7 IP 프로토콜

가천대학교 - 2019학년도 1학기 -

Contents

학습목표

- 네트워크 계층의 필요성과 역할을 이해한다.
- 라우팅 기능을 이해하고 관련 프로토콜을 알아본다.
- 혼잡 제어 기능을 이해한다.
- IP 프로토콜 헤더의 역할을 이해한다.

내용

- 네트워크 계층의 기능
- 라우팅 프로토콜
- IP 프로토콜

01_네트워크 계층의 기능

네트워크 계층

- 송수신 호스트 사이의 패킷 전달 경로를 선택하는 라우팅
- 라우팅 과정에서 일어나는 문제도 처리
  - 네트워크의 특정 지역에 트래픽이 몰리는 현상을 다루는 혼잡 제어
  - 라우터 사이의 패킷 중개 과정에서 다루는 패킷 분할과 병합

- 라우팅
  - 네트워크 구성 형태에 대한 정보는 라우팅 테이블Routing Table이라는 기억 장소에 보관
  - 이 정보를 이용해 패킷이 목적지까지 도달하기 위한 경로 선택
  - 라우팅 테이블 정보는 네트워크 관리자나 네트워크 자신의 판단으로 계속 변경

01_네트워크 계층의 기능

- 혼잡 제어
  - 혼잡 Congestion : 네트워크 패킷 수가 과도하게 증가하는 현상
  - 혼잡제어Congestion Control : 혼잡 현상을 예상하거나 제어하는 기능
  - 혼잡이 발생하면 네트워크 전체의 전송 속도가 급격히 떨어짐
  - 네트워크의 특정 지역에 혼잡이 발생하면 주위로 빠르게 확산될 가능성이 높음

- 패킷의 분할과 병합
  - 상위 전송 계층에서 송신을 요구한 데이터는 최종적으로 MAC 계층의 프레임 구조에 정의된 형식으로 캡슐화되어 물리적으로 전송
  - 전송 계층에서 보낸 데이터가 너무 크면 여러 개의 패킷으로 작게 쪼개어 전송
  - 분할Segmentation : 큰 데이터를 여러 패킷으로 나누는 과정
  - 병합Reassembly : 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 서비스Connectionless Service
  - 패킷의 전달순서, 패킷의 분실 여부 등에서 연결형 서비스보다 신뢰성이 떨어지는 전송 방식
  - 전송 계층에서 네트워크 계층의 비연결형 서비스를 이용할 때는 연결형 서비스를 이용하는 경우보다 자체적으로 오류 제어와 흐름 제어 기능을 더 많이 수행해야 함

  - 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

01_네트워크 계층의 기능

- 연결형 서비스Connection-oriented Service
  - 상대적으로 신뢰성이 높음
  - TCP : 전송 계층의 기능을 지원하는 연결형 프로토콜

01_네트워크 계층의 기능

라우팅Routing

- 패킷의 전송 경로를 지정

- 가상 회선 방식을 사용하는 연결형 서비스
  - 송수신 호스트 사이의 경로 선택은 연결이 설정되는 시점에 한 번만 결정
  - 이후의 패킷들은 이 경로를 따라 목적지까지 전달됨
  - 가상 회선 방식에서는 전송되는 모든 패킷이 동일 경로를 거치고, 패킷의 전달 순서도 일정하게 유지됨

- 비연결형 방식의 데이터그램 사용
  - 연결 설정 과정이 없기 때문에 송수신 호스트 사이에 고정 경로가 존재하지 않음
  - 전송 패킷마다 독립적인 전달 경로 선택

01_네트워크 계층의 기능

- 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. 패킷의 평균 지연 시간, 전체 네트워크의 성능에 대한 영향, 중개 과정에서 거치는 라우터 수의 최소화 등 고려

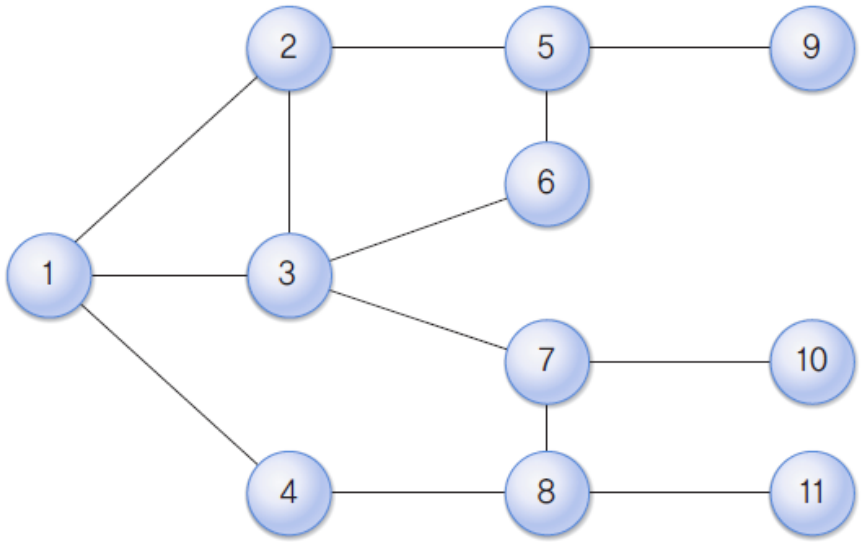
- 정적/동적 라우팅
  - 정적 라우팅Static Routing
    - 패킷 전송이 이루어지기 전에 경로 정보를 라우터가 미리 저장하여 중개
    - 단점 : 경로 정보의 갱신이 어려우므로, 네트워크 변화/네트워크 혼잡도 대처 부족
  - 동적 라우팅Dynamic Routing
    - 라우터의 경로 정보가 네트워크 상황에 따라 적절히 조절됨
    - 현재의 네트워크 링크 상태를 점검해 이를 새로운 경로 배정 시 적용해야 함
    - 각 라우터에서는 이웃 라우터의 존재 유무와 전송 지연 시간 등을 확인할 수 있어야 함
    - 각각의 라우터가 획득한 경로 정보를 다른 라우터들에 통보함으로써 네트워크의 최신 경로 정보를 신속하게 공유하고 갱신해야 함
    - 단점 : 경로 정보의 수집과 관리로 인한 성능 저하

01_네트워크 계층의 기능

- HELLO/ECHO 패킷
  - HELLO : 주변 라우터에 HELLO 패킷을 보내어 주변 경로 정보를 파악하는 용도
  - ECHO : 라우터 사이의 전송 지연 시간을 측정하는 용도
    - ECHO 패킷을 수신한 호스트는 송신 호스트에 즉각 회신하도록 설계됨
    - 측정값의 평균을 구해 해당 라우터까지의 전송 시간 유추
  - 임의의 라우터가 획득한 정보를 다른 라우터에 통보하여 경로 정보 공유
  - 경로 정보가 개별 라우터에 도착하는 시간이 서로 일치하기 않기 때문에 특정 시점에서 각각의 라우터가 바라보는 네트워크 상태는 같지 않을 수 있음
  - 여러 라우터에서 정보가 생성되는 경우에는 네트워크 내부의 경로 정보를 일관성있게 유지하기 어려움

01_네트워크 계층의 기능

- 라우팅 테이블Routing 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

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

그림 7-2 라우팅 테이블

01_네트워크 계층의 기능

- 라우팅 정보의 처리
  - 라우팅 정보가 네트워크 현재 상황을 정확히 반영할 수 있도록 관리
  - 소스 라우팅Source Routing
    - 패킷을 전송하는 호스트가 목적지 호스트까지 전달 경로를 스스로 결정하는 방식
    - 경로 정보를 전송 패킷에 기록함
    - 중간에 있는 라우터들은 라우팅 테이블을 따로 관리할 필요 없음
    - 데이터그램 방식과 가상 회선 방식에서 모두 이용함
    - 가상 회선 방식 : 연결의 초기화 과정에서 경로 정보를 담은 특수 연결 패킷 사용. 중간 라우터는 패킷의 경로 정보를 해석하여 전달 경로 선택
    - 데이터그램 방식 : 모든 패킷의 헤더에 경로 정보가 들어가므로 일반적이 데이터그램 방식에 비해 신뢰성 향상

01_네트워크 계층의 기능

- 분산 라우팅Distributed Routing
  - 라우팅 정보가 분산되는 방식, 패킷의 전송 경로에 위치한 각 라우터가 경로 선택에 참여함
  - 데이터그램 방식에서 많이 사용
  - 네트워크에 존재하는 호스트의 수가 많아질수록 다른 방식보다 효과적일 수 있음

- 중앙 라우팅Centralized Routing
  - RCCRouting Control Center라는 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 정보 관리부담이 줄어듬
  - 단점 : 네트워크 규모가 커질수록 RCC에 과중한 트래픽을 주어 전체 효율이 떨어짐

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

01_네트워크 계층의 기능

❖ 혼잡 제어

- 혼잡Congestion : 네트워크 성능 감소 현상이 급격하게 악화되는 현상
- 혼잡 제어Congestion 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

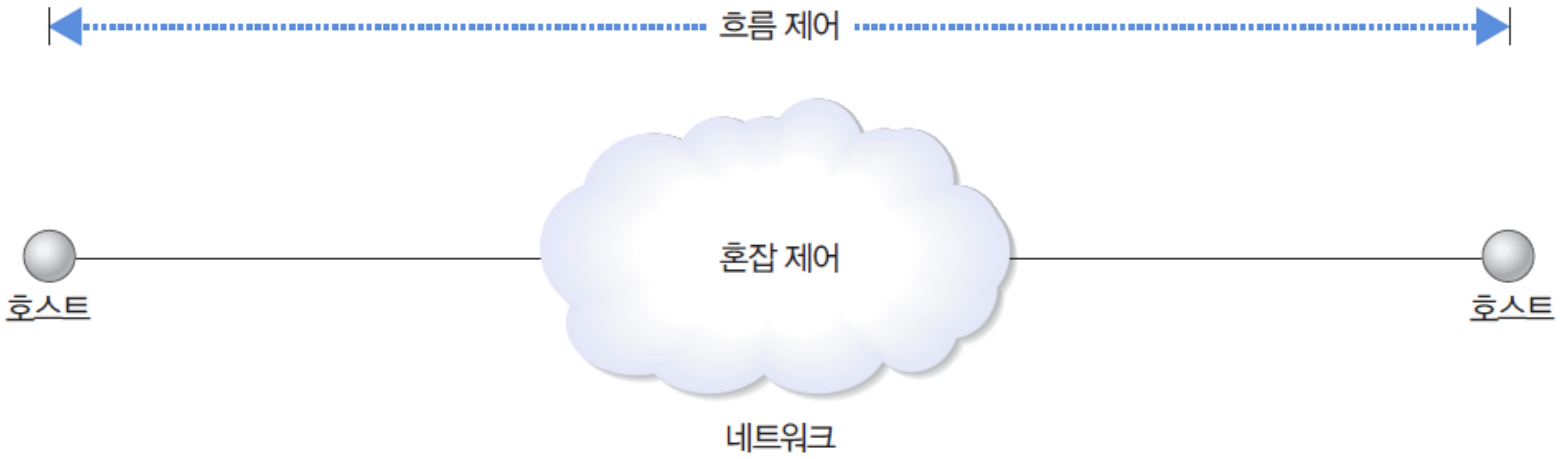


그림 7-3 흐름 제어와 혼잡 제어

01_네트워크 계층의 기능

- 혼잡의 원인
  - 기본적으로 네트워크의 처리 능력보다 지나치게 많은 패킷이 입력되면 혼잡 발생
  - 개별 라우터의 출력 선호를 통한 전송 용량이 부족해 전송하지 못한 패킷이 버퍼에 저장되고, 입력 선로로 들어오는 패킷이 늘면서 버퍼 용량은 더욱 부족해짐
  - 라우터의 내부 버퍼 용량이 부족이 심화되어 더 이상 패킷 보관 불가 초래
  - 송신 호스트는 타임아웃 동작을 통해 패킷 재전송
  - 네트워크로 송신되는 패킷 양 증가
  - 버퍼 용량을 늘리면 패킷 전송 지연도 증가
  - 패킷 전송 지연 시간이 송신 호스트가 설정한 타임아웃 시간보다 크면 패킷 재전송 과정 증가
  - 네트워크의 패킷 양과 중복 패킷 수신 현상 증가
  - 결과적으로 네트워크 혼잡도 증가 악순환 초래

01_네트워크 계층의 기능

- 혼잡의 원인
  - 초기 혼잡 과정에서 패킷의 전송 지연이 점점 증가할 때, 타임 아웃 시간이 작으면 혼잡도가 급격히 증가
  - 패킷 도착 순서가 다른 상황에서 패킷을 분실 처리하면 패킷 재전송 현상이 발생해 네트워크 혼잡도 증가
  - 수신 패킷에 대해 즉식 응답하는 방식은 수신 패킷 모두에 대해 개별 응답 패킷 발생. 의도적으로 피기배킹을 사용하면 응답 시간이 느려져 타임아웃 증가
  - 패킷 생존 시간을 작게 하면 패킷이 강제로 제거되어 타임아웃 증가
    - 수신 호스트에 도착할 가능성이 희박한 패킷의 생존 시간을 너무 크게 설정하면 네트워크에 불필요한 부하 발생
  - 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 버스트Burst 현상이 원인
  - 트래픽 성형Traffic Shaping : 송신 호스트가 전송하는 패킷의 발생 빈도가 네트워크에서 예측할 수 있는 전송률로 이루어지게 하는 기능
  - 리키 버킷Leaky Bucket 알고리즘

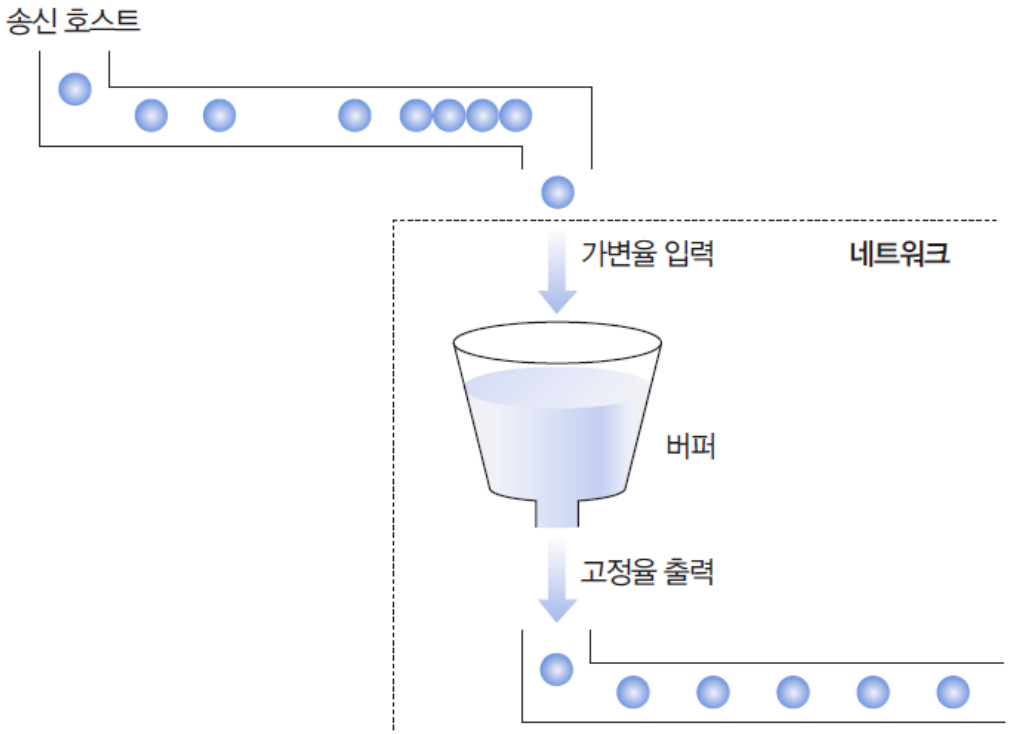


그림 7-4 리키 버킷 알고리즘

01_네트워크 계층의 기능

- 혼잡 제거
  - 가상회선 방식을 사용하는 서브넷에서 혼잡을 감지했을 때 가장 간단한 방법은 혼잡이 사라질 때까지 연결 설정을 불허
  - 실제 네트워크에서는 일부 지점에서 혼잡이 발생하는 경우가 많음
  - 특정 지역의 혼잡이 다른 지역으로 확대되지 않도록 하는 것이 중요
  - 혼잡 제거를 위해 호스트와 서브넷이 가상 회선 연결 과정에서 협상을 함.

    전송 과정에서 사용하는 대역을 미리 할당 받음 (자원 예약 방식)
    - 네트워크에서 수용 불가능한 정도로 트래픽이 발생하는 일을 사전에 예방함
    - 단점 : 전송 대역을 해당 사용자가 이용하지 않더라도 다른 사용자가 이용하지 못함

01_네트워크 계층의 기능

- 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

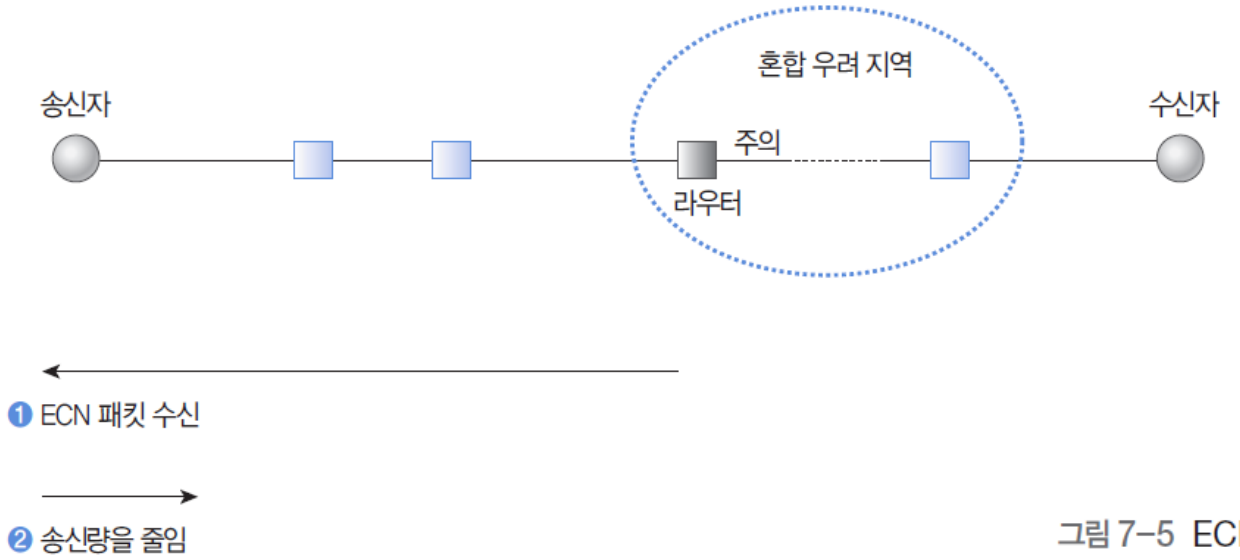


그림 7-5 ECN 패킷

  - 특정 라우터에서 주의 표시를 시작하면 이후 경로에 위치한 라우터에서도 주의 표시할 가능성이 높아짐. ECN 패킷이 여러 라우터에서 동시에 발생할 가능성이 높음
  - 최초로 ECN 패킷을 발생시킨 라우터는 전송되는 패킷의 헤더 내부에 ECN-Echo와 같은 임의의 표시를 하여 목적지까지 도착하는 동안 거치는 라우터가 ECN 패킷을 더 이상 발생하지 않도록 해야 함
  - ENC 패킷을 전달받은 송신 호스트는 정해진 비율에 따라 송신 패킷의 양을 줄여 전송
  - 임의의 시간 경과 후에도 ECN 패킷이 계속 들어오면 송신 패킷의 양을 추가로 줄임(9장 참조)

02_라우팅 프로토콜

❖ 간단한 라우팅 프로토콜

- 네트워크 거리 기준 : 라우터의 개수, 홉Hop의 수로 판단

- 최단 경로 라우팅
  - 패킷이 목적지에 도달할 때까지 라우터 수가 최소화될 수 있도록 경로 선택
  - 장점 : 간단한 형식으로 적용가능

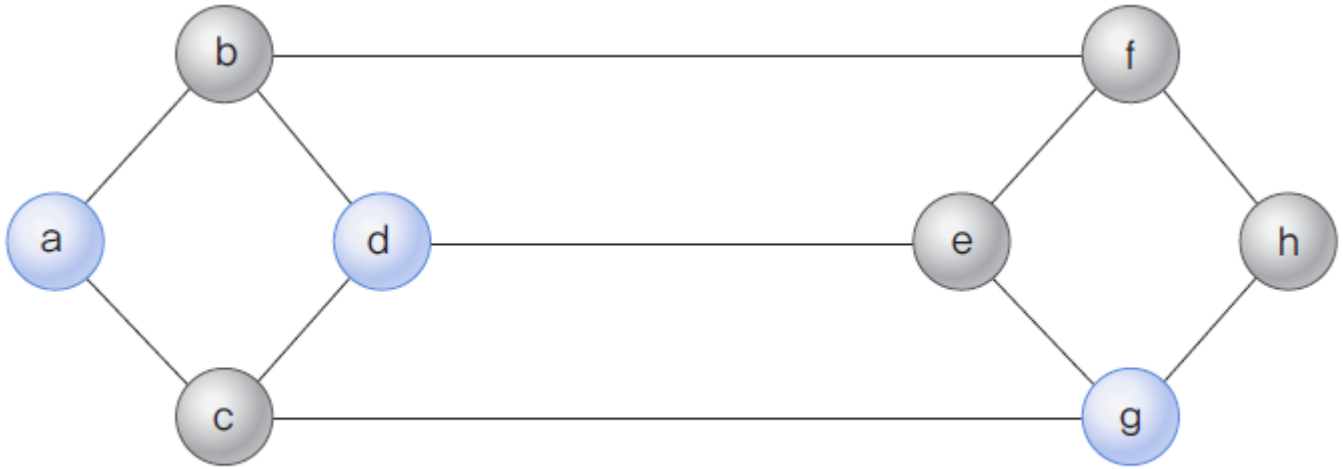


그림 7-6 최단 경로 라우팅

- 홉 수 외에 거리 기준이 될 수 있는 요소 : 패킷 전송 지연 시간, 전송 대역폭, 통신 비용 등

02_라우팅 프로토콜

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

❖ 거리 벡터Distance Vector 라우팅 프로토콜

- 라우터가 자신과 연결된 이웃 라우터와 라우팅 정보를 교환하는 방식
- 거리 벡터 프로토콜을 사용하는 호스트나 라우터 간 정보 교환
- 교환 정보 : 각각의 라우터에서 전체 네트워크에 소속되는 개별 네트워크까지 패킷을 전송하는데 걸리는 거리 정보
- 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 x와 연결된 이웃 네트워크에 대한 연결 정보를 보관

링크 벡터 L(x) = [포트(1), 포트(2), ..., 포트(m), ..., 포트(M)]

그림 7-7 링크 벡터 L(x)

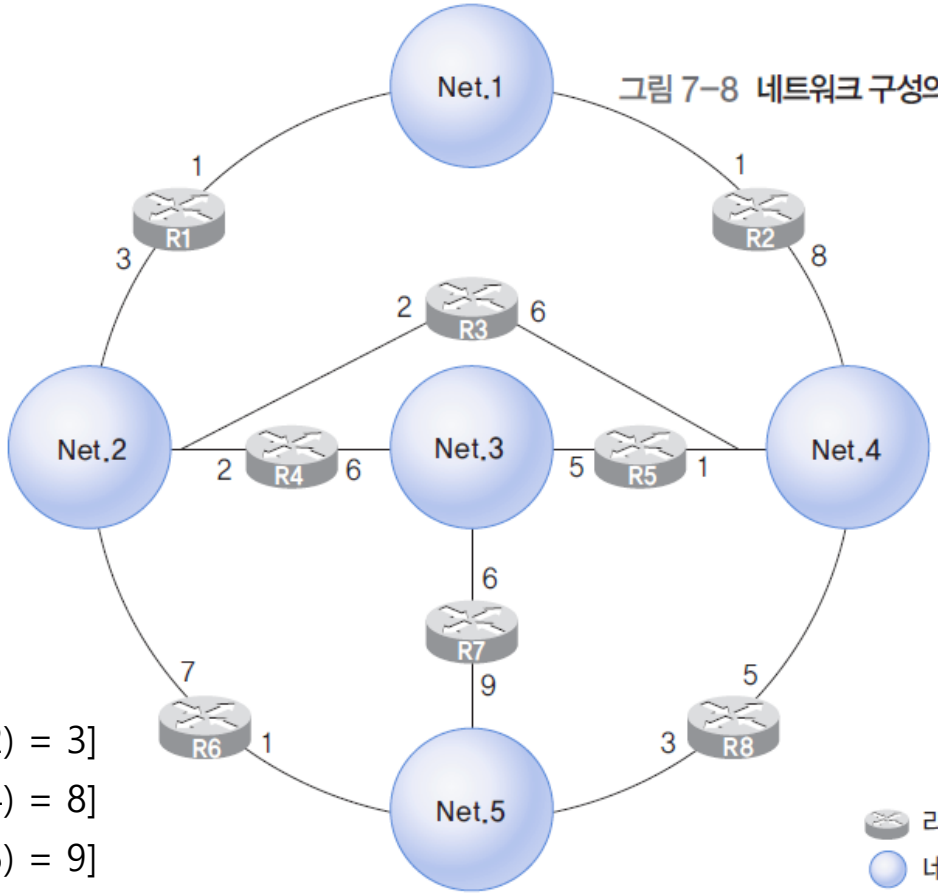


그림 7-8 네트워크 구성의 예

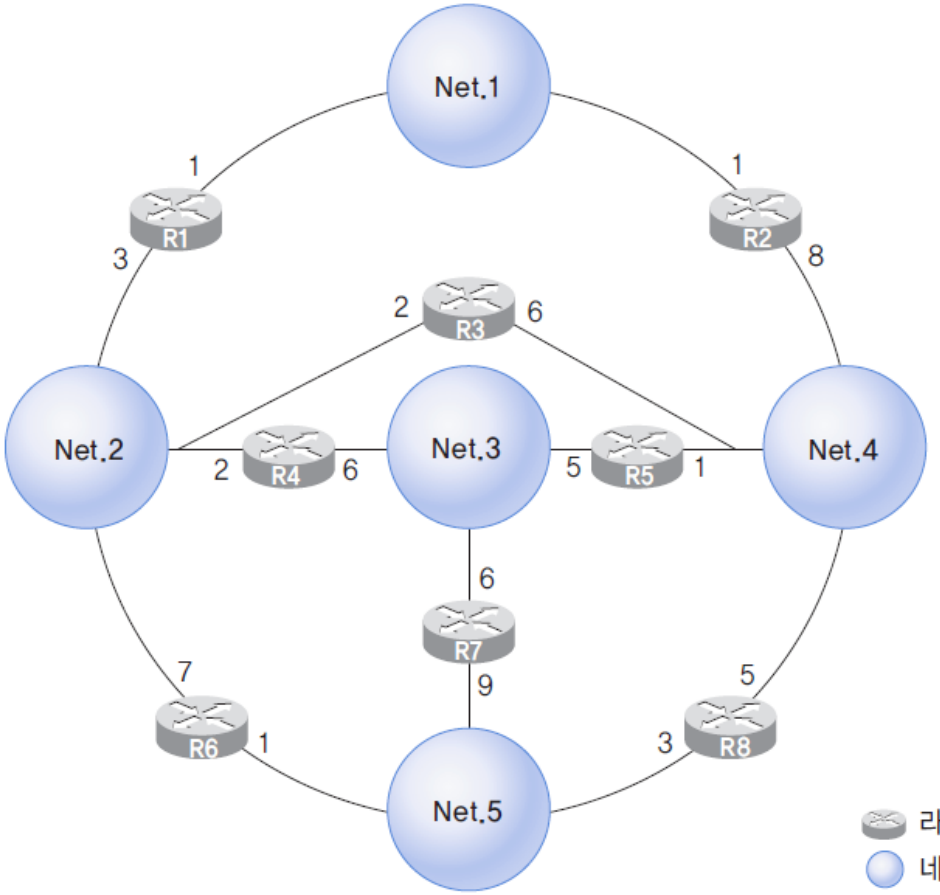
- L(R1) = [포트(Net.1) = 1, 포트(Net.2) = 3]
- L(R2) = [포트(Net.1) = 1, 포트(Net.4) = 8]
- L(R7) = [포트(Net.3) = 6, 포트(Net.5) = 9]

02_라우팅 프로토콜

- 거리 벡터
  - 전체 네트워크에 소속된 개별 네트워크들까지의 거리 정보를 관리

거리 벡터 D(x) = [거리(1), 거리(2), ..., 거리(n), ..., 거리(N)]

그림 7-9 거리 벡터 D(x)



- [그림 7-8]에서

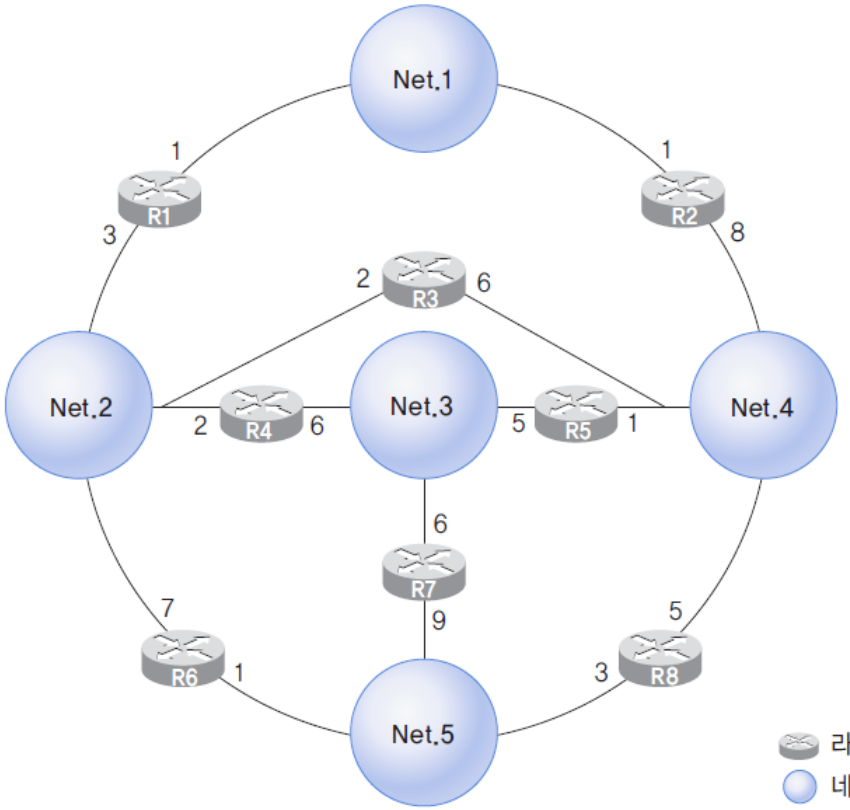
  D(R1)= [거리(Net.1) = 1,
  거리(Net.2) = 1,
  거리(Net.3) = 2,
  거리(Net.4) = 2,
  거리(Net.5) = 2]

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) = [홉(1), 홉(2), ... , 홉(n), ... , 홉(N)]

그림 7-10 다음 홉 벡터 H(x)



- [그림 7-8]에서

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

02_라우팅 프로토콜

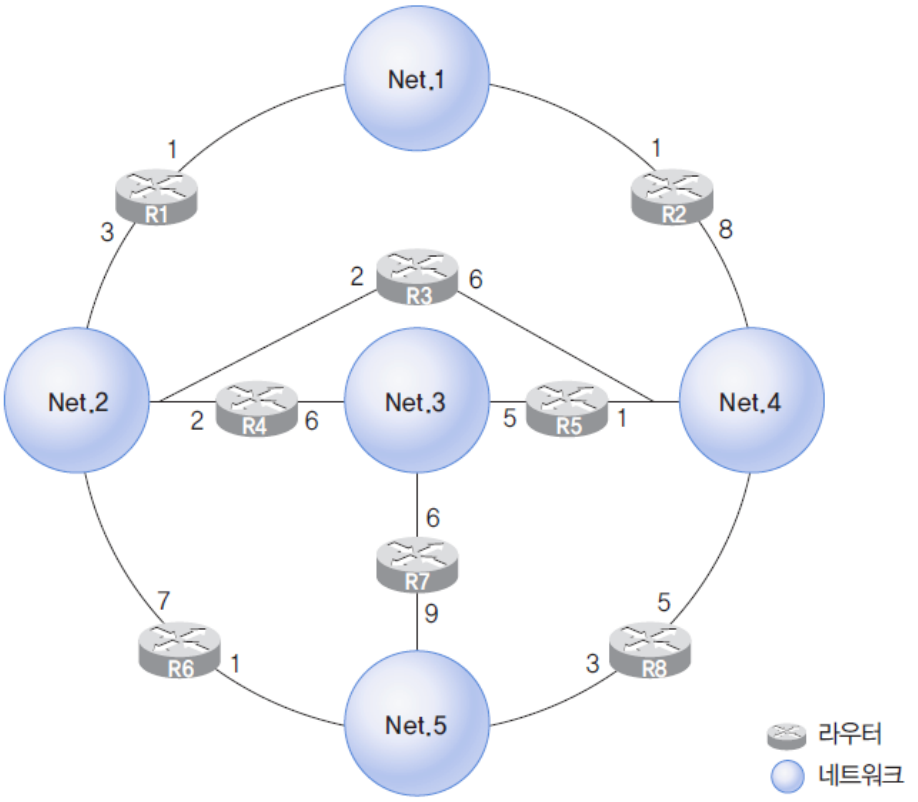
- RIP Routing Information Protocol 프로토콜
  - 거리 벡터 방식의 내부 라우팅 프로토콜 중에서 가장 간단하게 구현된 것
  - 소규모 네트워크 환경에 적합, 현재 가장 많이 사용하는 라우팅 프로토콜
  - RIP이 제대로 동작하려면 이웃 라우터가 제공하는 거리 벡터 정보가 임의의 짧은 시간 내에 모두 도착해야 함(현실적으로 구현 어려움)
  - RIP 패킷은 UDP 프로토콜 사용. 비신뢰성 전송을 제공하므로 RIP 패킷이 전송과정에 사라질 수도 있음
  - 라우팅 테이블 적용
    - 입력되는 거리 벡터 정보가 새로운 네트워크의 목적지 주소이면 라우팅 테이블에 적용
    - 입력되는 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



02_라우팅 프로토콜

- 임의의 시점에 거리 벡터 정보

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

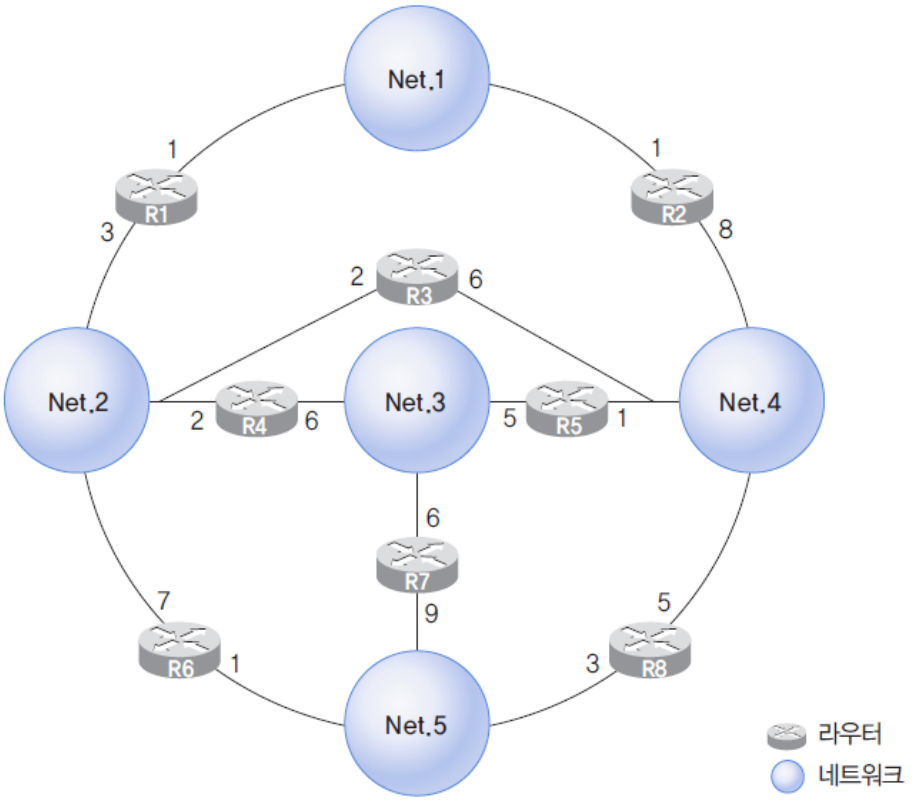
  R3 = [2, 1, 2, 1, 2]

  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



02_라우팅 프로토콜

- 벡터 정보를 교환하기 위해 다음과 같은 패킷 구조를 사용함
  - 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

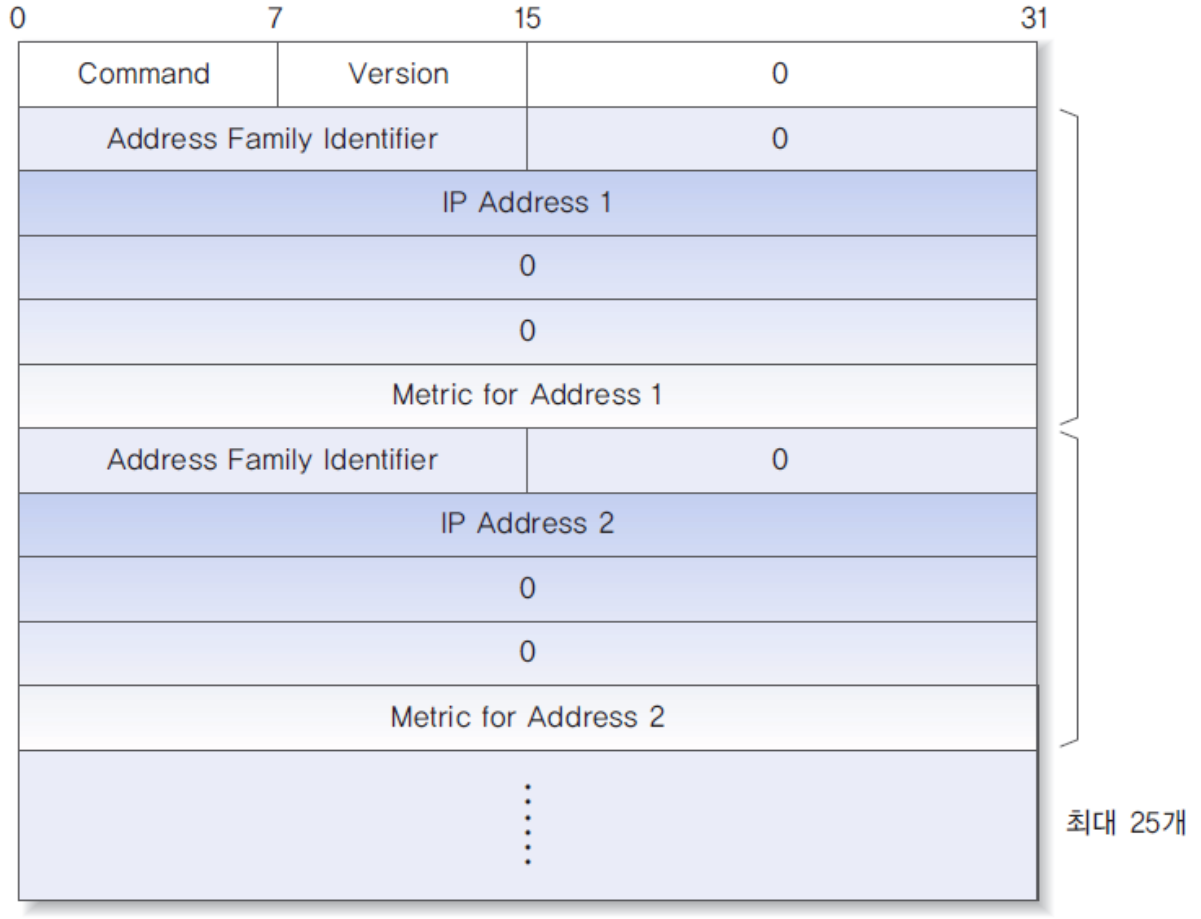


그림 7-11 RIP 패킷의 구조

02_라우팅 프로토콜

링크 상태Link 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
  - 거리 벡터 라우팅 프로토콜은 각 라우터가 상당한 양의 정보 전송을 요구 받고, 링크 상태가 많이 변하면 동작 과정에서 많은 시간 소요
  - 거리 벡터 라우팅 프로토콜은 주기적으로 정보 전달. 링크 상태 라우팅 프로토콜은 이웃 라우터와 연결된 상황에 변화가 있을 때만 정보 전달
- 플러딩Flooding 기법 : 임의의 라우터가 이웃한 모든 라우터에 정보를 전달하고, 다시 이들 라우터가 주변의 모든 라우터에 정보를 전달하는 방식으로 동작 (입력된 선로를 통해 출력되지 않도록 주의 필요)
- 예) OSPFOpen Shortest Path First
- 링크 상태 라우팅 프로토콜과 거리 벡터 라우팅 프로토콜의 가정
  - 각 라우터는 이웃 라우터의 주소 정보, 패킷 전송에 필요한 비용 정보를 알고 있음. 비용 종류는 패킷 전송 지연 등 여러가지 가능

02_라우팅 프로토콜

❖ 외부 라우팅 프로토콜

- 내부 라우팅 프로토콜
  - 거리 벡터 방식을 사용하는 RIP
  - 링크 상태 방식을 사용하는 OSPF
- 외부 라우팅 프로토콜
  - 경로 벡터 Path Vector 프로토콜 : 단순히 연결 가능한지에 대한 정보만 제공
- BGP Border Gateway Protocol : 서로 다른 종류의 자율시스템에서 동작하는 라우터가 라우팅 정보를 교환
  - TCP 프로토콜을 사용

표 7-3 TCP 프로토콜에서 제공하는 메시지의 종류

메시지	설명
Open	다른 라우터와 연관Relationship을 설정한다.
Update	라우팅 관련 정보를 전달한다.
KeepAlive	Open 메시지에 대한 응답 기능과 주변 라우터와의 연관을 주기적으로 확인한다.
Notification	오류 상태를 통보한다.

03_IP 프로토콜

IP Internet Protocol 프로토콜

- 인터넷 환경에서 네트워크 계층의 데이터 전송 프로토콜로 사용
- 호스트 주소 표기, 패킷 분할에 관한 기능 지원
- 단대단 형식의 오류 제어나 흐름 제어 기능은 제공하지 않음
- 전송 패킷이 수신 호스트에 100% 도착하는 것을 보장하지 않음
- IP 프로토콜에서 제공하지 않는 전송 오류 문제를 상위 계층에서 고려
- 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

03_IP 프로토콜

❖ IP 헤더 구조

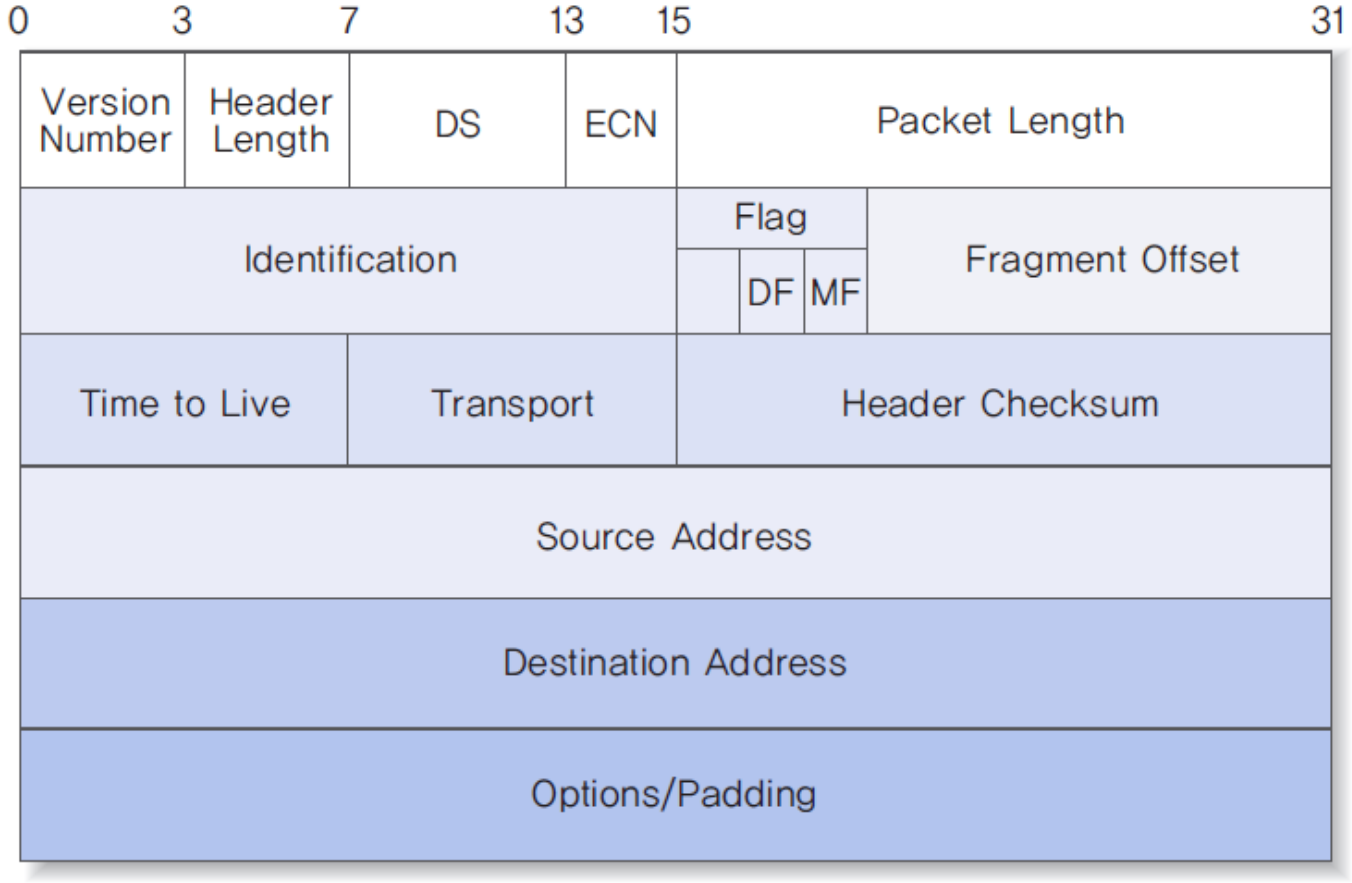
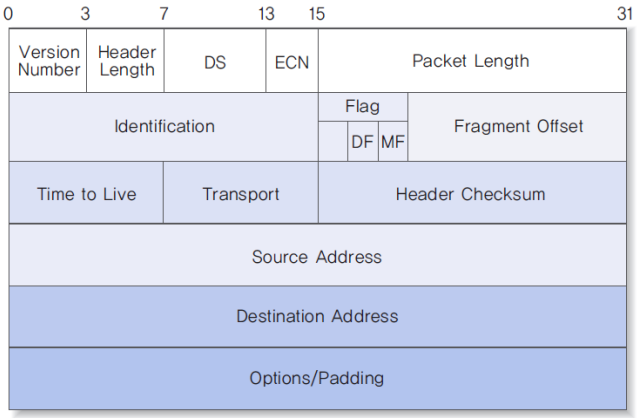


그림 7-12 IP 헤더의 구조

03_IP 프로토콜



- DS Differentiated Services /ECN Explicit Congestion Notification
  - 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

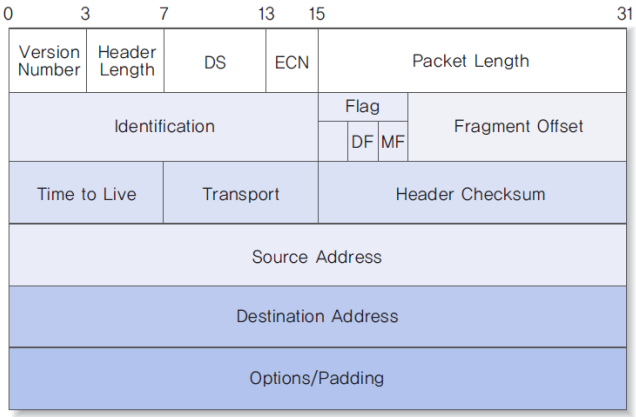
03_IP 프로토콜

- DS Differentiated Services
  - 사전에 서비스 제공자와 서비스 이용자 사이에 서비스 등급에 대해 합의
  - 동일한 DS 값을 갖는 트래픽들은 동일한 서비스 등급으로 처리됨
  - 차등 서비스의 기준이 되는 레이블 값으로 64개의 트래픽 클래스를 정의 가능
- ECN Explicit 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

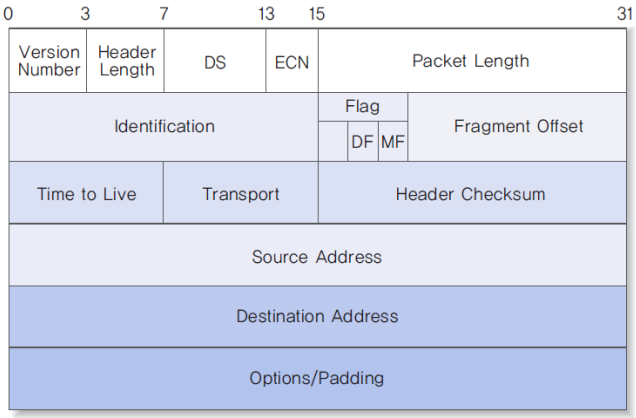
필드 값	의미
00	IP 패킷이 ECN 기능을 사용하지 않음을 의미한다.
01(ECT 1)	TCP 프로토콜도 ECN 기능을 지원한다는 의미이다.
10(ECT 0)	TCP 프로토콜도 ECN 기능을 지원한다는 의미이다.
11(CE: Congestion Experienced)	라우터가 송신 호스트에 혼잡을 통지할 때 사용한다.

03_IP 프로토콜



- 패킷 분할
  - Identification(식별자 혹은 구분자)
    - IP 헤더의 두 번째 워드에는 패킷 분할과 관련된 정보가 포함
    - Identification은 송신 호스트가 지정하는 패킷 구분자 기능을 수행함
    - 분할한 패킷에 동일한 고유번호 부여. 수신 호스트가 패킷 다시 병합 가능
    - 패킷을 분할하지 않으면 패킷을 전송할 때마다 이 필드의 값 하나씩 증가
  - DF Don't Fragment : 패킷이 분할되지 않도록 함
    - 값을 1로 지정하면 패킷 분할을 막을 수 있음
    - 수신 호스트가 패킷 병합 기능이 없을 때 사용
    - 중간 경유 네트워크에서 처리 가능한 패킷의 크기보다 큰 IP 패킷에 DF 필드가 설정되어 있으면 패킷을 버림
  - MF More Fragment
    - MF필드 값을 1로 지정하여, 분할 패킷이 뒤에 계속됨을 표시
    - 마지막 패킷은 MF 비트를 0으로 지정하여 분할 패킷이 더 없음을 표시
  - Fragment Offset(분할 옵셋)
    - 저장되는 값은 분할된 패킷의 내용이 원래의 분할 전 데이터에서 위치하는 상대 주소값
    - 값은 8바이트의 배수. 예) 값이 64라면 원래 데이터에서 64 x 8 = 512번째에 위치

03_IP 프로토콜



- 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

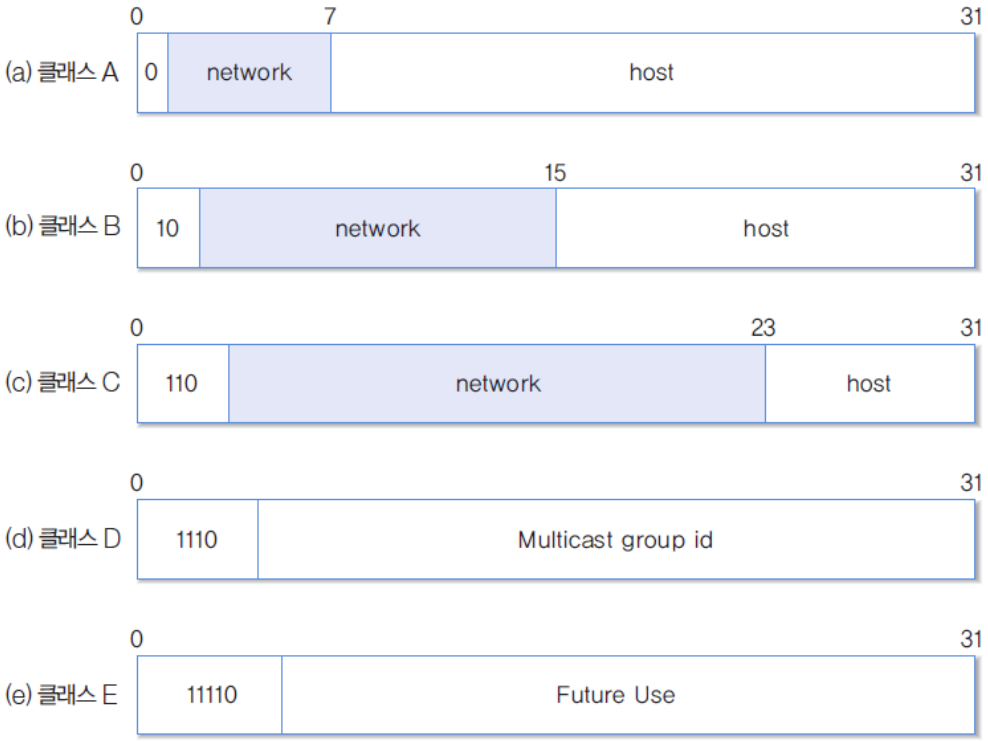


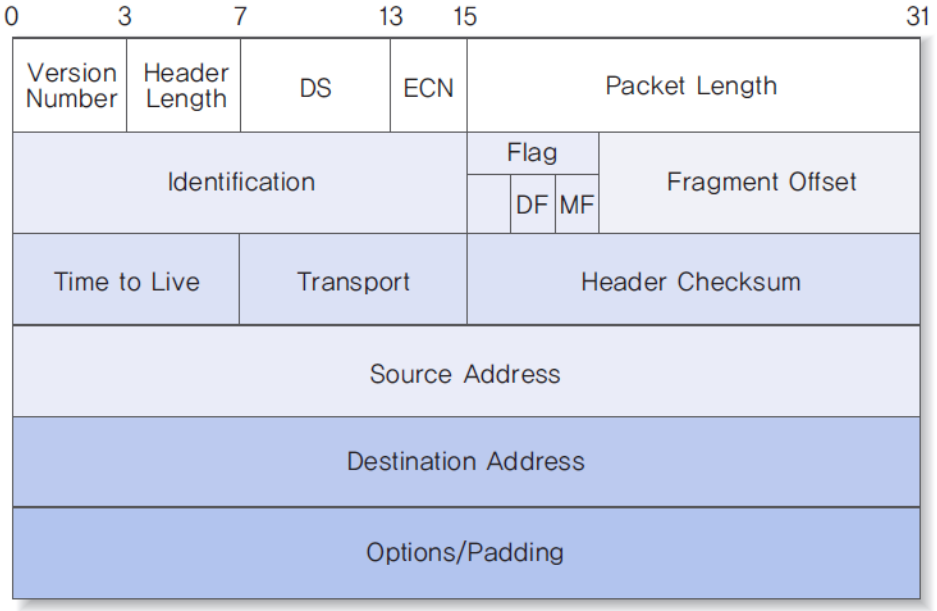
그림 7-13 IP 주소 체계

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

03_IP 프로토콜

- 기타 필드
  - Version Number(버전 번호) : IP 프로토콜의 버전 번호
    - IPv4, IPv6
  - Header Length(헤더 길이) : IP 프로토콜 헤더 길이를 32비트 워드 단위로 표시
    - 일반 패킷인 경우 Options, Padding을 제외하고 최소 5의 값을 가짐
  - 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) : IP 프로토콜에 데이터 전송을 요구한 전송계층의 프로토콜

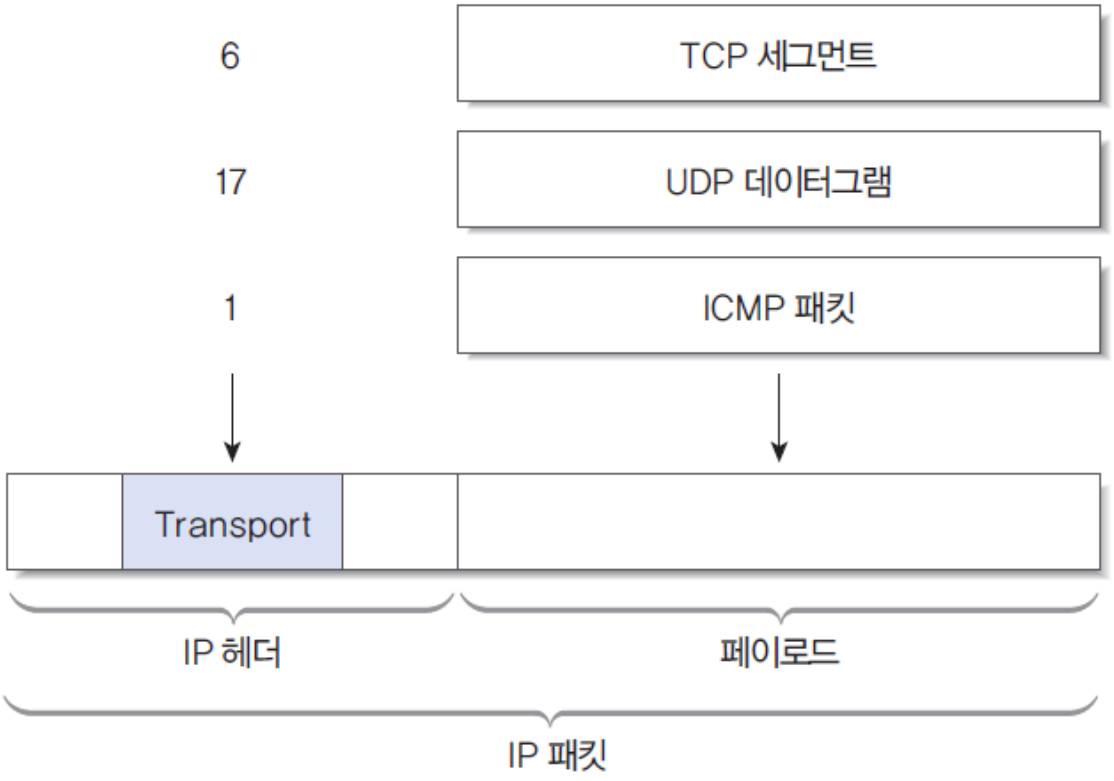
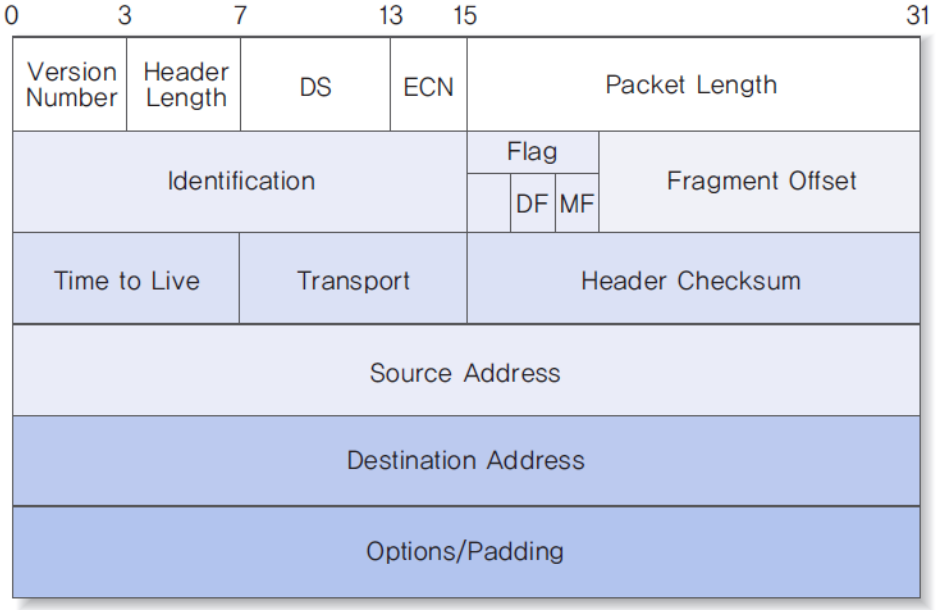


그림 7-14 Transport 필드

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



03_IP 프로토콜

❖ 패킷 분할

- 분할의 필요성

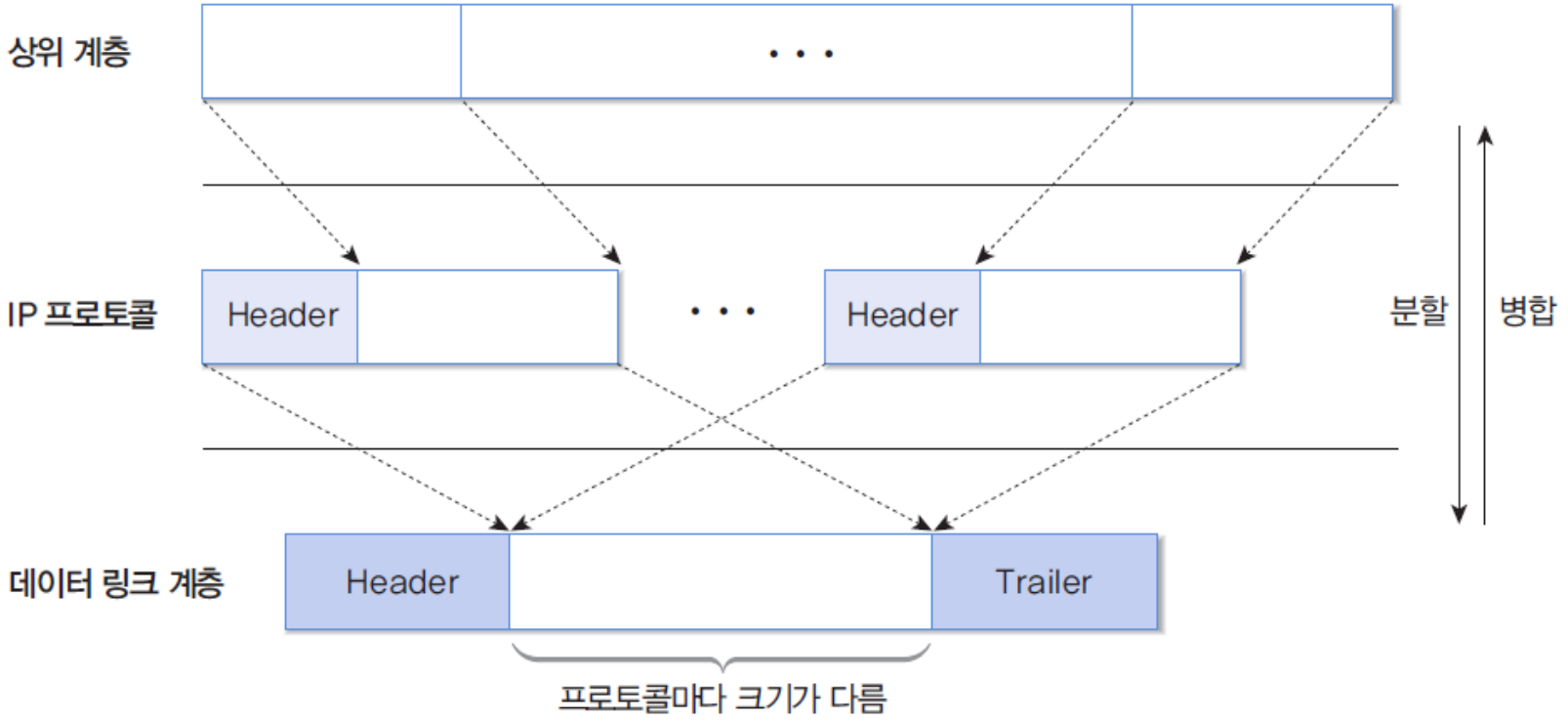


그림 7-15 패킷 분할의 필요성

- 전송경로에 위치한 라우터에서 패킷 분할
  - 라우터 좌우에 연결된 LAN이 서로 다를 수 있기 때문에 데이터 링크 계층에 위치한 프레임 크기가 프로토콜마다 달라짐

03_IP 프로토콜

- 분할의 예
  - IP 헤더를 제외한 전송 데이터의 크기는 380바이트
  - 패킷은 최대 크기가 128바이트라고 가정
    - IP 패킷 헤더는 20바이트
    - 분할 패킷에 보관 가능 데이터 최대 크기 : 108 bytes / 8 bytes => 정수 값 x 8 = 104 bytes
    - 380 bytes / 104 bytes = 3개 … 68 bytes => 패킷 수는 4개
    - 108 bytes / 8 bytes = 13

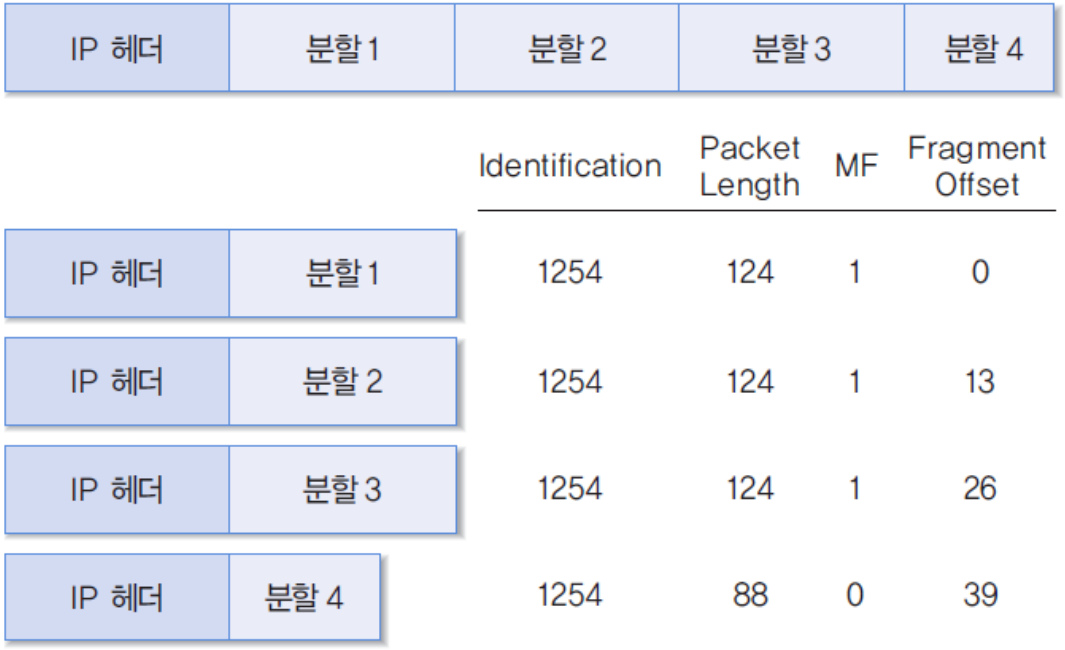


그림 7-16 패킷 분할의 예

03_IP 프로토콜

DHCP Dynamic Host Configuration Protocol 프로토콜

- IP 주소를 여러 컴퓨터가 공유해서 사용

(예)

- 대학 내 여러 실습실에서 1,000개의 컴퓨터가 설치되어 있는 경우
- 고정 IP를 사용하는 경우 1,000개의 IP 필요
- 늘 사용하는 것이 아니라 실습 때만 사용
- DHCP를 사용하여 현재 사용하는 컴퓨터에만 IP 주소를 자동으로 할당
  - DHCP 서버에 자동으로 할당 가능한 IP 주소를 풀Pool로 저장하여 관리
  - 클라이언트로부터 IP 주소 요청이 오면 풀에서 하나의 IP 주소 할당
  - 사용이 끝나면 다시 IP 주소를 풀로 반환하여 다른 호스트가 사용 가능

03_IP 프로토콜

- DHCP 메시지
  - IP 주소를 원하는 클라이언트는 DHCP 서버에 요청 메시지 전성
  - 서버는 이에 대한 응답 메시지 회신

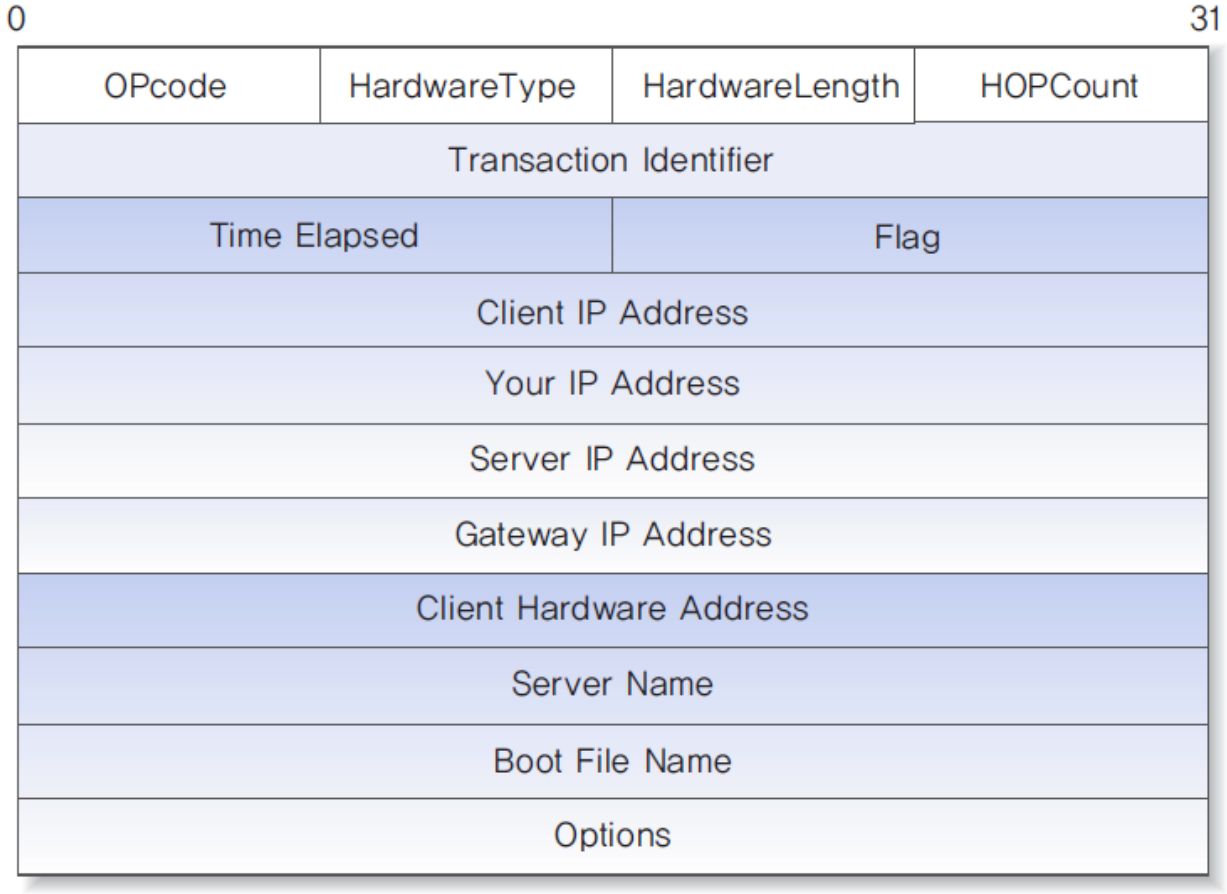


그림 7-17 DHCP 메시지

03_IP 프로토콜

- Opcode : 요청 메시지는 1, 응답 메시지는 2
- HardwareType : 이더넷 등과 같은 하위 계층의 하드웨어 유형
- HardwareLength : 하드웨어 주소의 길이
- HOPCount : 패킷이 전달되는 최대 홉의 수
- Transaction Identifier : 클라이언트의 요청이 있을 때 지정하는 임의의 숫자. 서버는 지정된 번호로 응답. DHCP 메시지는 브로드캐스팅 방식으로 전송되므로 클라이언트와 서버 간 논리적인 세션 형성을 목적으로 사용
- Time Elapsed : 클라이언트가 부팅된 이후의 경과 시간
- Flag : 현재 첫번째 비트만 사용. 유니캐스트인지 멀티캐스트인지 구분
- Client IP Address : 클라이언트가 자신의 IP 주소 지정. 모르면 0으로 표시
- Your IP Address : 서버가 응답 메시지로 권고해주는 클라이언트 IP 주소
- Server IP Address : 서버의 IP 주소. 모르면 0으로 표시
- Gateway IP Address : 클라이언트의 디폴트 라우터 IP 주소
- Client Hardware Address : 클라이언트의 하드웨어 주소
- Server Name : 서버의 도메인 네임. 64바이트
- Boot File Name : 추가 정보를 보관하고 있는 파일 경로명. 128바이트
- Options : 필요한 추가 정보. 64바이트

03_IP 프로토콜

- DHCP 프로토콜의 주요 메시지
  - DHCP_DISCOVER : 클라이언트가 DHCP 서버를 찾기 위해 전송하는 브로드캐스트 메시지
    송신자 주소는 0.0.0.0, 수신자 IP 주소는 브로드캐스팅 주소
  - DHCP_OFFER : 클라이언트의 DHCP_DISCOVER 메시지에 대한 응답으로 DHCP 서버가 응답하는 메시지
    Your IP Address 필드에 권고하는 IP 주소, Server IP Address 필드에 서버 IP 주소 지정
    클라이언트 IP 주소가 결정되지 않았으므로 수신자 IP 주소는 브로드캐스팅 주소
  - DHCP_REQUEST : 주소를 권고한 DHCP 서버에 DHCP_REQUEST 메시지를 전송하여 권고한 주소를 사용한다고 알림 (여러 서버로 부터 다수의 DHCP_OFFER를 받을 수도 있음)
  - DHCP_ACK : 권고한 IP 주소가 최종적으로 사용 가능한지 판단 후 사용 가능하면 DHCP_ACK 메시지를 전송
  - DHCP_NACK : 클라이언트가 DHCP_DISCOVER 과정을 다시 하도록 함 (DHCP 서버는 동일한 IP 주소를 여러 클라이언트에게 권고할 수도 있음)

03_IP 프로토콜

- DHCP 프로토콜의 동작 과정

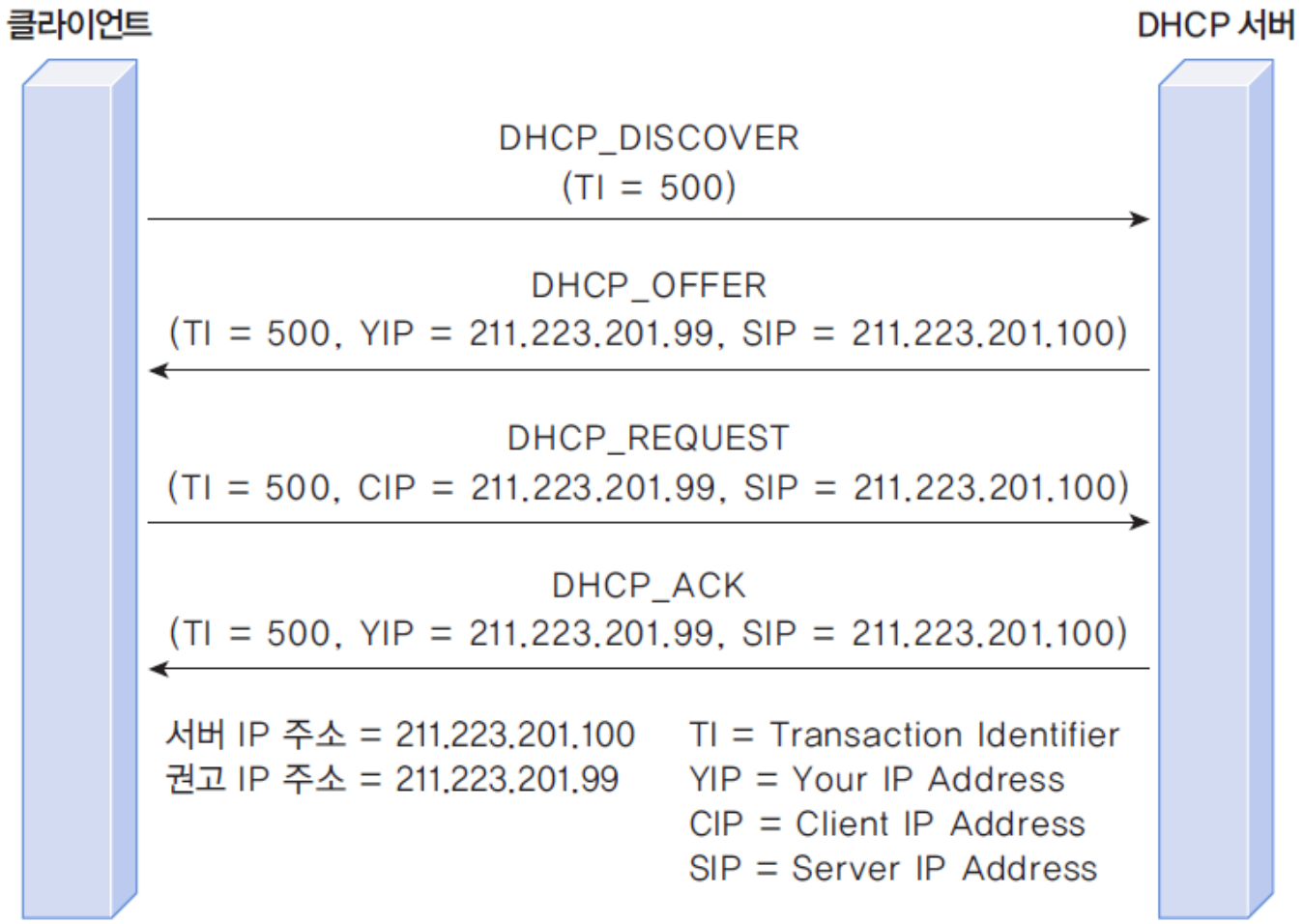


그림 7-18 DHCP 프로토콜의 동작 과정

03_IP 프로토콜

- UDP/IP 프로토콜의 캡슐화

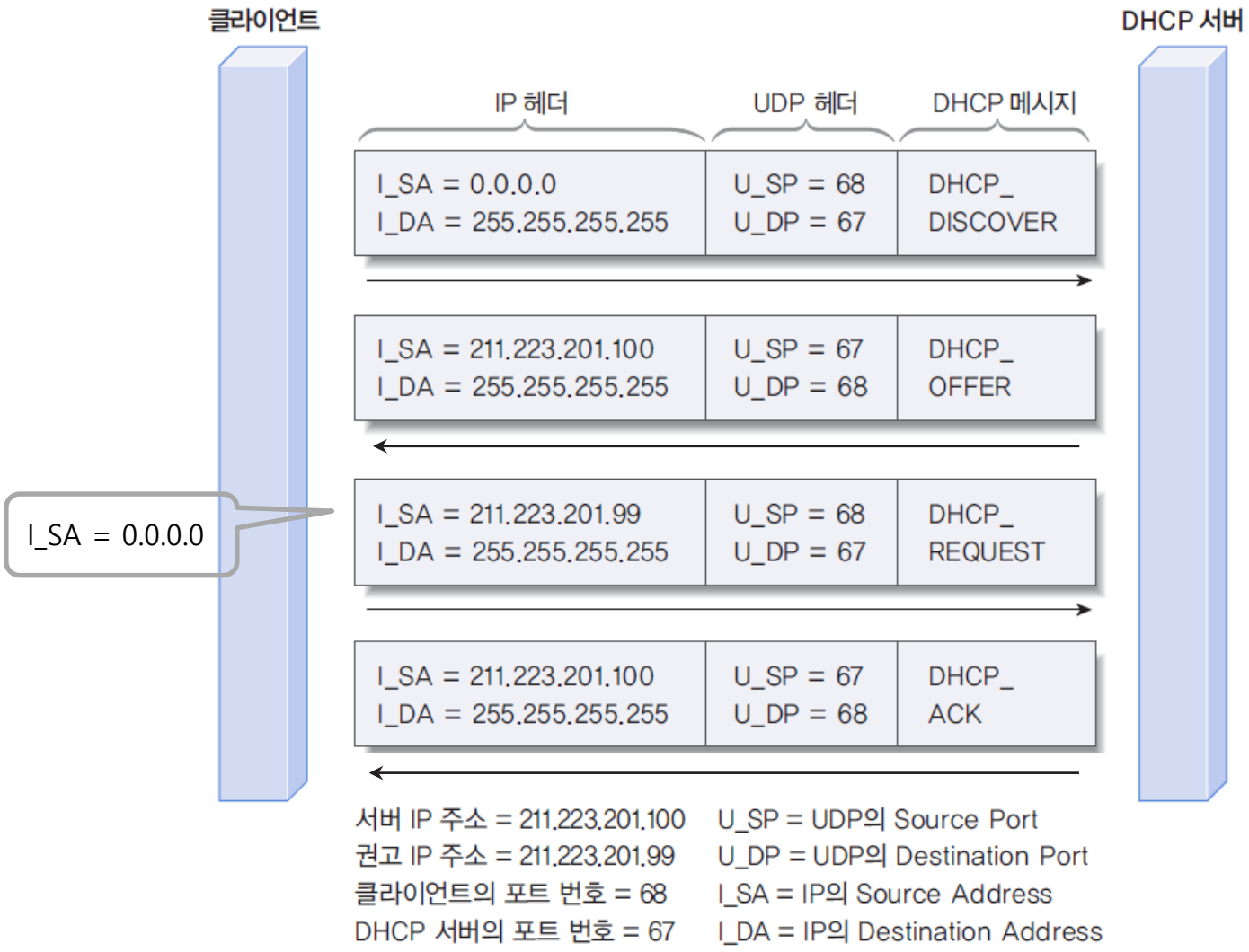


그림 7-19 UDP/IP 프로토콜의 캡슐화

Thank You