

TCP/IP 프로토콜

- 6장 IP 패킷의 전달과 포워딩, 7장 IPv4 -

서진우(jinwoo@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- IP 패킷의 전달과 포워딩
- IPv4

목 차

- IP 패킷의 전달과 포워딩
- IPv4

IP 주소의 전달과 포워딩

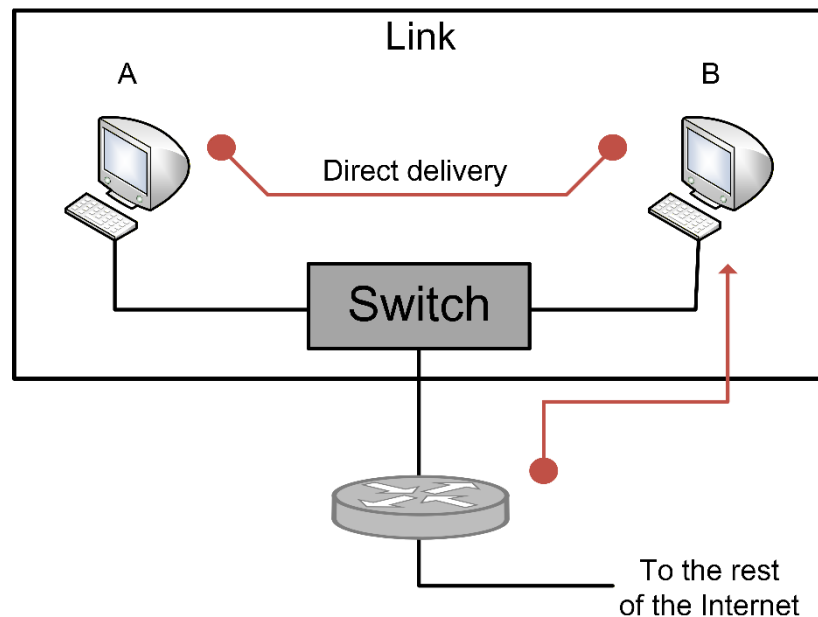
- IP 주소의 전달 방식

- 종류 (1/2)

- 직접 전달

- 패킷의 최종 목적지가 출발지와 같은 물리 네트워크에 연결되어 있을 때 사용하는 방식

- 마스크를 사용하여 목적지의 네트워크 주소를 추출하여 출발지와 비교



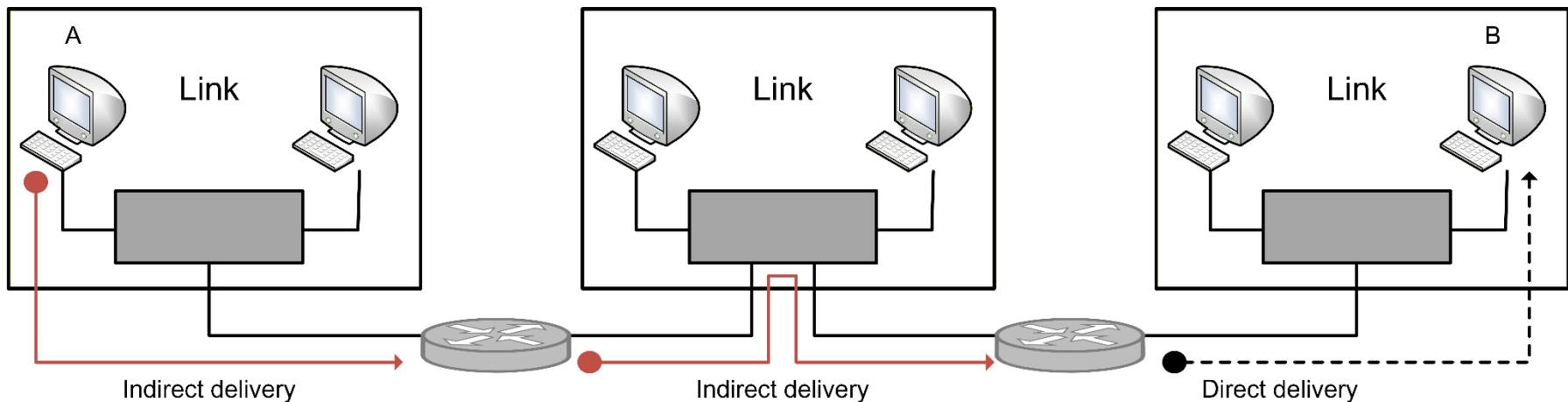
IP 주소의 전달과 포워딩

- IP 주소의 전달 방식

- 종류 (2/2)

- 간접 전달

- 목적지 호스트가 출발지와 같은 물리 네트워크에 있지 않을 때 사용
 - 패킷은 최종 목적지와 같은 물리적인 네트워크에 연결된 라우터에 도달할 때까지 여러 라우터를 경유하여 전달



IP 주소의 전달과 포워딩

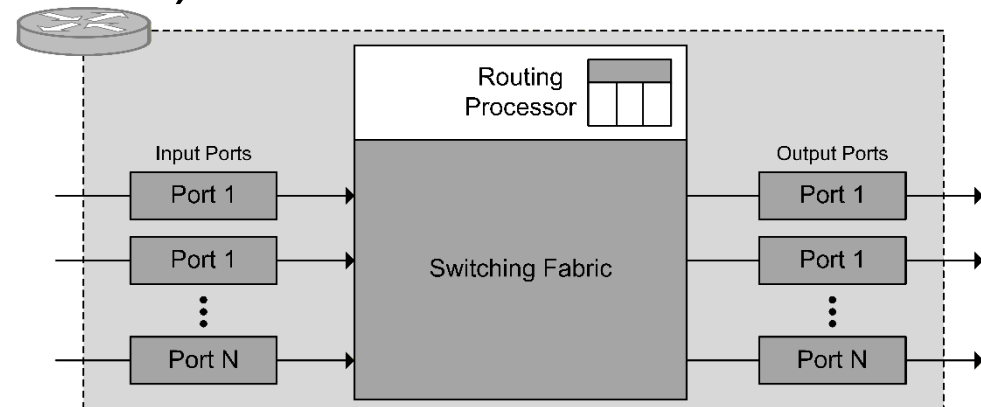
- 라우터

- 정의

- 둘 이상의 네트워크를 연결하고, 데이터 패킷이 목적지까지 최적의 경로를 지정해 주는 네트워크 장비

- 구성 요소

- 입력 포트(Input Port)
- 출력 포트(Output Port)
- 라우팅 처리기(Routing Processor)
- 교환 조직(Switching Fabric)



IP 주소의 전달과 포워딩

• 라우터의 구성 요소 (1/7)

*역캡슐화(Encapsulation)

TCP/IP 모델, OSI 모델에서 상위 계층으로 보낼 때 데이터에 헤더를 제거하여 전달하는 것.

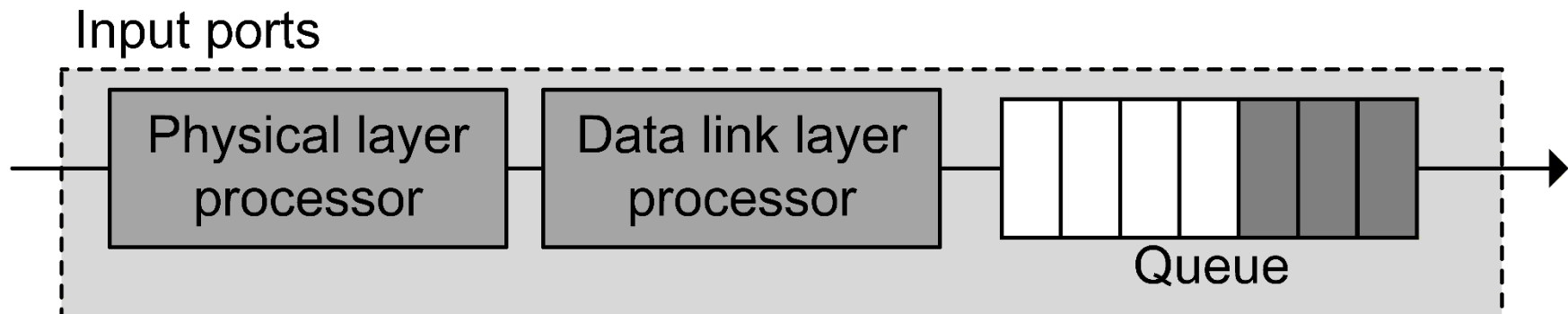
• 입력 포트(Input Port)

• 특징

- 수집된 신호로부터 비트들이 만들어지고 프레임으로부터 패킷이 역캡슐화
- 오류 탐지 및 수정

• 구성

- 물리 계층 프로세서, 데이터 링크 계층 프로세서, 큐



IP 주소의 전달과 포워딩

• 라우터의 구성 요소 (2/7)

• 출력 포트(Output Port)

• 특징

- 입력 포트와 같은 기능을 수행하나 순서는 반대
- 출력될 패킷이 큐에 들어오면 각 패킷을 프레임으로 캡슐화하여 전기 신호로 송신

• 구성

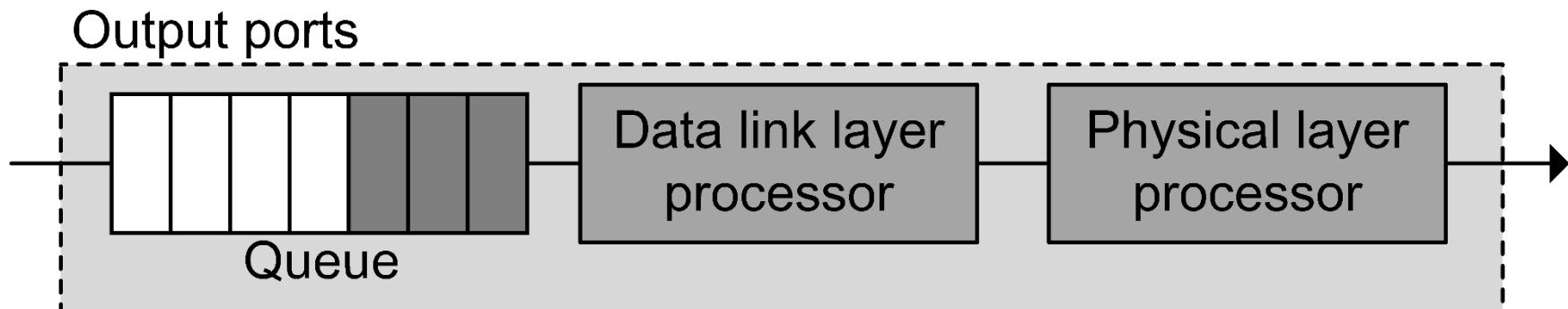
- 큐, 데이터 링크 계층 프로세서, 물리 계층 프로세서

*프레임(Frame)

데이터 링크 계층에서 데이터를 전송하기 위해 헤더와 트레일러를 붙인 데이터 단위.

*캡슐화(Capsulation)

TCP/IP 모델, OSI 모델에서 하위 계층으로 보낼 때 데이터에 헤더를 붙여 전달하는 것.



IP 주소의 전달과 포워딩

- 라우터의 구성 요소 (3/7)
 - 라우팅 처리기(Routing Processor)
 - 특징
 - 네트워크 계층의 기능을 수행하며 두뇌 담당
 - 패킷의 최적의 경로를 결정하고, 이를 기반으로 패킷을 적절한 인터페이스로 포워딩 담당
 - 패킷을 전송할 출력 포트번호와 다음 홉 주소를 찾기 위해 패킷의 주소를 참조

IP 주소의 전달과 포워딩

- 라우터의 구성 요소 (4/7)
 - 교환 조직(Switching Fabric)
 - 특징
 - 입력 큐에서 출력 큐로 패킷을 이동
 - 작업 수행 속도가 패킷 전달 전체 지연에 영향
 - 여러 유형에 따라 다른 방식으로 패킷을 전송
 - 종류
 - 크로스바 교환기(Cross-bar Switch)
 - 배년 교환기(Banyan Switch)
 - 배취-배년 교환기(Batcher-Banyan Switch)

IP 주소의 전달과 포워딩

- 라우터의 구성 요소 (5/7)

- 교환 조직(Switching Fabric)

- 종류 (1/3)

- 크로스바 교환기(Cross-bar Switch)

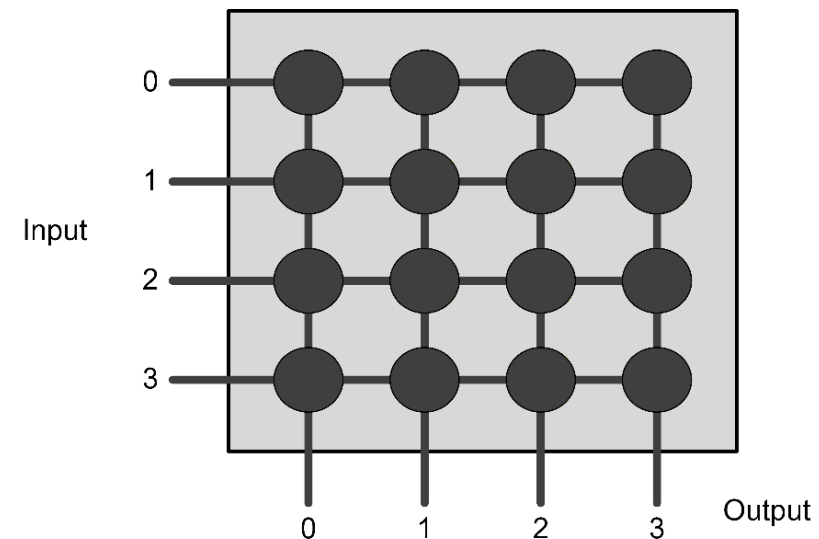
- 입력과 출력을 격자 형태의 교차점(Crosspoint)으로 연결하는 구조
 - N 개 입력, N 개 출력이면 $N \times N$ 개의 교차점이 필요
 - N 개 입력과 N 개 출력이 격자로 교차하는 지점마다 크로스포인트를 두고, 원하는 입력-출력 쌍의 교차점만 닫아 연결

- 장점

- 동시에 여러 입출력 쌍 연결 가능
 - 낮은 지연

- 단점

- 교차점 수가 N 제곱에 비례해 증가
 - 대규모 스위치 구현 시 비용과 복잡도 증가



IP 주소의 전달과 포워딩

- 라우터의 구성 요소 (6/7)

- 교환 조직(Switching Fabric)

- 종류 (2/3)

- 배년 교환기(Banyan Switch)

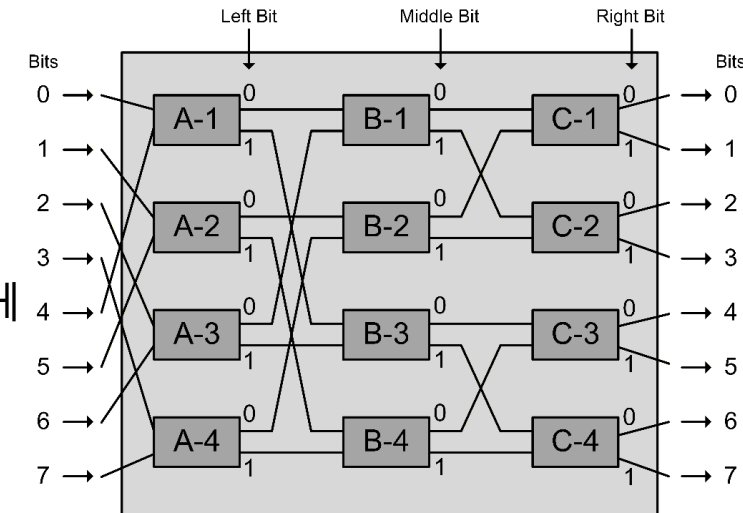
- 각 단계에 여러 개의 미세 교환기를 가지고 있는 다단계 교환기
 - 입력과 출력이 n 개인 경우, $\log_2 n$ 개의 단계와 $\frac{n}{2}$ 의 미세 교환기
 - 목적지 주소에 따라 자체적으로 경로 결정, 별도 제어 로직 불필요
 - e.g., $n = 8$ 인 목적지 주소 3비트 형식일 경우, 각 자리별 비트에 따라 결정

- 장점

- 교차점 수가 $N \log N$ 에 비례
 - 크로스바 교환기보다 하드웨어 낮은 복잡도

- 단점

- 서로 다른 입력의 패킷이 내부 링크를 동시에 요구 시 충돌 및 블로킹 발생 가능



IP 주소의 전달과 포워딩

• 라우터의 구성 요소 (7/7)

*논블로킹(Non-Blocking)

데이터가 아직 준비되지 않아도 요청 직후 제어권을 바로 반환하여 프로그램이 멈추지 않고 다른 일을 할 수 있게 하는 입출력 방식

• 교환 조직(Switching Fabric)

• 종류 (3/3)

• 배취-배년 교환기(Batcher-Banyan Switch)

• 배년 교환기의 문제점을 해결

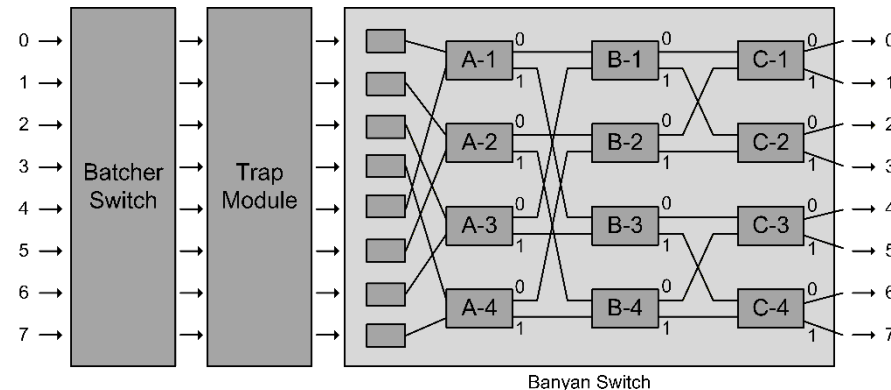
- 배취 스위치(Batcher Switch)를 통해 입력 값들을 목적지 주소 기준으로 정렬
- 트랩 모듈(Trap Module)을 통해 동일한 목적지를 가지고 있는 셀들에 대해 동시에 전송되지 않게 잡아두는 역할

• 장점

- 내부 충돌 문제 제거, 논블로킹(Non-Blocking) 동작 가능

• 단점

- 하드웨어 복잡도 추가



IP 주소의 전달과 포워딩

- 포워딩

- 정의

- 패킷을 목적지까지 전달하기 위해 다음 홉으로 전송하는 과정

- 종류

- 목적지 주소 기반 포워딩
 - IP가 비연결형 서비스로 사용될 때 수행
 - 레이블 기반 포워딩
 - IP가 연결형 서비스로 사용될 때 수행

IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩 (1/13)

- 정의

- 패킷의 목적지 주소를 기준으로 경로를 설정하는 포워딩

- 특징

- 목적지 주소를 기준으로 라우팅 테이블을 참조하여 다음 홉 결정
- 호스트나 라우터는 패킷 전송 전 라우팅 테이블을 참조하여 다음 홉 결정

- 문제점

- 목적지 정보 증가에 따른 라우팅 테이블 크기 증가
- 테이블 크기 증가로 인한 검색 효율 저하 등 문제 발생 가능

IP 주소의 전달과 포워딩

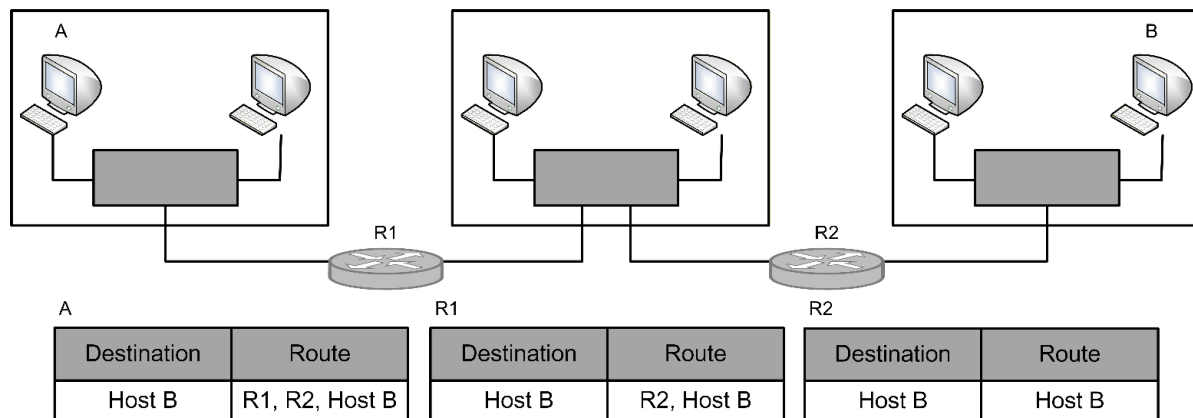
- 목적지 주소 기반 포워딩 (2/13)
 - 해결 방법
 - 라우팅 테이블 항목 구성 방식을 다르게 적용하여 테이블 크기 문제 해결
 - 네 가지 방법을 조합하여 테이블 크기 최소화
 - 방법
 - 다음 홉 방법
 - 네트워크 지정 방법
 - 호스트 지정 방법
 - 디폴트 방법

IP 주소의 전달과 포워딩

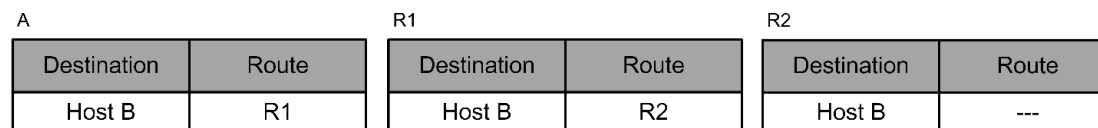
- 목적지 주소 기반 포워딩 (3/13)

- 다음 홉 방법

- 라우팅 테이블은 전체 경로에 대한 정보 대신 다음 홉 주소만 저장
 - 라우팅 테이블은 서로에 대한 일관성 유지 필요
 - e.g., (a)는 기존 방법, (b)는 다음 홉 방법



a. Routing tables based on route



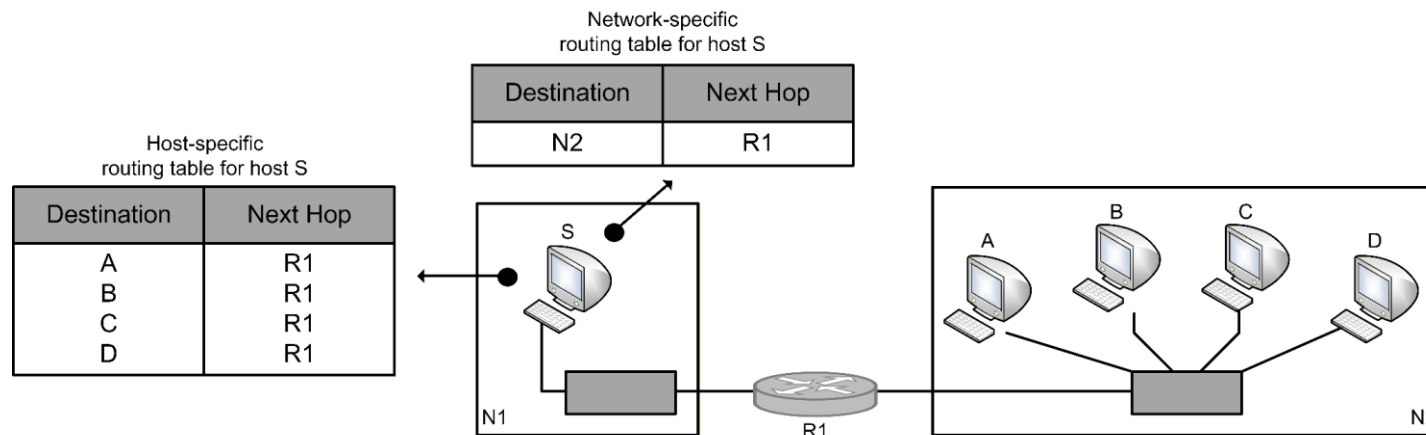
b. Routing tables based on next hop

IP 주소의 전달과 포워딩

• 목적지 주소 기반 포워딩 (4/13)

• 네트워크 지정 방법

- 같은 네트워크에 연결된 모든 호스트에 대해 각 호스트 별 엔트리를 갖는 대신 네트워크 자신의 주소를 정의하는 엔트리 하나만 소유
 - 같은 네트워크에 연결된 모든 호스트들은 하나의 엔티티로 취급
- 대부분의 목적지에 대해 기본적으로 사용
- 같은 네트워크에 속한 호스트들이 동일한 경로로 도달 가능할 때 사용

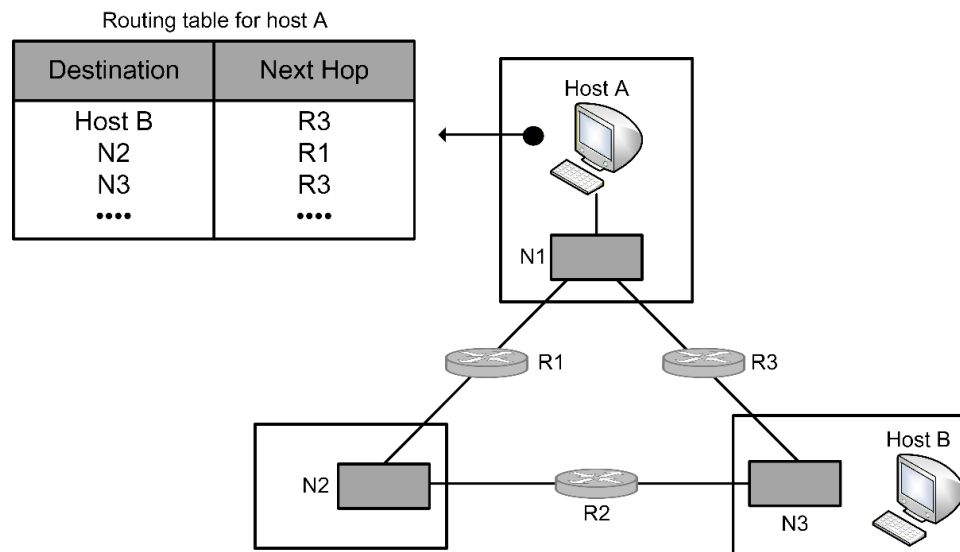


IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩 (5/13)

- 호스트 지정 방법

- 라우팅 테이블에 목적지 호스트 주소 저장
- 경로를 점검하거나 보안성을 제공하기 위한 경우 사용
- 관리자가 라우팅 테이블을 제어할 때 효율적
- 특정 호스트 하나에 대해서만 예외적으로 다른 경로를 적용해야 할 때 사용

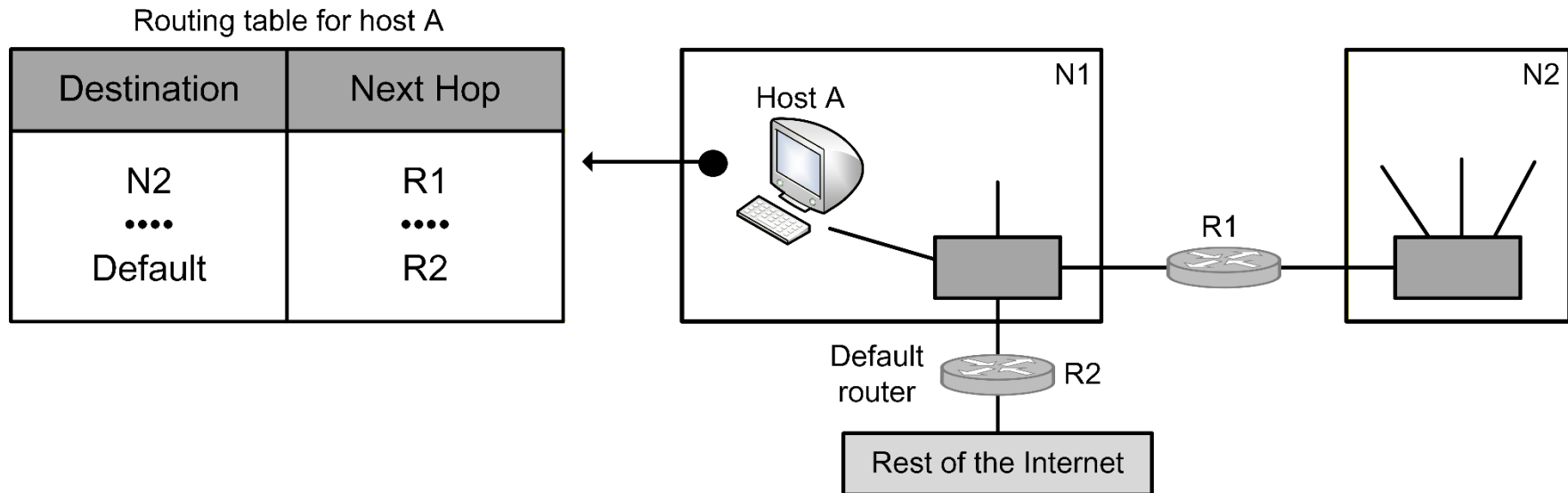


IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩 (6/13)

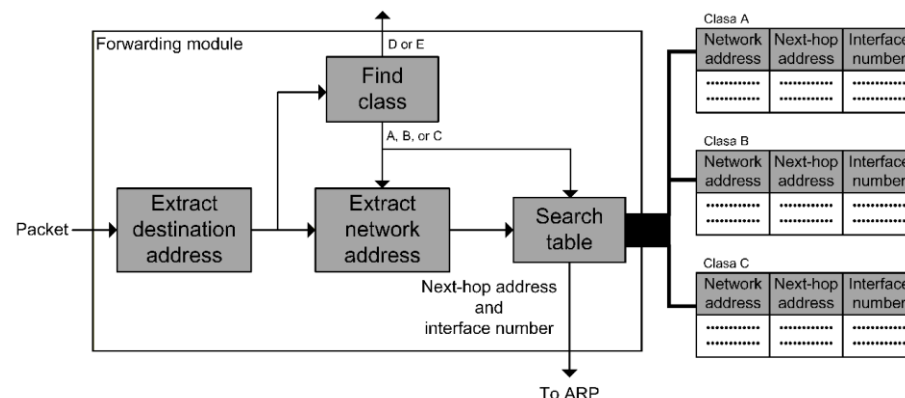
- 디폴트 방법

- 일부 목적지에 대한 엔트리를 따로 정의하고, 자주 보내지는 경로를 디폴트로 지정하는 방식
- 정의되지 않은 나머지 모든 목적지는 디폴트 라우터로 전송



IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩 (7/13)
 - 클래스기반 주소체계에서의 포워딩 (1/2)
 - 서브네팅(Subnetting)이 없는 경우의 포워딩
 - 동작 순서
 1. 패킷의 목적지 주소 추출
 2. 목적지 주소의 복사본을 오른쪽으로 28비트 이동하여 주소의 클래스 추출
 - 상위 4비트를 이용해 네트워크 ID와 호스트 ID 파악
 - 네트워크 주소 추출
 3. 클래스와 네트워크 주소를 사용하여 다음 홉 주소 파악
 4. 다음 홉 주소와 인터페이스번호를 ARP에게 전달
 - ARP는 브로드캐스트로 MAC 주소 파악

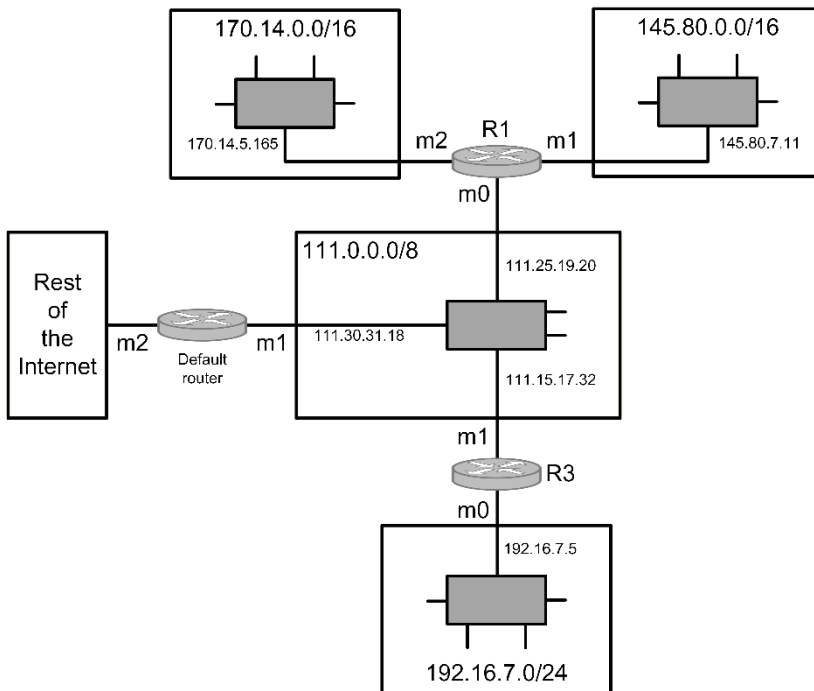


IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩

- 예제 6.1

그림 6.8은 가상의 네트워크를 보여준다. 라우터 R1의 라우팅 테이블을 보여라.



(그림 6.8)

정답)

Class A

Network address	Next-hop address	Interface
111.0.0.0	-----	m0

Class B

Class B

Network address	Next-hop address	Interface
145.80.0.0	-----	m1
170.14.0.0	-----	m2

Network address	Next-hop address	Interface
192.16.7.0	111.15.17.32	m0

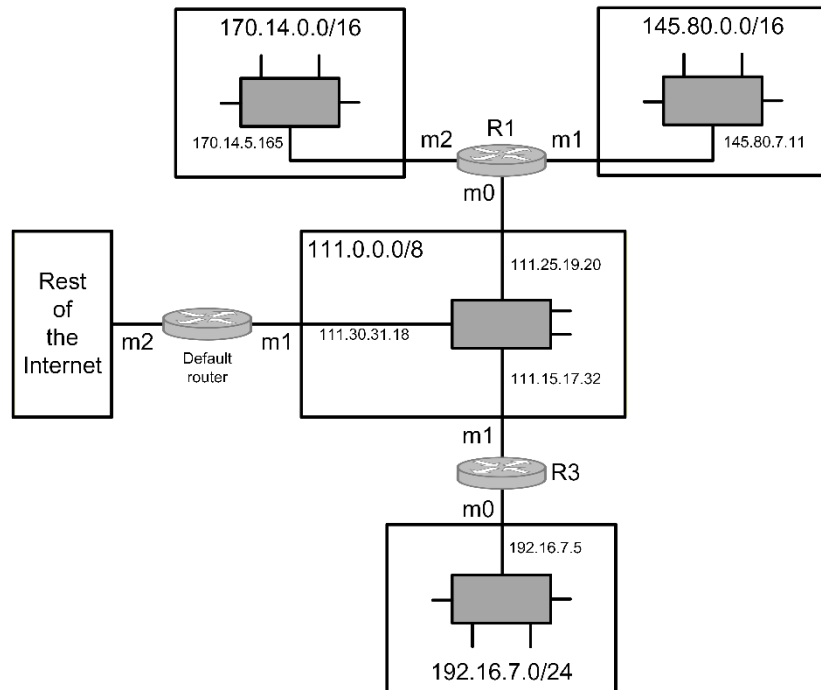
Default: 111.30.31.18, m0

IP 주소의 전달과 포워딩

• 목적지 주소 기반 포워딩

• 예제 6.2

그림 6.8에서 라우터 R1은 목적지 주소가 192.16.7.14인 패킷을 받았다. 패킷은 어떻게 포워딩 되는지 보여라.



(그림 6.8)

- 192.16.7.14 주소를 2진법으로 변환
 - 11000000 00010000 00000111 00001110₍₂₎
- 주소의 복사본이 오른쪽으로 28비트 쉬프트
 - 00000000 00000000 00000000 00001100₍₂₎
 - 12₍₁₀₎
- 목적지 네트워크는 클래스 C
- 목적지 주소에서 왼쪽 24비트만 보면 192.16.7.0으로 클래스 C의 테이블이 탐색되고 첫 번째 줄에서 부합되는 엔트리가 발견

Class A

Network address	Next-hop address	Interface
111.0.0.0	-----	m0

Class C

Network address	Next-hop address	Interface
192.16.7.0	111.15.17.32	m0

Class B

Network address	Next-hop address	Interface
145.80.0.0	-----	m1
170.14.0.0	-----	m2

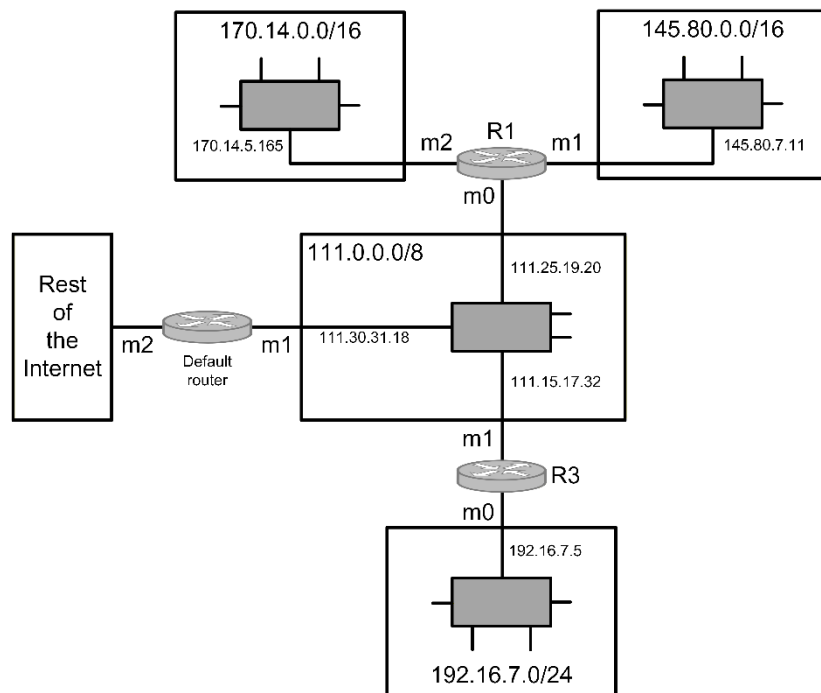
Default: 111.30.31.18, m0

IP 주소의 전달과 포워딩

• 목적지 주소 기반 포워딩

• 예제 6.3

그림 6.8에서 라우터 R1은 목적지 주소가 167.24.160.5인 패킷을 받았다. 패킷은 어떻게 포워딩 되는지 보여라.



(그림 6.8)

• 풀이

- 167.24.160.5 주소를 2진법으로 변환
 - 10100111 00011000 10100000 00000101
- 주소의 복사본이 오른쪽으로 28비트 쉬프트
 - 00000000 00000000 00000000 00001010₍₂₎
 - 10₍₁₀₎
- 목적지 네트워크는 클래스 B
- 목적지 주소에서 왼쪽 16비트만 보면 167.24.0.0으로 클래스 B의 테이블이 탐색되지만 부합되는 엔트리 발견 실패
- 목적지 네트워크는 인터넷상에 위치하므로 패킷은 디폴트 라우터에게 전달
- 다음 홉 주소 111.30.31.18과 인터페이스 번호 m0가 ARP에게 전달 되어, MAC 주소를 알아낸 후 프레임을 전송

Class A		
Network address	Next-hop address	Interface
111.0.0.0	-----	m0

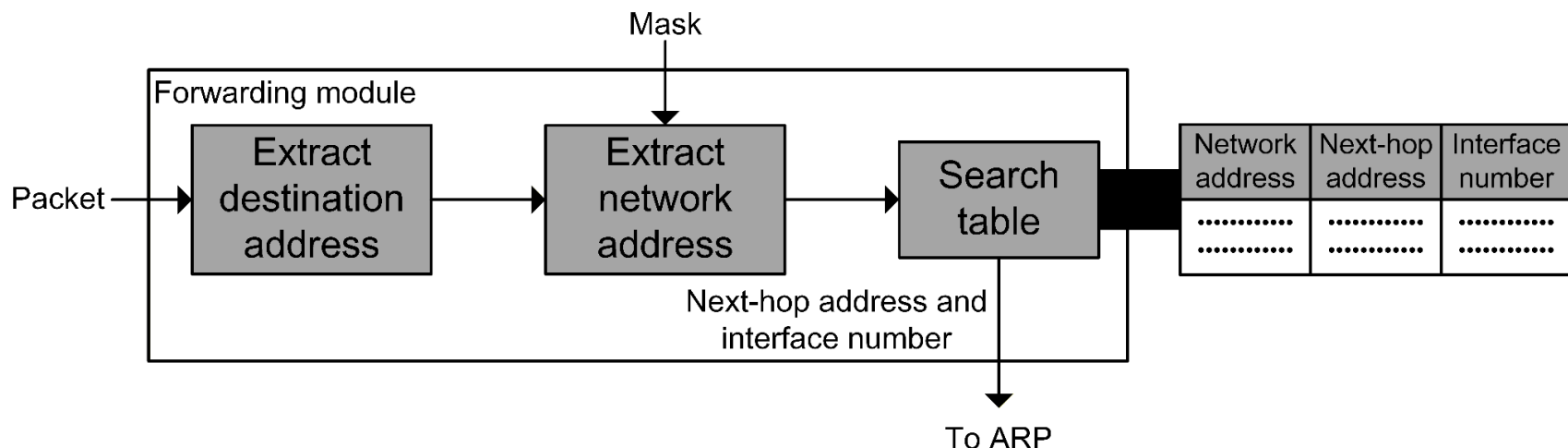
Class C		
Network address	Next-hop address	Interface
192.16.7.0	111.15.17.32	m0

Class B		
Network address	Next-hop address	Interface
145.80.0.0	-----	m1
170.14.0.0	-----	m2

Default: 111.30.31.18, m0

IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩 (8/13)
- 클래스기반 주소체계에서의 포워딩 (2/2)
 - 서브네팅(Subnetting)이 있는 경우의 포워딩
 - 동작 순서
 1. 목적지 주소를 추출
 2. 목적지 주소와 서브넷 마스크를 사용하여 서브넷 주소를 추출
 3. 테이블에서 서브넷 주소를 탐색하여 다음 홉 주소와 인터페이스 번호 추출
 4. 다음 홉 주소와 인터페이스 번호를 ARP에게 전달
 5. ARP는 브로드캐스트로 MAC 주소 파악

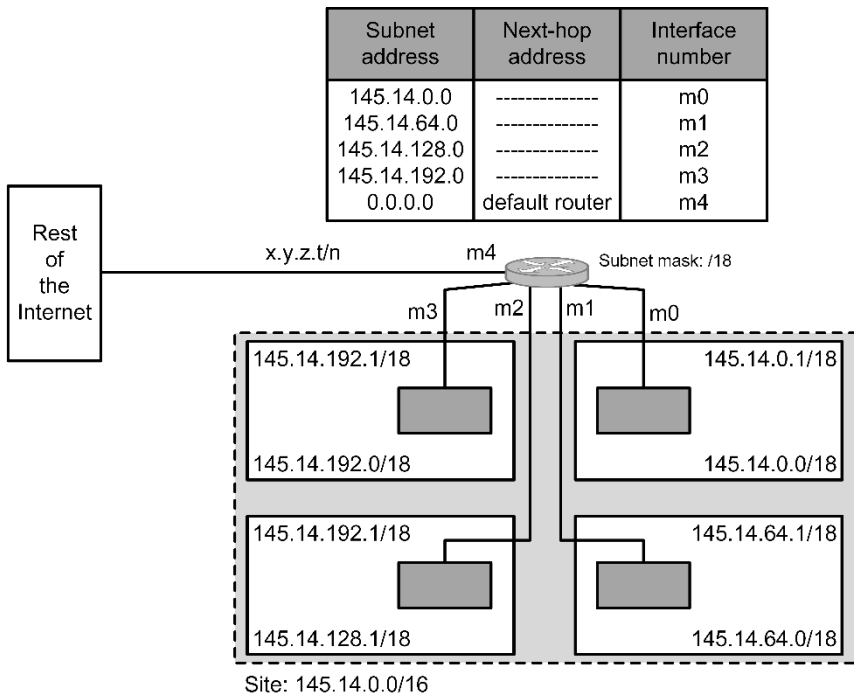


IP 주소의 전달과 포워딩

• 목적지 주소 기반 포워딩

• 예제 6.4

그림 6.11의 라우터가 목적지 주소가 145.14.32.78인 패킷을 받았다. 패킷은 어떻게 포워딩 되는지 보여라.



(그림 6.11)

• 풀이

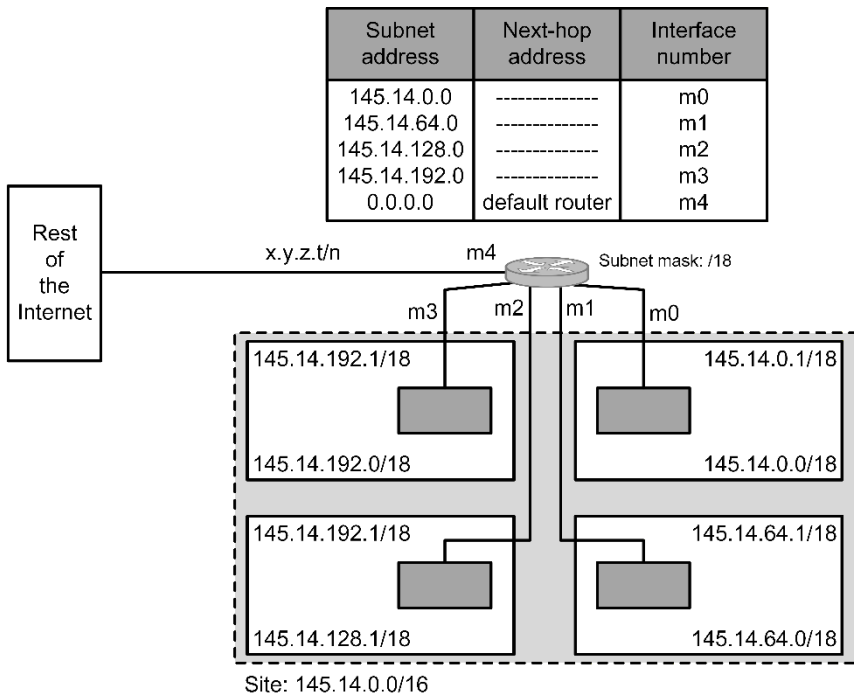
- 마스크는 /18
- 마스크를 적용한 서브넷 주소는 145.14.0.0
- 테이블 탐색 결과
 - 인터페이스 m0
 - 다음 홉 주소는 목적지 주소(145.14.32.78)
- 다음 홉 주소 145.14.32.78과 인터페이스 m0가 ARP에게 전달
- ARP는 브로드캐스트로 MAC 주소 파악

IP 주소의 전달과 포워딩

• 목적지 주소 기반 포워딩

• 예제 6.5

그림 6.11에서 145.14.0.0 네트워크 내의 호스트가 주소 7.22.67.91인 호스트에 보낼 패킷을 가지고 있다. 패킷은 어떻게 전달되는지 보여라.



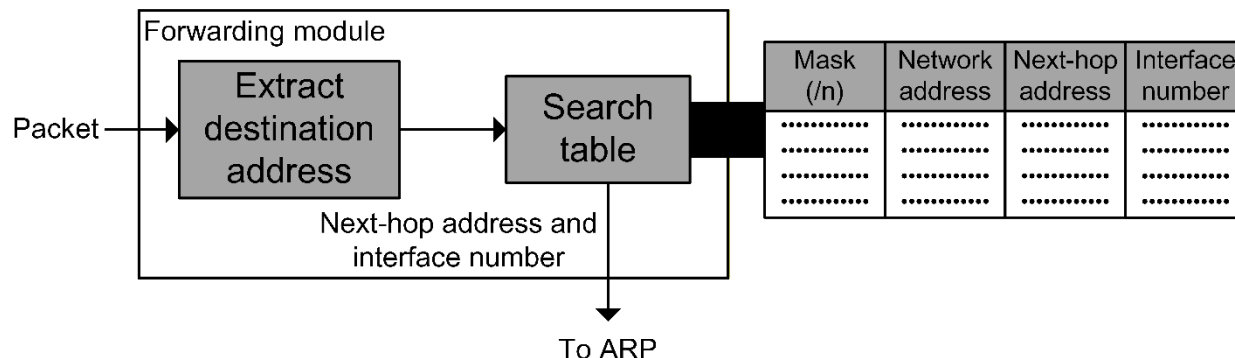
(그림 6.11)

• 풀이

- 라우터는 패킷을 받아 마스크 /18을 적용
- 네트워크 주소 7.22.64.0
- 테이블을 탐색하지만 실패
- 라우터는 디폴트 라우터의 주소를 사용하여 패킷을 라우터에 전송
 - ARP는 같은 로컬 네트워크(같은 서브넷) 안에 있는 호스트의 MAC 주소를 알아낼 때 사용

IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩 (9/13)
 - 클래스없는 주소체계에서의 포워딩 (1/4)
 - 정의
 - 가변 길이의 서브넷 마스크를 사용하여 네트워크를 구분하여 포워딩하는 방식
 - 문제점
 - 클래스 기반 주소 체계와 달리, 목적지 주소만으로는 네트워크 주소를 파악하기 어려움
 - 해결책
 - 주소 표현 시 마스크(/n)를 포함하는 열 추가

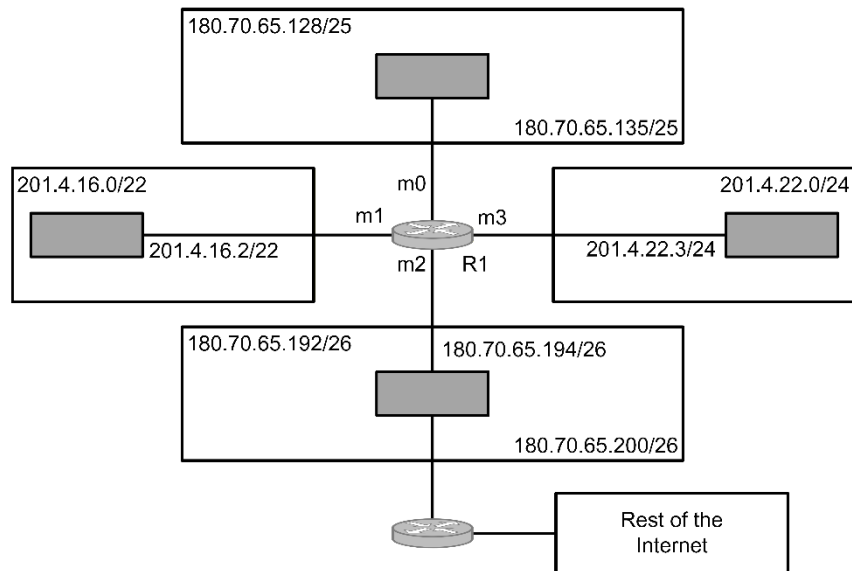


IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩

- 예제 6.7

그림 6.13의 구성을 사용하여 라우터 R1의 라우팅 테이블을 만들라



(그림 6.13)

- 풀이

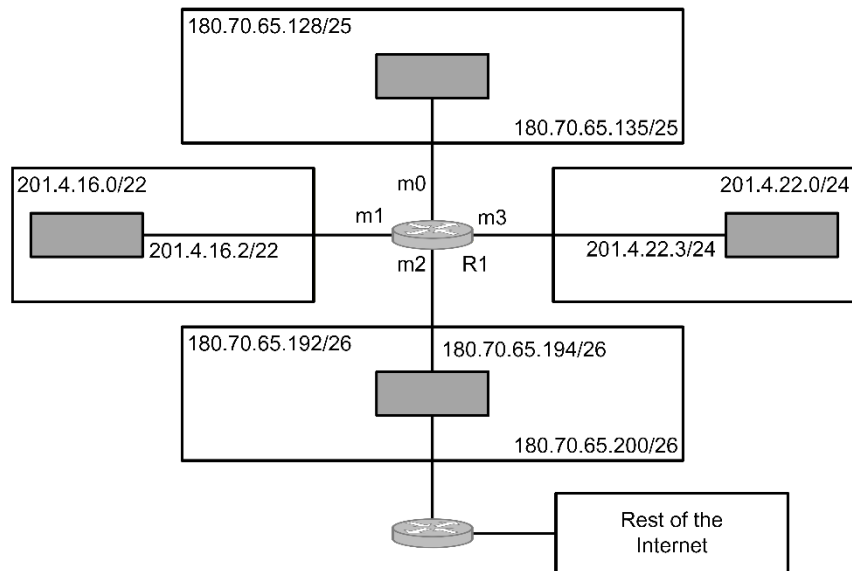
Mask	Network address	Next Hop	Interface
/26	180.70.65.192	-	m2
/25	180.70.65.128	-	m0
/24	201.4.22.0	-	m3
/22	201.4.16.0		m1
Default	Default	180.70.65.200	m2

IP 주소의 전달과 포워딩

• 목적지 주소 기반 포워딩

• 예제 6.8

그림 6.13에서 R1에 목적지 주소가 180.70.65.140인 패킷이 도착한 후의 포워딩 과정을 설명하라



(그림 6.13)

• 풀이

- 목적지 주소에 첫 번째 마스크 /26 적용
 - 결과는 180.70.65.192
 - 해당하는 네트워크 주소와 모순
- 목적지 주소에 두 번째 마스크 /25 적용
 - 결과는 180.70.65.128
 - 해당하는 네트워크 주소와 부합
 - 다음 홉 주소와 인터페이스 번호 m0이 ARP에 전달

R1 라우팅 테이블

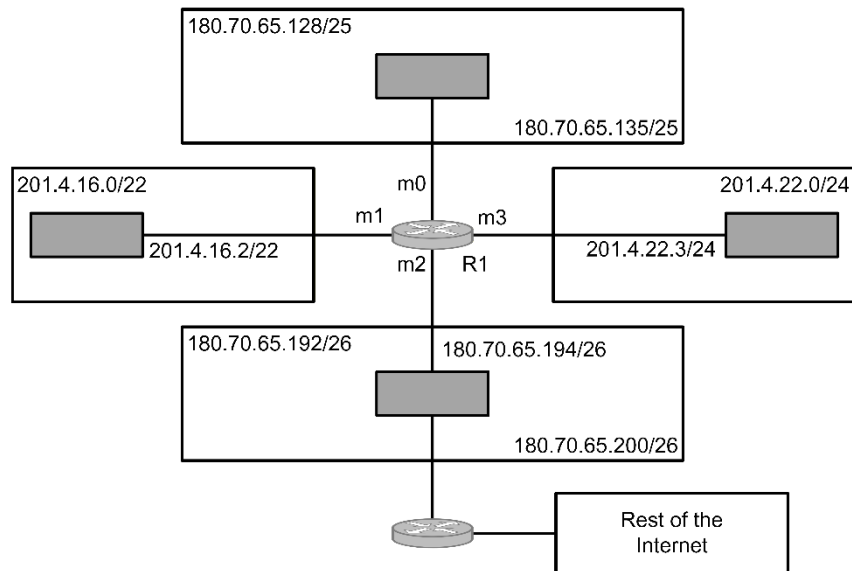
Mask	Network address	Next Hop	Interface
/26	180.70.65.192	-	m2
/25	180.70.65.128	-	m0

IP 주소의 전달과 포워딩

• 목적지 주소 기반 포워딩

• 예제 6.9

그림 6.13에서 R1에 목적지 주소가 201.4.22.35인 패킷이 도착하면 이 패킷이 어떻게 처리되는지 보여라



(그림 6.13)

• 풀이

- 첫 번째 마스크부터 차례대로 적용
- 세 번째 마스크인 /24 적용
 - 201.4.22.0으로 R1 라우팅 테이블의 주소와 동일

R1 라우팅 테이블

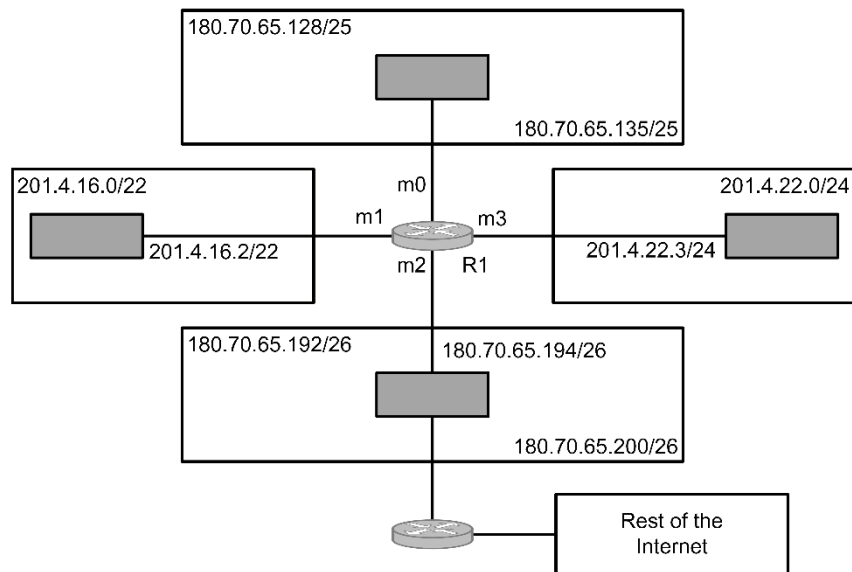
Mask	Network address	Next Hop	Interface
/26	180.70.65.192	-	m2
/25	180.70.65.128	-	m0
/24	201.4.22.0	-	m3

IP 주소의 전달과 포워딩

• 목적지 주소 기반 포워딩

• 예제 6.10

그림 6.13에서 R1에 목적지 주소가 18.24.32.78인 패킷이 도착하면 이 패킷이 어떻게 처리되는지 보여라



(그림 6.13)

• 풀이

- 목적지 주소에 모든 마스크가 적용되지만 라우팅 테이블의 주소와는 부합하지 않음
- 테이블 끝에 도달 시
 - 다음 홉 주소 180.70.65.200과 인터페이스 번호 m2를 ARP에 전달

R1 라우팅 테이블

Mask	Network address	Next Hop	Interface
/26	180.70.65.192	-	m2
/25	180.70.65.128	-	m0
/24	201.4.22.0	-	m3
/22	201.4.16.0		m1
Default	Default	180.70.65.200	m2

IP 주소의 전달과 포워딩

• 목적지 주소 기반 포워딩

• 예제 6.11

라우팅 테이블의 내용만을 알면 라우터의 구성을 알 수 있을 건인가? 라우터 R1의 라우팅 테이블이 표 6.2에 주어져 있다. 토폴로지를 그릴 수 있는가?

Mask	Network address	Next Hop	Interface
/26	140.6.12.64	180.14.2.5	m2
/24	130.4.8.0	190.17.6.2	m1
/16	110.70.0.0	-----	m0
/16	180.14.0.0	-----	m2
/16	190.17.0.0	-----	m1
Default	Default	110.70.4.6	m0

(표 6.2)

- 풀이
 - 라우팅 테이블 만으로는 토폴로지에 대한 일부 정보만 알 수 있으며 모든 정보는 불가능
 - R1 인터페이스: m0, m1, m2
 - 세 개의 네트워크에 직접 연결

IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩(10/13)

- 클래스없는 주소체계에서의 포워딩 (2/4)

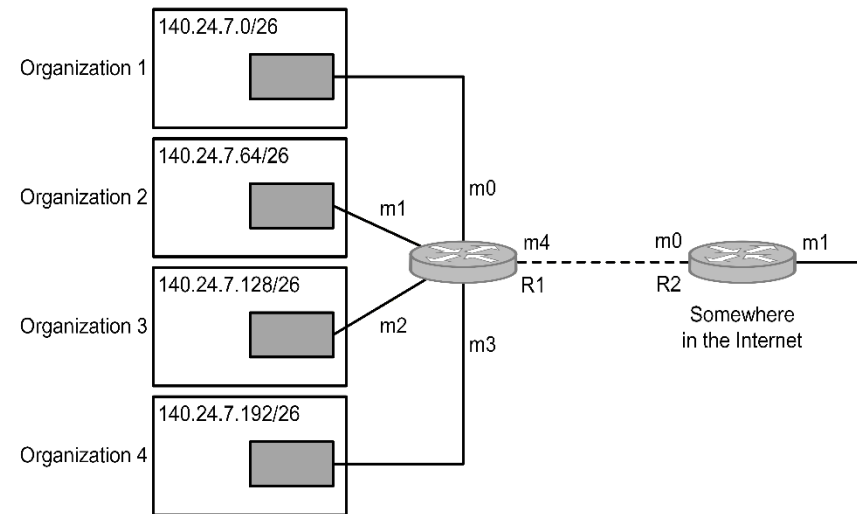
- 주소 집단화(Address Aggregation)

- 정의

- 여러 개의 작은 네트워크 주소를 하나의 큰 엔트리로 묶어 라우팅 테이블 크기를 줄이는 기법

- 특징

- 라우팅 테이블의 크기 증가 및 탐색 시간 증가 문제를 해결하기 위해 사용



Mask (/n)	Network address	Next-hop address	Interface number
/26	140.24.7.0	-----	m0
/26	140.24.7.64	-----	m1
/26	140.24.7.128	-----	m2
/26	140.24.7.192	-----	m3
/0	0.0.0.0	default router	m4

Routing table for R1

Mask (/n)	Network address	Next-hop address	Interface number
/24	140.24.7.0	-----	m0
/0	0.0.0.0	default router	m1

Routing table for R2

IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩 (11/13)

- 클래스없는 주소체계에서의 포워딩 (3/4)

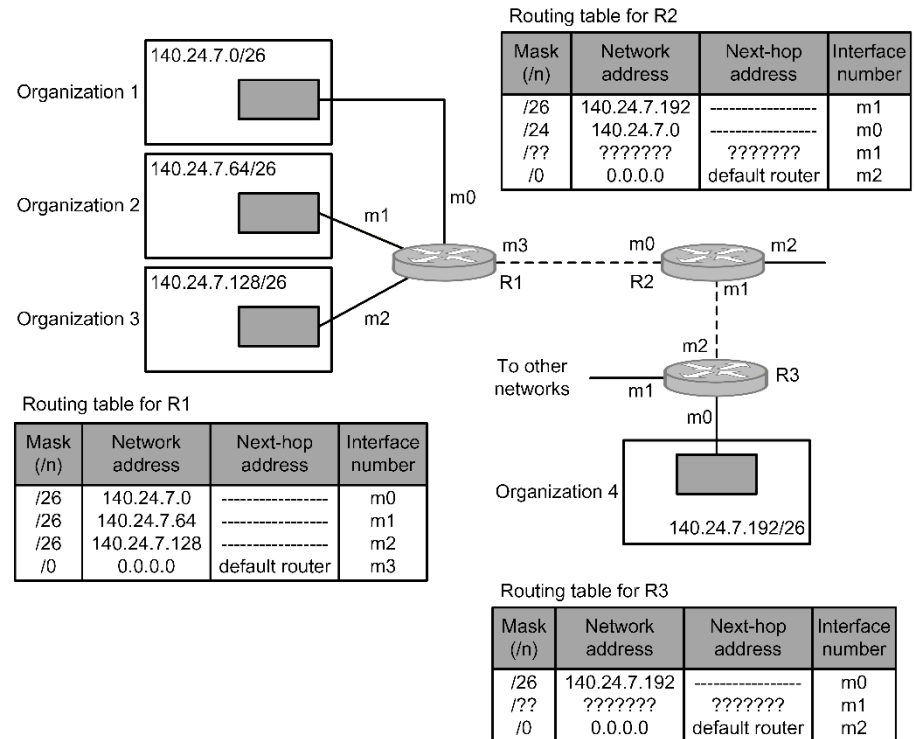
- 가장 긴 마스크 부합(Longest Mask Switching)

- 정의

- 여러 마스크가 동시에 일치할 경우, 가장 긴 마스크를 우선 적용하는 라우팅 규칙

- 특징

- 라우팅 테이블이 가장 긴 마스크에서 가장 짧은 마스크 순으로 정렬
 - e.g., /27, /26, /24 순



IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩 (12/13)

- 클래스없는 주소체계에서의 포워딩 (4/4)

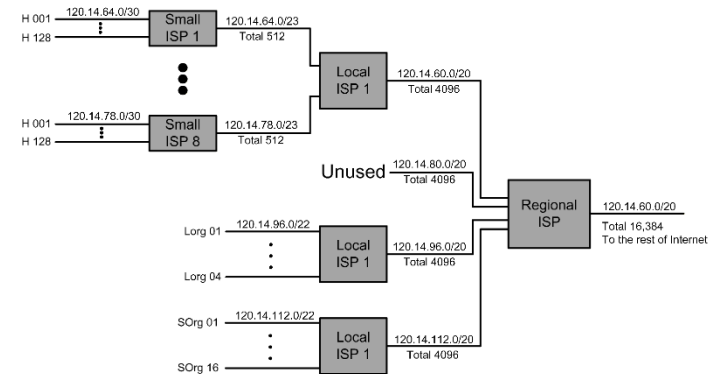
- 계층적 라우팅(Hierarchical Routing)

- 정의

- 백본, 지역 ISP, 로컬 ISP처럼 계층 구조로 네트워크를 나누어 라우팅하는 방식

- 특징

- 인터넷 구조와 같이 계층 구조의 개념을 반영
 - 상위 계층으로 갈수록 더 큰 주소 블록을 하나의 엔트리로 요약 가능



- 지리적 라우팅(Geographical Routing)

- 정의

- 계층적 라우팅의 개념을 지리적 단위(대륙/국가 등)로 확장한 라우팅 방식

- 특징

- 전체 주소 공간을 대륙 단위의 큰 블록으로 분할
 - 외부 ISP는 각 대륙에 대해 단 하나의 엔트리만으로 라우팅 가능

IP 주소의 전달과 포워딩

- 목적지 주소 기반 포워딩 (13/13)
 - 라우팅 테이블 탐색 알고리즘
 - 클래스기반 주소체계에서의 탐색
 - 라우팅 테이블은 리스트 구조
 - 동작 원리
 1. 라우팅 테이블은 각 클래스 당 한 개씩 전부 세 개의 테이블로 분할
 2. 패킷이 도착하면 라우터는 네트워크 마스크를 적용하여 해당하는 테이블 탐색
 - 클래스없는 주소체계에서의 탐색
 - 가장 긴 부합 방법(longest Mask Match)
 - 동작 원리
 1. 라우팅 테이블은 각 마스크 별로 한 개씩 정의된 여러 개의 테이블로 분할
 2. 라우터는 먼저 가장 긴 마스크 시도
 3. 만약 목적지 주소가 이 테이블 내에 발견되면, 탐색 종료
 4. 만약 발견되지 않는다면, 다음 마스크 탐색

IP 주소의 전달과 포워딩

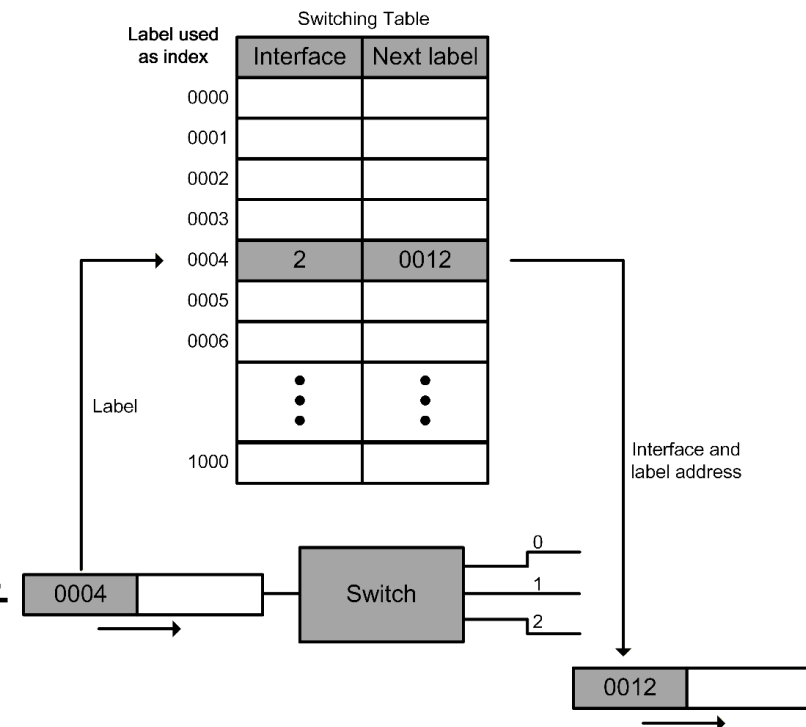
- 레이블 기반 포워딩 (1/3)

- 정의

- 연결형 네트워크에서 패킷에 부착된 레이블(Label)을 기반으로 목적지를 결정하는 포워딩

- 특징

- 스위칭 테이블(Switching Table) 기반 포워딩
 - 기존 라우팅 테이블 대신 레이블을 사용하는 스위칭 테이블 사용
- 인덱스 기반의 접근 방식
 - 패킷에 부착된 고정 길이의 레이블 번호 자체를 테이블의 배열 인덱스로 활용



IP 주소의 전달과 포워딩

• 레이블 기반 포워딩 (2/3)

*루핑(Looping)

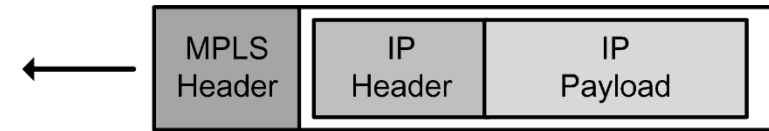
패킷이 네트워크 상에서 목적지에 도달하지 못하고 끝없이 순환하는 현상.

• MPLS (Multi-Protocol Label Switching) (1/2)

• 정의

- IP 헤더와 IP 페이로드로 구성된 기존 패킷에 MPLS 헤더를 추가하는 포워딩 방식

• 헤더구조



• Label

- 20비트로, 라우터 내 라우팅 테이블의 인덱스로 사용

- 16~1,048,561 사용 가능

- 탐색 없이 직접 접근 가능

• Exp (Experimental field)

- 3비트로, 패킷의 우선순위 지정

• S (Bottom-of-Stack)

- 1비트로, 레이블을 중첩하여 사용할 경우 사용되는 필드

- 0과 1 중 하나로, 값이 1이면 마지막 레이블임을 표시

• TTL (Time-To-Live)

- 8비트로, 루핑 방지용도 사용되는 값으로 방문할 수 있는 최대 홉 수를 의미

0	20	24	30
Label			
Exp		S	TTL
Label			
Exp		S	TTL
...			
Label			
Exp		S	TTL

IP 주소의 전달과 포워딩

- 레이블 기반 포워딩 (3/3)

*트래픽 엔지니어링(Traffic Engineering)
트래픽을 효율적으로 분산하기 위해 네트워크 경로를
최적화하는 기술.

- MPLS (Multi-Protocol Label Switching) (2/2)

- 특징

- 라우팅 테이블 크기 감소

- 레이블(Label)을 인덱스로 직접 접근하므로 IP 목적지 탐색 과정 생략
 - 다수 목적지를 하나의 레이블 경로로 묶어 관리 가능

- 효율성 증가

- IP 헤더 분석 및 경로 재탐색 없이, 미리 설정된 레이블 경로로 즉시 전달
 - 트래픽 상황에 따라 경로를 유연하게 재설정 가능
 - 혼잡 또는 장애 발생, 트래픽 엔지니어링(Traffic Engineering) 정책 변경 시
 - 새로운 레이블 경로를 생성하거나 백업 경로로 전환하여 우회 전달

- 다양한 네트워크 환경에서 사용 가능

- Exp 필드로 트래픽별 QoS를 차등 적용하여 지연에 민감한 트래픽과 일반 트래픽을 하나의 망에서 함께 서비스
 - S 필드로 레이블을 여러 겹 쌓아 여러 목적을 동시에 처리 가능
 - e.g., 고객 구분과 경로 제어를 각각의 레이블로 동시 수행

목 차

- IP 패킷의 전달과 포워딩
- IPv4

IPv4

• 인터넷 프로토콜(IP, Internet Protocol)

• 정의

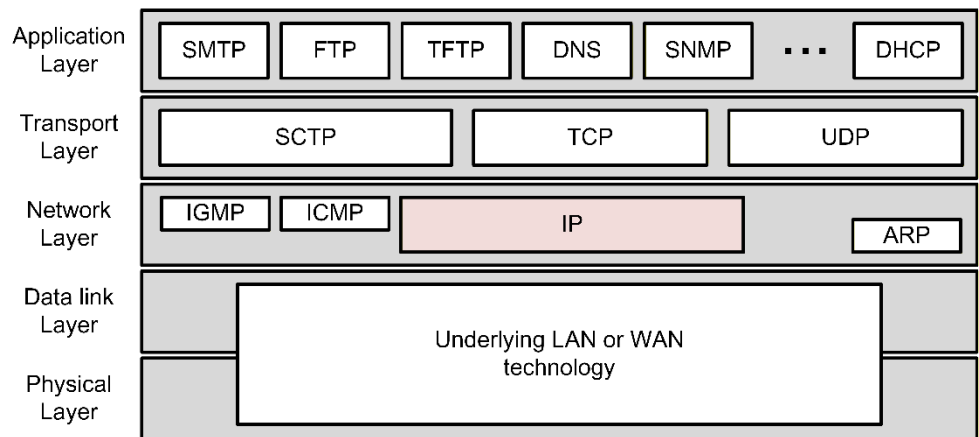
- 네트워크 계층에서 TCP/IP 프로토콜이 사용하는 전송 메커니즘

• 특징

- 비연결형 데이터그램 프로토콜
 - 패킷마다 독립적으로 처리되며 사전 경로 설정 없음
 - 최선의 노력(Best Effort) 전달만 수행하며 신뢰성 보장하지 않음
- 네트워크 상황에 따라 오류, 분실, 순서 역전, 지연 등 발생 가능

*SMTP: Simple Mail Transfer Protocol
*FTP: File Transfer Protocol
*TFTP: Trivial File Transfer Protocol
*DNS: Domain Name System
*SNMP: Simple Network Management Protocol
*DHCP: Dynamic Host Configuration Protocol

*SCTP: Stream Control Transmission Protocol
*TCP: Transmission Control Protocol
*UDP: User Datagram Protocol
*IGMP: Internet Group Management Protocol
*ICMP: Internet Control Message Protocol
*ARP: Address Resolution Protocol



IPv4

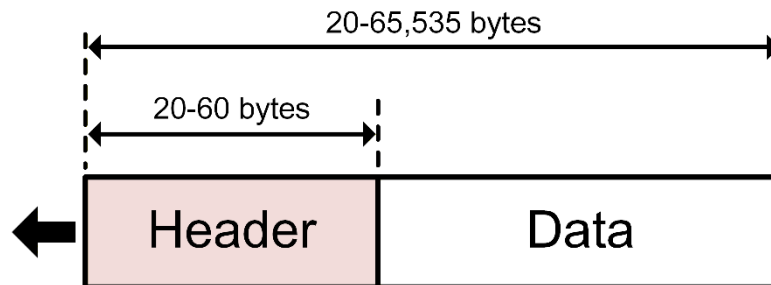
- 데이터그램(Datagram)

- 정의

- 패킷 교환 네트워크에서 독립적으로 전송되는 데이터의 기본 단위

- 특징

- 가변 길이이며, 헤더와 데이터로 구분
 - 헤더는 기본 20바이트이며, 옵션 포함 시 최대 60바이트로 구성
 - 라우팅과 전달에 필요한 정보 포함
 - e.g., 경로 기록, 타임스탬프, 소스 라우팅, 라우터 경고
 - 데이터는 헤더에 따라 0~65,515바이트로 구성
 - 상위 계층에서 내려온 데이터



IPv4

- 데이터그램(Datagram)

- 헤더 구성요소 (1/14)

- 버전(VER, VERsion)

- IP 프로토콜 버전 의미

- e.g., 0100으로 IPv4를 나타냄

- 헤더 길이(HLEN, Header LENgth)

- 데이터그램 헤더의 전체 길이를 4바이트 단위로 나타냄

- e.g., 헤더에 옵션이 없다면 20바이트이므로 필드의 값은 5($5 \times 4 = 20$)

0	3	4	7	8	15	16	31
VER 4 bits		HLEN 4 bits		Service type 8 bits		Total length 16 bits	
Identification 16 bits					Flags 3 bits	Fragmentation offset 13 bits	
Time to live 8 bits		Protocol 8 bits		Header checksum 16 bits			
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

IPv4

- 데이터그램(Datagram)

- 헤더 구성요소 (2/14)

- 서비스 유형(Service Type) (1/2)

- IP 헤더의 초기 설계

- TOS (Type Of Service)이라 불림

- 데이터그램이 어떻게 처리되어야 하는가를 의미

- 3비트

- 0~7단계로 패킷의 중요도를 결정하며 값이 클수록 높은 우선순위

- 4비트

- TOS 조건으로 서비스의 품질 요구사항 지정

- 1비트

- 사용하지 않고 0으로 고정

0	3	4	7	8	15	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits			Total length 16 bits		
Identification 16 bits					Flags 3 bits	Fragmentation offset 13 bits	
Time to live 8 bits		Protocol 8 bits			Header checksum 16 bits		
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

IPv4

• 데이터그램(Datagram)

*IETF (Internet Engineering Task Force)
국제 인터넷 표준화 기구로, 인터넷의 기술 표준과 운영 규칙을
개발하는 국제적인 개방형 표준화 기구.

• 헤더 구성요소 (3/14)

• 서비스 유형(Service Type) (2/2)

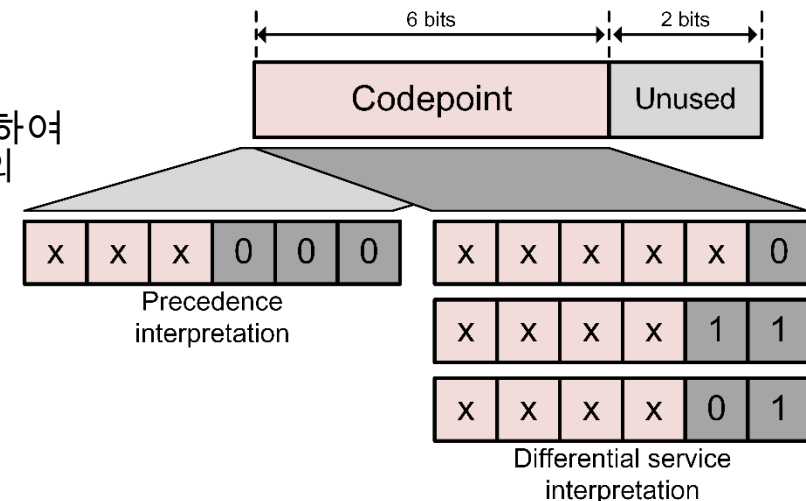
- IETF(Internet Engineering Task Force)의 의미 수정
 - 현재 차별화 서비스(Differentiated Service)의 집합이라 불림
 - 6비트는 코드포인트(Codepoint) 필드이며, 마지막 2비트는 사용하지 않음

• 코드포인트(Codepoint)에 따른 분류

- 오른쪽 세 비트가 0일 경우
 - 왼쪽 세 비트는 우선순위 의미(0~7)
- 오른쪽 세 비트가 모두 0이 아닐 경우
 - 표1과 같이 인터넷이나 지역 당국에 의하여
부여된 우선순위에 따라 56(64-8)가지의
서비스 정의

Codepoint	Assigning Authority
XXXXX0	Internet
XXXXX11	Local
XXXXX01	Temporary or experimental

표1. 코드포인트의 값



IPv4

- 데이터그램(Datagram)

- 헤더 구성요소 (4/14)

- 전체 길이(Total Length)

- 데이터그램의 총 길이를 바이트 단위로 나타냄
 - 헤더와 데이터를 포함

- 상위 계층으로부터 받은 데이터에서 데이터의 길이를 구할 때 사용

- 데이터의 길이 = 전체 길이 - 헤더 길이(HLEN x 4)

0	3	4	7	8	15	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits				
Identification 16 bits					Flags 3 bits	Fragmentation offset 13 bits	
Time to live 8 bits		Protocol 8 bits		Header checksum 16 bits			
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

IPv4

- 데이터그램(Datagram)

*단편화(Fragmentation)

큰 패킷을 데이터 링크 계층의 최대 전송 단위 크기에 맞춰 여러 개의 작은 조각으로 나누어 전송하는 기술.

- 헤더 구성요소 (5/14)

- 식별(Identification)

- 데이터그램의 단편화와 재조립과 관련된 필드
 - 발신지로부터 나온 데이터그램을 유일하게 식별
 - 같은 식별자 값을 가지는 모든 단편은 하나의 데이터그램으로 재조립
 - 유일성을 보장하기 위해 IP 프로토콜은 카운터를 사용하여 데이터그램에 고유 식별 번호 부여
 - 카운터 값은 양의 정수 값으로 초기화
 - IP 프로토콜이 데이터그램을 보낼 때 카운터의 현재 값을 식별자 필드에 복사하고 카운터 값을 1 증가
 - 카운터 값이 주기억장치에 유지되는 한 유일성 보장

0	3	4	7	8	15	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits			Total length 16 bits		
Identification 16 bits					Flags 3 bits	Fragmentation offset 13 bits	
Time to live 8 bits		Protocol 8 bits			Header checksum 16 bits		
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

IPv4

• 데이터그램(Datagram)

*ICMP (Internet Control Message Protocol)

IP 네트워크에서 오류 보고, 네트워크 진단, 제어 메시지 전송을 위해 사용하는 네트워크 계층 프로토콜.

• 헤더 구성요소 (6/14)

• 플래그(Flags)

- 데이터그램의 단편화와 재조립과 관련된 필드

- 단편화의 여부 의미

- 첫 번째 비트는 미사용

- 두 번째 비트는 Do not fragment 비트

- 1이면 데이터그램을 단편화 금지

- 0이면 데이터그램 단편화 가능

- 단편화 필요 시 DF=1이면 데이터그램 폐기 및 ICMP 오류 메시지 전송

- 세 번째 비트는 More fragment 비트

- 1이면 뒤에 단편 존재

- 0이면 마지막 단편

D: Do not fragment

M: More fragments

	D	M
--	---	---

0	3	4	7	8	15	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits				
Identification 16 bits					Flags 3 bits	Fragmentation offset 13 bits	
Time to live 8 bits		Protocol 8 bits			Header checksum 16 bits		
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

IPv4

• 데이터그램(Datagram)

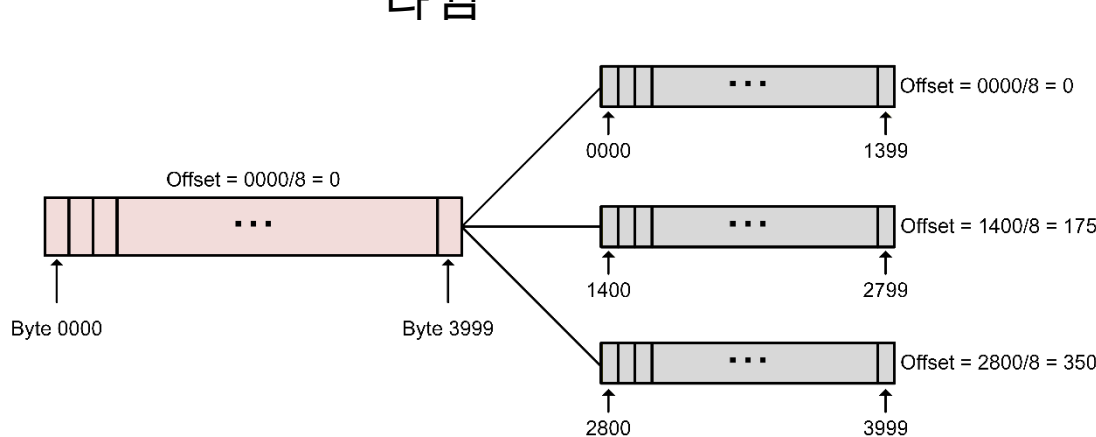
*MTU(Maximum Transfer Unit)
전송할 패킷의 크기가 라우터나 네트워크가 한 번에 보낼 수 있는 최대 크기.

• 헤더 구성요소 (7/14)

• 단편화 오프셋(Fragmentation Offset)

- 데이터그램의 단편화와 재조립과 관련된 필드
- 전체 데이터그램 내에서 단편의 상대적 위치를 의미
- 원래 데이터그램(Original Datagram) 내에서 데이터의 오프셋을 8바이트 단위로 나타냄

- e.g., MTU가 1400이고 크기가 4000바이트인 데이터가 세 개의 단편으로 나뉜다면, 0~1399번, 1400~2799번, 2800~3999번으로 총 세 개 단편으로 나뉨



0	3	4	7	8	15	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits			Total length 16 bits		
Identification 16 bits					Flags 3 bits	Fragmentation offset 13 bits	
Time to live 8 bits		Protocol 8 bits			Header checksum 16 bits		
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

IPv4

• 데이터그램(Datagram)

*홉(Hop)

컴퓨터 네트워크에서 출발지와 목적지 사이에 위치한 경로의 한 부분.

• 헤더 구성요소 (8/14)

• 수명(Time to live)

- 데이터그램이 방문할 수 있는 최대 홉 수를 의미
 - 발신지가 데이터그램을 보낼 때 수명 필드에 숫자를 저장
 - 숫자는 대략 두 호스트 사이에 위치한 라우터 수의 두 배
 - 데이터그램을 처리하는 각 라우터는 이 필드의 값을 1씩 감소
 - 0이 되면 라우터는 데이터그램 폐기
- 데이터그램이 오랫동안 라우터들 사이에서 돌아다니는 상황 방지
 - 라우터 큐의 불필요한 점유 방지
- 발신지가 의도적으로 패킷이 전달되는 범위를 제한할 때 사용 가능
 - e.g., 발신지가 패킷이 동일 네트워크 내에서만 전달되기를 원한다면 1로 설정

0	3	4	7	8	15	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits			Total length 16 bits		
Identification 16 bits					Flags 3 bits	Fragmentation offset 13 bits	
Time to live 8 bits		Protocol 8 bits			Header checksum 16 bits		
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

IPv4

• 데이터그램(Datagram)

• 헤더 구성요소 (9/14)

• 프로토콜(Protocol)

- IP 계층의 서비스를 사용하는 상위 계층 프로토콜 종류 식별
- IP 데이터그램은 여러 종류의 상위 계층 프로토콜 캡슐화 가능
- 목적지 호스트가 데이터그램을 받은 후 해당 상위 계층 프로토콜로 역다중화(Demultiplexing)하는 데 사용
 - 다중화(Multiplexing)
 - 상위 계층 데이터를 IP 데이터그램으로 캡슐화하여 전달
 - 역다중화(Demultiplexing)
 - Protocol 필드 값을 확인하여 해당 상위 계층 프로토콜(TCP, UDP, ICMP 등)로 전달

*캡슐화(Capsulation)

상위 계층의 데이터에 하위 계층의 헤더를 붙여 전송 가능한 형식으로 변환하는 과정.

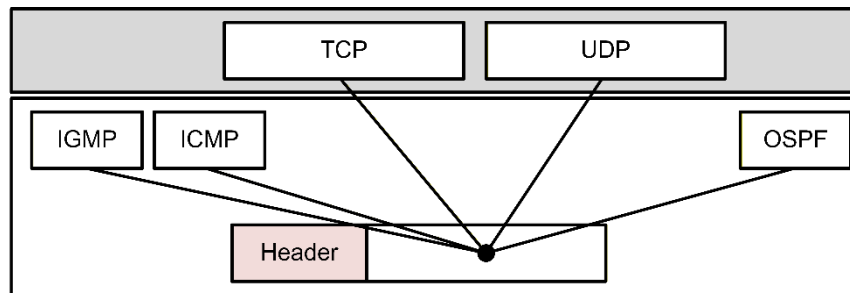
*세그먼트(Segment)

전송 계층에서 데이터를 일정한 크기로 자른 단위.

Value	Protocol
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Transport Layer

Network Layer



0	3	4	7	8	15	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits				
Identification 16 bits				Flags 3 bits	Fragmentation offset 13 bits		
Time to live 8 bits	Protocol 8 bits		Header checksum 16 bits				
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

IPv4

- 데이터그램(Datagram)

- 헤더 구성요소

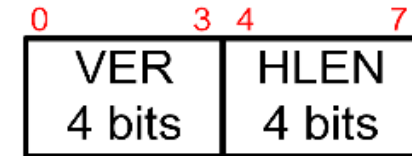
- 예제 7.1

처음 8비트가 01000010인 IP 패킷이 도착하였다. 이때 수신자는 이 패킷을 폐기한다. 이유를 설명하라.

- 풀이

- 패킷에 오류 존재

- 왼쪽 4비트(0100)은 버전으로 정상
 - 다음 4비트(0010)은 헤더의 길이를 나타내지만 $2 \times 4 = 8$ 로 오류
 - 헤더의 최소 바이트 수는 20바이트



- 예제 7.2

IP 패킷에서 HLEN의 값이 이진수로 1000이다. 이 패킷에는 옵션이 몇 바이트 있는가?

- 풀이

- HLEN의 값은 8이고 이것은 헤더 내 바이트의 총 수가 $32(8 \times 4)$ 임을 나타냄
 - 처음 20바이트는 주 헤더이므로 옵션은 12바이트

IPv4

- 데이터그램(Datagram)

- 헤더 구성요소

- 예제 7.3

IP 패킷에서 HLEN의 값이 16진수로 5이고 전체 길이 필드의 값이 16진수로 0028이다. 이 패킷은 몇 바이트의 데이터를 전송하고 있는가?

- 풀이
 - HLEN의 값이 5이므로 헤더는 $20(5 \times 4)$ 바이트이고 옵션은 없음을 알 수 있음
 - 전체 길이는 40바이트이므로 $40 - 20$ 인 20바이트의 데이터가 전송되고 있음을 알 수 있음

- 예제 7.4

처음 16진수 디지트(digits)들이 45000028000100000102...과 같은 패킷이 도착하였다. 이 패킷은 폐기되기 전에 몇 홉을 지날 수 있는가? 이 패킷은 어떤 상위 계층 프로토콜 데이터를 전송하고 있는가?

- 풀이
 - 수명 필드를 찾기 위해서 처음 8바이트(16개의 16진 디지트)를 넘어 탐색
 - 수명 필드는 9번째 바이트이며 이는 01
 - 패킷이 한 홉만 갈 수 있다는 것을 의미
 - 프로토콜 필드는 다음 바이트로서 02
 - 상위 계층 프로토콜이 IGMP임을 알 수 있음

Value	Protocol
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

IPv4

- 데이터그램(Datagram)

- 헤더 구성요소 (10/14)

- 검사합(Checksum) (1/2)

- 패킷의 전달 중에 발생할 수 있는 오류 검출을 위해 사용되는 값
 - 발신지와 목적지에서 검사합 계산 수행

0	3	4	7	8	15	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits			Total length 16 bits		
Identification 16 bits					Flags 3 bits	Fragmentation offset 13 bits	
Time to live 8 bits		Protocol 8 bits			Header checksum 16 bits		
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

• 데이터그램(Datagram)

• 헤더 구성요소 (11/14)

• 검사합(Checksum) (2/2)

• 발신지의 검사합 계산

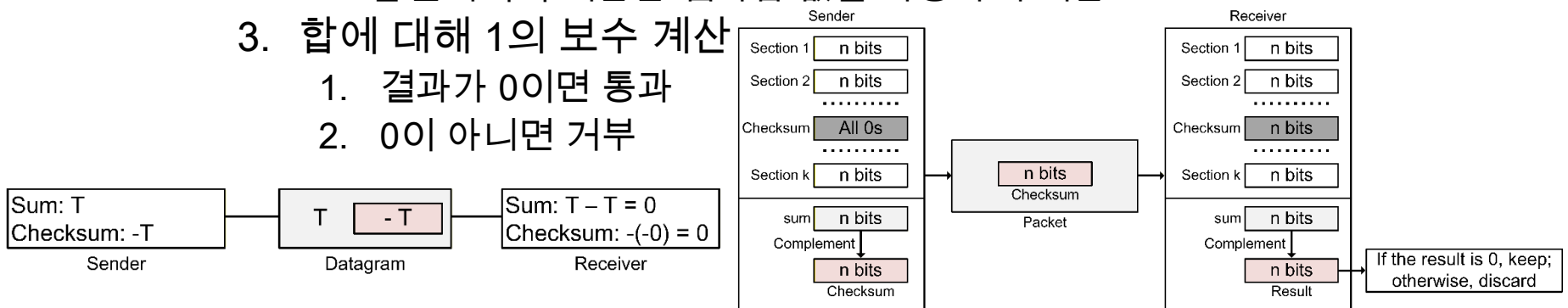
1. 패킷을 크기가 n 비트인 k 개의 조각으로 분할(주로 $n = 16$)
2. 모든 조각을 1의 보수 덧셈을 사용하여 합을 계산
 - 검사합의 값은 0
3. 합에 대해 1의 보수를 취하여 검사합 계산

• 목적지의 검사합 계산

1. 수신된 패킷을 k 개의 조각으로 분할
2. 모든 조각을 1의 보수 덧셈을 사용하여 합 계산
 - 발신지에서 계산한 검사합 값을 사용하여 계산
3. 합에 대해 1의 보수 계산
 1. 결과가 0이면 통과
 2. 0이 아니면 거부

*1의 보수 덧셈

2진수를 더할 때 자릿수를 넘긴 자리올림을 다시 하위 비트에 더해주는 덧셈 방식



IPv4

- 데이터그램(Datagram)

- 헤더 구성요소

- 예제 7.5

다음 IP 데이터그램의 발신지에서의 검사합을 16비트로 나누어 계산하라.

- 풀이

- 4, 5 and 0 → 01000101 00000000
- 28 → 00000000 00011100
- 1 → 00000000 00000001
- 0 and 0 → 00000000 00000000
- 4 and 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				

IPv4

- 데이터그램(Datagram)

- 헤더 구성요소

- 예제 7.6

다음 IP 데이터그램의 목적지에서의 검사합을 16비트로 나누어 계산하라.

- 풀이

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

- sum → 11111111 11111111
- Checksum → 00000000 00000000

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

IPv4

- 데이터그램(Datagram)

- 헤더 구성요소 (12/14)

- 발신지 주소(Source Address)

- 발신지의 IP 주소를 의미

- IP 데이터그램이 발신지에서 목적지까지 전달되는 동안 고정

- 목적지 주소(Destination Address)

- 목적지의 IP 주소를 의미

- IP 데이터그램이 발신지에서 목적지까지 전달되는 동안 고정

0	3	4	7	8	15	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits			Total length 16 bits		
Identification 16 bits					Flags 3 bits	Fragmentation offset 13 bits	
Time to live 8 bits		Protocol 8 bits			Header checksum 16 bits		
Source IP address							
Destination IP address							
Options + padding (0 to 40 bytes)							

IPv4

- 데이터그램(Datagram)

- 헤더 구성요소 (13/14)

- 옵션 (1/2)

- 헤더의 가변 부분으로 최대 40바이트
 - 네트워크를 시험하거나 디버그하기 위해 사용

0		3		4		7		8		15		16		31	
VER 4 bits		HLEN 4 bits		Service type 8 bits				Total length 16 bits							
Identification 16 bits								Flags 3 bits		Fragmentation offset 13 bits					
Time to live 8 bits				Protocol 8 bits				Header checksum 16 bits							
Source IP address															
Destination IP address															
Options + padding (0 to 40 bytes)															

- 옵션 형식

- 유형(Type)

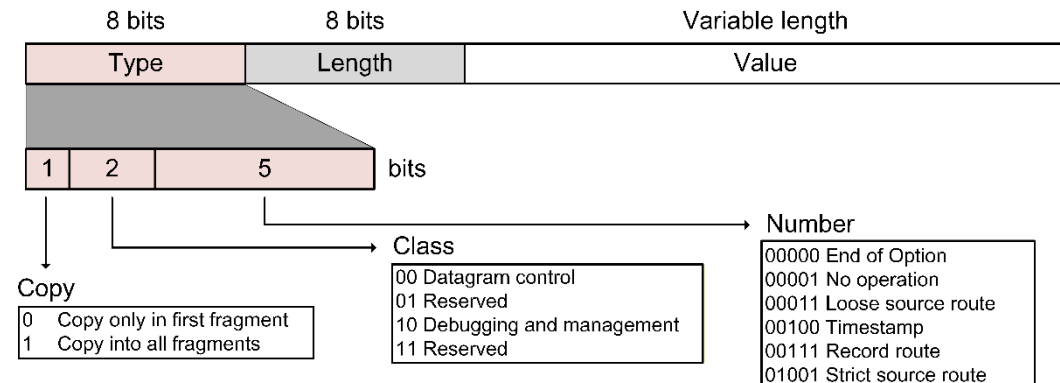
- 복사(Copy): 단편화 시 단편에 옵션을 포함시킬 것인지 제어
 - 클래스(Class): 옵션의 일반적인 목적 정의
 - 번호(Number): 옵션의 유형 정의

- 길이(Length)

- 유형, 길이, 값 필드를 포함한 옵션 전체 길이

- 값(Value)

- 특정 옵션이 필요로 하는 데이터 포함



IPv4

- 데이터그램(Datagram)

- 헤더 구성요소 (14/14)

- 옵션 (2/2)

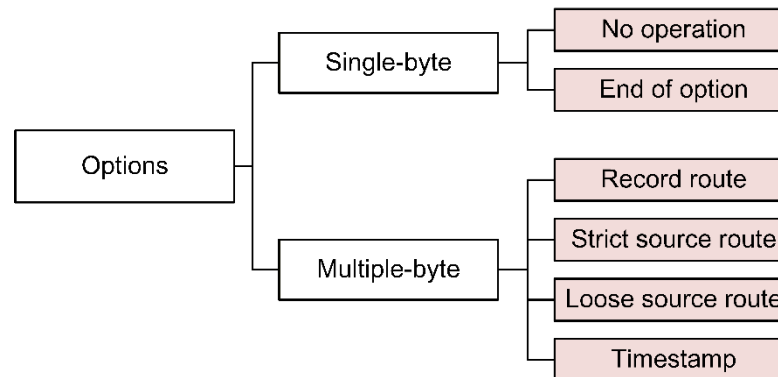
- 옵션 유형

- 1바이트 옵션

- 1바이트로 구성되는 옵션
 - 추가적인 데이터 없이 형식(Type) 필드만 사용
 - 주로 패딩과 같은 간단한 기능을 수행

- 다중 바이트 옵션

- 2바이트 이상으로 구성되는 옵션
 - Type, Length, Data 필드로 구성
 - 경로 기록, 타임스탬프 등 복잡한 기능을 수행



IPv4

• 옵션 유형 (1/9)

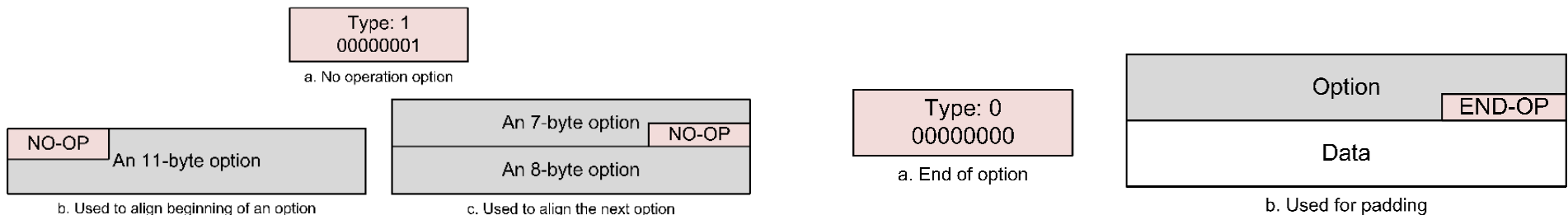
• 1바이트 옵션

• 무연산 옵션(No operation)

- 옵션들 사이의 여백을 채워주는 목적으로 사용
 - Type은 00000001
 - 16비트나 32비트 경계에 위치시키기 위해 사용

• 옵션 종료 옵션(End of Option)

- 옵션의 필드 끝의 패딩 목적으로 사용
 - Type은 00000000
 - 마지막 옵션으로만 사용 가능
- 최대 1개만 사용 가능



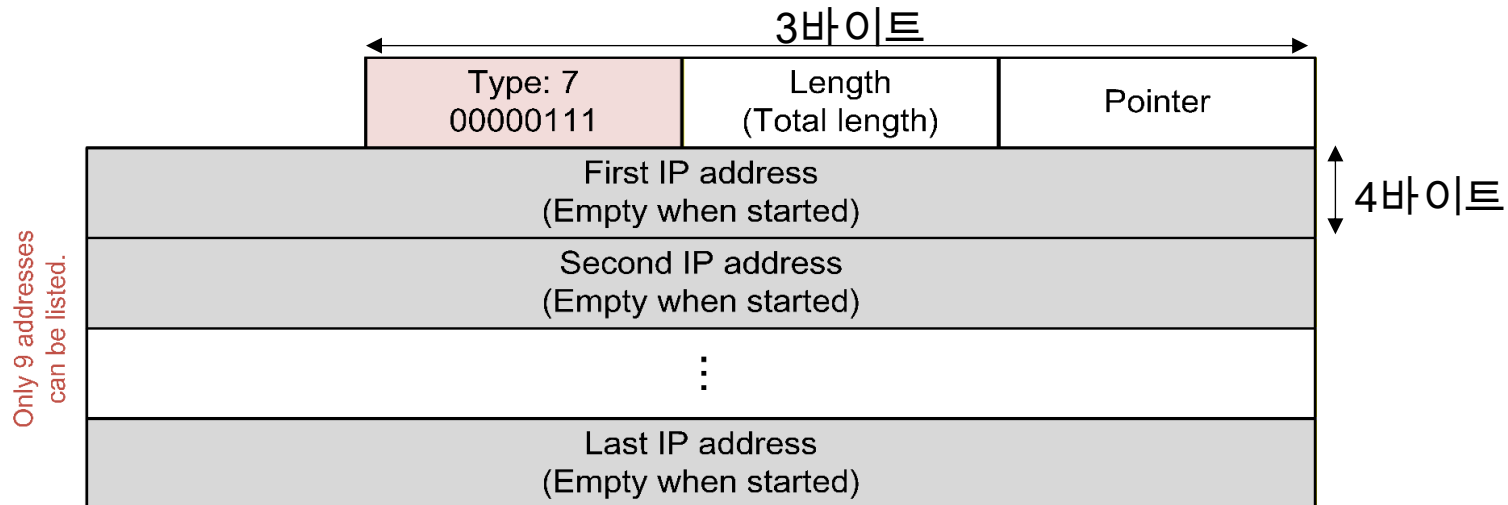
IPv4

- 옵션 유형 (2/9)

- 다중 바이트 옵션

- 경로 기록 옵션(Record Route) (1/2)

- 데이터그램을 처리한 라우터들을 기록하는 옵션
 - 최대 9개까지 기록 가능
 - 옵션 길이 40바이트 중 37바이트 사용 가능
 - 9개가 넘어가면 기록하지 않음
 - 포인터는 첫 번째 빈 엔트리의 바이트 번호를 포함하는 옵션 정수
 - 처음에 첫 번째 필드를 가리키므로, 기본값 4



IPv4

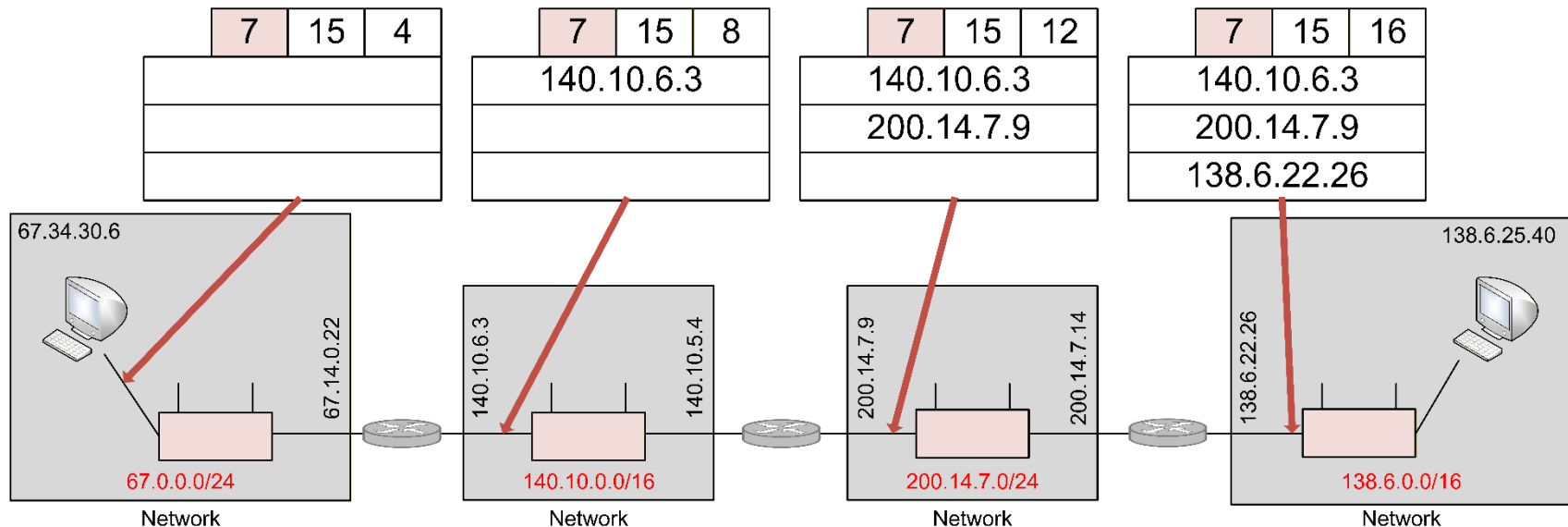
- 옵션 유형 (3/9)

- 다중 바이트 옵션

- 경로 기록 옵션(Record Route) (2/2)

- 예시

1. 처음 포인터 필드는 기본값 4를 가짐
2. 데이터그램이 이동할 때 라우터는 포인터 값과 길이 값을 비교
3. 라우터를 지날 때 마다 지나온 경로를 기록
4. 포인터 필드는 기록할 때마다 4 증가시키고 다음 경로로 이동



IPv4

- 옵션 유형 (4/9)

- 다중 바이트 옵션

- 엄격한 발신지 경로 옵션(Strict Source Route) (1/2)

- 발신지가 지정한 경로상의 모든 라우터를 순서대로 차례대로 거쳐야만 전달되는 옵션
 - Type은 10001001
 - 특정 네트워크 우회, 지연 최소화, 처리율 향상 등의 목적으로 사용
 - 폐기 후 오류 메시지 전송 조건
 - 지정된 라우터를 방문할 수 없는 경우
 - 목적지 도착 시 지정된 경로를 모두 수행하지 못한 경우

Only 9 addresses can be listed.	Type: 137 10001001	Length (Total length)	Pointer
	First IP address (Filled when started)		
	Second IP address (Filled when started)		
	⋮		
	Last IP address (Filled when started)		

IPv4

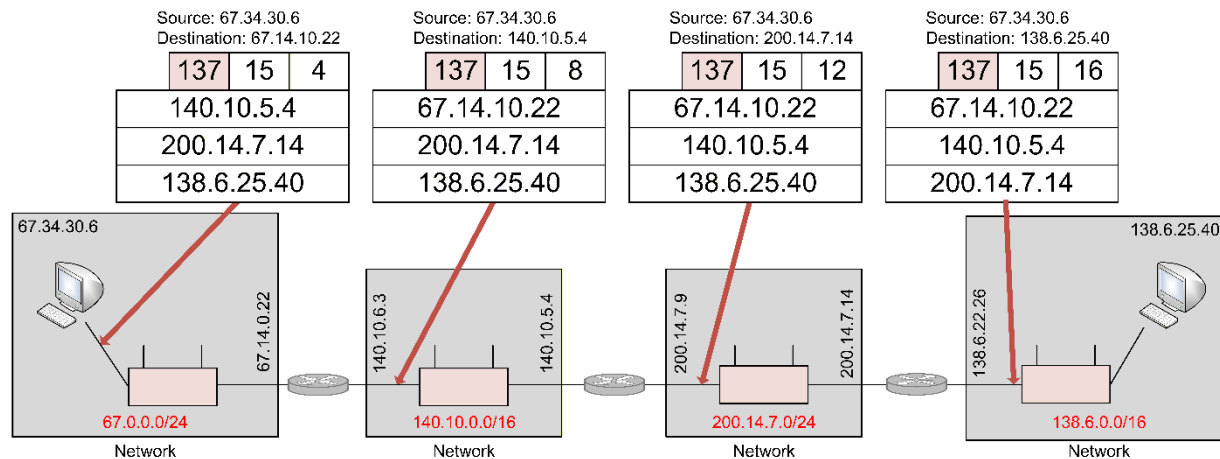
- 옵션 유형 (5/9)

- 다중 바이트 옵션

- 엄격한 발신지 경로 옵션(Strict Source Route) (2/2)

- 예시

- 경로가 미리 지정된 것 이외에 경로 기록 옵션과 동일
 - 데이터그램이 이동할 때 라우터는 포인터의 값과 길이 값을 비교
 - 포인터의 값보다 길이 값이 크면, 미리 지정된 라우터를 모두 방문을 의미
 - 포인터의 값이 더 작다면, 라우터는 포인터가 가리키고 있는 IP 주소와 라우터에서 데이터그램이 들어온 IP 주소 비교
 - 두 값이 같다면, 현재의 IP 주소를 나가는 IP 주소로 대체하고 포인터 값을 4 증가 시킨 후 데이터그램 전달



IPv4

- 옵션 유형 (6/9)

- 다중 바이트 옵션

- 느슨한 발신지 경로 옵션(Loose Source Route)

- 발신지가 지정한 필수 라우터들을 거치되, 라우터 간 사이의 경유지는 자유롭게 선택할 수 있는 옵션

- Type은 10000011

- 엄격한 발신지 경로와 달리, 리스트에 정의한 라우터 이외에 다른 라우터도 방문 가능

- 옵션 필드에 적힌 경로들 외의 중간 경로는 유동적으로 선택 가능

Only 9 addresses can be listed.	Type: 131 10000011	Length (Total length)	Pointer
	First IP address (Filled when started)		
	Second IP address (Filled when started)		
	⋮		
	Last IP address (Filled when started)		

IPv4

- 옵션 유형 (7/9)

- 다중 바이트 옵션

- 타임스탬프(Timestamp) (1/3)

- 라우터가 데이터그램을 처리하는 시간을 기록하는 옵션

- Type은 01000100

- 시간은 세계 표준시(Universal Time) 자정으로부터 millisecond 단위로 표시

- 하나의 라우터에서 다른 라우터까지 가는데 걸리는 시간 추정 가능

- 모든 라우터가 세계 표준시를 사용하지만 각 시계는 정확히 동기화 되지 않을 수 있기에 추정만 가능

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

IPv4

- 옵션 유형 (8/9)

- 다중 바이트 옵션

- 타임스탬프(Timestamp) (2/3)

- 추가된 구조

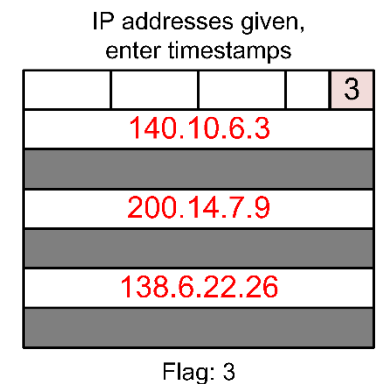
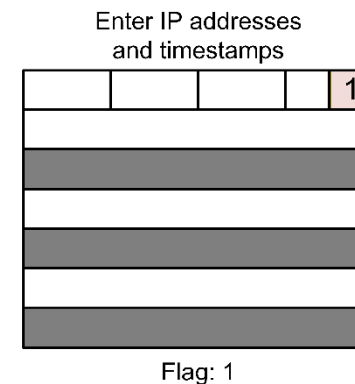
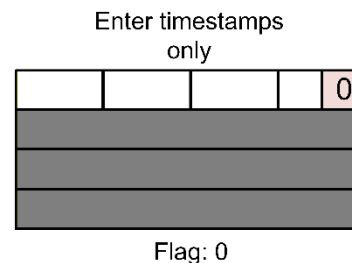
- 오버플로우(O-flow)

- 필드가 더이상 없어 타임스탬프를 기록하지 못한 라우터의 수 기록

- 플래그(Flags)

- 방문된 라우터의 동작 명세
 - 0일 경우, 각 라우터는 주어진 필드에 타임스탬프만 추가
 - 1일 경우, 라우터는 출력 인터페이스 IP와 타임스탬프 기록
 - 3일 경우, 라우터는 주어진 IP 주소와 입력 인터페이스 IP 주소를 비교해 같을 경우 IP 주소에 출력 인터페이스 IP 주소를 덮어쓰고 타임스탬프 추가

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



IPv4

- 옵션 유형 (9/9)

- 다중 바이트 옵션

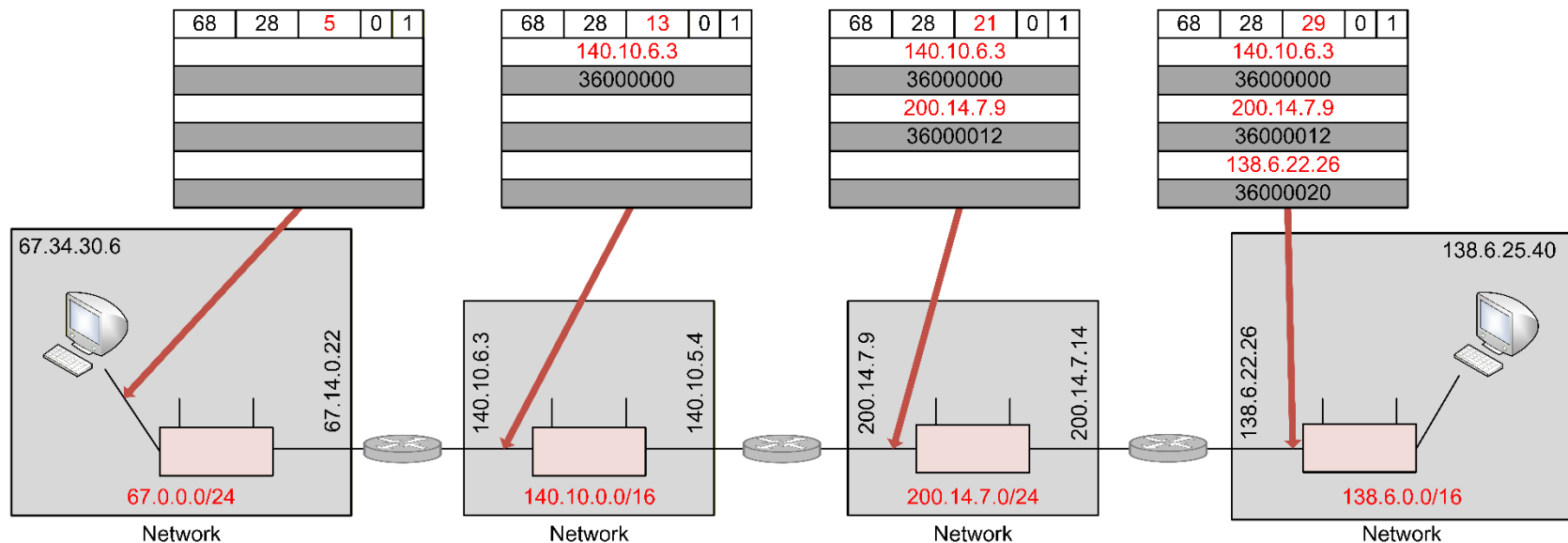
- 타임스탬프(Timestamp) (3/3)

- 예시

- 1. 라우터는 플래그 값을 확인

- 플래그가 1이므로, 포인터가 가리키는 필드에 IP 주소와 타임스탬프 값을 기록

- 2. 포인터 값을 8 증가시키고 데이터그램을 다음 경로로 이동



IPv4

• 옵션 유형

• 예제 7.7

다음에 대해 설명하라.

1. 6개의 옵션 중 어느 것이 각 단편에 복사되어야 하는가?
2. 6개의 옵션 중 어느 것이 데이터그램 제어에 사용되고 어떤 것이 디버깅과 관련이 있는가?

- 1) 풀이
 - 옵션의 코드의 왼쪽에서 첫 번째 비트로 판단
 - 0이면 복사 안함, 1이면 복사
- 2) 풀이
 - 옵션의 코드의 왼쪽에서 두 번째와 세 번째 비트로 판단
 - 00이면 데이터그램 제어
 - 10이면 디버깅

옵션	유형	단편 복사 여부	데이터그램 제어/디버깅
무연산	00000001	X	데이터그램 제어
옵션 종료	00000000	X	데이터그램 제어
경로 기록	00000111	X	데이터그램 제어
엄격한 경로 기록	10001001	O	데이터그램 제어
느슨한 경로 기록	10000011	O	데이터그램 제어
타임스탬프	01000100	X	디버깅

IPv4

• 단편화(Fragmentation) (1/3)

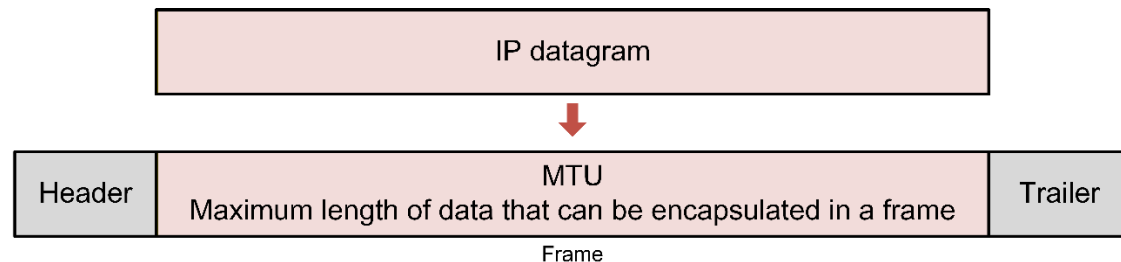
*MTU(Maximum Transfer Unit)
전송할 패킷의 크기가 라우터나 네트워크가 한 번에 보낼 수 있는 최대 크기.

• 정의

- 전송할 패킷의 크기가 MTU보다 클 때, 패킷을 작은 조각으로 나누는 과정

• 특징 (1/2)

- MTU 값에 따라 단편화 크기는 다름
 - MTU 값은 물리 네트워크 프로토콜에 따라 다름
 - e.g., Ethernet LAN은 1500바이트, FDDI LAN은 4352바이트, PPP는 296바이트
- 단편화 시 헤더의 일부 필드 값만 변경
 - e.g., 식별자 필드는 유지되며, 플래그 필드와 단편화 오프셋 필드 변동

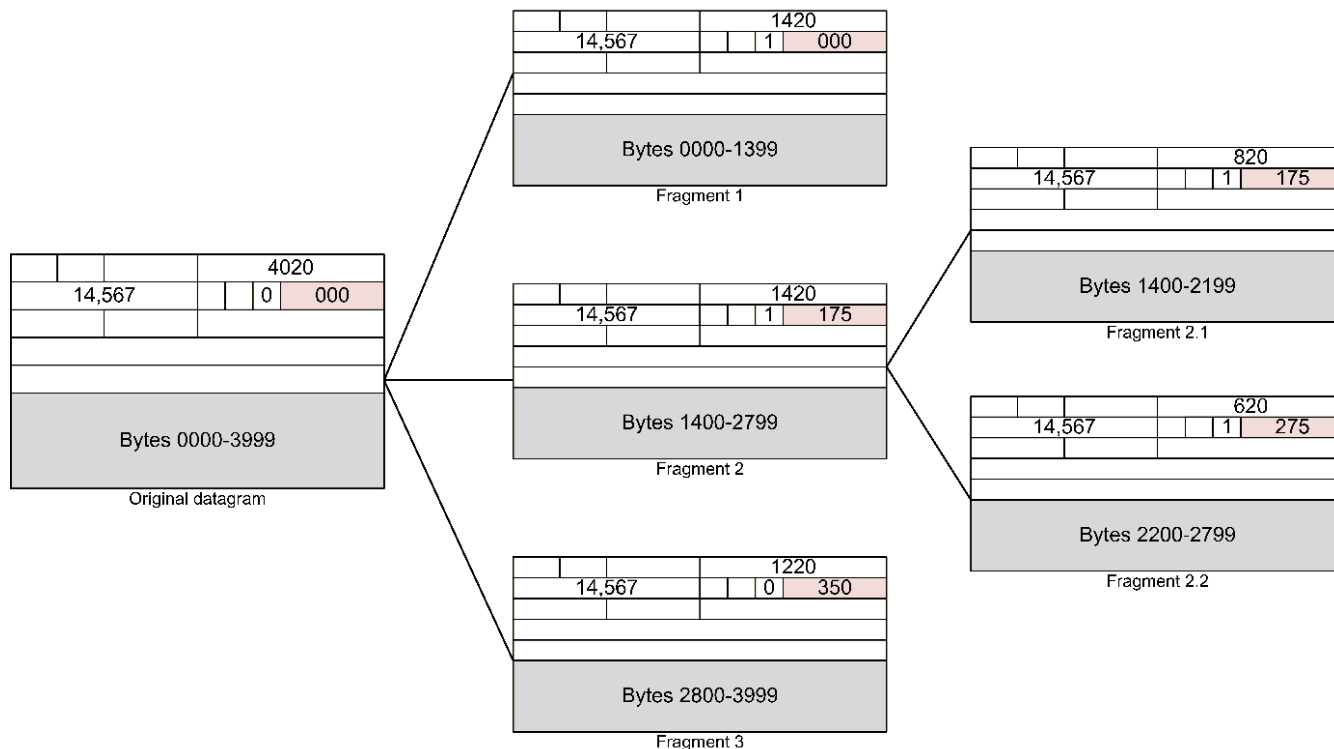


IPv4

- 단편화(Fragmentation) (2/3)

- 특징 (2/2)

- 단편화 된 것이 다시 단편화 될 수 있음
 - 오프셋 필드의 값은 항상 원래 데이터그램에서 상대적인 위치로 결정
- 단편 순서가 역전되어도 모든 단편이 존재하면 재조립 가능

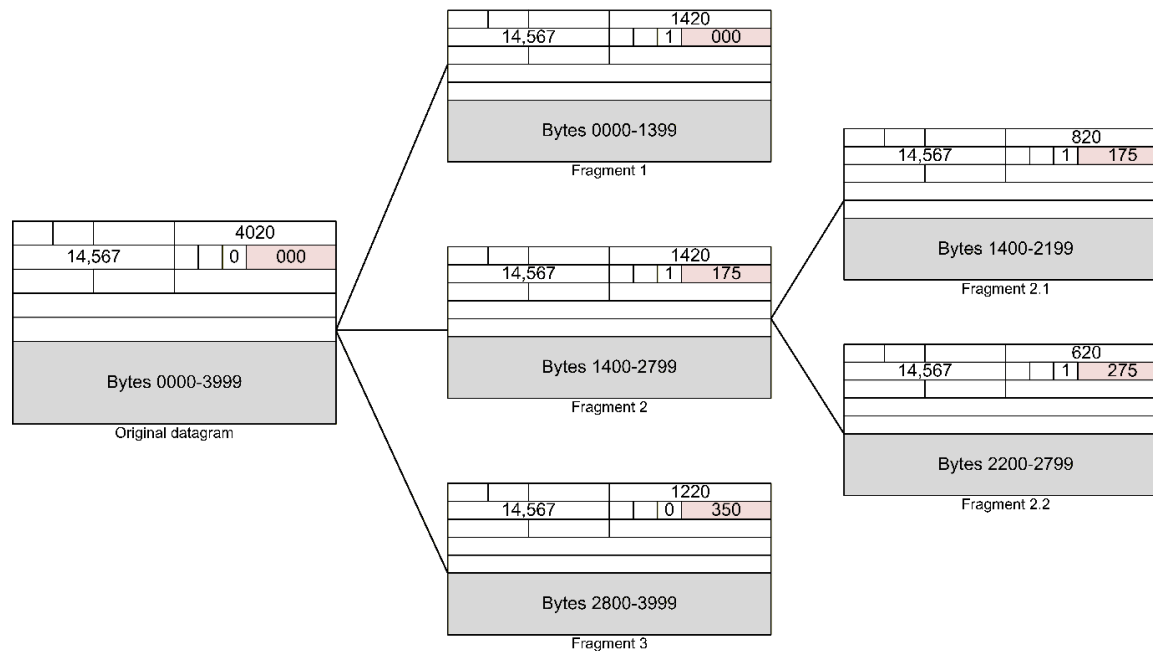


IPv4

• 단편화(Fragmentation) (3/3)

• 재조립 과정

- 모든 단편의 식별자 값을 확인하여 동일한 데이터그램인지 판단
- 단편화 오프셋을 기준으로 순서를 정렬하고 데이터 연속성 확인
- MF(More Fragment) 값을 확인하여 마지막 단편 여부 판단
- MF 값이 0인 단편 수신 시 모든 단편을 결합하여 재조립 완료



• 단편화(Fragmentation)

• 예제 7.8

각 상황 별로 패킷이 단편화 되었는지, 되었다면 첫 번째 단편인지, 마지막 단편 혹은 중간 단편인지 설명하라.

1. M 비트 값이 0인 패킷이 도착하였다.
2. M 비트 값이 1인 패킷이 도착하였다.
3. M 비트 값이 1이고 단편화 오프셋이 0인 패킷이 도착하였다.

- 1) 풀이
 - M 비트가 0이면 단편이 더 이상 없다는 것을 의미
 - 단편은 마지막 단편
 - M 비트가 0이라는 것만 가지고 원래의 패킷이 단편화 되었는지는 알 수 없음
 - 단편화 되지 않은 패킷은 마지막 단편으로 고려
- 2) 풀이
 - M 비트가 1이면 단편이 더 존재한다는 것을 의미
 - 이 패킷은 첫 번째 또는 중간 단편 중 하나
 - 이를 알기 위해서는 단편화 오프셋과 같은 정보 필요
 - M 비트가 1이라는 것은 단편화가 되었음을 의미
- 3) 풀이
 - M 비트가 1이므로 첫 번째 또는 중간 단편
 - 오프셋이 0이므로 첫 번째 단편

IPv4

• 단편화(Fragmentation)

• 예제 7.9

각 상황에 맞추어 첫 번째 바이트의 번호는 무엇인지, 마지막 바이트의 번호를 알 수 있는지 설명하라.

1. 오프셋 값이 100인 패킷이 도착하였다
2. 오프셋 값이 100이고 HLEN이 5이며 전체 길이가 100인 패킷이 도착하였다.
 - 1) 풀이
 - 첫 번째 바이트의 번호를 구하기 위해 오프셋 값을 8로 곱하여 계산
 - 첫 번째 바이트 번호는 800
 - 데이터의 길이를 모르기 때문에 마지막 바이트의 번호는 미지수
 - 2) 풀이
 - 첫 번째 바이트의 번호를 구하기 위해 오프셋 값을 8로 곱하여 계산
 - 첫 번째 바이트 번호는 800
 - 전체 길이는 100바이트이고 헤더 길이는 20바이트(5x4)이므로 데이터그램 내에는 80바이트 데이터 존재
 - 첫 번째 바이트 번호가 800이므로 마지막 바이트 번호는 879

IPv4

- ATM 상 IP (1/6)

*스위칭(Switching)

데이터를 목적지까지 전달할 때, 미리 정해진 경로 정보(테이블)를 참조해 빠르게 다음 목적지로 넘겨주는 방식.

- 개요 (1/3)

- ATM (Asynchronous Transfer Mode)

- 정의

- 네트워크에서 데이터를 셀(Cell)로 나누어 전송하는 방식
 - e.g., 전화 시스템, 실시간 고품질 화상회의, 방송 송출망

- 특징

- 고속 통신망에 적합한 기술
 - 고정 길이 53바이트 셀 단위로 데이터 전송
 - 스위치 간 가상 회선(Virtual Circuit) 기반의 경로 설정

- IP와의 차이점

- 연결 지향형 방식으로 사전에 전송 경로(VC) 설정
 - 고정 길이의 셀 단위로 고속 전송
 - 복잡한 테이블 조회 없이 자동 교환 처리하는 L2 스위칭 방식

IPv4

• ATM 상 IP (2/6)

• 개요 (2/3)

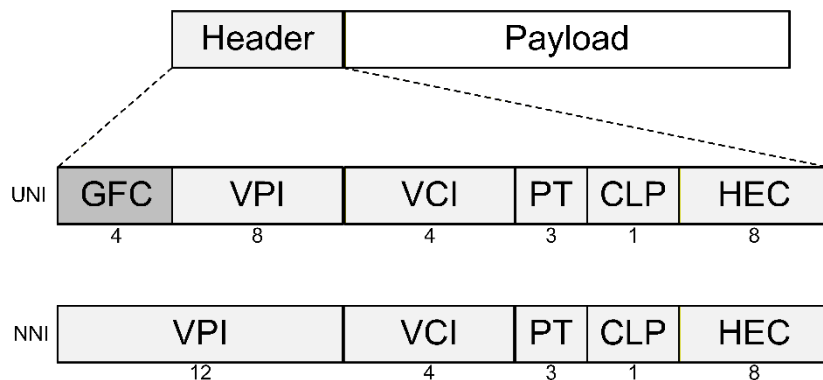
• ATM 셀(ATM Cell)

• 정의

- ATM 네트워크 상에서 데이터를 전달하는 최소 단위

• 구조

- 5바이트 헤더와 48바이트 데이터로 구성
- 헤더는 UNI(User-Network Interface)와 NNI(Network-Network Interface) 환경에 따라 차이 존재



Name	Size	Description
GFC (Generic Flow Control)	4bits	UNI에서만 사용 하며, 사용자와 네트워크 사이 흐름을 제어하는데 사용
VPI (Virtual Path Identifier)	8~12bits	가상 경로를 식별하는 번호
VCI (Virtual Channel Identifier)	16bits	가상 채널을 식별하는 번호
PT (Payload Type)	3bits	페이로드가 어떤 데이터인지 구분
CLP (Cell Loss Priority)	1bits	셀 손실 우선순위 표시
HEC (Header Error Control)	8bits	ATM 헤더 전용 오류 검출/수정 코드

IPv4

- ATM 상 IP (3/6)

- 개요 (3/3)

- AAL (ATM Adaptation Layer)

- 정의

- ATM 네트워크에서 상위 계층 데이터를 ATM 셀 형식으로 변환하기 위한 계층

- 종류

- AAL1

- 고정 비트 전송률 및 실시간 전송 지원
 - 송수신 장치 시계 동기화로 끊임 없는 실시간 전송

- AAL2

- 가변 비트 전송률 및 실시간 전송 지원

- AAL3/4

- 연결형, 비연결형 가변 비트 전송률 모두 지원
 - 오류 검출 기능 포함하며 데이터 통신 용도

- AAL5

- AAL3/4 대비 오버헤드가 감소한 간소화 버전
 - IP over ATM 등 오늘날 고속 데이터 통신에 주로 사용

IPv4

- ATM 상 IP (4/6)

- 정의

- IP 패킷을 ATM 네트워크의 셀 단위로 변환하여 전송하는 방식

- 필요성

- 기존 IP 서비스를 ATM 기반 광대역 네트워크에서도 제공하기 위해 사용
 - 기존 IP 서비스는 LAN 또는 점대점 WAN 환경 기반

- 특징

- IP 패킷을 여러 개의 ATM 셀로 분할하여 전송
 - AAL5를 이용하여 IP 패킷을 ATM 셀로 변환하고 목적지에서 재조립
 - IP 주소 체계와 ATM 주소 체계가 독립적으로 존재
 - 두 주소 체계 간 매핑을 위한 주소 변환 프로토콜 필요
 - e.g., ATMARP (ATM Address Resolution Protocol)
 - 매핑된 ATM 주소로 가상회선을 설정하고 셀 라우팅 수행

IPv4

- ATM 상 IP (5/6)

- 셀 라우팅

- 정의

- ATM 셀이 목적지까지 전달되도록 경로를 설정하는 과정

- 주소

- IP 주소

- IP 계층에서 라우터를 정의
 - ATM 네트워크와 라우터에 정의된 IP 주소는 무관

- 물리 주소

- ATM 네트워크 관리자에 의해 정의된 20바이트의 주소
 - ATM 네트워크와 연관되어 있으며 인터넷과는 무관
 - 연결 설정 동안 사용

- 가상 회선 식별자(VCI, Virtual Circuit Identifier)

- ATM 연결 설정 과정에서 생성되는 식별자로, ATM 스위치가 셀을 전달할 때 사용하는 주소
 - 데이터 전송에 사용

IPv4

• ATM 상 IP (6/6)

• 셀 라우팅

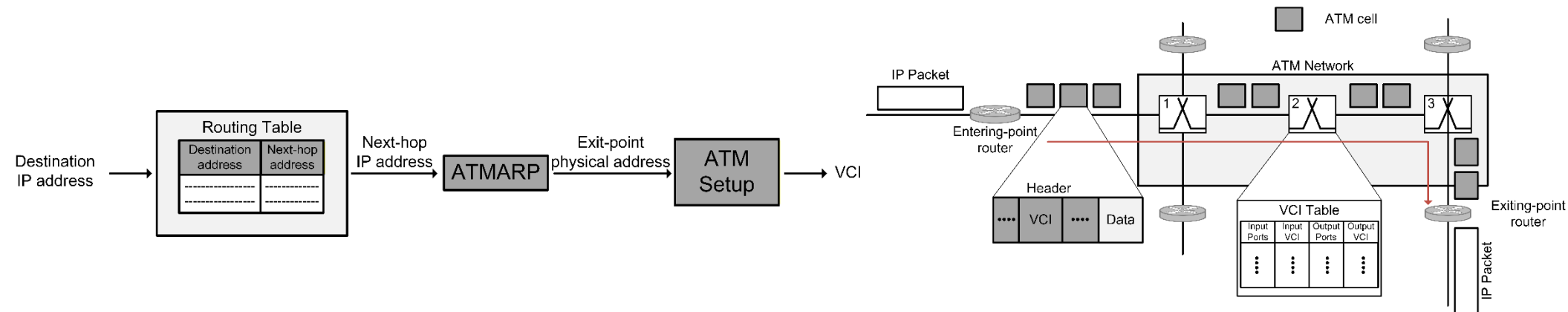
• 주소 결합

• 목적

- 셀 라우팅을 위한 가상 회선 식별자를 목적지 IP 주소로부터 결정

• 과정

1. 진입점 라우터(Entering-point router)가 목적지 IP 주소를 확인하고 라우팅 테이블에서 다음 홉 IP 주소를 탐색
2. ATMARP를 이용하여 다음 홉 IP 주소에 대응하는 ATM 물리 주소를 획득
3. ATM 연결을 설정하는 과정에서 ATM 물리 주소와 VCI를 매핑
4. ATM 셀은 매핑된 VCI를 이용하여 ATM 네트워크를 통해 전달



IPv4

- 보안(Security) (1/7)
 - 보안 문제점
 - 패킷 스니핑(Packet Sniffing)
 - 침입자가 IP 패킷을 도청하여 복사본 생성 가능
 - 내용을 수정하지 않는 수동적 공격이며 탐지 어려움
 - 패킷 암호화로 대응 가능
 - e.g., IPsec, TLS/SSL
 - 패킷 수정(Packet Modification)
 - 공격자가 패킷을 가로채 내용 변조 후 재전송 가능
 - 데이터 무결성 메커니즘으로 탐지 가능
 - e.g., 해시 함수 기반 체크섬, HMAC, 전자서명
 - IP 스푸핑(IP Spoofing)
 - 공격자가 다른 발신지 주소로 위장한 IP 패킷 생성 가능
 - 발신지 인증(Origin Authentication) 메커니즘으로 방지 가능
 - e.g., IPsec AH, 디지털 인증서 기반 상호 인증

IPv4

- 보안(Security) (2/7)
 - IPsec (Internet Protocol security)
 - 정의
 - 네트워크 계층에서 IP 패킷에 인증, 무결성, 기밀성을 제공하기 위한 보안 프로토콜 집합
 - 필요성
 - IP는 기본적으로 암호화 기능이 없어 전송 데이터의 기밀성 보장이 필요
 - IP는 데이터 변조 여부 확인과 송/수신자 인증 기능이 없어 무결성과 인증 보장이 필요
 - 특징
 - 네트워크 계층에서 IP 패킷을 보호하여 보안 통신 채널 제공
 - 가상 사설망(VPN)에서 주로 사용

IPv4

- 보안(Security) (3/7)
- IPsec (Internet Protocol security)
 - 운영 모드
 - 목적
 - 통신 대상과 환경에 따라 보호할 IP 패킷 범위를 결정하기 위해 사용
 - 특징
 - IPsec 연결 설정(IKE 협상, Internet Key Exchange) 시 먼저 결정
 - 선택된 모드에 따라 SA(Security Association)가 생성
 - 설정된 SA는 통신 중 유지되며, 보호 방식 변경 시 새로운 SA 생성 필요
 - 종류
 - 전송 모드(Transport Mode)
 - 터널 모드(Tunnel Mode)

*IKE (Internet Key Exchange)

IPsec 통신에 필요한 보안 알고리즘과 키를 협상하는 프로토콜.

*SA (Security Association)

IPsec 통신에서 사용할 보안 정책과 키 정보를 저장한 논리적 연결.

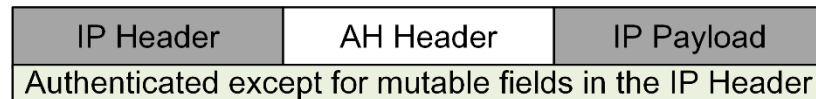
IPv4

- 보안(Security) (4/7)
 - IPsec (Internet Protocol security)
 - 운영 모드
 - 전송 모드(Transport Mode)
 - 정의
 - 원본 IP 헤더는 유지하고 IP Payload 부분에 보안 기능을 적용하는 방식
 - 특징
 - 호스트-호스트 간에 주로 사용
 - IP 헤더를 그대로 사용하여 오버헤드가 적음
 - 터널 모드(Tunnel Mode)
 - 정의
 - 원래 IP 패킷 전체를 암호화 및 인증하고, 새로운 IP 헤더를 추가하여 전송하는 방식
 - 특징
 - 라우터-라우터 또는 호스트-라우터 간 VPN에 주로 사용
 - 내부 IP 주소를 보호할 수 있음

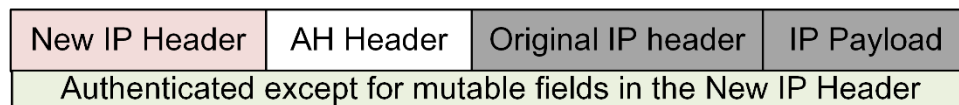
IPv4

- 보안(Security) (5/7)
 - IPsec (Internet Protocol security)
 - 핵심 프로토콜 (1/2)
 - AH(Authentication Header) 프로토콜
 - 전송 모드
 - 변경 가능한 IP Header 필드를 제외한 IP Packet 전체에 대한 무결성과 인증 제공
 - 데이터 암호화 기능은 제공하지 않음
 - 터널 모드
 - 원본 IP Packet 전체와 새로운 IP Header 일부를 제외한 전체 데이터에 대한 무결성과 인증 제공
 - 데이터 암호화 기능은 제공하지 않음

AH
Transport Mode

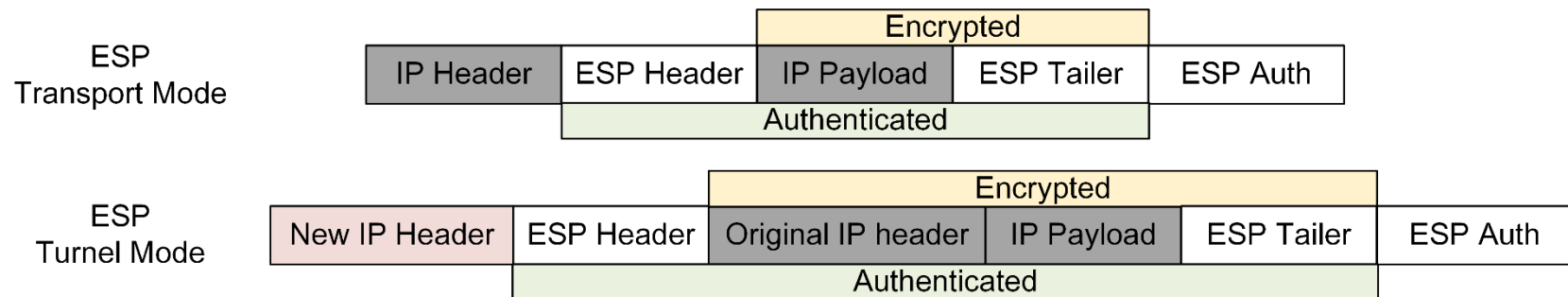


AH
Tunnel Mode



IPv4

- 보안(Security) (6/7)
 - IPsec (Internet Protocol security)
 - 핵심 프로토콜 (2/2)
 - ESP(Encapsulating Security Payload) 프로토콜
 - 전송 모드
 - IP Payload를 암호화하고 ESP Header와 암호화 데이터를 인증
 - 데이터 기밀성, 무결성 및 발신지 인증 제공
 - 터널 모드
 - 원본 IP Packet 전체를 암호화하고 ESP Header와 암호화 데이터를 인증
 - 내부 IP 주소 보호 및 VPN 환경에서 주로 사용



IPv4

- 보안(Security) (7/7)
 - IPsec (Internet Protocol security)
 - 제공 서비스
 - 알고리즘과 키의 결정
 - IKE를 통해 암호 알고리즘과 세션 키를 협상
 - 패킷 암호화
 - 협상된 암호 알고리즘과 키를 이용해 전송 패킷을 암호화
 - 패킷 스니핑 공격 무력화
 - 데이터 무결성
 - 전송 중 패킷이 변조되지 않았음을 보장
 - 무결성 검증 실패 시 수신 패킷 폐기
 - 패킷 수정 공격 방지
 - 발신지 인증
 - 패킷이 위장자에 의해 생성되지 않았음을 확인
 - IP 스푸핑 공격 방지

IPv4

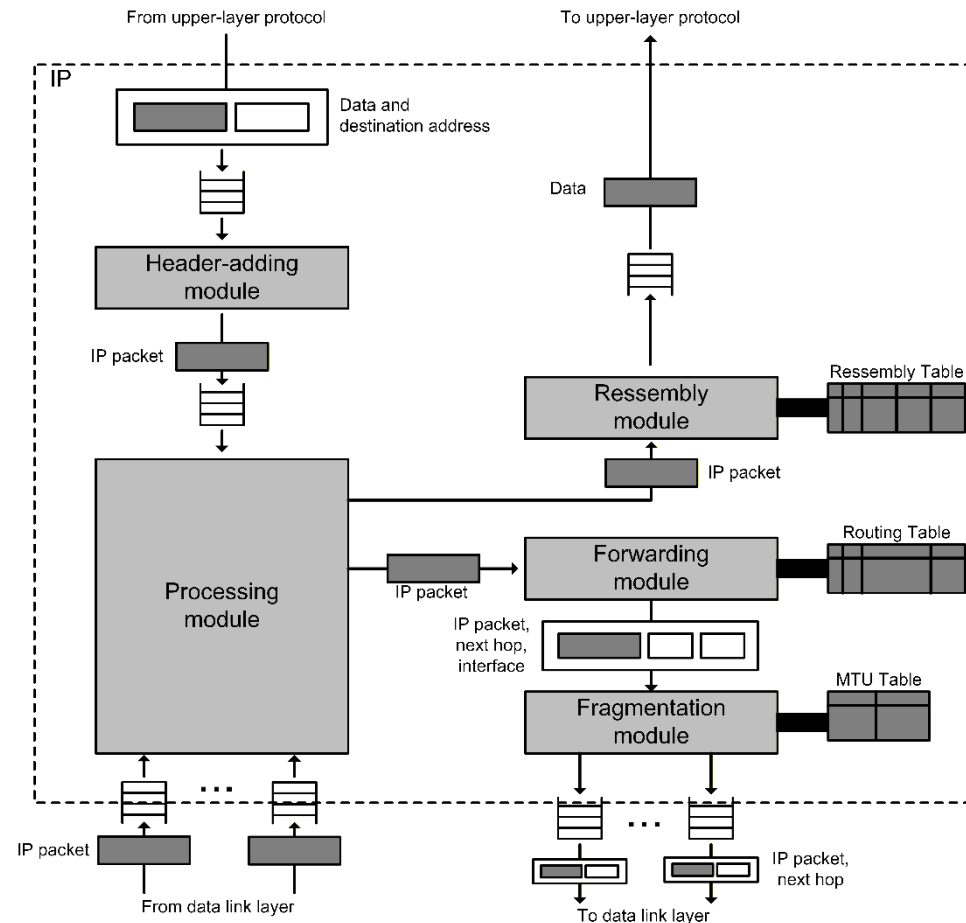
- IP 패키지

- 정의

- IP 패킷을 다음 계층으로 전달하기 위한 처리 과정 전체

- 구성요소

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



IPv4

• IP 패키지 구성요소 (1/10)

• 큐(Queue)

• 정의

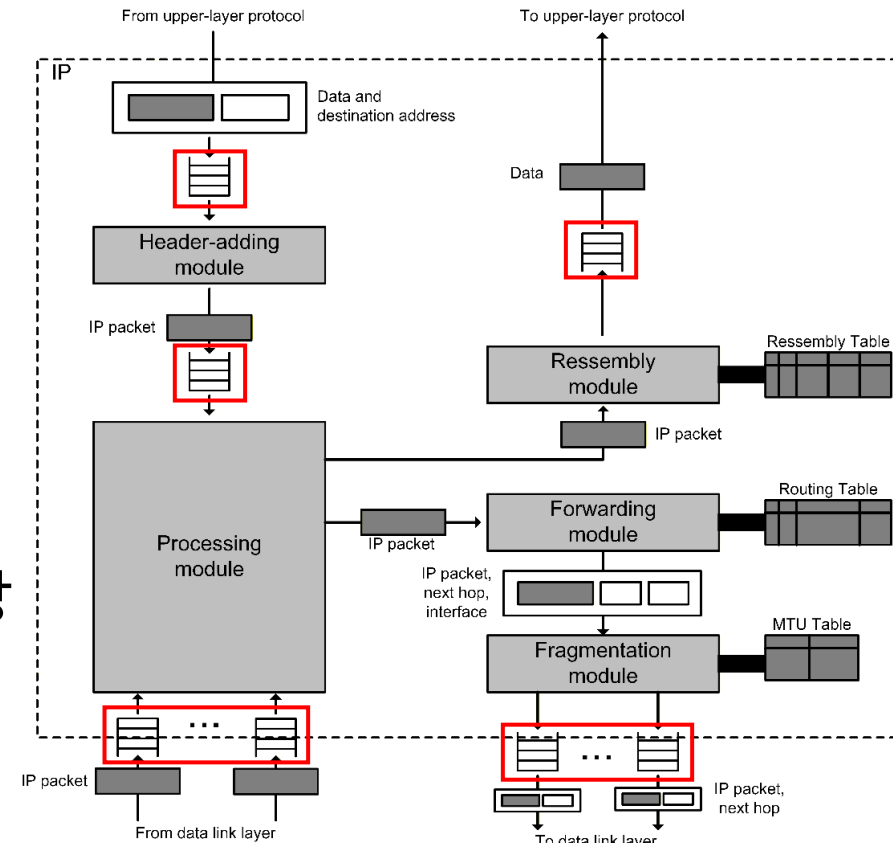
- 먼저 들어온 데이터가 먼저 나가는 선입선출 구조

• 입력 큐

- 데이터링크 계층이나 상위 계층 프로토콜로부터 온 데이터그램 저장
- 처리 모듈은 입력 큐로부터 데이터그램을 가져옴

• 출력 큐

- 데이터링크 계층이나 상위 계층 프로토콜로 가는 데이터그램 저장
- 단편화 모듈과 재조립 모듈은 출력 큐에 데이터그램을 넣음



IPv4

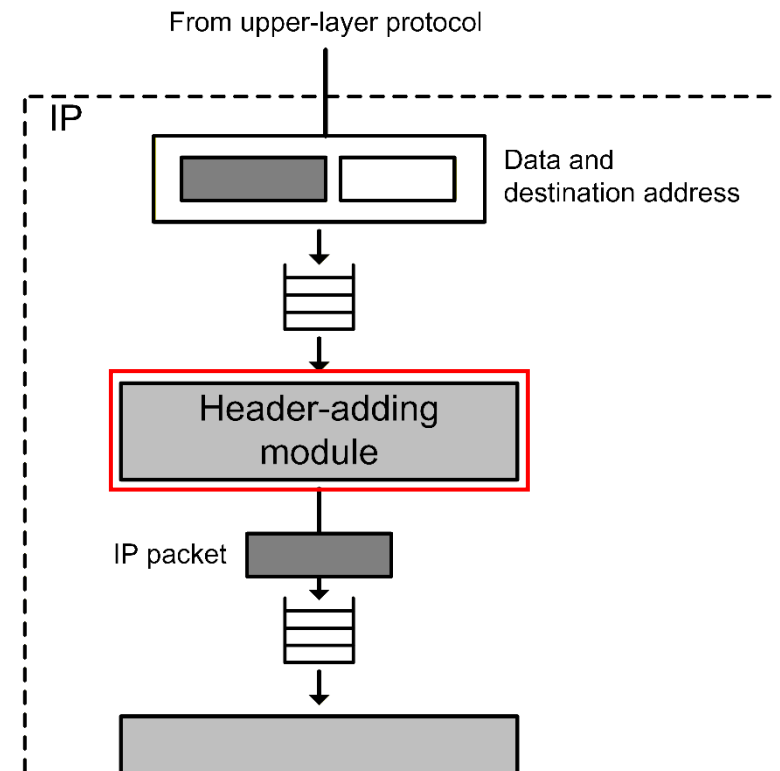
- IP 패키지 구성요소 (2/10)

- 헤더 추가 모듈(Header-adding Module)

- 동작 원리

1. 상위 계층 프로토콜로부터 데이터와 목적지 IP 주소를 받음
2. IP 데이터그램 내에 데이터를 캡슐화
3. 체크섬을 계산하고 체크섬 필드에 추가
4. 목적에 맞는 큐에 완성된 IP 데이터그램을 전달

```
IP_Adding_Module (data, destination)
{
    Encapsulate data in an IP datagram
    Calculate checksum and insert it in the checksum field
    Send data to the corresponding queue
    Return
}
```



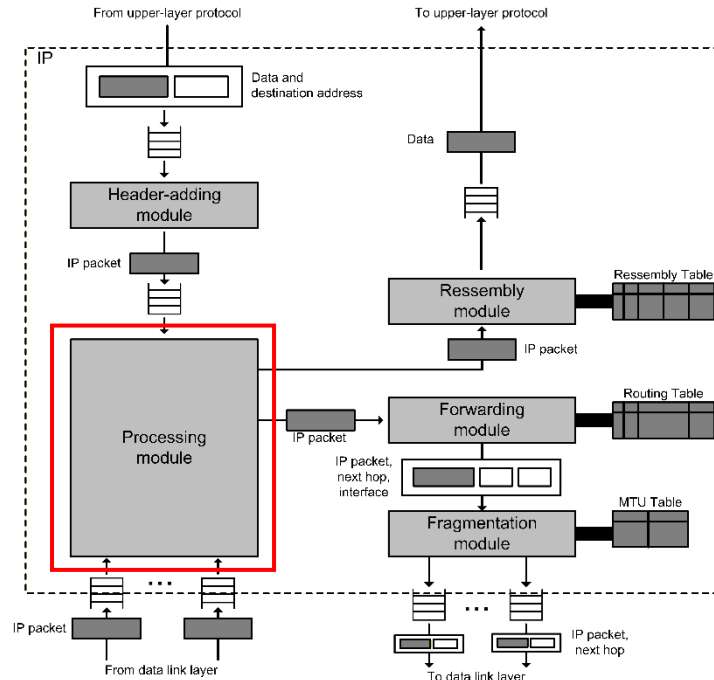
IPv4

• IP 패키지 구성요소 (3/10)

• 처리 모듈(Processing Module) (1/2)

• 동작 원리

1. 입력 큐에서 하나의 데이터그램 제거
2. 목적지 주소가 로컬 주소와 일치하는지 확인
 - 일치하면 재조립 모듈로 전달 후 종료



IP_Processing_Module (Datagram)

```
{  
    Remove one datagram from one of the input queues  
    if (destination address matches a local address) {  
        send the datagram to the reassembly module  
        Return  
    }  
    if (machine is a router) {  
        Decrement TTL  
    }  
    if (TTL less than or equal to zero) {  
        Discard the datagram  
        Send an ICMP error message  
        Return  
    }  
    Send the datagram to the forwarding module.  
    Return  
}
```

IPv4

- IP 패키지 구성요소 (4/10)

- 처리 모듈(Processing Module) (2/2)

- 동작 원리

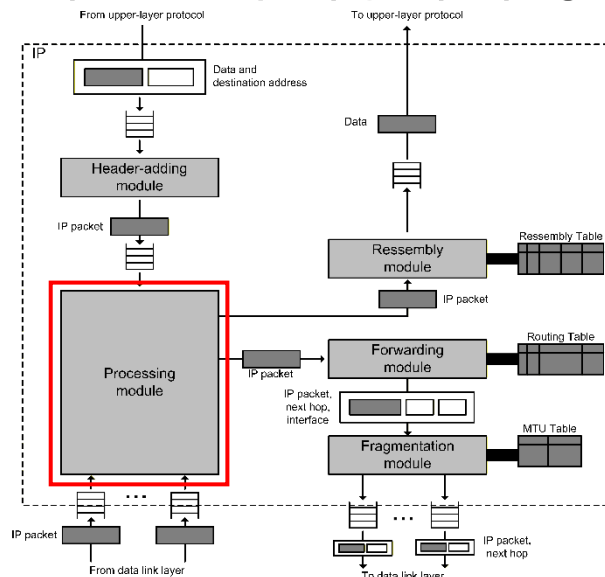
- 3. 현재 장치가 라우터이면

- TTL 값 1 감소

- 4. TTL 값이 0 이하인지 확인

- 0 이하이면 데이터그램 폐기 하고 ICMP 오류 메시지 전송 후 종료

- 5. 위 조건에 해당하지 않으면 포워딩 모듈로 데이터그램 전달 후 종료



IP_Processing_Module (Datagram)

```
{  
  Remove one datagram from one of the input queues  
  if (destination address matches a local address) {  
    send the datagram to the reassembly module  
    Return  
  }  
  if (machine is a router) {  
    Decrement TTL  
  }  
  if (TTL less than or equal to zero) {  
    Discard the datagram  
    Send an ICMP error message  
    Return  
  }  
  Send the datagram to the forwarding module.  
  Return  
}
```

IPv4

- IP 패키지 구성요소 (5/10)

- 라우팅 테이블(Routing Table)

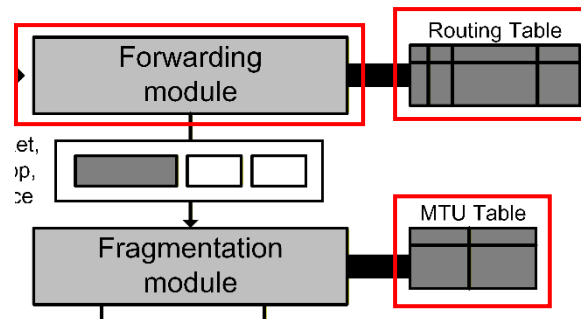
- 패킷의 다음 홉 주소를 결정하기 위해 포워딩 모듈에서 사용

- 포워딩 모듈(Forwarding Module)

- 처리 모듈로부터 전달되어야 하는 IP 패킷을 받아 처리
 - 보내져야 할 노드의 주소와 패킷이 보내져야 하는 인터페이스의 번호를 찾아서 패킷과 함께 단편화 모듈로 전송

- MTU 테이블(MTU Table)

- 단편화 모듈이 특정 인터페이스의 MTU를 찾기 위해 사용
- 인터페이스와 MTU의 두 열로 구성



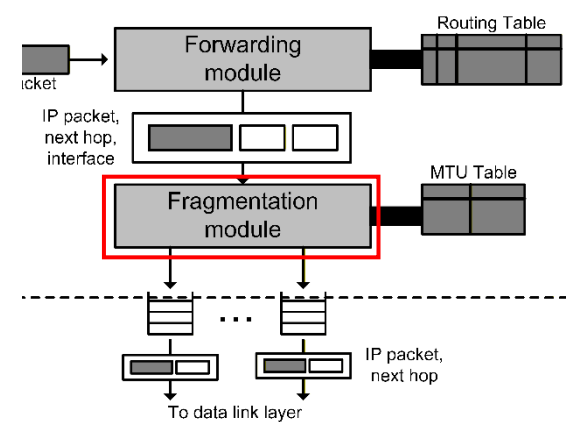
IPv4

• IP 패키지 구성요소 (6/10)

• 단편화 모듈(Fragmentation Module)

• 동작 원리

1. 포워딩 모듈로부터 IP 데이터그램을 받음
2. 데이터그램의 크기 추출
3. MTU보다 크기가 큰 경우
 - D 비트가 설정된 경우
 - 데이터그램 폐기 및 ICMP 오류 메시지 전달 후 종료
 - D 비트가 설정되지 않은 경우
 - 최대 사이즈 계산하고 단편으로 분할
 - 각 단편에 헤더와 각 단편에 요구되는 옵션 추가
 - 단편 전달 후 종료
4. MTU보다 작은 경우
 - 데이터그램 전송 후 종료



```
IP_Fragmentation_Module (datagram)
{
    Extract the size of datagram
    if (size > MTU of the corresponding network) {
        if (D bit is set) {
            Discard datagram
            Send an ICMP error message
            Return
        }
        else {
            Calculate maximum size
            Divide the segment into fragments
            Add header to each fragment
            Add required options to each fragment
            Send fragment
            Return
        }
    }
    else {
        Send the datagram
    }
    Return
}
```

IPv4

• IP 패키지 구성요소 (7/10)

• 재조립 테이블(Reassembly Table)

- 재조립 모듈에 의하여 사용
- 필드

- 상태(State)

- 재조립 테이블의 상태를 의미
- 사용 중일 경우 IN-USE, 사용하지 않을 경우 FREE

- 발신지 주소(Source Address)

- 데이터그램의 발신지 주소

- 데이터그램 ID (Datagram ID)

- 데이터그램과 이 데이터그램에 속하는 단편들을 정의하는 번호

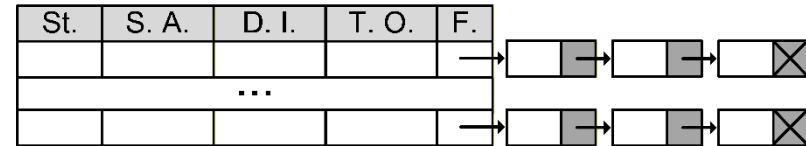
- 타임 아웃(Time-out)

- 미리 결정된 시간으로 모든 단편이 이 시간 내에는 도착

- 단편(Fragments)

- 단편들의 연결 리스트에 대한 포인터

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



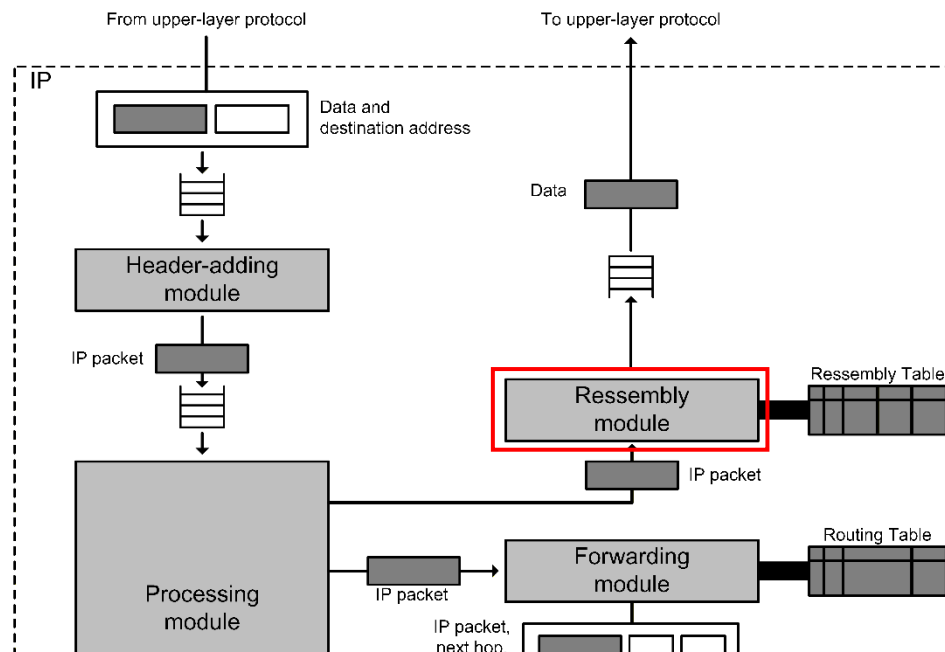
IPv4

• IP 패키지 구성요소 (8/10)

• 재조립 모듈(Reassembly Module) (1/3)

• 동작 원리

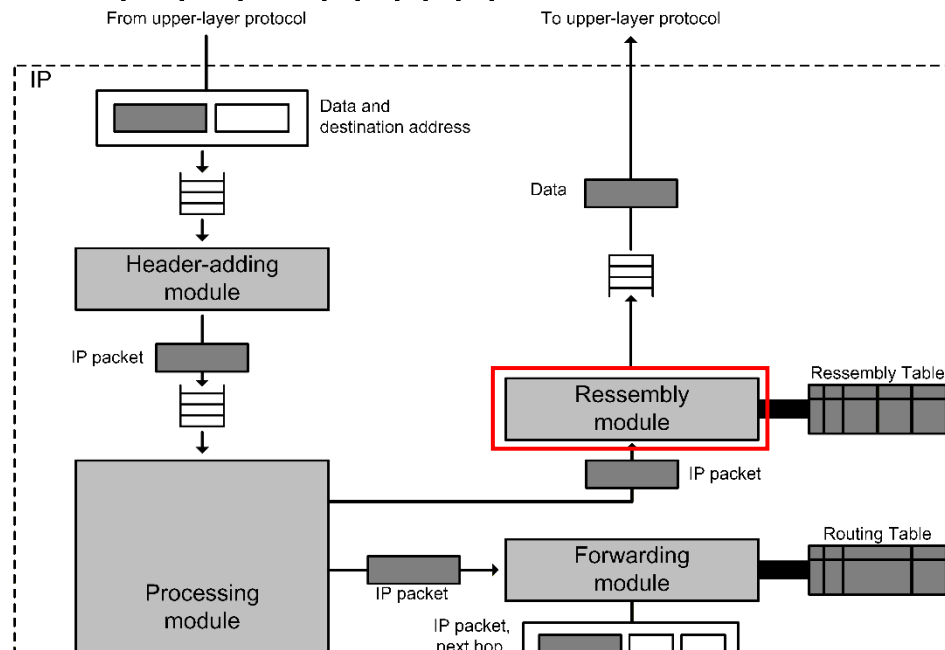
1. 처리 모듈로부터 최종 목적지에 도착한 데이터그램 단편을 받음
2. 단편의 오프셋 값이 0이면서 M비트가 0인지 확인
 - 두 값이 0이면 데이터그램을 적절한 큐에 보내고 종료



```
IP_Reassembly_Module (datagram)
{
    if (offset value = 0 AND M = 0) {
        Send datagram to the appropriate queue
        Return
    }
    Search the reassembly table for the entry
    if (entry not found) {
        Create a new entry
    }
    Insert datagram into the linked list
    if (all fragments have arrived) {
        Reassemble the fragment
        Deliver the fragment to upper-layer protocol
        Return
    }
    else{
        if (time-out expired) {
            Discard all fragments
            Send an ICMP error message
        }
    }
    Return
}
```

IPv4

- IP 패키지 구성요소 (9/10)
- 재조립 모듈(Reassembly Module) (2/3)
 - 동작 원리
- 3. 재조립 테이블에서 항목에 대해 검색
 - 해당 항목을 찾지 못하면 새로운 항목을 생성
- 4. 생성한 항목의 연결리스트에 데이터그램 삽입



```
IP_Reassembly_Module (datagram)
{
    if (offset value = 0 AND M = 0) {
        Send datagram to the appropriate queue
        Return
    }
    Search the reassembly table for the entry
    if (entry not found) {
        Create a new entry
    }
    Insert datagram into the linked list
    if (all fragments have arrived) {
        Reassemble the fragment
        Deliver the fragment to upper-layer protocol
        Return
    }
    else{
        if (time-out expired) {
            Discard all fragments
            Send an ICMP error message
        }
    }
    Return
}
```

IPv4

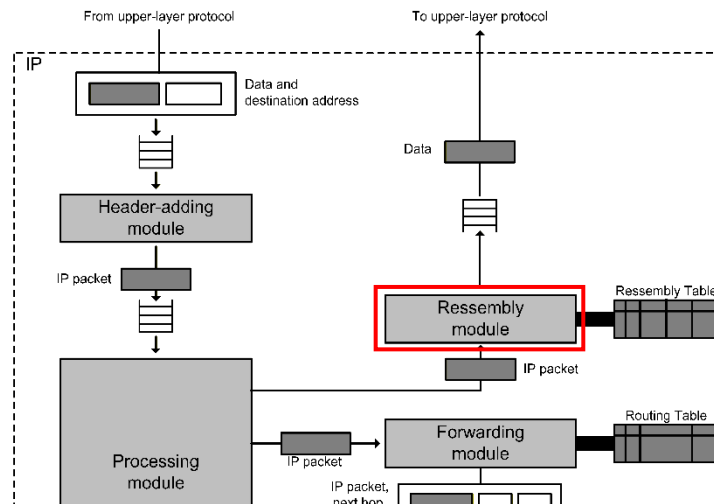
• IP 패키지 구성요소 (10/10)

• 재조립 모듈(Reassembly Module) (3/3)

• 동작 원리

5. 모든 단편이 도착했는지 확인

- 모든 단편이 도착했으면 단편을 재조립하고 상위 계층 프로토콜에 단편을 전송하고 종료
- 단편들이 모두 도착하지 않았다면, 타임아웃이 발생했는지 확인
 - 타임아웃이 발생했다면 모든 단편을 폐기하고 ICMP 오류 메시지를 전송하고 종료



```
IP_Reassembly_Module (datagram)
{
    if (offset value = 0 AND M = 0) {
        Send datagram to the appropriate queue
        Return
    }
    Search the reassembly table for the entry
    if (entry not found) {
        Create a new entry
    }
    Insert datagram into the linked list
    if (all fragments have arrived) {
        Reassemble the fragment
        Deliver the fragment to upper-layer protocol
        Return
    }
    else{
        if (time-out expired) {
            Discard all fragments
            Send an ICMP error message
        }
    }
    Return
}
```

Thanks!

서진우(jinwoo@pel.sejong.ac.kr)