

IP 프로토콜

편집: 홍익대학교(세종) 김혜영

01_네트워크 계층의 기능

❖ 네트워크

■ 라우팅

- 라우팅 테이블 : 네트워크 구성 형태에 관한 정보를 관리
- 라우팅 : 송수신 호스트 사이의 패킷 전달 경로를 선택하는 과정

■ 혼잡 제어

- 혼잡 : 네트워크에 패킷 수가 과도하게 증가되는 현상
- 혼잡 제어 : 혼잡의 발생을 예방하거나 제거하는 기능

■ 패킷의 분할과 병합

- 상위 전송 계층에서 송신을 요구한 데이터는 최종적으로 MAC 계층의 프레임 구조에 정의
- 된 형식으로 캡슐화되어 물리적으로 전송
- 패킷 분할 : 데이터를 여러 패킷으로 나누는 과정
- 패킷 병합 : 목적지에서 분할된 패킷을 다시 모으는 과정



01_네트워크 계층의 기능

❖ 연결형 서비스와 비연결형 서비스

- 연결형: 데이터 전송 전에 데이터의 전송 경로를 미리 결정
- 비연결형: 데이터의 전송 경로를 사전에 결정하지 않고 패킷 단위로 결정



그림 7-1 연결형 · 비연결형 서비스



01_네트워크 계층의 기능

- 비연결형 서비스 Connectionless Service
 - 패킷의 전달 순서
 - 패킷이 서로 다른 경로로 전송되므로 도착 순서가 일정하지 않음
 - 상위 계층에서 순서를 재조정해야 함
 - 패킷 분실 가능성
 - 패킷의 100% 도착을 보장하지 않음
 - 상위 계층에서 패킷 분실 오류를 복구해야 함
 - 인터넷 환경의 예
 - IP : 네트워크 계층의 기능을 지원하는 비연결형 프로토콜
 - UDP : 전송 계층의 기능을 지원하는 비연결형 프로토콜
- 연결형 서비스 Connection-oriented Service
 - 상대적으로 신뢰성이 높음
 - TCP : 전송 계층의 기능을 지원하는 연결형 프로토콜



❖ 라우팅 Routing

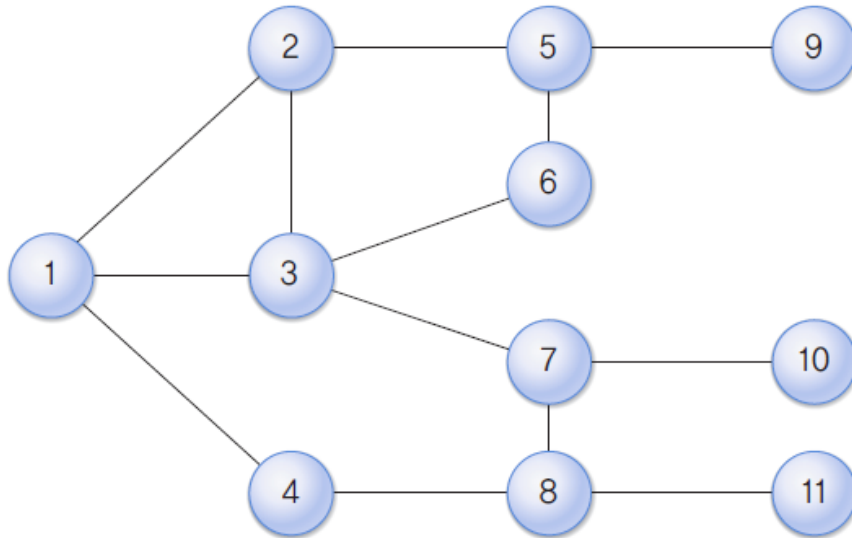
- 패킷의 전송 경로를 지정
- 전송 경로 결정시 고려 사항
 - 공평 원칙 : 다른 패킷의 우선 처리를 위해 다른 패킷이 손해를 보면 안됨
 - 효율 원칙 : 전체 네트워크의 효율성에 대해 고려해야 함
- 정적/동적 라우팅
 - 정적 라우팅 Static Routing
 - 패킷 전송이 이루어지기 전에 경로 정보를 라우터가 미리 저장하여 중개
 - 단점 : 경로 정보의 갱신이 어려우므로, 네트워크 변화/네트워크 혼잡도 대처 부족
 - 동적 라우팅 Dynamic Routing
 - 라우터의 경로 정보가 네트워크 상황에 따라 적절히 조절됨
 - 단점 : 경로 정보의 수집과 관리로 인한 성능 저하
- HELLO/ECHO 패킷
 - HELLO: 주변 라우터에 HELLO 패킷을 보내어 주변 경로 정보를 파악하는 용도
 - ECHO: 라우터 사이의 전송 지연 시간을 측정하는 용도



01_네트워크 계층의 기능

■ 라우팅 테이블 Routing Table

- 패킷 전송 과정에서 라우터들이 경로를 쉽게 찾도록 하는 가장 기본적인 도구
- 필수 정보 : 목적지 호스트, 다음 홉
 - 목적지 호스트 : 패킷의 최종 목적지가 되는 호스트 주소
 - 다음 홉 : 목적지 호스트까지 패킷을 전달하기 위한 인접 경로



(a) 네트워크 연결 구성의 예

그림 7-2 라우팅 테이블

목적지	홉
1	-
2	2
3	3
4	4
5	2
6	3
7	3
8	4
9	2
10	3
11	4

(b) 호스트 1의 라우팅 테이블



01_네트워크 계층의 기능

■ 라우팅 정보의 처리

● 소스 라우팅 Source Routing

- 패킷을 전송하는 호스트가 목적지 호스트까지 전달 경로를 스스로 결정하는 방식
- 경로 정보를 전송 패킷에 기록함
- 데이터그램 방식과 가상 회선 방식에서 모두 이용함

● 분산 라우팅 Distributed Routing

- 라우팅 정보가 분산되는 방식, 패킷의 전송 경로에 위치한 각 라우터가 경로 선택에 참여함
- 네트워크에 존재하는 호스트의 수가 많아질수록 다른 방식보다 효과적일 수 있음

● 중앙 라우팅 Centralized Routing

- RCC라는 특별한 호스트를 사용해 전송 경로에 관한 모든 정보를 관리하는 방식
- RCC로부터 목적지 호스트까지 도착하기 위한 경로 정보를 미리 얻음
- 장점 : 경로 정보를 특정 호스트가 관리하기 때문에 경로 정보를 관리부담이 줄어듦
- 단점 : RCC에 과중한 트래픽을 주어 전체 효율이 떨어짐

● 계층 라우팅 Hierarchical Routing

- 분산 라우팅 기능과 중앙 라우팅 기능을 적절히 조합하는 방식
- 네트워크 규모가 계속 커지는 환경에 효과적



01_네트워크 계층의 기능

❖ 혼잡 제어

- 혼잡^{Congestion} : 네트워크 성능 감소 현상이 급격하게 악화되는 현상
- 혼잡 제어^{Congestion Control} : 혼잡 문제를 해결하기 위한 방안
 - 흐름 제어 : 송신,수신 호스트 사이의 논리적인 점대점 전송 속도를 다룸
 - 혼잡 제어 : 서브넷에서 네트워크의 전송 능력 문제를 다룸



그림 7-3 흐름 제어와 혼잡 제어



01_네트워크 계층의 기능

■ 혼잡의 원인

- 초기 혼잡 과정에서 타임 아웃 시간이 작으면 혼잡도가 급격히 증가
- 패킷 도착 순서가 다른 상황에서 패킷을 분실 처리하면 타임아웃 증가
- 의도적으로 피기배킹을 사용하면 응답 시간이 느려져 타임아웃 증가
- 패킷 생존 시간을 작게 하면 패킷이 강제로 제거되어 타임아웃 증가

• 라우팅 알고리즘

- 혼잡이 발생하지 않는 경로를 배정하도록 설계
- 혼잡이 발생하는 경로를 선택하면 혼잡이 주변으로 확대됨



01_네트워크 계층의 기능

■ 트래픽 성형

- 혼잡은 트래픽이 특정 시간에 집중되는 버스트현상이 원인
- 트래픽 성형Traffic Shaping : 송신 호스트가 전송하는 패킷의 발생 빈도가 네트워크에서 예측할 수 있는 전송률로 이루어지게 하는 기능
- 리키 버킷Leaky Bucket 알고리즘

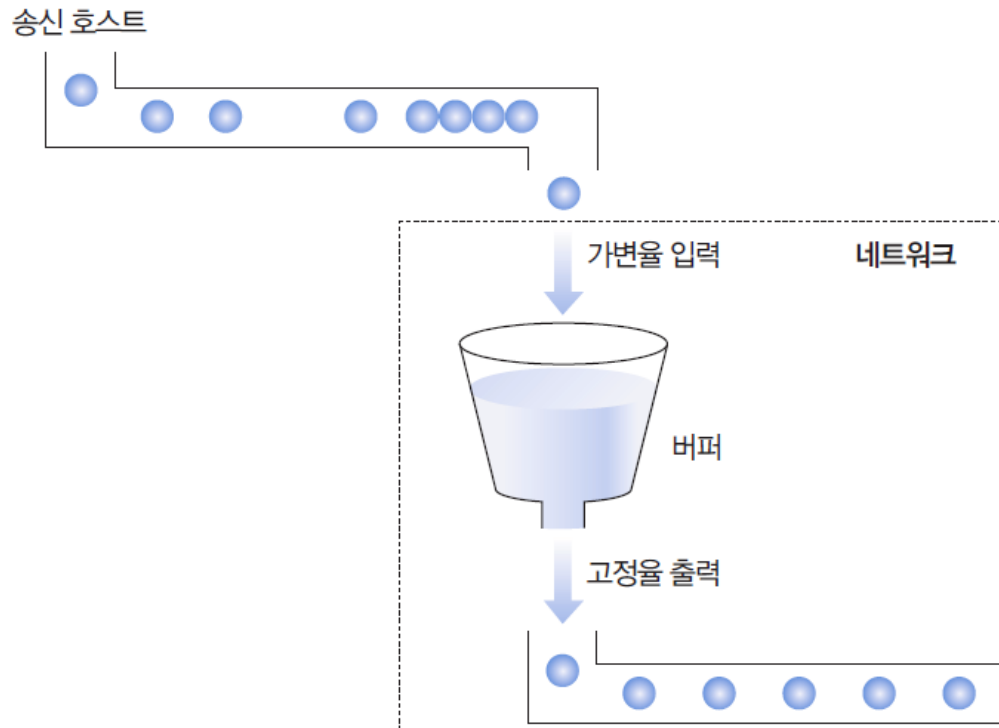


그림 7-4 리키 버킷 알고리즘



01_네트워크 계층의 기능

■ 혼잡 제거

- 특정 지역의 혼잡이 다른 지역으로 확대되지 않도록 하는 것이 중요
- 혼잡 제거를 위해 호스트와 서브넷이 가상 회선 연결 과정에서 협상을 함 (자원 예약 방식)
 - 네트워크에서 수용 불가능한 정도로 트래픽이 발생하는 일을 사전에 예방함
 - 단점 : 전송 대역을 해당 사용자가 이용하지 않더라도 다른 사용자가 이용하지 못함
- ECN^{Explicit Congestion Notification} 패킷
 - 라우터는 트래픽의 양을 모니터해 출력 선로의 사용 정도가 한계치를 초과하면 주의 표시를 함
 - 주의 표시한 방향의 경로는 혼잡이 발생할 가능성이 높기 때문에 특별 관리함

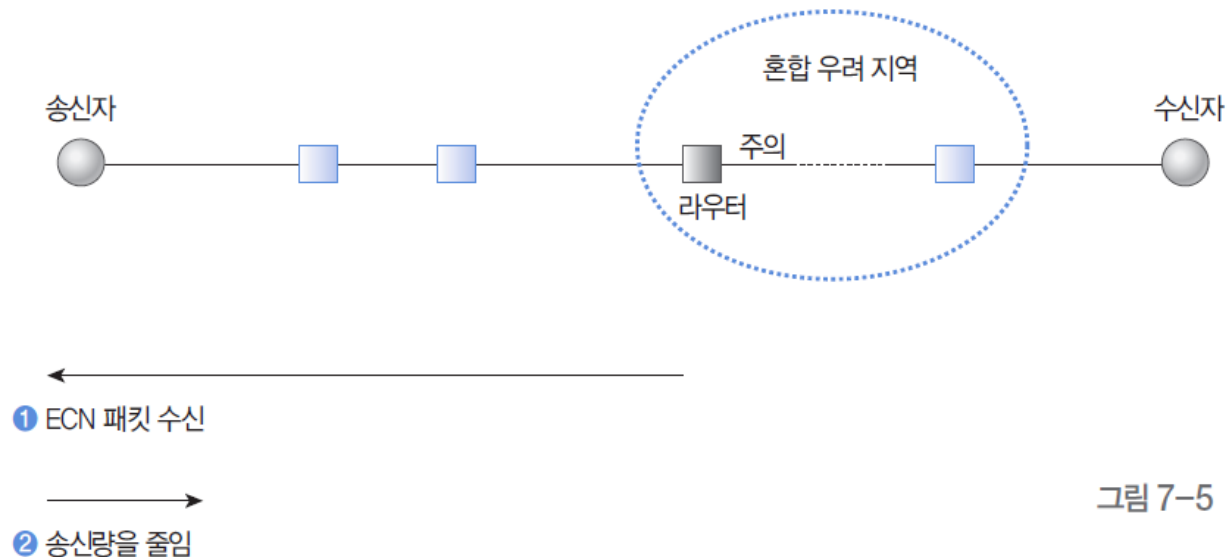


그림 7-5 ECN 패킷



❖ 간단한 라우팅 프로토콜

- 네트워크 거리 기준 : 라우터의 개수, 홉 Hop의 수로 판단
- 최단 경로 라우팅
 - 패킷이 목적지에 도달할 때까지 라우터 수가 최소화될 수 있도록 경로 선택
 - 장점 : 간단한 형식으로 적용가능

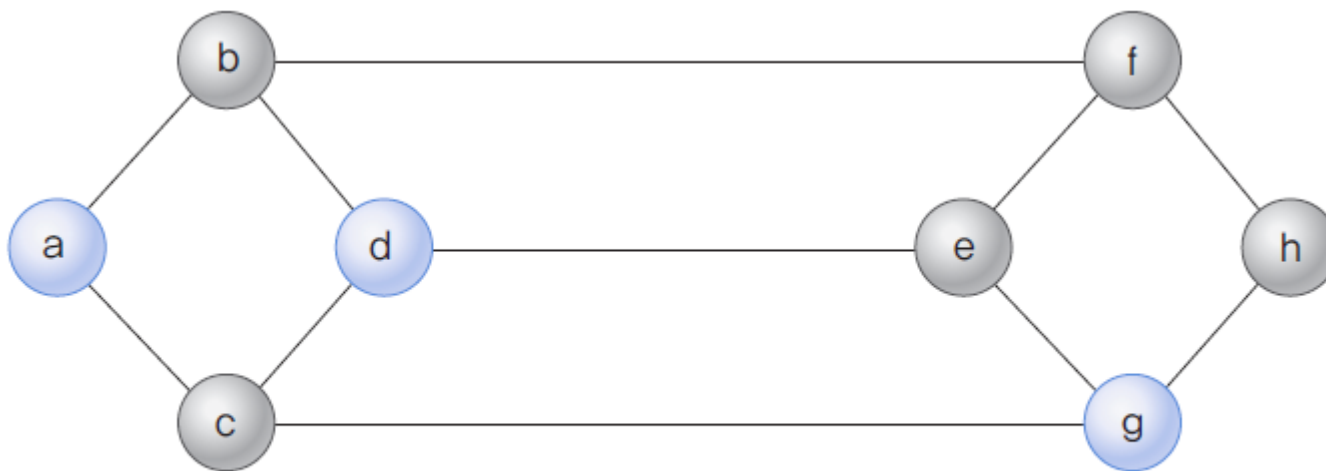


그림 7-6 최단 경로 라우팅



02_라우팅 프로토콜

■ RIP Routing Information Protocol 프로토콜

- 거리 벡터 방식의 내부 라우팅 프로토콜 중에서 가장 간단하게 구현된 것
- 소규모 네트워크 환경에 적합, 현재 가장 많이 사용하는 라우팅 프로토콜
- 라우팅 테이블 적용
 - 새로운 네트워크의 목적지 주소이면 라우팅 테이블에 적용
 - 거리 벡터 정보가 기존 정보와 비교하여 목적지까지 도착하는 지연이 더 적으면 대체
 - 라우터로부터 거리 벡터 정보가 들어왔을 때, 라우팅 테이블에 해당 라우터를 다음 홉으로 하는 등록 정보가 있으면 새로운 정보로 수정



02_라우팅 프로토콜

- 라우터 R1의 라우팅 테이블
 - 목적지 Net.4: 다음 홉 R4
 - 개선의 여지가 있음

표 7-1 수정 전 라우터 R1의 라우팅 테이블

목적지 네트워크	다음 홉	거리
Net.1	-	1
Net.2	-	1
Net.3	R4	2
Net.4	R4	3
Net.5	R6	2

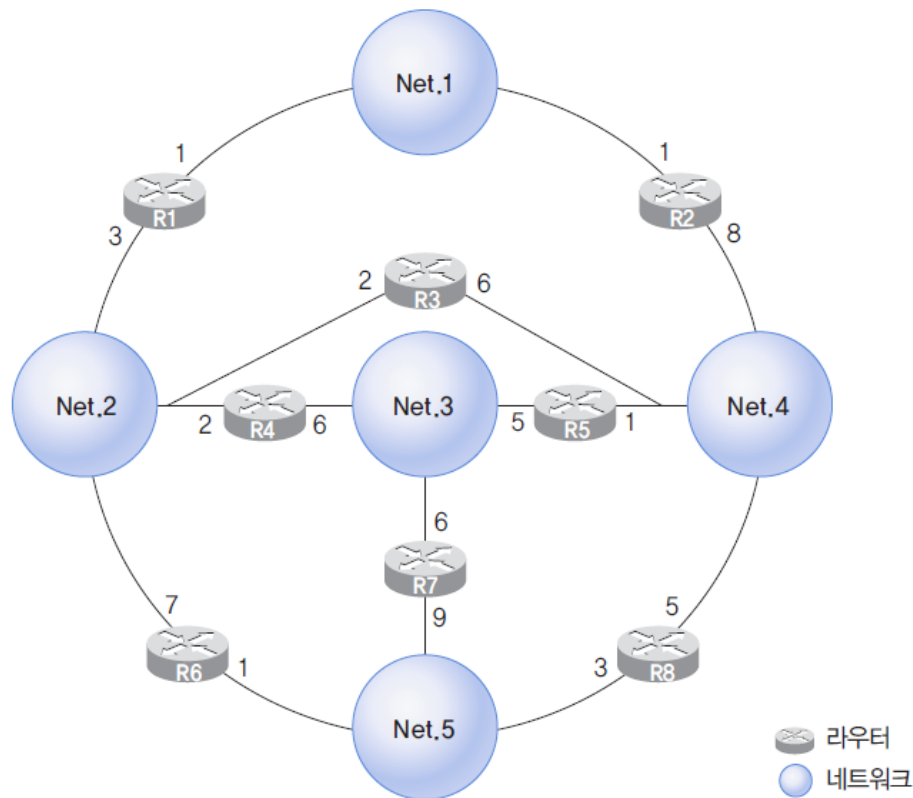


그림 7-8 네트워크 구성의 예



02_라우팅 프로토콜

- 임의의 시점에 거리 벡터 정보

R2 = [1, 2, 2, 1, 2]

R3 = [2, 1, 2, 1, 2]

R4 = [2, 1, 1, 2, 2]

R6 = [2, 1, 2, 2, 1]

표 7-2 수정 후 라우터 R1의 라우팅 테이블

목적지 네트워크	다음 홉	거리
Net.1	–	1
Net.2	–	1
Net.3	R4	2
Net.4	R3	2
Net.5	R6	2



03_IP 프로토콜

■ IP 프로토콜의 주요 특징

- 비연결형 서비스를 제공
- 패킷을 분할/병합하는 기능을 수행
- 데이터 체크섬은 제공하지 않고, 헤더 체크섬만 제공
- Best Effort 원칙에 따른 전송 기능을 제공

❖ IP 헤더 구조

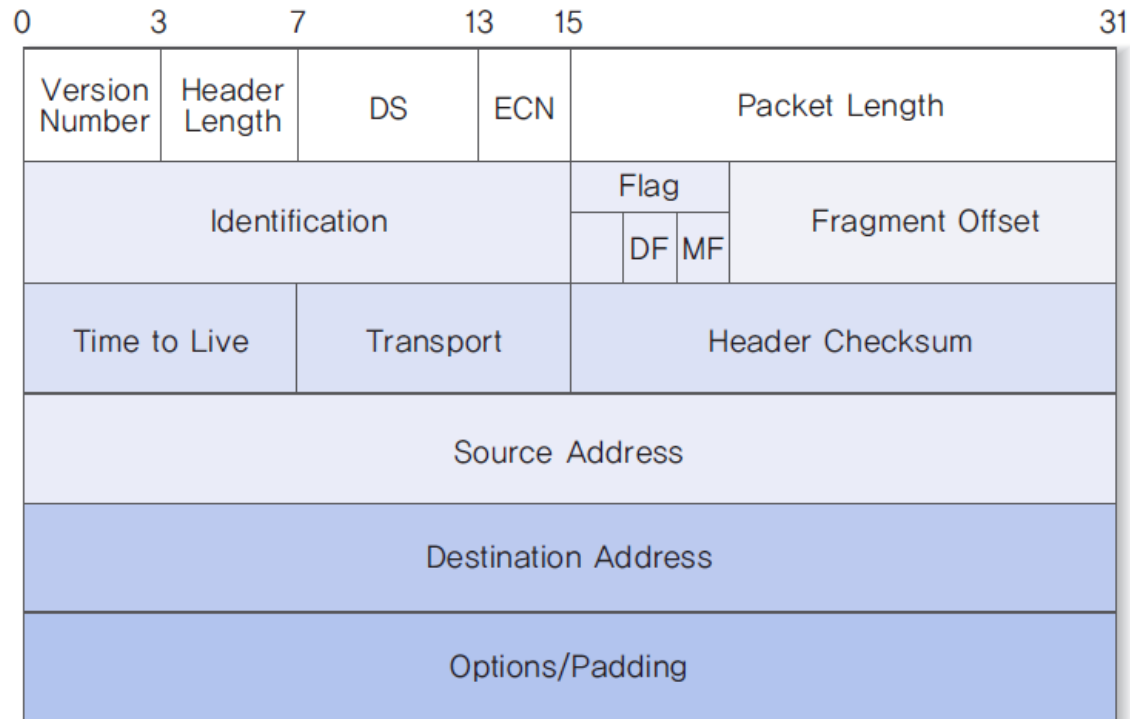


그림 7-12 IP 헤더의 구조



■ DS/ECN

• Service Type 필드

- 우선순위, 지연, 전송률, 신뢰성 등의 값을 지정할 수 있음
- IP 프로토콜이 사용자에게 제공하는 서비스의 품질에 관련된 내용을 표현

표 7-4 Service Type

비트 번호	각 비트의 값	
	0	1
0 ~ 2	우선순위(111 : 가장 높음)	
3	보통의 지연	낮은 지연
4	보통의 전송률	높은 전송률
5	보통의 신뢰성	높은 신뢰성
6 ~ 7	예약	

- Service Type 필드는 6비트의 DS 필드와 2비트의 ECN 필드로 새로 정의됨



03_IP 프로토콜

■ DSDifferentiated Services

- 사전에 서비스 제공자와 서비스 이용자 사이에 서비스 등급에 대해 합의
- 동일한 DS 값을 갖는 트래픽들은 동일한 서비스 등급으로 처리됨

■ ECNExplicit Congestion Notification

- ECT 0과 ECT 1은 동일한 의미
- ECN 기능을 위하여 TCP 프로토콜의 헤더에 ECE 필드와 CWR 필드가 추가

표 7-5 ECN 필드 값의 의미

필드 값	의미
00	IP 패킷이 ECN 기능을 사용하지 않음을 의미한다.
01(ECT 1)	TCP 프로토콜도 ECN 기능을 지원한다는 의미이다.
10(ECT 0)	TCP 프로토콜도 ECN 기능을 지원한다는 의미이다.
11(CE: Congestion Experienced)	라우터가 송신 호스트에 혼잡을 통지할 때 사용한다.



■ 패킷 분할

- Identification(식별자 혹은 구분자)
 - IP 헤더의 두 번째 워드에는 패킷 분할과 관련된 정보가 포함.
 - Identification은 송신 호스트가 지정하는 패킷 구분자 기능을 수행함
- DFDon't Fragment : 패킷이 분할되지 않도록 함
- MFMore Fragment
 - MF필드 값을 1로 지정하여, 분할 패킷이 뒤에 계속됨을 표시
 - 마지막 패킷은 MF 비트를 0으로 지정하여 분할 패킷이 더 없음을 표시
- Fragment Offset(분할 오프셋)
 - 저장되는 값은 분할된 패킷의 내용이 원래의 분할 전 데이터에서 위치하는 상대 주소값
 - 값은 8바이트의 배수



■ 주소 관련 필드

- Source Address : 송신 호스트의 IP 주소
- Destination Address : 수신 호스트의 IP
- network(네트워크) : 네트워크 주소
- host(호스트) : 네트워크 주소가 결정되면 하위의 호스트 주소를 의미하는 host 비트 값을 개별 네트워크의 관리자가 할당



03_IP 프로토콜

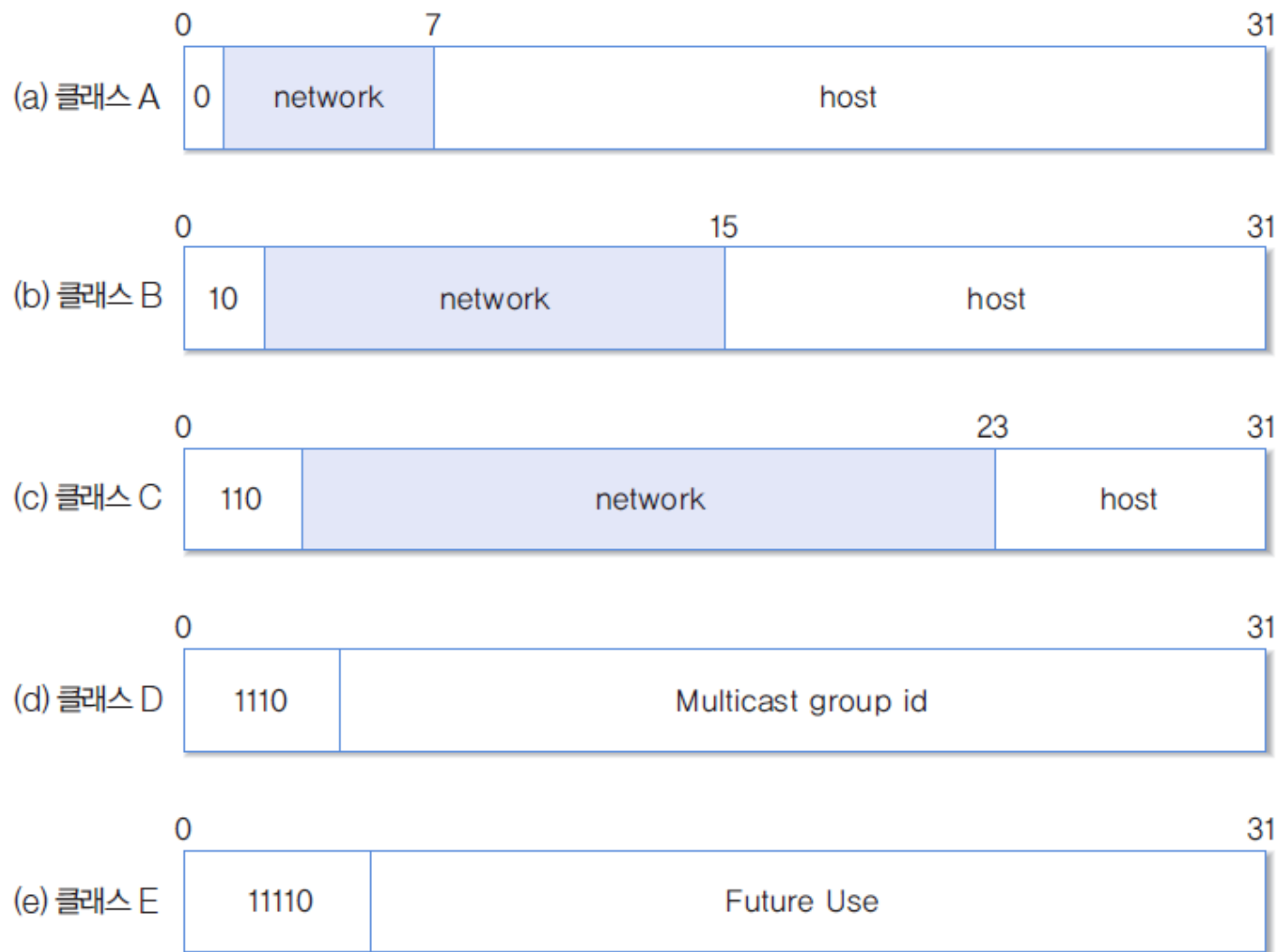


그림 7-13 IP 주소 체계



표 7-6 IP 주소 값에 따른 주소 체계

IP 주소 값	주소 체계
0.0.0.0 ~ 127.255.255.255	클래스 A의 주소 대역
128.0.0.0 ~ 191.255.255.255	클래스 B의 주소 대역
192.0.0.0 ~ 223.255.255.255	클래스 C의 주소 대역
224.0.0.0 ~ 239.255.255.255	클래스 D의 주소 대역
240.0.0.0 ~ 255.255.255.255	클래스 E의 주소 대역

■ 기타 필드

- Version Number(버전 번호) : IP 프로토콜의 버전 번호
- Header Length(헤더 길이) : IP 프로토콜 헤더 길이를 32비트 워드 단위로 표시
- Packet Length(패킷 길이) : IP 헤더를 포함하여 패킷의 전체 길이
- Time To Live(생존 시간) : 패킷의 생존 시간, 라우터를 거칠 때마다 1씩 감소되며 0이 되면 네트워크에서 강제로 제거



03_IP 프로토콜

- Transport(전송 프로토콜) : [그림 7-14]와 같이 IP 프로토콜에 데이터 전송을 요구한 전송계층의 프로토콜

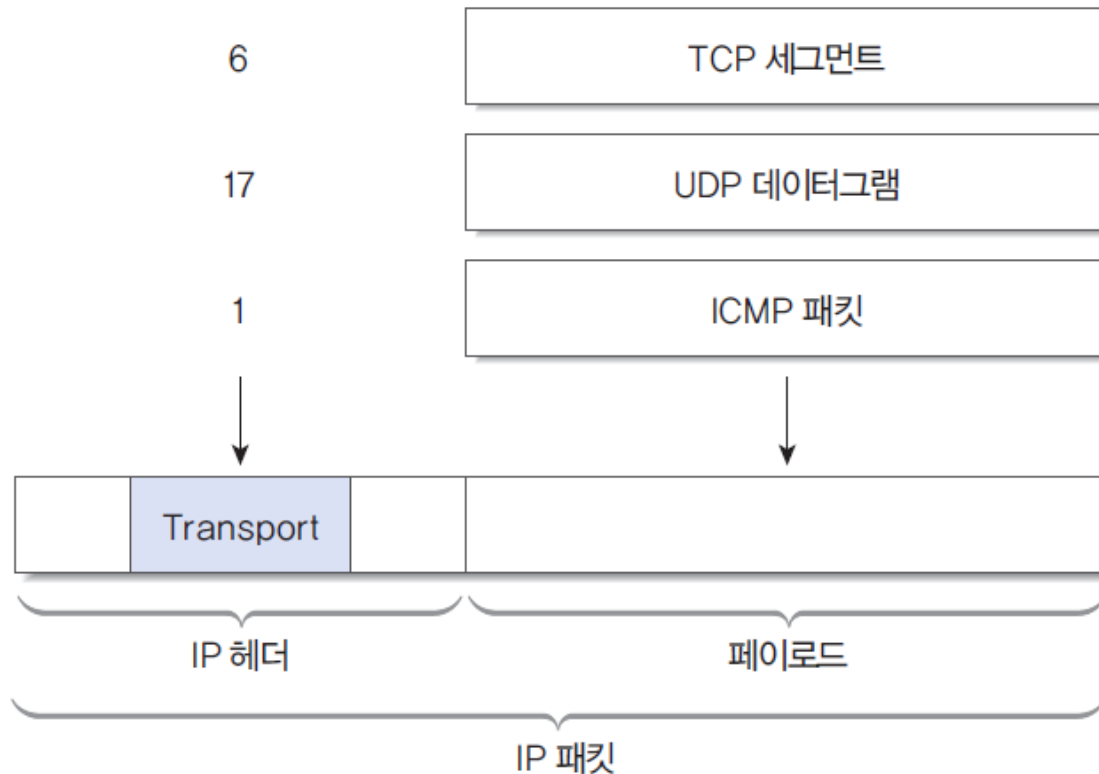


그림 7-14 Transport 필드



03_IP 프로토콜

- Header Checksum(헤더 체크섬) : 전송 과정에서 발생할 수 있는 헤더 오류를 검출하는 기능
- Options(옵션) : 네트워크 관리나 보안처럼 특수 용도로 이용할 수 있음
- Padding(패딩) : IP 헤더의 크기는 16비트 워드의 크기가 4의 배수가 되도록 설계



❖ 패킷 분할

■ 분할의 필요성

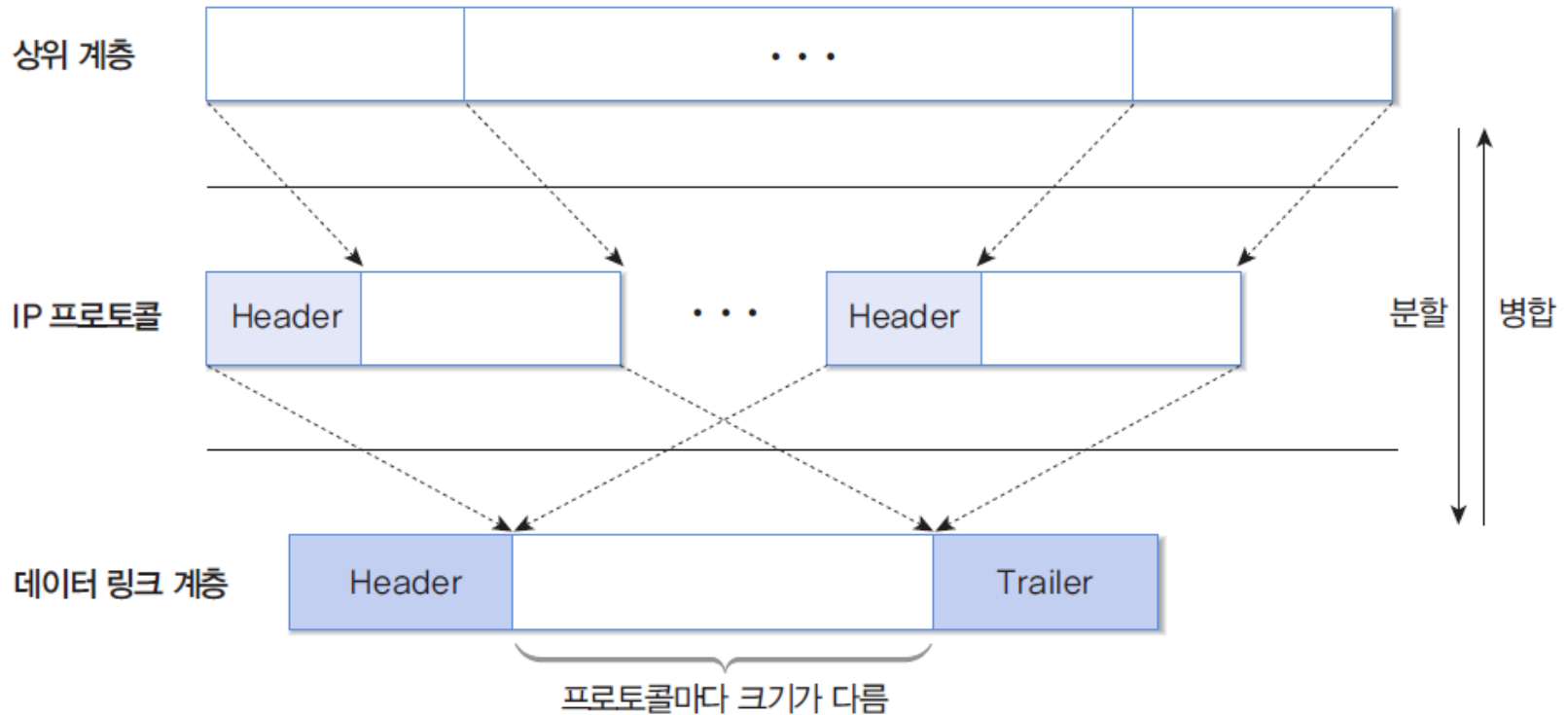


그림 7-15 패킷 분할의 필요성



03_IP 프로토콜

■ 분할의 예

- IP 헤더를 제외한 전송 데이터의 크기는 380바이트
- 패킷은 최대 크기가 128바이트라고 가정

IP 헤더	분할 1	분할 2	분할 3	분할 4	
		Identification	Packet Length	MF	Fragment Offset
IP 헤더	분할 1	1254	124	1	0
IP 헤더	분할 2	1254	124	1	13
IP 헤더	분할 3	1254	124	1	26
IP 헤더	분할 4	1254	88	0	39

그림 7-16 패킷 분할의 예



❖ ARP 프로토콜

- IP 주소와 MAC 주소 사이의 변환을 담당
- MAC 주소
 - 송신 호스트의 IP 주소 : 송신 호스트의 하드 디스크에서 얻을 수 있음
 - 수신 호스트의 IP 주소 : 사용자가 제공
 - 송신 호스트의 MAC 주소 : 송신 호스트의 LAN 카드에서 얻을 수 있음
 - 수신 호스트의 MAC 주소 : IP 주소를 매개변수로 하여 ARP 프로토콜로 얻음
- ARP 프로토콜
 - 특정 호스트의 IP 주소로 부터 MAC 주소를 제공하는 프로토콜
 - ARP request라는 특수 패킷을 브로드캐스팅
 - IP 주소에 해당하는 호스트만 ARP reply로 MAC 주소를 회신
 - 효율 향상을 위해 캐시 기능을 제공



■ IPv6의 주요 변경 사항

- 주소 공간 확장 : 공간이 32비트에서 128비트로 확장
- 헤더 구조 단순화 : 오류제어 등의 오버헤드를 줄여 프로토콜의 전송 효율 향상
- 흐름 제어 기능 지원 : 일정 범위 내에서 예측 가능한 데이터 흐름을 지원
실시간 멀티미디어 응용 환경을 수용



❖ IPv6 헤더 구조

- 9개의 기본 필드를 지원, 총 40바이트 중에서 32바이트는 주소 공간으로 할당, 8바이트만 프로토콜 기능

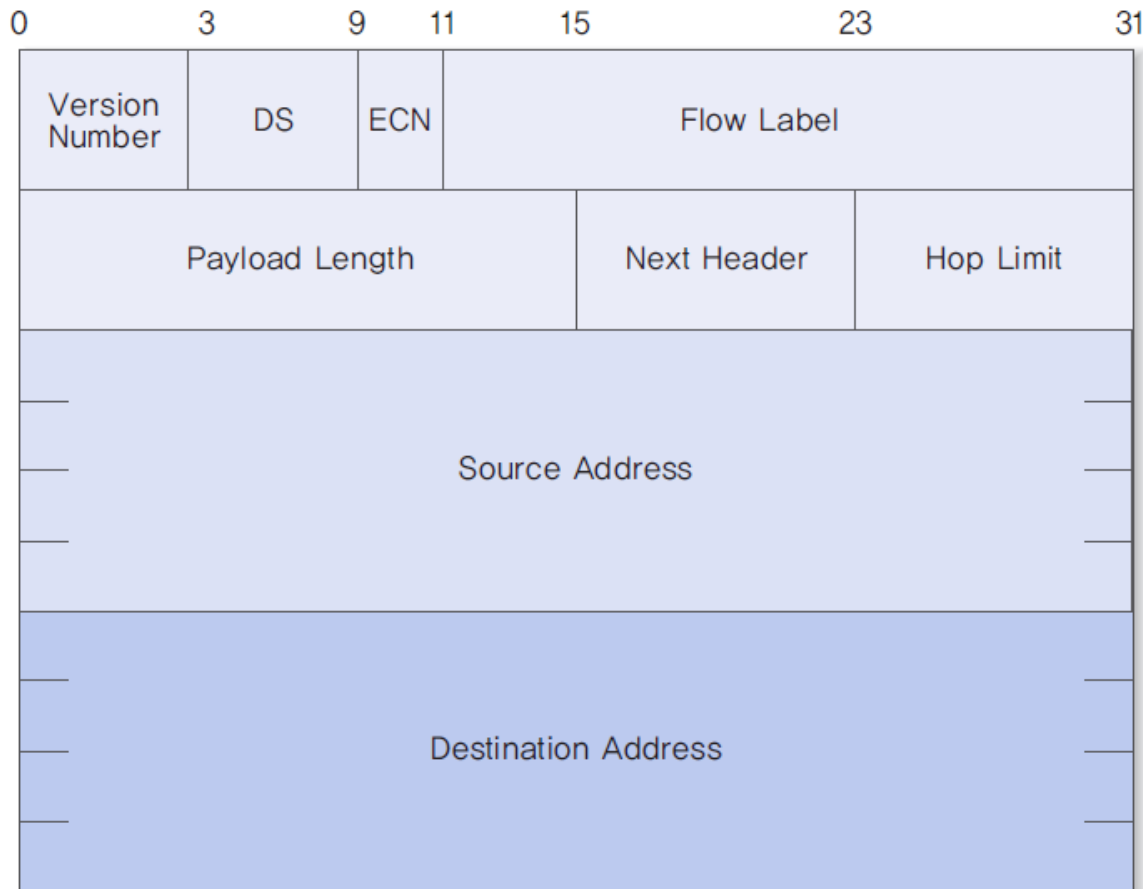


그림 8-1 IPv6 기본 헤더의 구조



❖ IPv6 주소

■ 주소 표현

- 128비트, 16비트의 숫자 8개를 콜론(:)으로 구분

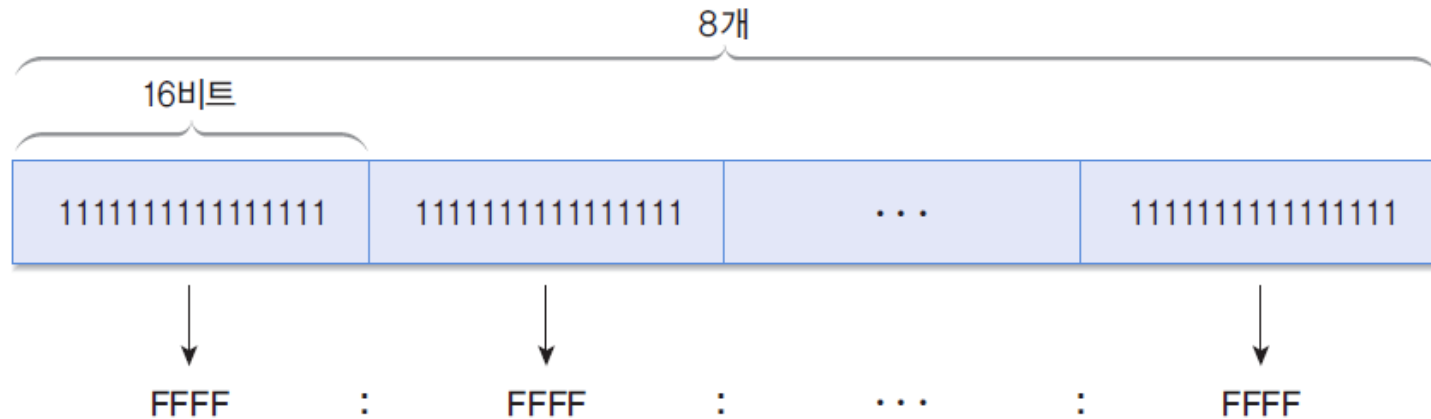


그림 8-2 IPv6의 주소 표현

• 축약표시

X:X:X:X:X:X:d,d,d,d

- X는 16비트이므로 총 96(16×6)비트, d는 8비트이므로 총 32(8×4)비트
- 즉, 전체 크기는 IPv6의 주소 크기와 동일한 128(96+32)비트



01_IPv6 프로토콜

- 주소 공간
 - IPv6의 주소 공간

표 8-1 IPv6의 주소 공간

상위 비트	용도	상위 비트	용도
0000 0000	예약(IPv4 공간 지원 포함)	100	비할당
0000 0001	비할당	101	비할당
0000 001	OSI NSAP 주소 공간	110	비할당
0000 010	Novell Netware IPX 주소 공간	1110	비할당
0000 011	비할당	1111 0	비할당
0000 01	비할당	1111 10	비할당
0001	비할당	1111 110	비할당
001	유니 캐스트 주소 공간	1111 1110 0	비할당
010	비할당	1111 1110 10	Link 지역 주소 공간
010	비할당	1111 1110 11	Site 지역 주소 공간
011	비할당	1111 1111	멀티캐스트 주소 공간



03_기타 네트워크 계층 프로토콜

■ ARP의 필요성

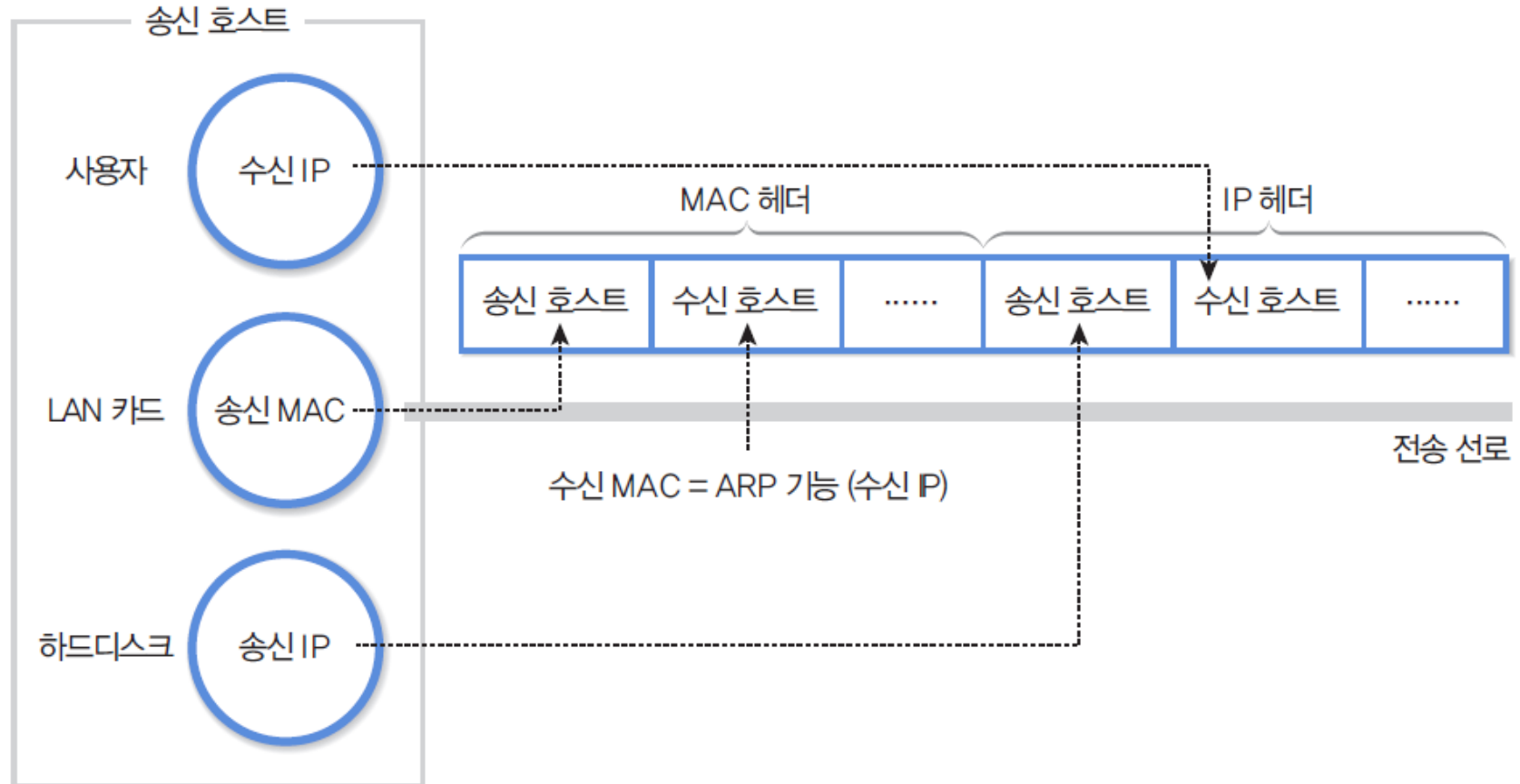


그림 8-7 ARP의 필요성



03_기타 네트워크 계층 프로토콜

- RARP(Reverse Address Resolution Protocol) 프로토콜의 필요성
 - 하드 디스크가 없는 시스템은 자신의 IP 주소를 알 수 없음
 - 특정 호스트의 MAC 주소로 부터 IP 주소를 제공하는 프로토콜

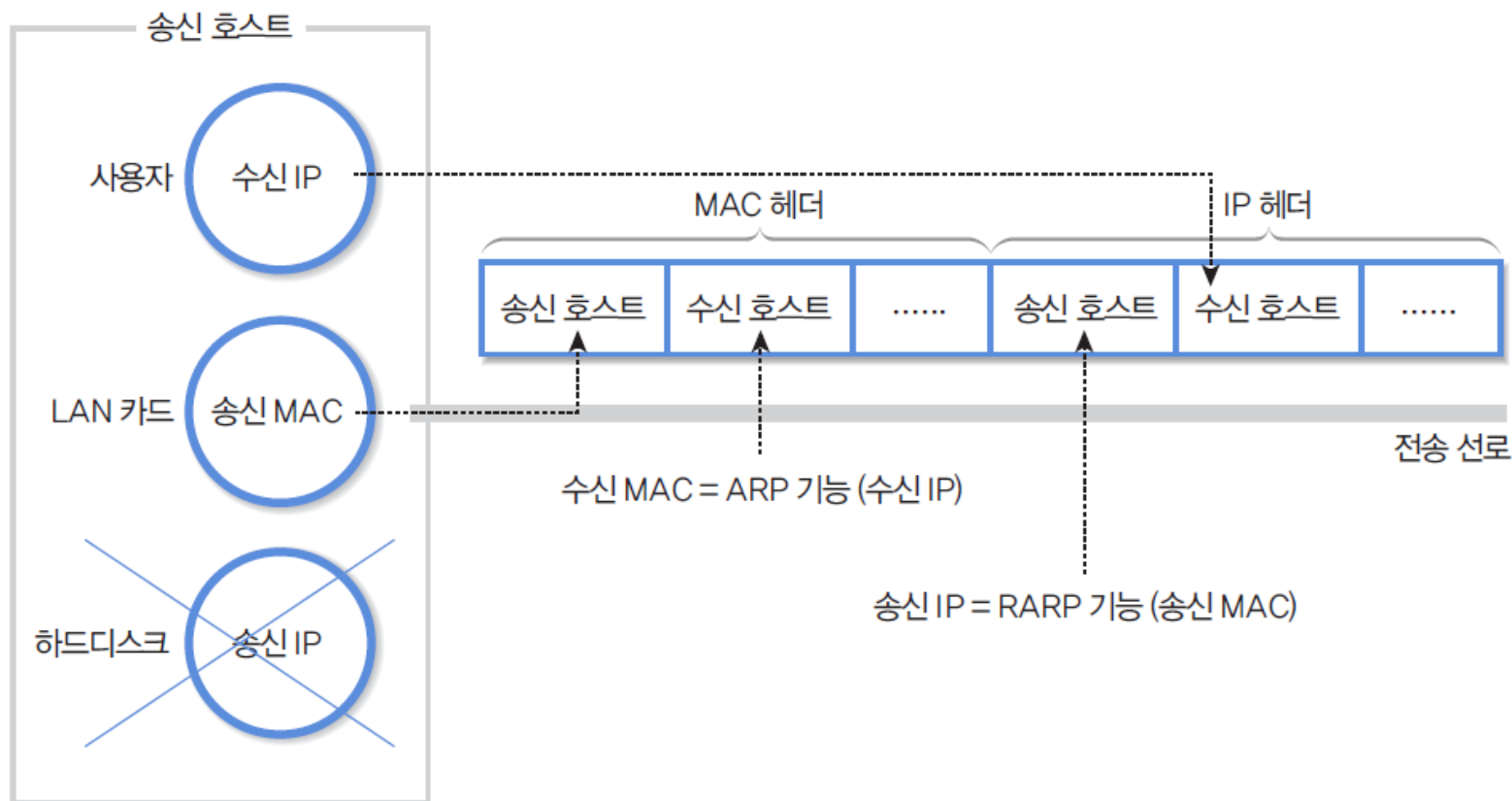


그림 8-8 RARP의 필요성

03_기타 네트워크 계층 프로토콜

❖ ICMP Internet Control Message Protocol 프로토콜

- 인터넷 환경에서 오류에 관한 처리를 지원
- ICMP 메시지
 - 오류 보고 메시지 Error-Reporting Message : IP 패킷을 전송하는 과정에서 발생하는 문제를 보고하는 것이 목적

표 8-2 오류 보고 메시지

메시지	설명
DESTINATION UNREACHABLE	수신 호스트가 존재하지 않거나, 존재해도 필요한 프로토콜이나 포트 번호 등이 없어 수신 호스트에 접근이 불가능한 경우에 발생한다. IP 헤더의 DF 필드가 설정된 패킷을 라우터가 분할해야 하는 경우에도 해당 패킷을 버리고 이 메시지를 회신해준다.
SOURCE QUENCH	네트워크에 필요한 자원이 부족하여 패킷이 버려지는 경우에 발생한다. 예를 들면, 전송 경로에 있는 라우터에 부하가 많이 걸려 패킷이 버려지는 경우이다. 이 메시지를 이용해 송신 호스트에 혼잡 가능성을 경고함으로써, 패킷을 송신하는 호스트가 데이터를 천천히 전송하도록 알릴 수 있다.
TIME EXCEEDED	패킷의 TTL Time To Live 필드 값이 0이 되어 패킷이 버려진 경우에 주로 발생한다. 기타 시간 초과 현상에 의해 패킷이 버려진 경우도 이에 해당한다.



03_기타 네트워크 계층 프로토콜

- 질의 메시지 Query Message : 라우터 혹은 다른 호스트들의 정보를 획득하려는 목적

표 8-3 질의 메시지

메시지	설명
ECHO REQUEST, ECHO REPLY	유닉스 ^{Unix} 의 ping 프로그램에서 네트워크의 신뢰성을 검증하기 위하여 ECHO REQUEST 메시지를 전송하고, 이를 수신한 호스트는 ECHO REPLY를 전송해 응답한다. 특정 호스트가 인터넷에서 활성화되어 동작하는지 확인할 수 있다.
TIMESTAMP REQUEST, TIMESTAMP REPLY	두 호스트 간의 네트워크 지연을 계산하는 용도로 사용한다.



03_기타 네트워크 계층 프로토콜

- ICMP 헤더 형식
 - 오류 보고 메세지

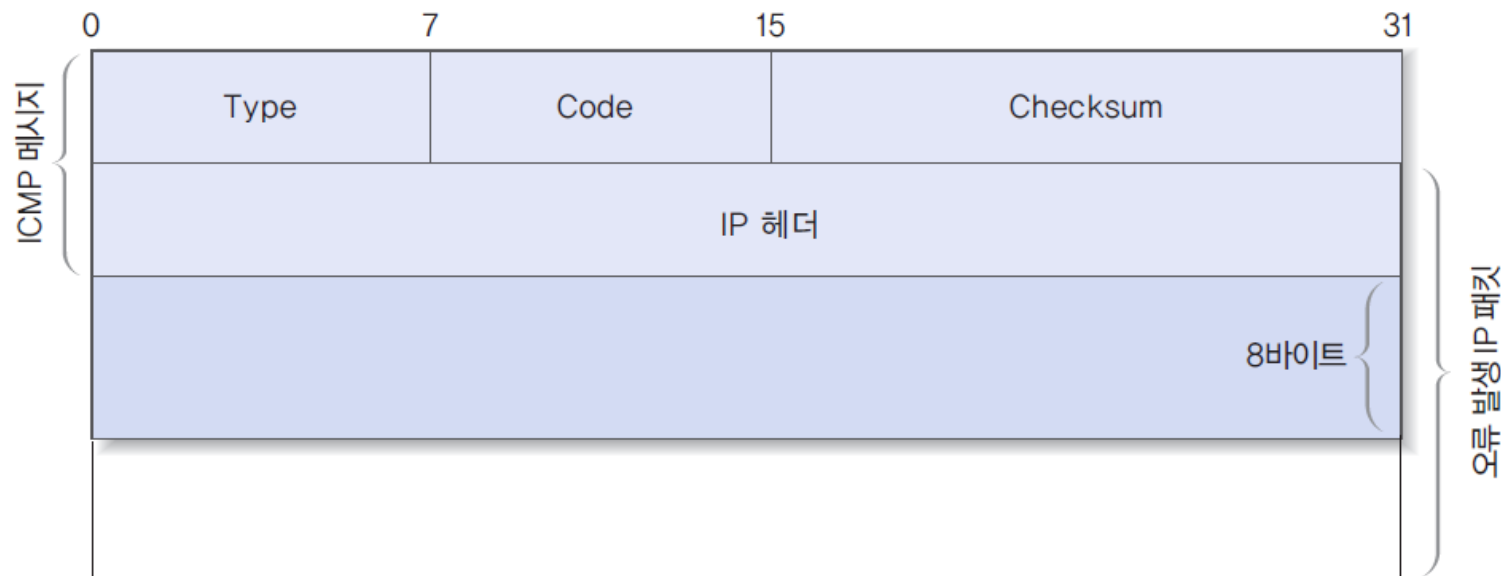


그림 8-9 ICMP 메시지: 오류 보고 메시지

- 오류 원인을 제공한 IP 패킷의 헤더와 이어지는 8바이트의 정보가 오류 보고 메시지에 포함됨
- Type(유형) : 1바이트 크기로 메시지의 종류를 구분
- Code(코드) : 메시지 내용에 대한 자세한 정보를 제공하는 매개변수 값
- Checksum(체크섬) : ICMP 전체 메시지에 대한 체크섬 기능을 지원



03_기타 네트워크 계층 프로토콜

- 질의 메시지
 - Identifier와 Sequence Number 필드를 사용하여 메시지를 구분하는 기능이 사용

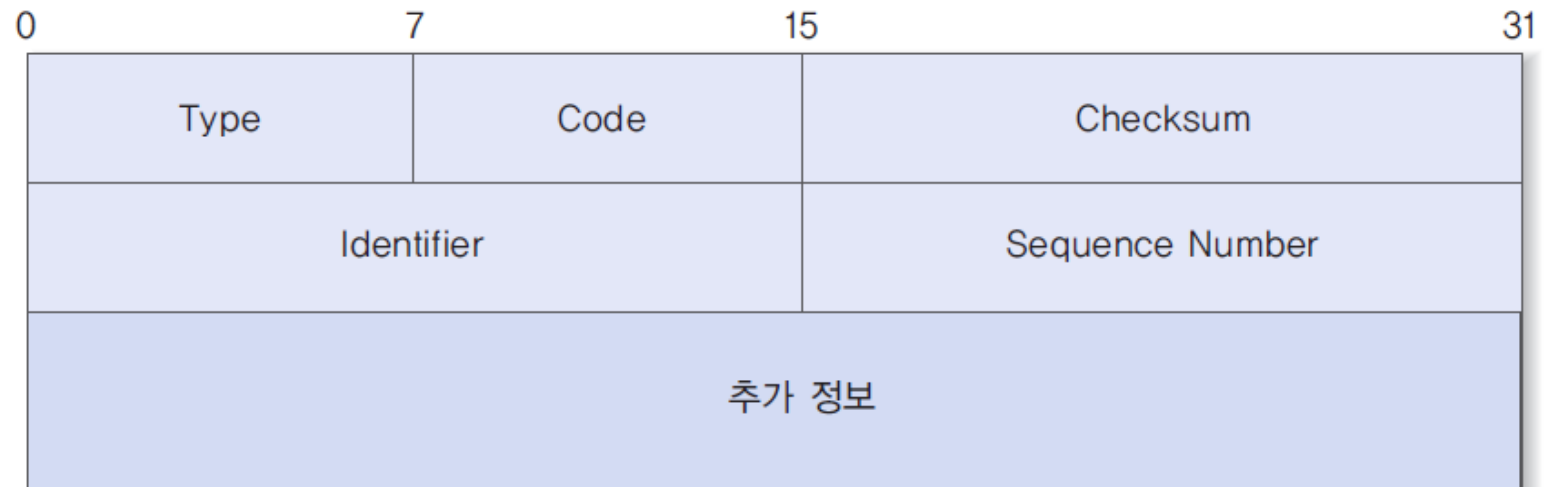


그림 8-10 ICMP 메시지 : 질의 메시지



03_기타 네트워크 계층 프로토콜

■ ICMP 메시지 전송

- ICMP는 기능적으로 IP 프로토콜과 같은 계층의 역할을 수행
- ICMP 메시지는 IP 프로토콜에 캡슐화되어 전송

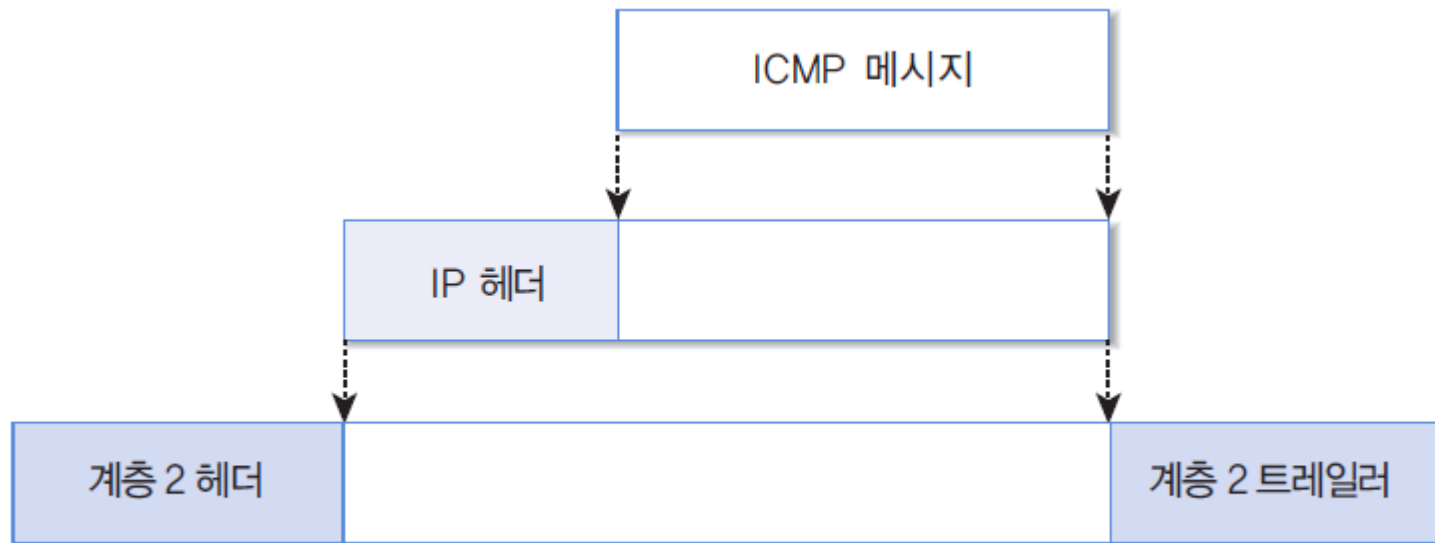


그림 8-11 ICMP 메시지의 전송



❖ IGMP 프로토콜

- 멀티캐스팅 Multicasting : 특정 그룹의 모든 호스트에 메시지를 전송하는 방식
- 멀티캐스트 라우팅 Multicast Routing : 멀티캐스팅에 필요한 라우팅 알고리즘

■ 그룹 관리

- 그룹 관리의 주요 기능 : 그룹의 생성.제거, 전송 호스트의 그룹 참가.탈퇴 등
- 멀티캐스팅 기능
 - 다중 호스트를 표시하는 멀티캐스트 그룹 주소 표기 방법의 통일
 - 라우터가 멀티캐스트 주소와 이 그룹에 속하는 호스트 사이의 연관성 처리
 - 멀티캐스트 라우팅 알고리즘은 그룹의 모든 멤버에게 가장 짧은 경로를 선택하는 기능 제공

■ IGMP Internet Group Management Protocol

- 멀티캐스트 그룹에 가입하거나 탈퇴할 때 사용하는 프로토콜
- 멀티캐스트 그룹에 가입한 호스트와 라우터 사이에 멤버 정보를 교환하는 용도
- 질의 메시지 : 멀티캐스트 라우터가 그룹 정보를 얻기 위하여 호스트에 전달
- 보고 메시지 : 질의의 응답으로 호스트가 보고 메시지를 회신



03_기타 네트워크 계층 프로토콜

- IGMP 헤더의 구조

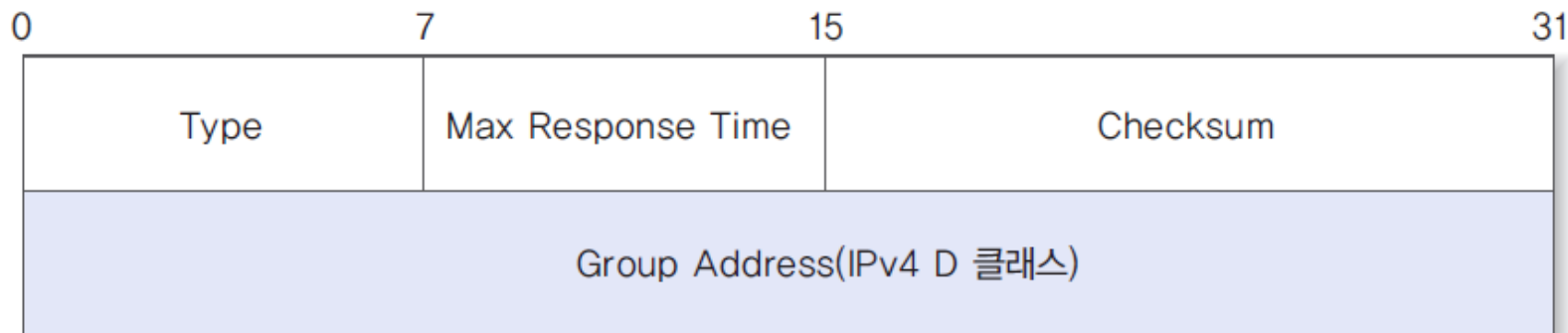


그림 8-12 IGMP 헤더의 구조

- Type(유형) : 0x11 - 멀티캐스트 라우터가 전송한 질의 메시지
0x16 - 호스트가 전송하는 보고 메시지
0x17 - 그룹 탈퇴에 관한 메시지
- Max Response Time(최대 응답 시간) : 질의에 대한 보고 메시지가 전송되는 최대응답시간
- Checksum(체크섬) : IP 프로토콜에서 사용하는 알고리즘과 동일한 방식 (오류 검출용으로 이용)
- Group Address(그룹 주소) : 질의 메시지는 0, 보고 메시지에는 호스트가 가입을 원하는 그룹 주소를 표기



03_기타 네트워크 계층 프로토콜

■ IGMP 동작 과정

- 그룹 가입 : 해당 멀티캐스트 주소를 표기한 IGMP 보고 메시지를 전송
- 그룹 유지 : IGMP 보고 메시지를 사용해 IGMP 질의에 응답해야함
- 그룹 탈퇴 : 라우터의 질의 메시지에 대해 호스트의 보고 메시지 응답이 없음

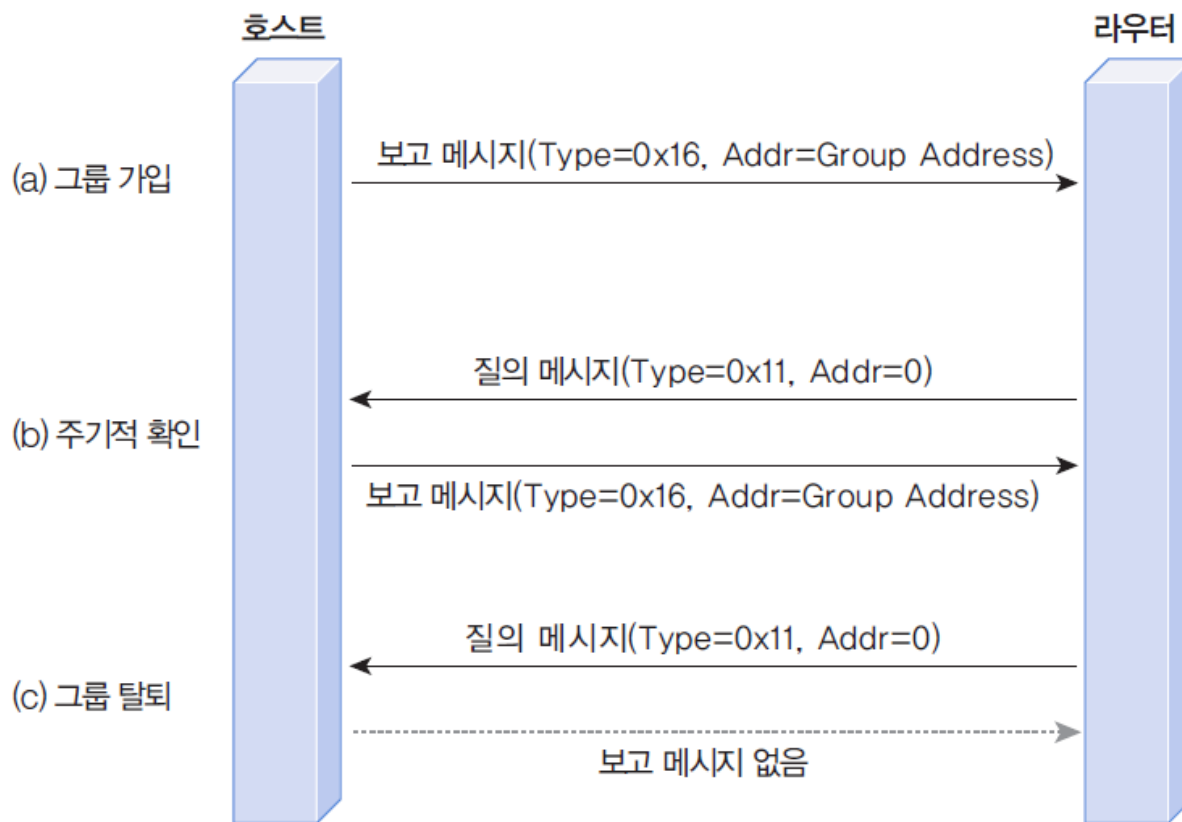


그림 8-13 IGMP 동작 과정



03_기타 네트워크 계층 프로토콜

■ IGMP 메시지의 전송

- IGMP는 IP 패킷에 캡슐화되어 보내짐

즉, IGMP 메시지는 IP 프로토콜의 데이터로 처리되기 때문에 IP 패킷의 헤더에 실려서 계층 2 프로토콜로 전달됨

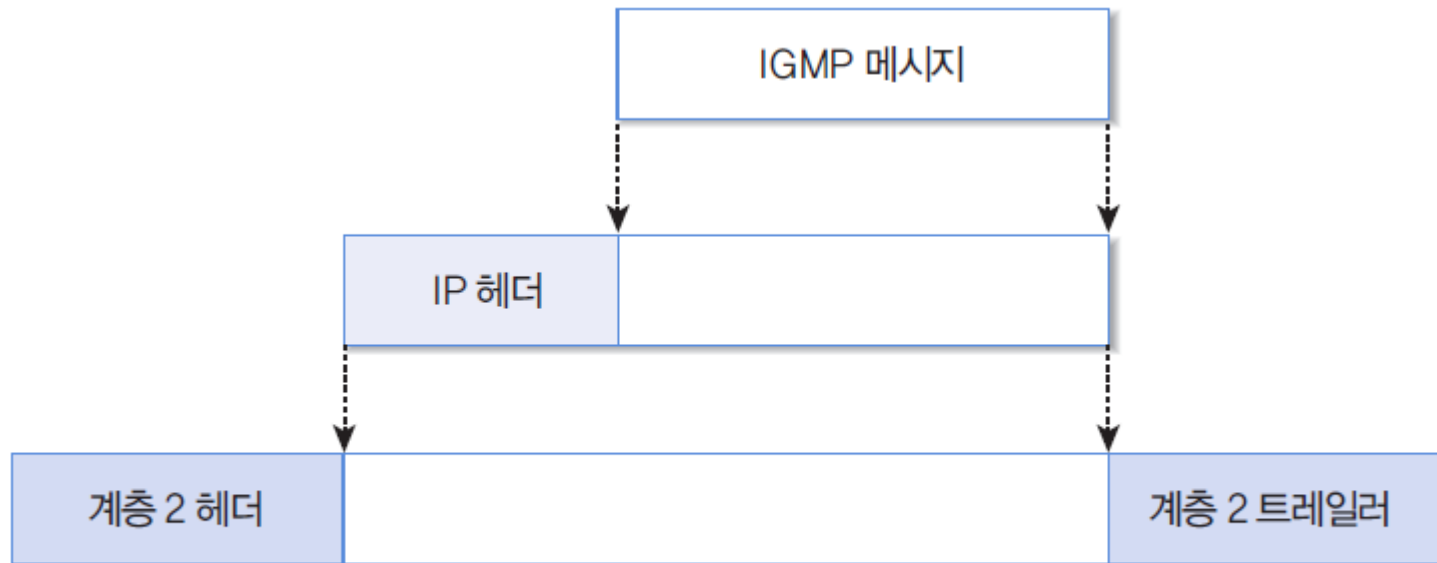


그림 8-14 IGMP 메시지의 전송



참조

02_라우팅 프로토콜

■ 플러딩 Flooding

- 라우터가 자신에게 입력된 패킷을 출력 가능한 모든 경로로 중개하는 방식
- 패킷이 무한히 만들어질 수 있으므로 생존 시간으로 제한
- 특별한 목적으로만 사용

❖ 거리 벡터 라우팅 프로토콜

- 라우터가 자신과 연결된 이웃 라우터와 라우팅 정보를 교환하는 방식
- 필수 정보
 - 링크 벡터 : 이웃 네트워크에 대한 연결 정보
 - 거리 벡터 : 개별 네트워크까지의 거리 정보
 - 다음 홉 벡터 : 개별 네트워크로 가기 위한 다음 홉 정보



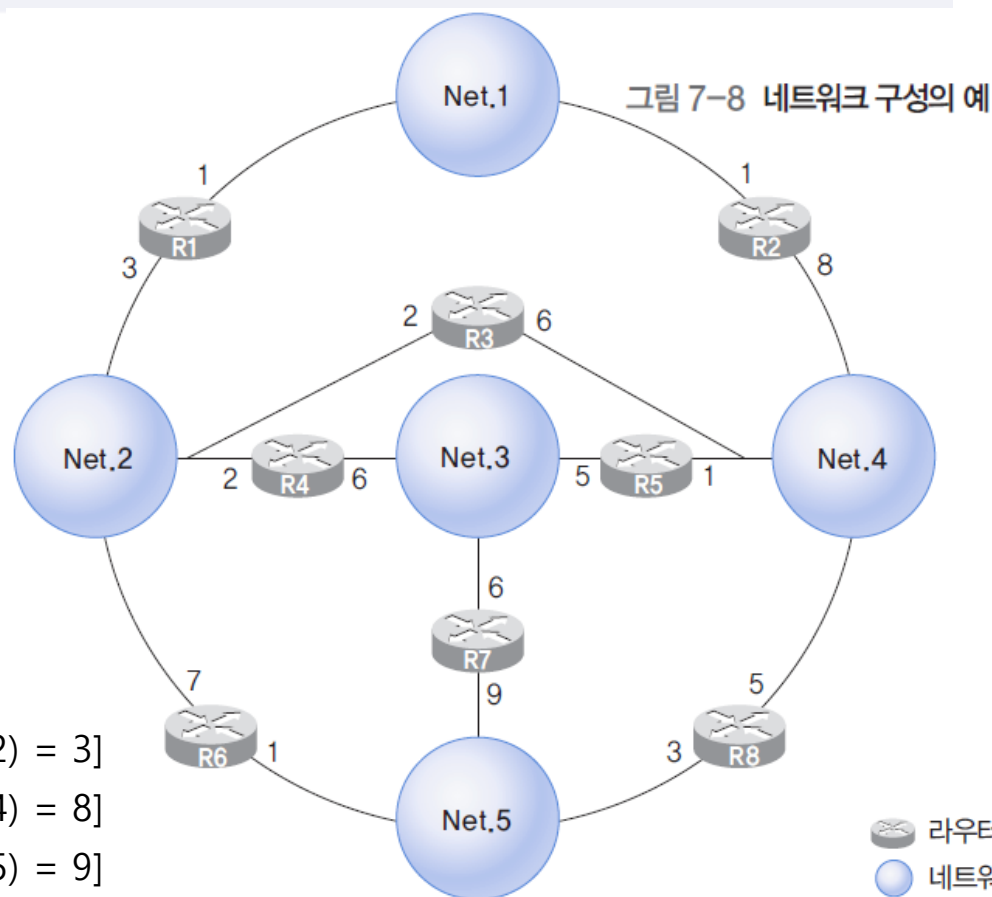
02_라우팅 프로토콜

■ 링크 벡터

- 링크 벡터 $L(x)$: 라우터 x 와 연결된 이웃 네트워크에 대한 연결 정보를 보관

링크 벡터 $L(x) = [\text{포트}(1), \text{포트}(2), \dots, \text{포트}(m), \dots, \text{포트}(M)]$

그림 7-7 링크 벡터 $L(x)$



- $L(R1) = [\text{포트}(\text{Net.1}) = 1, \text{포트}(\text{Net.2}) = 3]$
- $L(R2) = [\text{포트}(\text{Net.1}) = 1, \text{포트}(\text{Net.4}) = 8]$
- $L(R7) = [\text{포트}(\text{Net.3}) = 6, \text{포트}(\text{Net.5}) = 9]$

02_라우팅 프로토콜

- 거리 벡터

- 전체 네트워크에 소속된 개별 네트워크들까지의 거리 정보를 관리

거리 벡터 $D(x) = [\text{거리}(1), \text{거리}(2), \dots, \text{거리}(n), \dots, \text{거리}(N)]$

그림 7-9 거리 벡터 $D(x)$

- [그림 7-8]에서

$D(R1) = [\text{거리}(\text{Net.1}) = 1,$
 $\text{거리}(\text{Net.2}) = 1,$
 $\text{거리}(\text{Net.3}) = 2,$
 $\text{거리}(\text{Net.4}) = 2,$
 $\text{거리}(\text{Net.5}) = 2]$



02_라우팅 프로토콜

- 다음 홉 벡터

- 다음 홉 벡터 $H(x)$ 는 개별 네트워크까지 패킷을 전송하는 경로에 있는 다음 홉 정보를 관리

다음 홉 벡터 $H(x) = [\text{홉}(1), \text{홉}(2), \dots, \text{홉}(n), \dots, \text{홉}(N)]$

그림 7-10 다음 홉 벡터 $H(x)$

- [그림 7-8]에서

$H(R1) = [\text{다음 홉}(\text{Net.1}) = -,$
 $\text{다음 홉}(\text{Net.2}) = -,$
 $\text{다음 홉}(\text{Net.3}) = R4,$
 $\text{다음 홉}(\text{Net.4}) = R3,$
 $\text{다음 홉}(\text{Net.5}) = R6]$



02_라우팅 프로토콜

- 벡터 정보를 교환하기 위해 다음과 같은 패킷 구조를 사용함

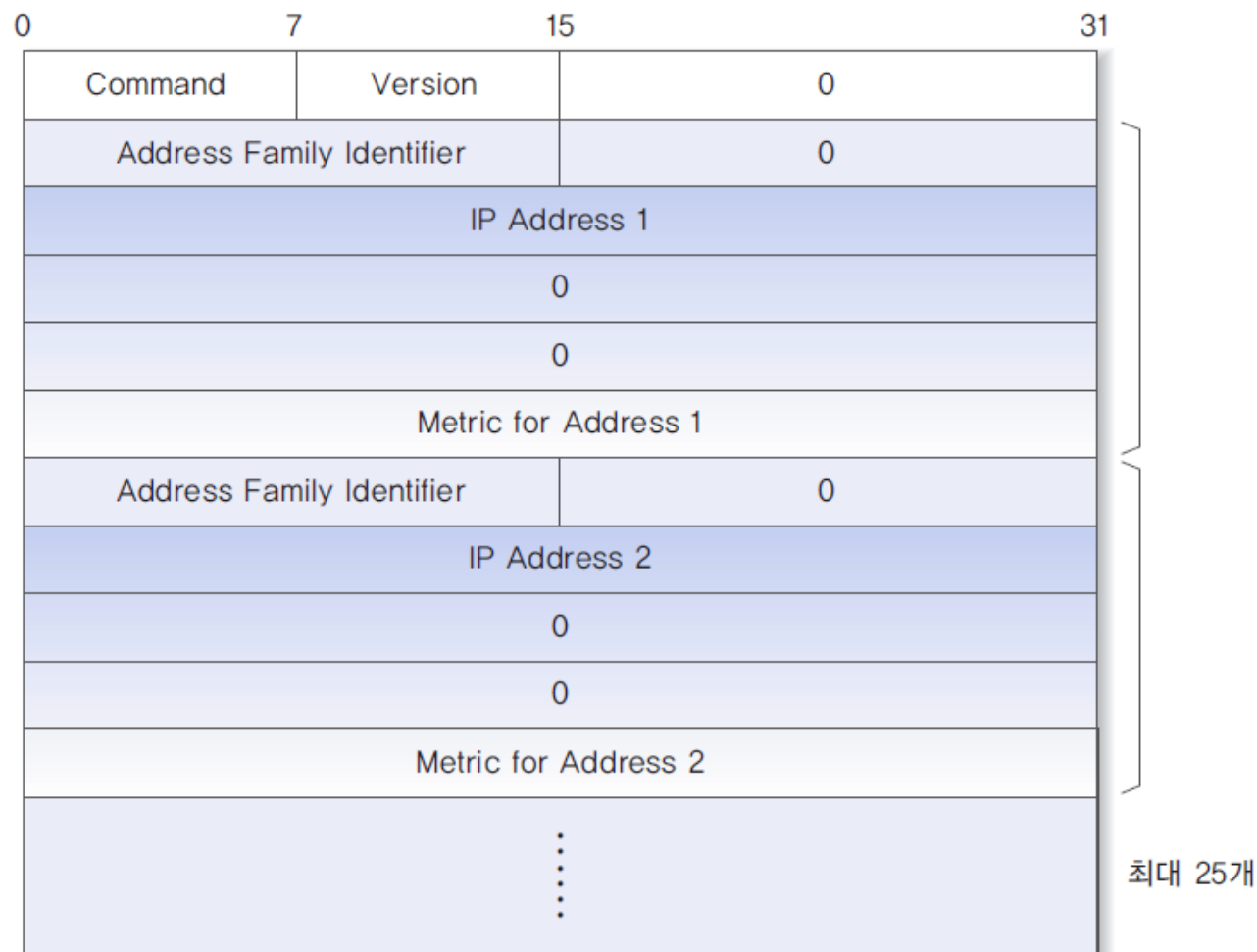


그림 7-11 RIP 패킷의 구조



02_라우팅 프로토콜

- Command(명령) : 값이 1이면 RIP 요청을, 2이면 RIP 응답을 의미.
- Version(버전) : RIP 프로토콜의 버전 번호
- Address Family Identifier(주소 패밀리 구분자) : IP 프로토콜의 주소는 2로 설정
- IP Address(IP 주소) : 특정한 네트워크를 지칭하는 용도로 사용되기 때문에 IP 주소의 네트워크 부분의 값만 사용하고, 호스트 부분은 0으로 채움
- Metric(거리) : 해당 라우터에서 목적지 네트워크까지의 거리



❖ 링크 상태 Link State 라우팅 프로토콜

- 개별 라우터가 이웃 라우터까지의 거리 정보를 구한 후, 이를 네트워크에 연결된 모든 라우터에 통보
- 거리 벡터 방식과 반대
- 거리 벡터 라우팅 프로토콜의 단점을 보완하기 위한 방식
- 플러딩 Flooding 기법 : 임의의 라우터가 이웃한 모든 라우터에 정보를 전달하고, 다시 이들 라우터가 주변의 모든 라우터에 정보를 전달하는 방식으로 동작
- 예) OSPF Open Shortest Path First



❖ 외부 라우팅 프로토콜

- 내부 라우팅 프로토콜
 - 거리 벡터 방식을 사용하는 RIP
 - 링크 상태 방식을 사용하는 OSPF
- 외부 라우팅 프로토콜
 - 경로 벡터 프로토콜: 단순히 연결 가능한지에 대한 정보만 제공
- BGP
 - TCP 프로토콜을 사용

표 7-3 TCP 프로토콜에서 제공하는 메시지의 종류

메시지	설명
Open	다른 라우터와 연관 ^{Relationship} 을 설정한다.
Update	라우팅 관련 정보를 전달한다.
KeepAlive	Open 메시지에 대한 응답 기능과 주변 라우터와의 연관을 주기적으로 확인한다.
Notification	오류 상태를 통보한다.



❖ DHCP 프로토콜

- IP 주소를 여러 컴퓨터가 공유해서 사용
- DHCP 메시지

0		31	
OPcode	HardwareType	HardwareLength	HOPCount
Transaction Identifier			
Time Elapsed		Flag	
Client IP Address			
Your IP Address			
Server IP Address			
Gateway IP Address			
Client Hardware Address			
Server Name			
Boot File Name			
Options			

그림 7-17 DHCP 메시지

- DHCP 프로토콜의 주요 메시지
 - DHCP_DISCOVER : 클라이언트가 DHCP 서버를 찾기 위해 전송하는 브로드캐스트 메시지
 - DHCP_OFFER : 클라이언트의 DHCP_DISCOVER 메시지에 대한 응답으로 DHCP 서버가 응답하는 메시지
 - DHCP_REQUEST : 주소를 권고한 DHCP 서버에 DHCP_REQUEST 메시지를 전송하여 권고한 주소를 사용한다고 알림
 - DHCP_ACK : 권고한 IP 주소가 최종적으로 사용 가능한지 판단후 사용 가능하면D HCP_ACK 메시지를 전송
 - DHCP_NACK : 클라이언트가 DHCP_DISCOVER 과정을 다시 하도록함



03_IP 프로토콜

- DHCP 프로토콜의 동작 과정

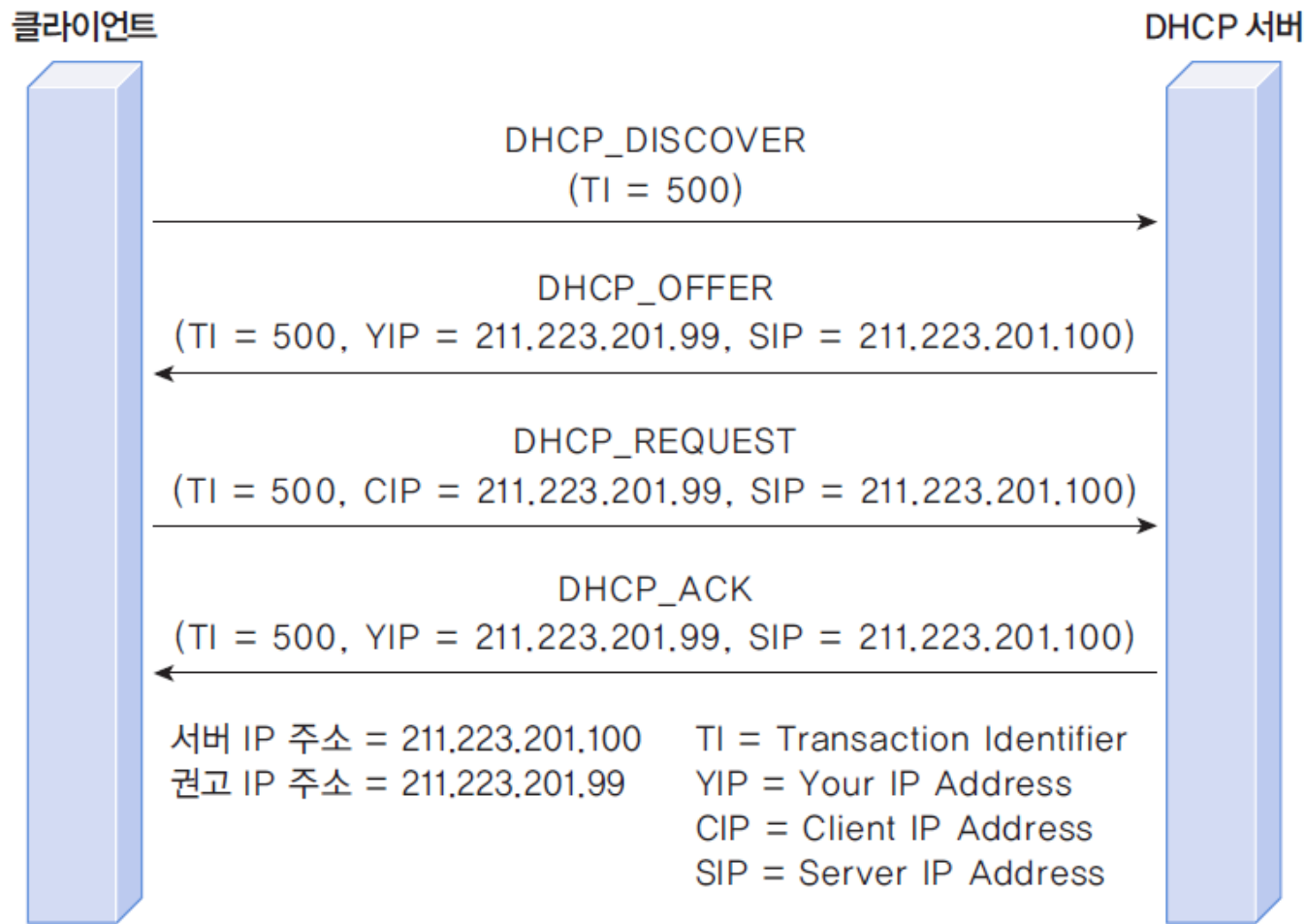


그림 7-18 DHCP 프로토콜의 동작 과정



03_IP 프로토콜

- UDP/IP 프로토콜의 캡슐화

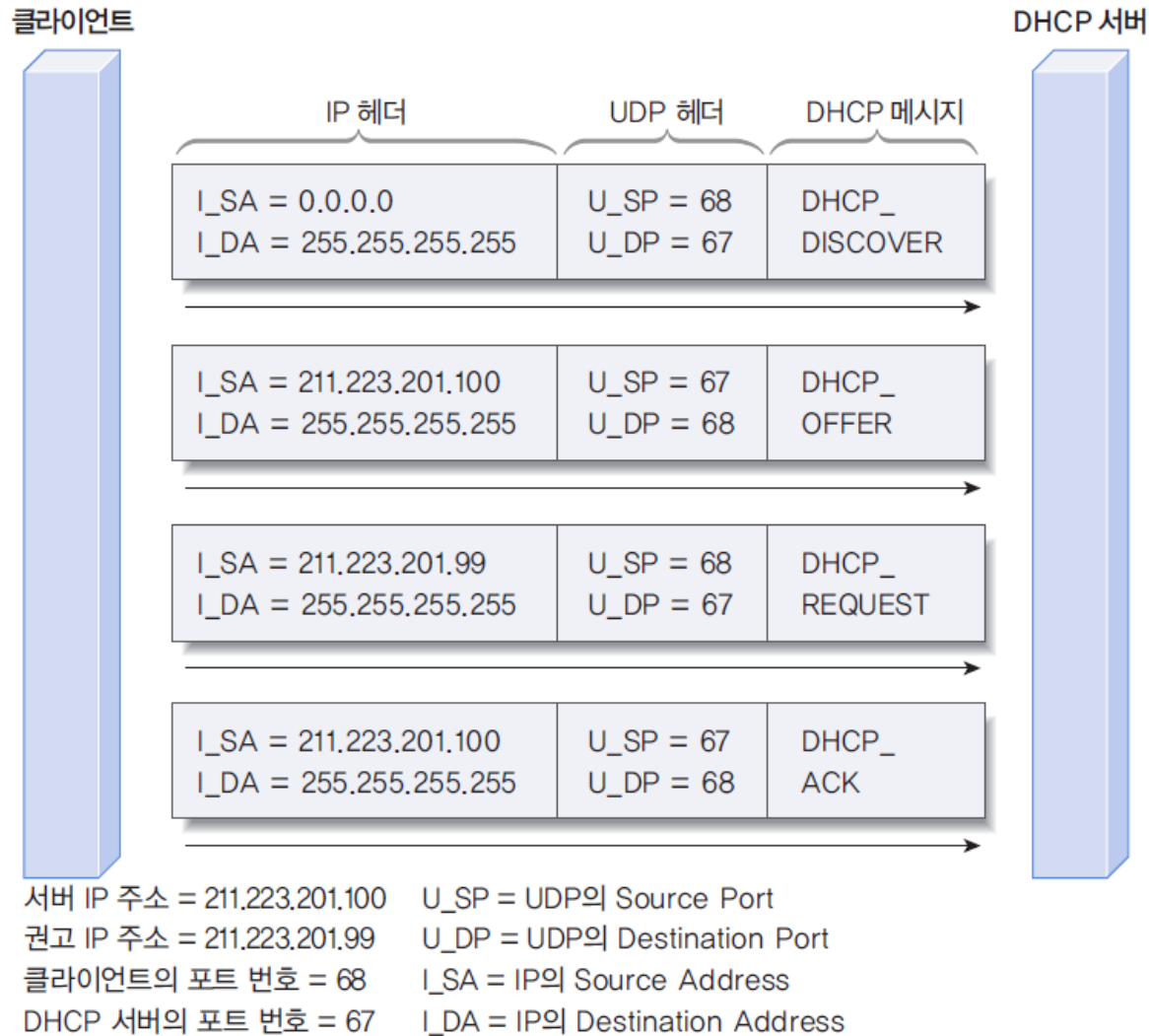


그림 7-19 UDP/IP 프로토콜의 캡슐화



❖ 터널링 원리

- 상이한 전송 수단
 - IP 프로토콜을 교체하는 방식
(버스→배→버스)

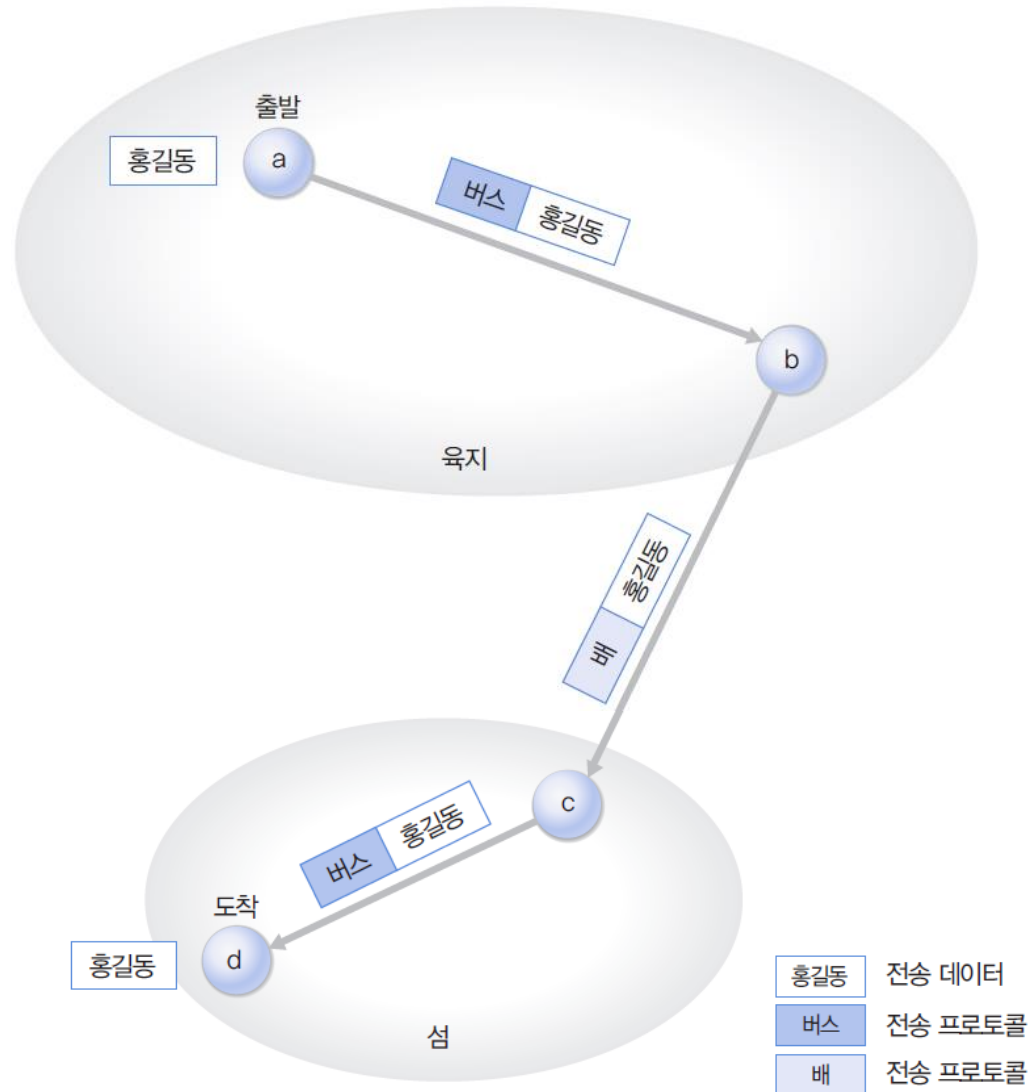


그림 8-3 상이한 전송 수단



02_이동 IP 프로토콜

■ 터널링 방식

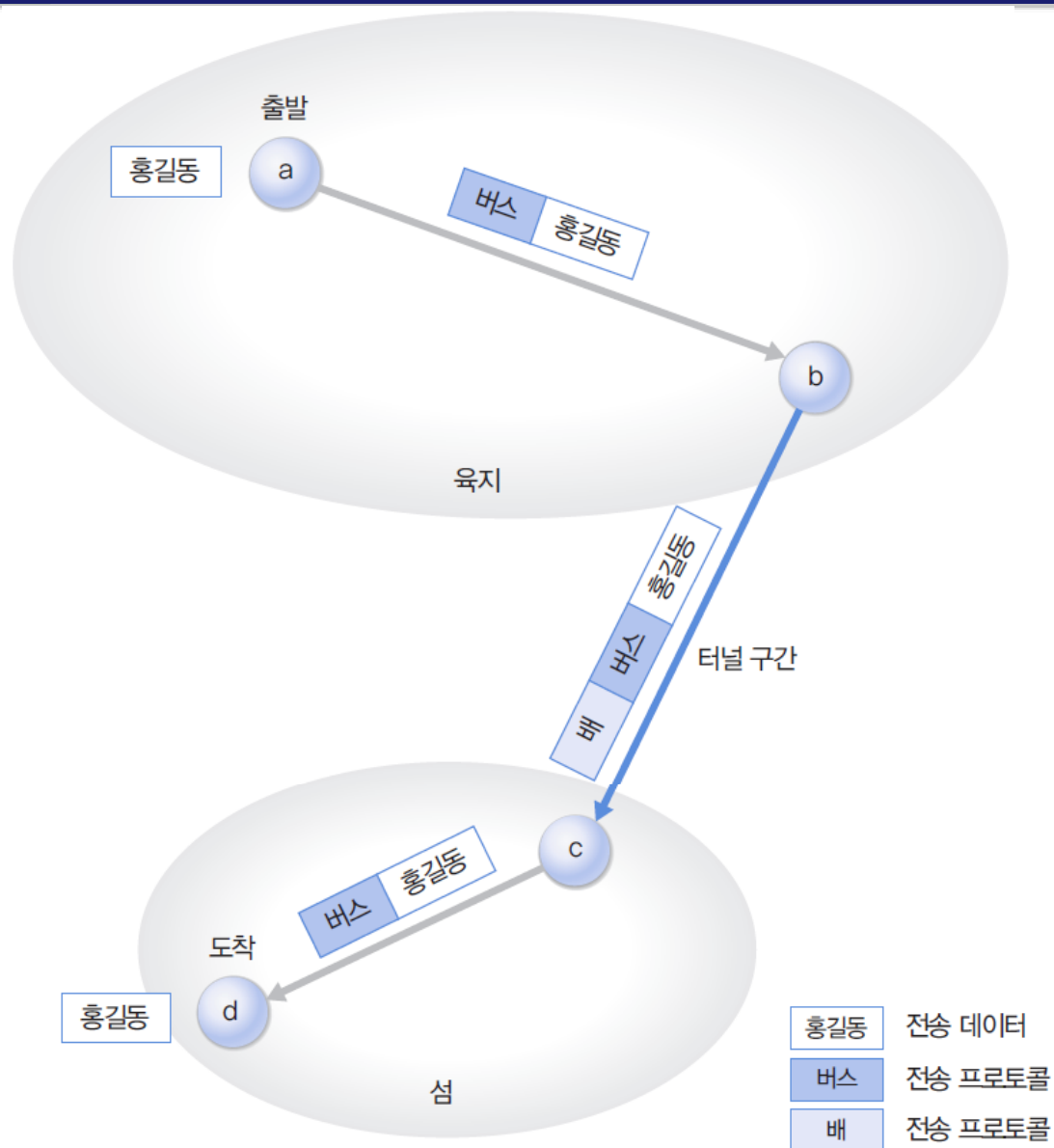


그림 8-4 IP 터널링의 원리



❖ IP 터널링

- 무선 호스트가 움직일 때 이동 IP 프로토콜의 기본 동작 원리
 - 이동 호스트의 움직이면 새로운 위치를 관장하는 포린 에이전트 Foreign Agent FAnew로부터 COA Care of Address를 얻음
 - 이 주소는 이동 호스트의 홈 에이전트 Home Agent HA에 등록되어 FAnew와 HA 사이에 터널을 형성
 - HA로 라우팅된 패킷을 이동 호스트에 전달하려면 새로 형성된 터널을 통해 FAnew로 전달

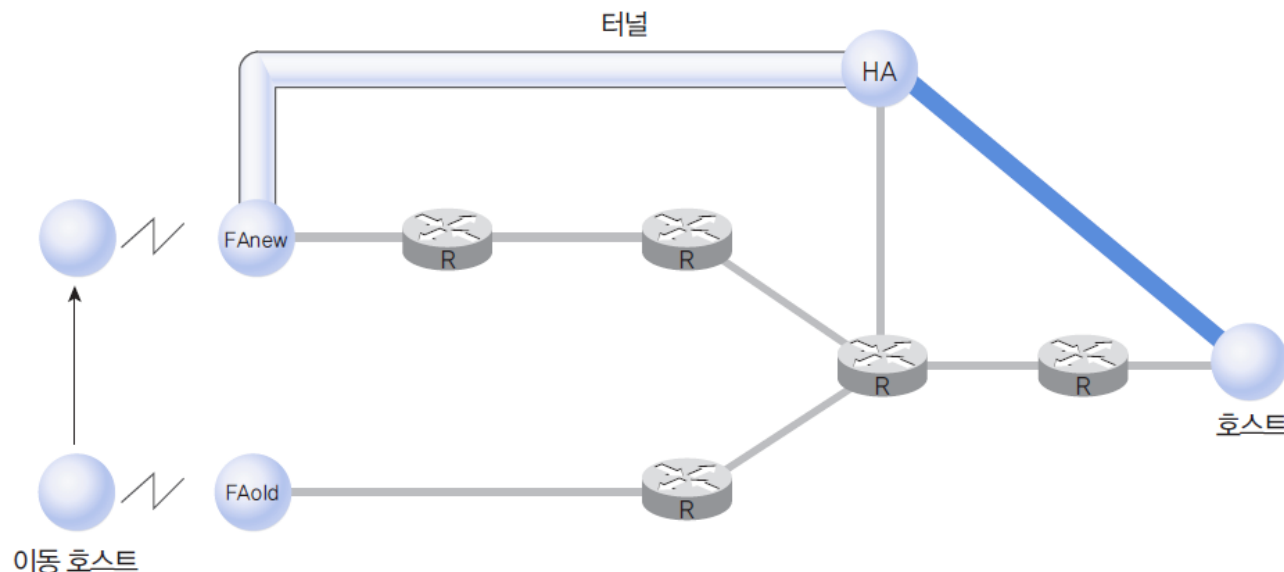


그림 8-5 이동 IP 프로토콜의 기본 동작 원리

02_이동 IP 프로토콜

- 이동 호스트에는 고유 IP 주소인 홈 주소 Home Address HA가 할당, 호스트 위치가 바뀌어도 변하지 않음. 홈 에이전트와 밀접한 관련
- COA는 이동 호스트가 새로 이동한 지역에서 일시적으로 할당된 IP 주소 호스트가 이동할 때마다 새로운 COA가 할당되고 기존 COA는 회수되는 과정이 반복됨
- 송신 호스트에서 이동 호스트까지 패킷 전달 과정
 - 이동 호스트를 목적지로 하는 패킷은 HA에게 전달됨
 - HA는 FA와의 터널을 이용해 FA에게 패킷을 전달함
 - FA는 이동 호스트에게 패킷 전달함
- 홈 에이전트와 이동 에이전트 사이에 설정되는 터널 Tunnel은 원 IP 패킷을
- 목적지까지 전송하기 위한 중간 단계의 새로운 경로임



02_이동 IP 프로토콜

- 터널구간 라우팅 처리

- 원 IP 패킷을 데이터로 취급하는 새로운 형태의 IP 캡슐 패킷이 구성되어 전달. 원 패킷의 Destination Address 필드에는 이동 호스트의 홈 주소가 들어감
- 홈 에이전트에서는 원 패킷을 이동 호스트에 전달하려고 그림처럼 캡슐 패킷으로 변경

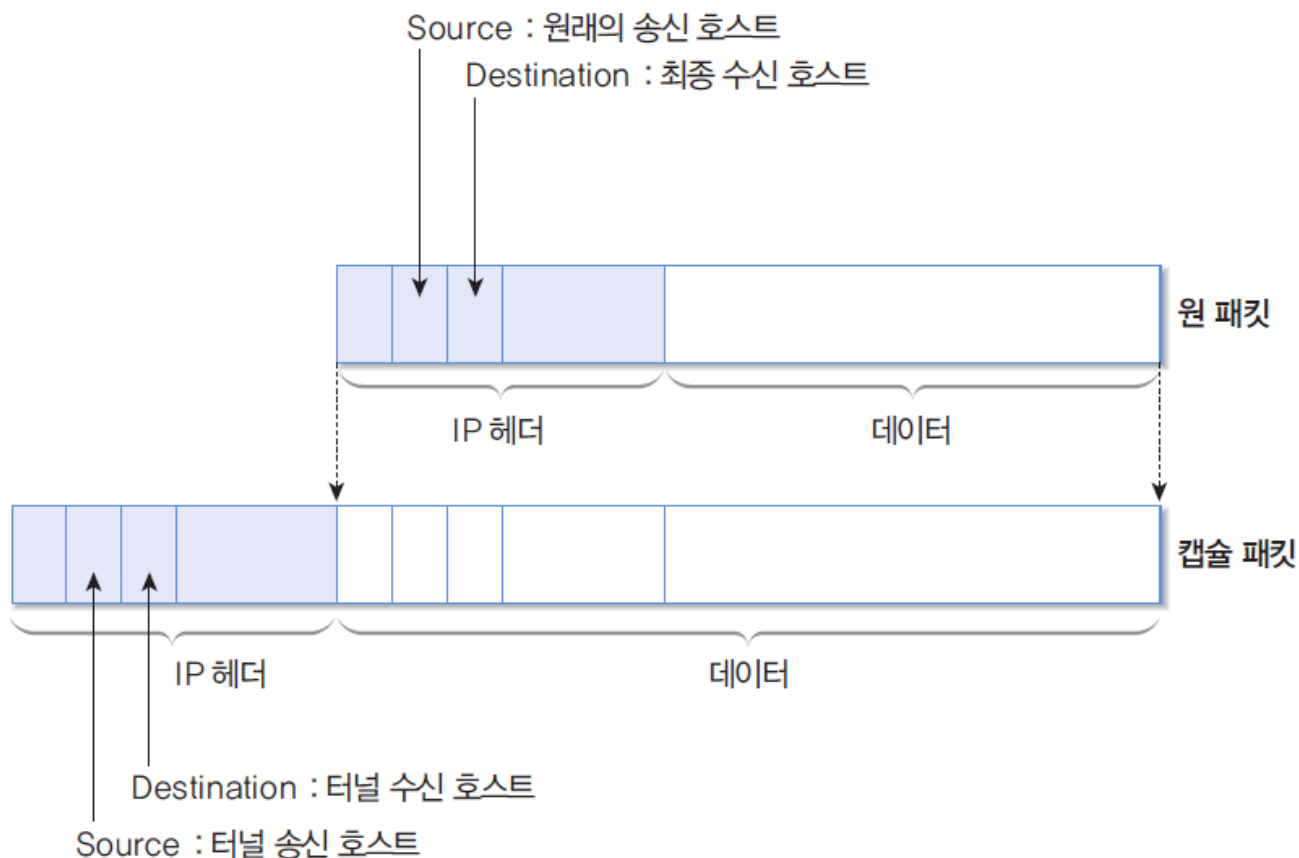


그림 8-6 IP 터널

