는가를 지정한다.
- 단편화 오프셋은 13bit로 구성된다.
- IP 데이터그램은 최대 65,535바이트(2^{16} 바이트)인 것에 반해 그 안에서의 위치(몇 바이트짜인가)를 13bit로 지정하기 위해 단편화 오프셋의 값에 4바이트를 곱한 값이 실제 페이로드의 위치가 된다.
- 다음으로 ⑩ [Time To Live]는 TTL이라 불리며 IP 패킷의 수명을 나타내는 1 바이트 필드이다.
- TTL 값은 송신 측에서 맨 처음 설정하며 라우터나 계층3 스위치 등의 중계 장치를 통과할 때마다 1씩 감소된다.
- 그리고 통신 도중에 TTL 값이 "0"이 되면 그 패킷은 폐기하게 되어 있다.

7. 인터넷 프로토콜(IP) 덤프 분석

- 또한 ⑪ [protocol] 필드는 IP 다음 계층의 헤더를 1 바이트로 나타낸다.
- [Protocol] 필드의 대표적인 값으로 “6”은 TCP, “17”은 UDP 등이 있다.
- 표처럼 IP는 4계층 이후의 여러 프로토콜을 페이로드(payload)로써 전달할 수 있다.

프로토콜 번호	캡슐화된 프로토콜
1	ICMP
6	TCP
17	UDP
50	ESP

7. 인터넷 프로토콜(IP) 덤프 분석

- 이어지는 ⑫ [Header Checksum]은 IP 헤더의 내용을 확인하기 위한 필드로 헤더에 오류가 없는지를 검사한다.
- 앞서 설명한 바와 같이 현재 와이어샤크는 디폴트에서는 검사합 계산을 하지 않는다.
- 따라서 검사합 정보에 [validation disabled]라고 나타냄과 동시에 아래의 줄에 ⑬ [Header Checksum : Status Unverified]라고 나타난다.
- 검사합에 문제가 없으면 와이어샤크가 [correct](올바르다)를 추가한다.
- 또한 여기에 와이어샤크의 판단으로 [Good : True](정상) 및 [Bad: False](이상)가 추가된다.
- 그 다음은 ⑭ [Source] 필드의 발신지 IP 주소와 ⑮ [Destination] 필드의 목적지 IP 주소이다.
- IP 주소는 발신지와 목적지 모두 4바이트로 지정된다.
- 16진수의 1바이트를 10진수로 고치고 점으로 구분해 나타낸다.



Thank You
