

TCP/IP

– Internet Protocol

Internet Protocol





Overview

❖ 연결 유형

▶ Connection - oriented

- ↳ 송수신을 위한 연결통로를 만들고 데이터를 전송
- ↳ 각 패킷에 대해 라우터는 경로를 다시 계산할 필요가 없다.
- ↳ 하나의 메시지에 속하는 모든 패킷이 전달된 후 연결은 종료된다.

▶ Connectionless

- ↳ 각 패킷은 상호 독립적으로 취급하며 패킷들 사이에 아무 관계가 없다.
 - ↳ 같은 목적지로 전달됨에도 불구하고 서로 다른 경로를 통하여 전달 될 수 있다.
-
- ▶ IP는 비연결형 프로토콜로서 비연결형 서비스를 제공한다



Internetworking issues (1/4)

❖ Network service

- ▶ LAN은 일반적으로 connectionless service를 제공하고 WAN은 connection-oriented service를 제공하는데, 이러한 LAN과 WAN이 서로 연결되는 인터넷에서는 서비스를 연결형으로 제공할 것인가 아니면 비연결형으로 제공할 것인가를 결정해야 함

❖ Protocol

- ▶ 각 서브 네트마다 사용하는 프로토콜 구조가 다른데, 인터넷에서는 각 서브 네트의 고유한 프로토콜을 유지하면서도 서로 통신할 수 있도록 해주는 연결 방법 필요



Internetworking issues (2/4)

❖ Addressing

- ▶ 각 sub network마다 사용하는 주소 구조가 다르므로 이러한 기존의 주소 체계는 인터넷에서는 사용 곤란하므로 제 3의 공통 주소 구조 필요
- ✖ Ethernet을 통해 인터넷에 접속된 노드: (48bit의 Ethernet Hardware 주소 + 32 bit IP 주소)



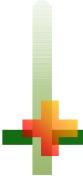
Internetworking issues (3/4)

❖ Routing

- ▶ 각 서브넷마다 **routing** 기법이 다른데, 이러한 서브넷이 서로 연결된 인터넷에서 어떻게 목적지까지 정확하게 패킷을 전달할 수 있는가에 대한 고려
 - ↳ 각 서브넷의 고유한 **routing** 기법은 그대로 유지해야 한다.
 - ↳ 서브넷 내에서는 고유한 물리주소를 사용하여 **routing**
 - ↳ 망과 망을 연결하는 라우터에서는 IP 주소를 사용하여 **routing**

❖ Quality of Service

- ▶ 자신이 접속된 서브넷에게 자신이 원하는 **QoS**(원하는 지연시간 한계, 비용 한계, 우선순위, 보안 수준, 허용 에러율)를 요청하고 각 서브넷은 종단시스템의 요구 사항을 재 조정하는 기능이 있음
- ▶ 다양한 **QoS** 방식을 갖는 서브넷이 연결된 인터넷에서 종단 시스템은 자신에게 허용될 수 있는 **QoS**를 알 수 있고 또한 지정할 수 있어야 한다.



Internetworking issues (4/4)

- ❖ Maximum packet size

- ▶ 각 서브넷마다 지원하는 패킷의 최대 길이가 다르므로 인터넷에서 패킷이 각 서브넷을 경유할 때 중간 노드에서는 **segmentation & reassembly** 과정이 필요

- ❖ Flow control & Congestion control

- ▶ **flow control:** 서로 다른 서브넷에 접속된 두 노드 사이에도 송신부의 전송 속도가 수신부의 처리 속도를 초과하지 않도록 하는 흐름제어 기법 필요
 - ▶ **congestion control:** 인터넷을 구성하는 특정 서브넷에 패킷이 너무 많이 몰리는 현상이 발생하지 않도록 해야 함.

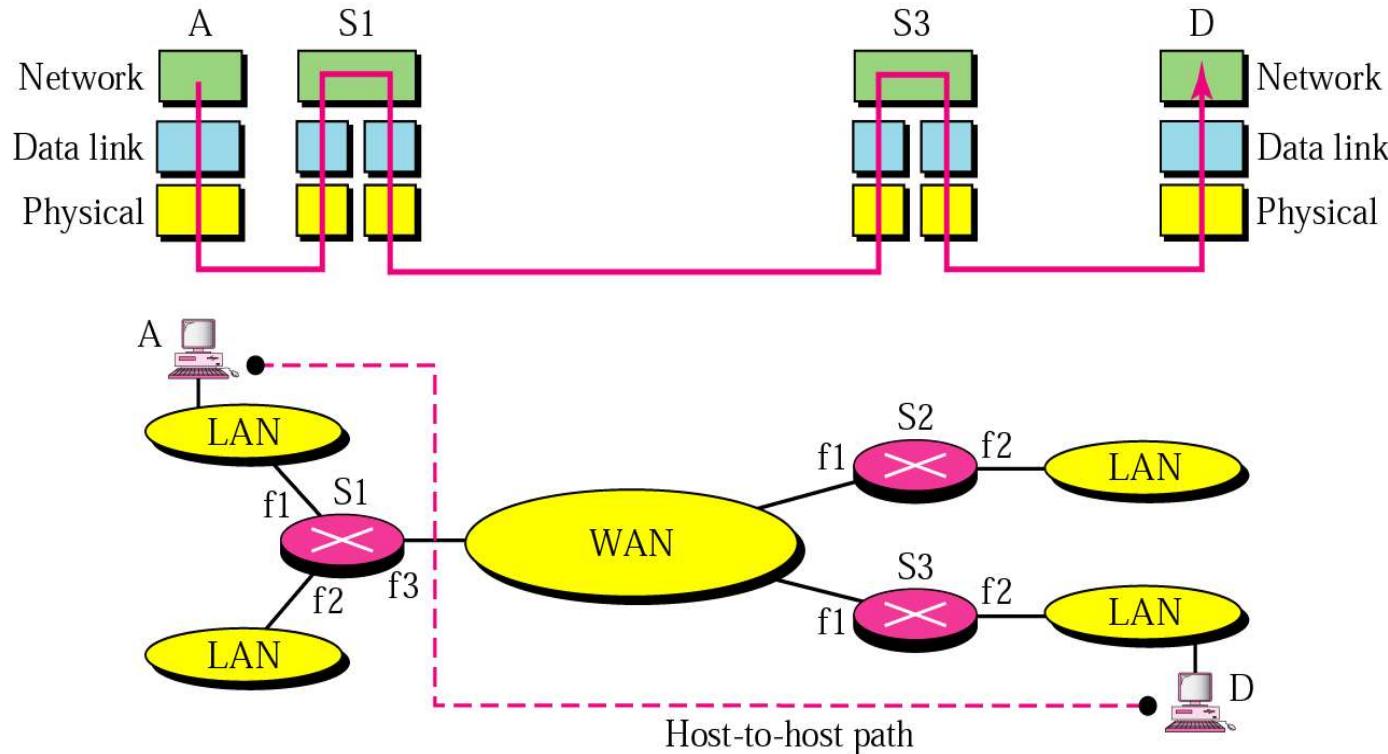
- ❖ Error reporting

- ▶ 각 서브넷마다 **error reporting**하는 방식이 다른데, 여러 개의 서브넷이 연결된 인터넷 상에서 공통된 **error reporting** 기법 필요(Ex: ICMP)

Internetworking (1/4)

❖ 네트워크 층의 필요성

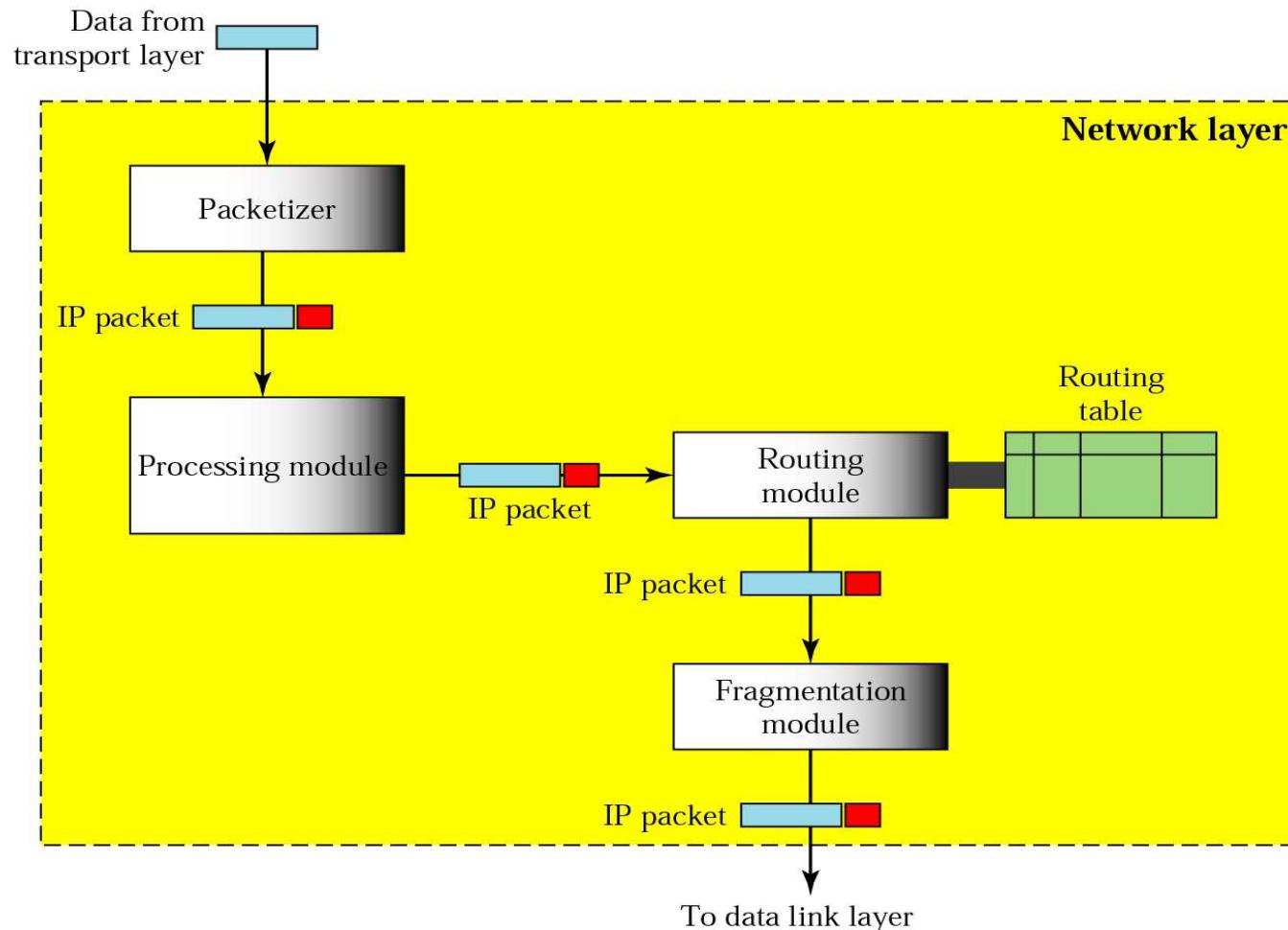
- ▶ 여러 링크를 통한 전달 문제 해결
- ▶ 호스트 대 호스트 전송 책임
- ▶ 라우터 또는 교환기를 통한 패킷 라우팅 책임



<Fig. 네트워크 계층의 필요성>

Internetworking (2/4)

❖ 네트워크 계층에서의 역할 - 출발 노드



<Fig. 출발지 노드의 Network Layer>



Packetizer Module

- ❖ 헤더 추가 모듈

- ▶ 상위 계층으로 부터 받은 데이터와 목적지 IP 주소를 받은 뒤 IP헤더를 추가하여 IP 데이터그램 내에 캡슐화 한다.

Receive: data, destination address

- 1. Encapsulate the data in an IP datagram.**
- 2. Calculate the checksum and insert it in the checksum field**
- 3. Send the data to the corresponding queue.**
- 4. Return.**



Processing Module

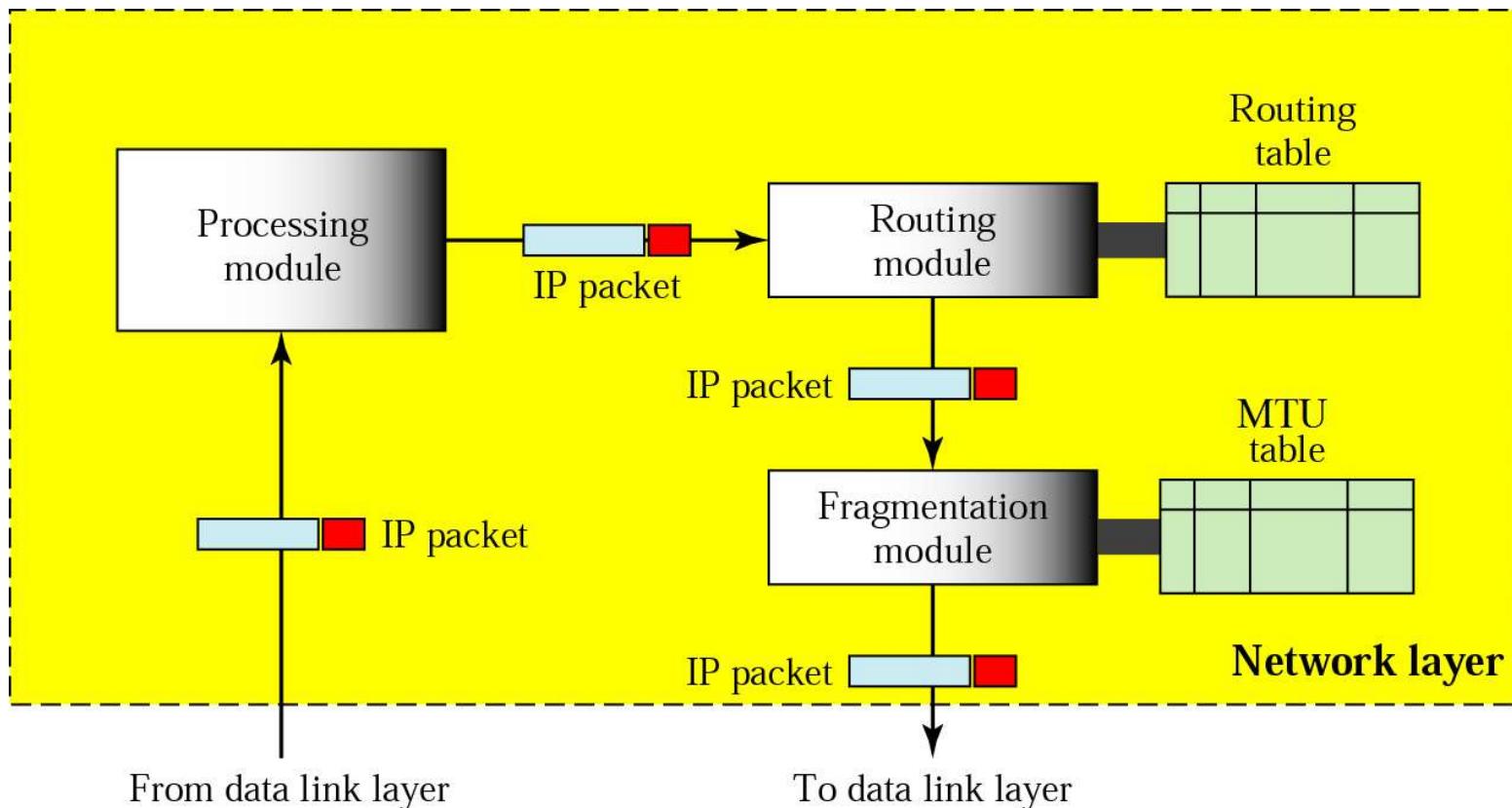
❖ 처리 모듈

- ▶ IP패키지의 핵심으로 먼저 목적지 주소가 루프팩 패킷인지 혹은 최종 목적지에 도착하였는지 검사한다. 이럴 경우 재조립 모듈로 보내진다.
- ▶ 만약 노드가 라우터라면 TTL을 1 감소 시킨다. 이 값이 0 이하면 데이터 그램은 폐기된다.

- 1. Remove datagram from one of the input queues.**
- 2. If(destination address is 127.X.Y.Z or matches one of the local addresses)**
 - 1. Send the datagram to the reassembly module.**
 - 2. Return.**
- 3. If(machine is a router)**
 - 1. Decrement TTL**
- 4. If(TTL less than or equal to zero)**
 - 1. Discard the datagram**
 - 2. Send an ICMP error message**
 - 3. Return**
- 5. Send the datagram to the forwarding module.**
- 6. Return**

Internetworking (3/4)

❖ 네트워크 계층에서의 역할 - 중간 노드



<Fig. 중간 노드의 Network Layer>



Fragmentation Module

❖ 단편화 모듈

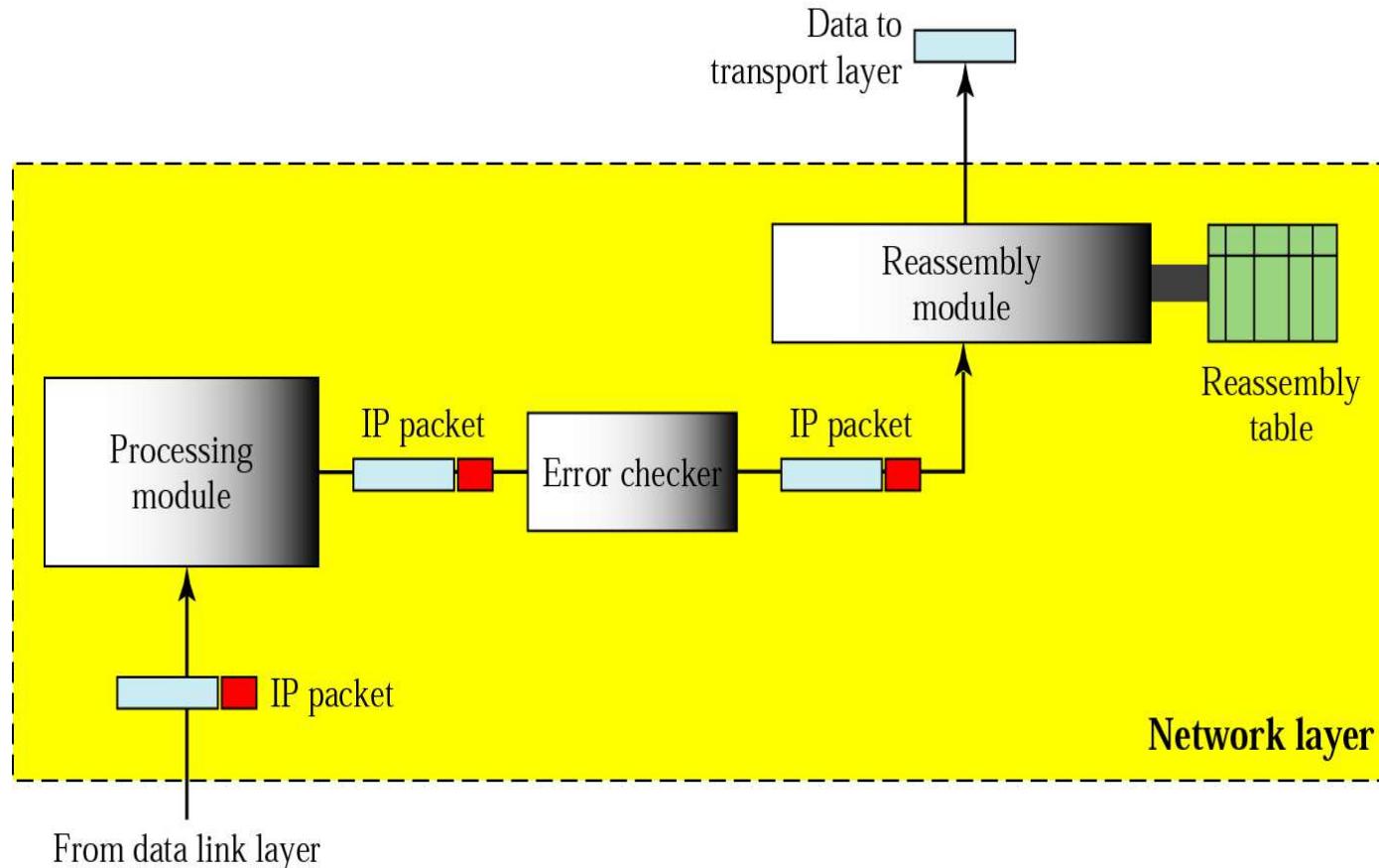
- ▶ 단편화 모듈은 MTU 테이블을 참조하여 해당하는 인터페이스 번호의 MTU를 찾는다. 만약 데이터 그램의 길이가 더 크다면 단편화하고 각 단편에 헤더를 붙인 후 주소 해석과 전달을 위하여 ARP 패키지에 보낸다.

Receive: an IP packet from routing module

- 1. Extract the size of the datagram.**
- 2. If(size>MTU of the corresponding network)**
 - 1. If[D(do not fragment) bit is set]**
 - 1. Discard the datagram**
 - 2. Send an ICMP error message(see Chapter 9)**
 - 3. Return**
 - 2. Else**
 - 1. Calculate the maximum size**
 - 2. Divide the datagram into fragments**
 - 3. Add header to each fragment**
 - 4. Add required options to each fragment**
 - 5. Send the datagrams**
 - 6. Return**
 - 3. Else**
 - 1. Send the datagram**
 - 4. Return**

Internetworking (1/4)

❖ 네트워크 계층에서의 역할 - 목적지 노드



<Fig. 목적지 노드의 Network Layer>

Reassembly Module

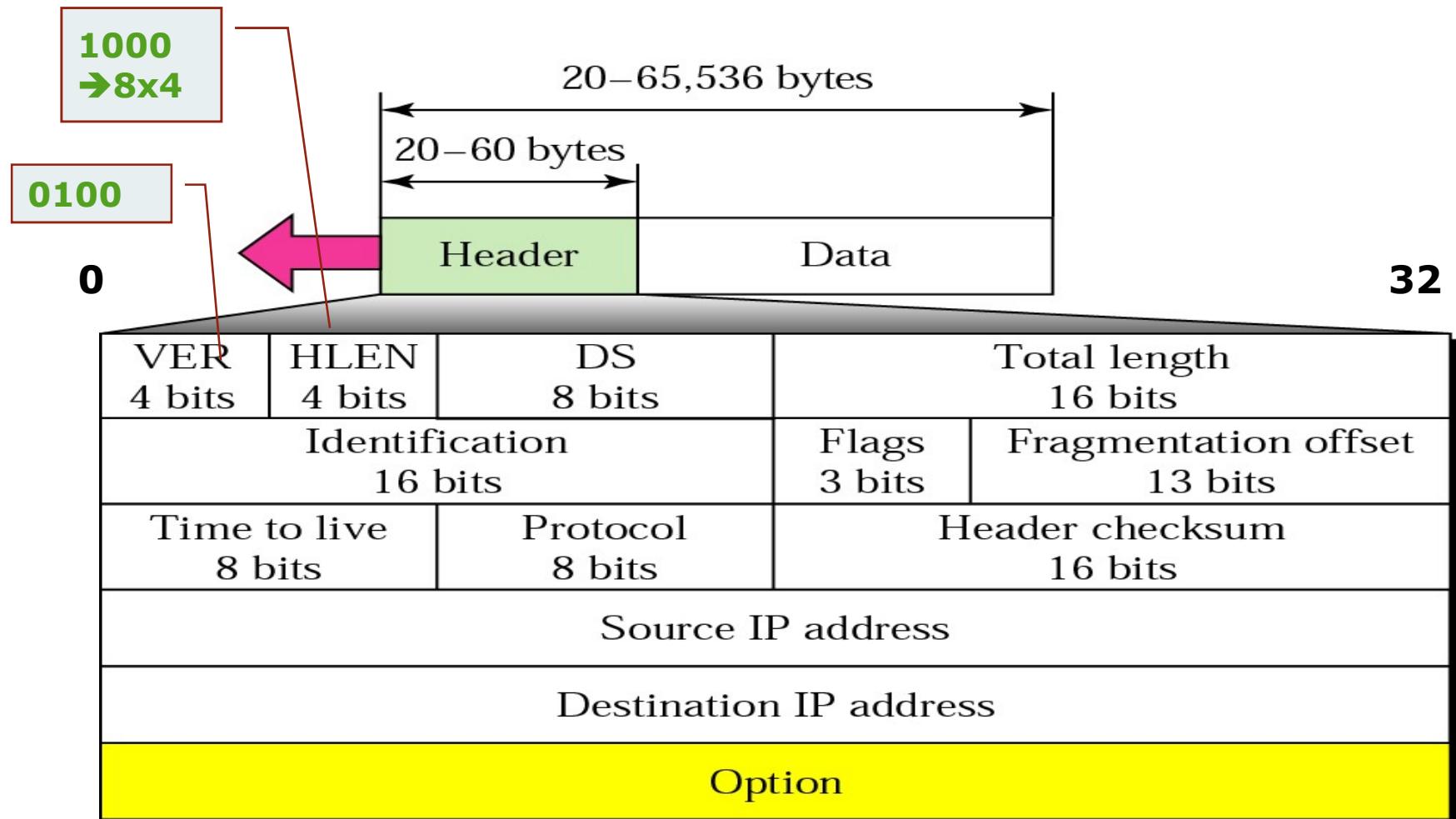
❖ 재조립 모듈

- ▶ 재조립 테이블을 사용하여 단편들을 재조립한다.
- ▶ 재조립 모듈은 단편이 속한 데이터그램을 찾고 같은 데이터그램에 속한 단편의 순서를 맞추고 모든 단편이 다 들어온 후 재조립 하는 것이다.
- ▶ 타임아웃이 만료 되었는데도 아직 들어오지 않는 단편이 있으면 모듈의 모든 단편을 폐기 한다.

Receive: an IP datagram from the processing module

1. **If(offset value is zero and the M bit is 0)**
 1. Send the datagram to the appropriate queue.
 2. Retrun.
2. Serarch the reassembly table for the corresponding entry.
3. **If(not found)**
 1. Create a new entry.
4. **Insert the fragment at the appropriate place in the linked list.**
 1. **If(all fragments have arrived)**
 1. Reassemble the fragments.
 2. Deliver the datagram to the corresponding upper layer protocol.
 3. Return.
 2. **Else**
 1. Check the time-out.
 2. **If(time-out expored)**
 1. check the time-out

IP 데이터그램 구조 (1/6)





IP 데이터그램 구조 (2/6)

❖ 헤더 내의 필드

- ▶ VER: IP 버전 정의
- ▶ HLEN: 헤더의 전체 길이를 4 바이트 단위로 표시(4비트)
- ▶ DS: 차별 서비스(differentiated services)
 - ↳ 서비스 품질(QoS)을 목적으로 데이터그램 등급을 정의
 - ↳ DSCP(6bit)-RFC 2474, ECN(2bit)-RFC 3168
- ▶ total length
 - ↳ IP 데이터그램의 총 길이(헤더+ 데이터)를 바이트 단위로 정의
- ▶ Identification, Flags, Fragmentation offset
 - ↳ 단편화에 사용

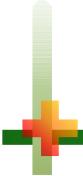


IP 데이터그램 구조 (3/6)

❖ 헤더 내의 필드 (Cont.)

▶ 수명(TTL; Time to Alive)

- ↳ 최대 흡(라우터)의 수를 제어
- ↳ 발신지 호스트가 데이터그램을 송신할 때,
숫자 하나(두 호스트 간의 최대 경로 수의 약 2배)를 저장
- ↳ 각 라우터는 이 값을 1씩 감소
- ↳ 이 값이 0이 될 때까지 라우팅을 수행, 0이 되면, 라우터는 데이터그램을 폐기
- ↳ IP_TTL, IP_MULTICAST_TTL 소켓 옵션으로 변경할 수 있다.



IP 데이터그램 구조 (4/6)

❖ 헤더 내의 필드 (Cont.)

▶ 프로토콜(protocol)

- ↳ IP층의 서비스를 사용하는 상위 계층 프로토콜(TCP, UDP, ICMP, IGMP) 정의
- ↳ IP 데이터그램이 전달되어야 하는 최종 목적지의 프로토콜 기술

value	protocol
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

IP 데이터그램 구조 (5/6)

❖ 헤더 내의 필드 (Cont.)

▶ checksum

- ↳ 데이터가 아닌 오직 헤더 부분에 대한 것만 검사
- ↳ 상위 계층의 프로토콜은 전체 패킷을 검사하기 위한 checksum을 소유
- ↳ IP 패킷의 헤더는 각 라우터를 지날 때마다 변경되지만, 데이터는 변하지 않음

4	5	0	28			
		1	0 0			
4	17	0				
10.12.14.5						
12.6.7.9						
4, 5, and 0 → 0100010100000000						
28 → 00000000000011100						
1 → 00000000000000001						
0 and 0 → 00000000000000000						
4 and 17 → 0000010000010001						
0 → 00000000000000000						
10.12 → 0000101000001100						
14.5 → 0000111000000101						
12.6 → 0000110000000110						
7.9 → 0000011100001001						
Sum → 0111010001001110						
Checksum → 1000101110110001						



IP 데이터그램 구조 (6/6)

- ❖ 헤더 내의 필드 (Cont.)
 - ▶ 발신지 주소(source address)
 - ↳ 발신지 IP 주소를 정의
 - ▶ 목적지 주소(destination address)
 - ↳ 목적지 IP 주소를 정의
 - ▶ 옵션(option)
 - ↳ 시험이나 오류 제거를 위해 사용

IP 단편화 (1/4)

❖ 단편화(fragmentation)

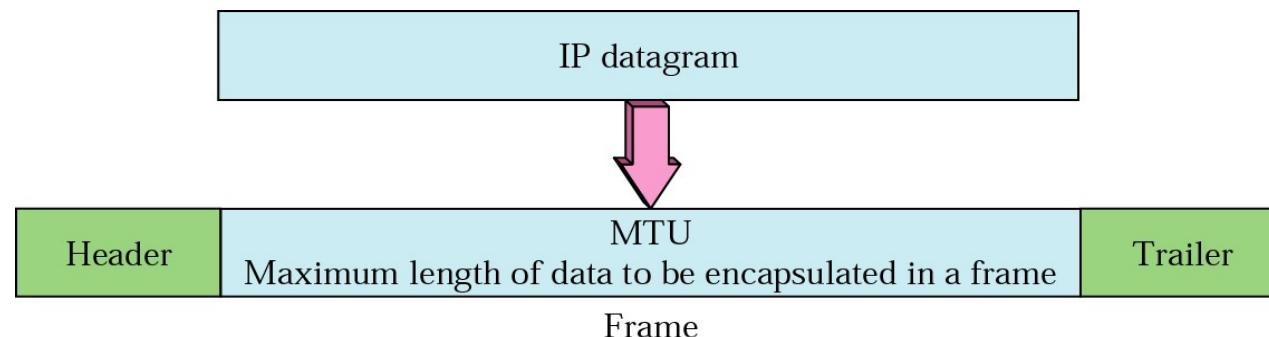
▶ 요구사항

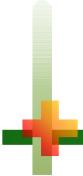
- ↳ 데이터그램은 여러 다른 네트워크를 통해 전송
- ↳ 네트워크가 사용하는 프로토콜에 따라 프레임 형식과 크기가 서로 다름
- ↳ 라우터는 프레임을 역캡슐화하고, 처리한 다음 다시 캡슐화함

▶ 최대 전송 단위(MTU; Maximum Transfer Unit)

- ↳ 각 네트워크에서 전달되는 최대 전송 길이로 망마다 틀리다.
- ↳ 데이터그램이 프레임으로 캡슐화될 때의 최대 길이
- ↳ 최대 길이는 **65,535 바이트**로 정의

↳ MTU





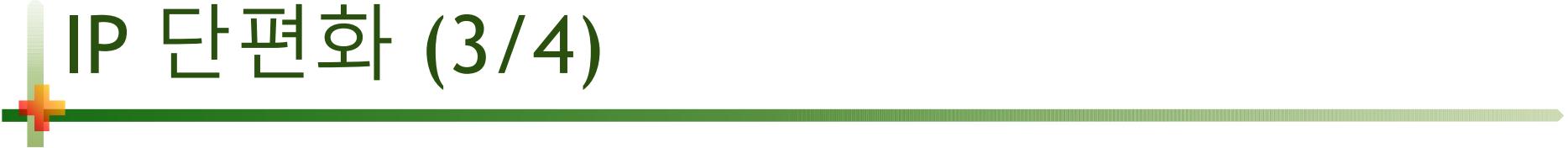
IP 단편화 (2/4)

❖ 단편화(fragmentation) (Cont.)

- ▶ 물리적인 네트워크들을 위해서, 네트워크를 통과할 수 있도록 데이터그램을 쪼개야 함
- ▶ 단편(fragment)은 더 작은 MTU 네트워크를 만나면, 다시 단편화 됨
- ▶ 데이터그램은 경로상에 있는 어떤 라우터에서도 단편화될 수 있음
- ▶ 데이터그램의 재조립은 오직 목적지 호스트에서만 가능

Protocol	MTU
Hyperchannel	65,535
Token Ring(16Mbps/4Mbps)	17,914/4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

<Table. 여러 종류 네트워크들의 MTU>



IP 단편화 (3/4)

❖ 단편화 요소 값

▶ 식별자(identification)

- ↳ 발신지 호스트로부터 생성된 데이터그램을 식별
- ↳ 데이터그램의 모든 단편들은 식별번호(id)를 전송

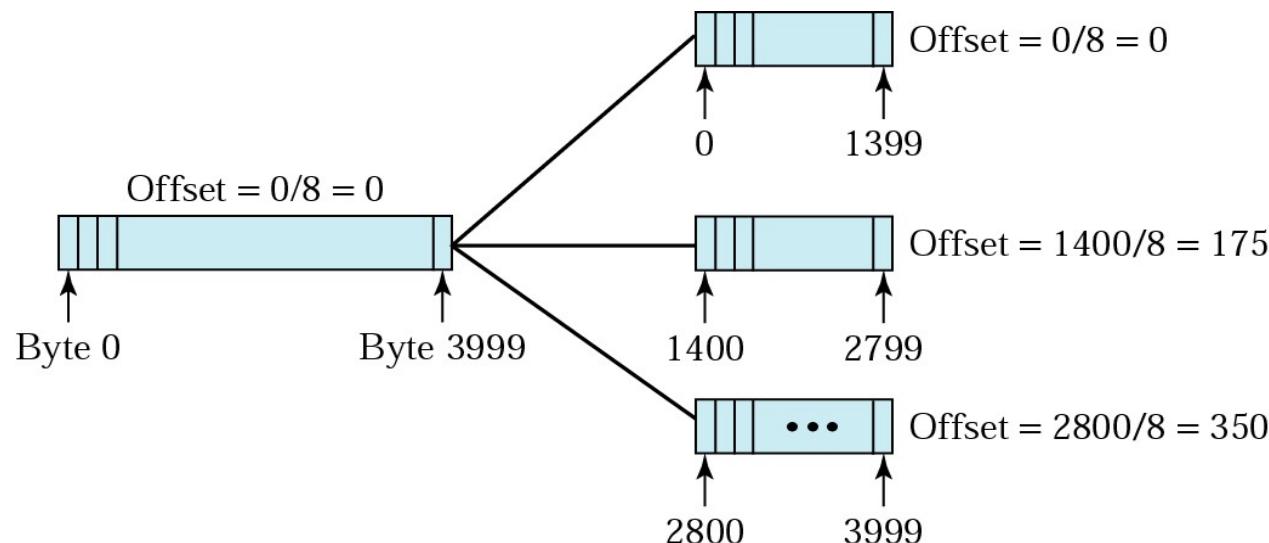
▶ 플래그(flags)

- ↳ 3 비트로 구성되며, 첫번째 비트는 예비임
- ↳ 두번째 비트는 데이터그램을 단편화하지 말라는 비트임
 - ▶ 만약 비트가 1인 경우, 단편화 불가능
 - ▶ 만약 비트가 0인 경우, 단편화 가능
- ↳ 세번째 비트는 추가적인 단편임
 - ▶ 만약 비트가 1인 경우, 마지막 단편이 아님을 의미
 - ▶ 만약 비트가 0인 경우, 마지막 단편이거나 단편이 오직 하나임을 의미

IP 단편화 (4/4)

❖ 단편화 옵셋(fragmentation offset)

- ▶ 13 비트 필드
- ▶ 전체 데이터그램에서 단편의 상대적인 위치
- ▶ 원래 데이터그램에서 8 바이트로 계산한 데이터의 옵셋
- ▶ 단편화 옵셋의 예 (MTU == 1400)



<Fig. 단편화의 예>

주소 지정 (1/11)

❖ 인터넷 주소

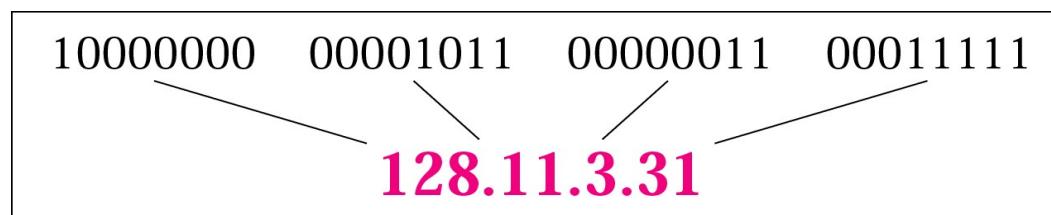
- ▶ 호스트간 전달을 위한 주소, 라우팅 방식 필요
- ▶ 모든 장치에 전세계적으로 통신이 가능한 장치들의 유일한 식별 방법으로 식별자는 인터넷 주소 또는 IP 주소로 호칭
- ▶ 호스트, 라우터 연결을 유일하고 전체적으로 정의하는 32비트 2진 주소

❖ 2진 표기법

- ▶ 주소를 읽기 쉽게 옥텟(8비트) 사이에 공간 삽입
- ▶ 32비트 주소, 4옥텟 주소, 4바이트 주소

❖ 점-10진 표

01110101 10010101 00011101 11101010

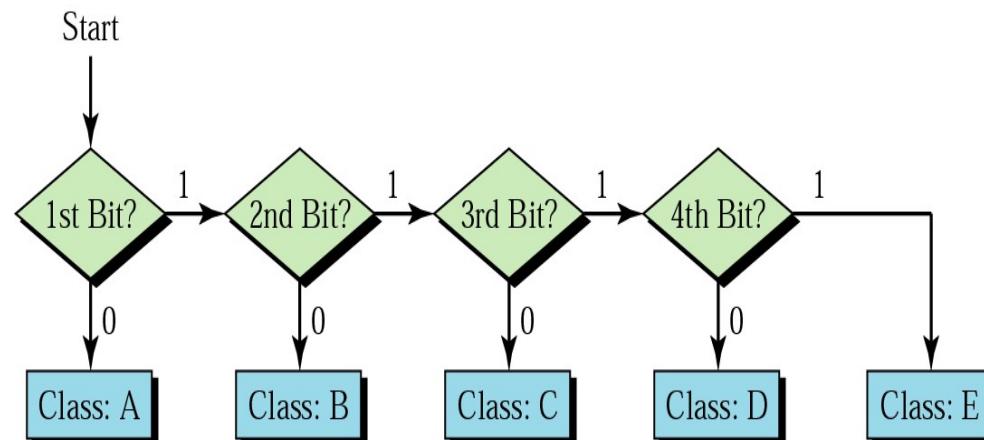


주소 지정 (2/11)

❖ 클래스 구분 분류

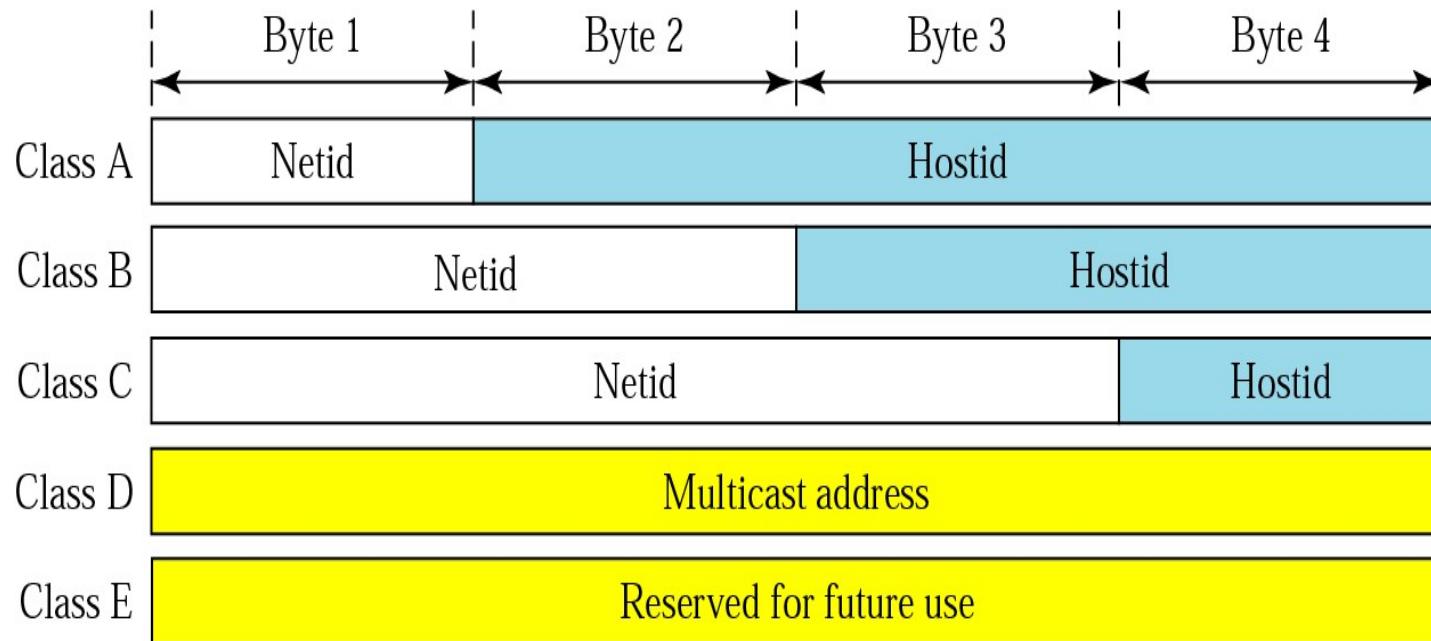
▶ 2진 표기법에서 클래스 구별, 10진 표기법 구별

	First byte	Second byte	Third byte	Fourth byte	
Class A	0				0 to 127
Class B	10				128 to 191
Class C	110				192 to 223
Class D	1110				224 to 239
Class E	1111				240 to 255



주소 지정 (3/11)

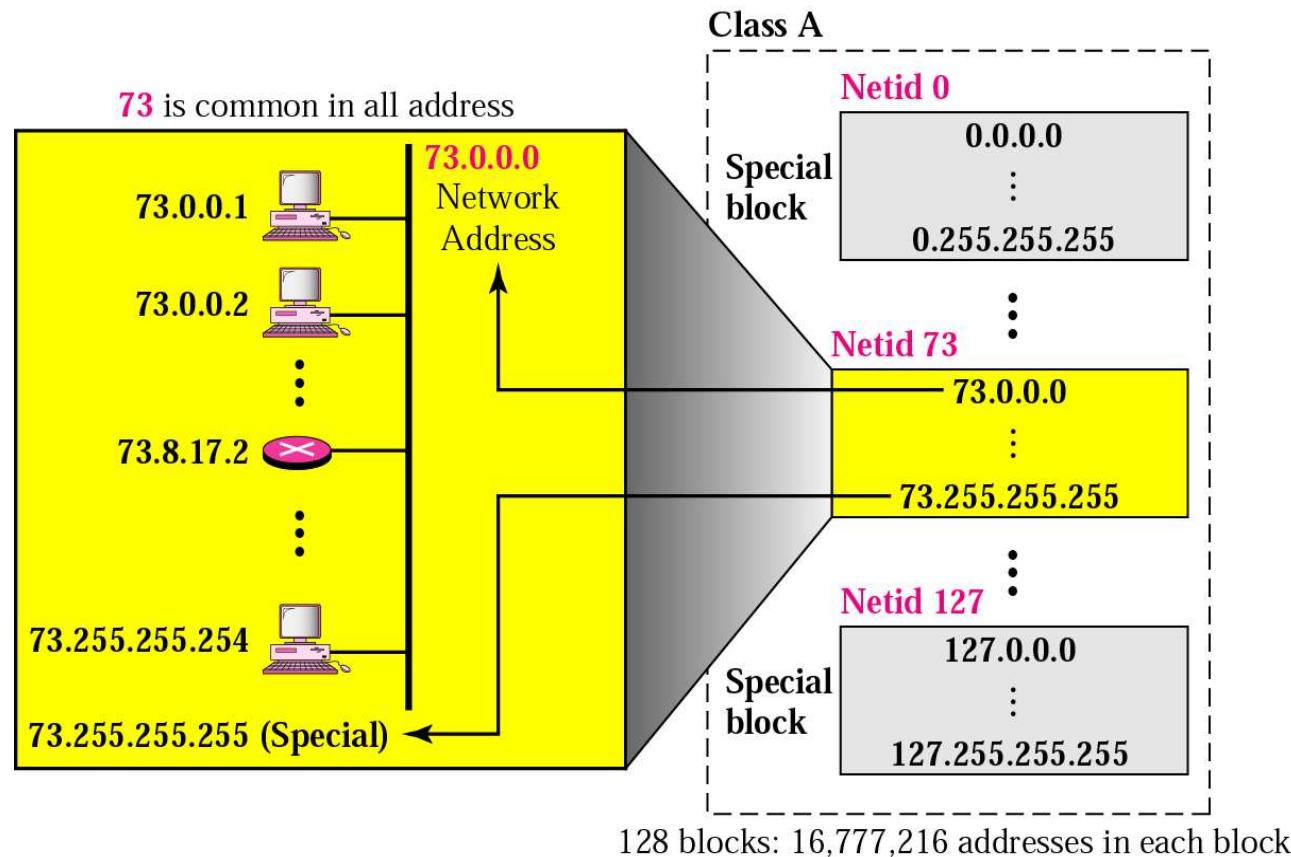
❖ netid와 hostid



주소 지정 (4/11)

❖ 클래스 A

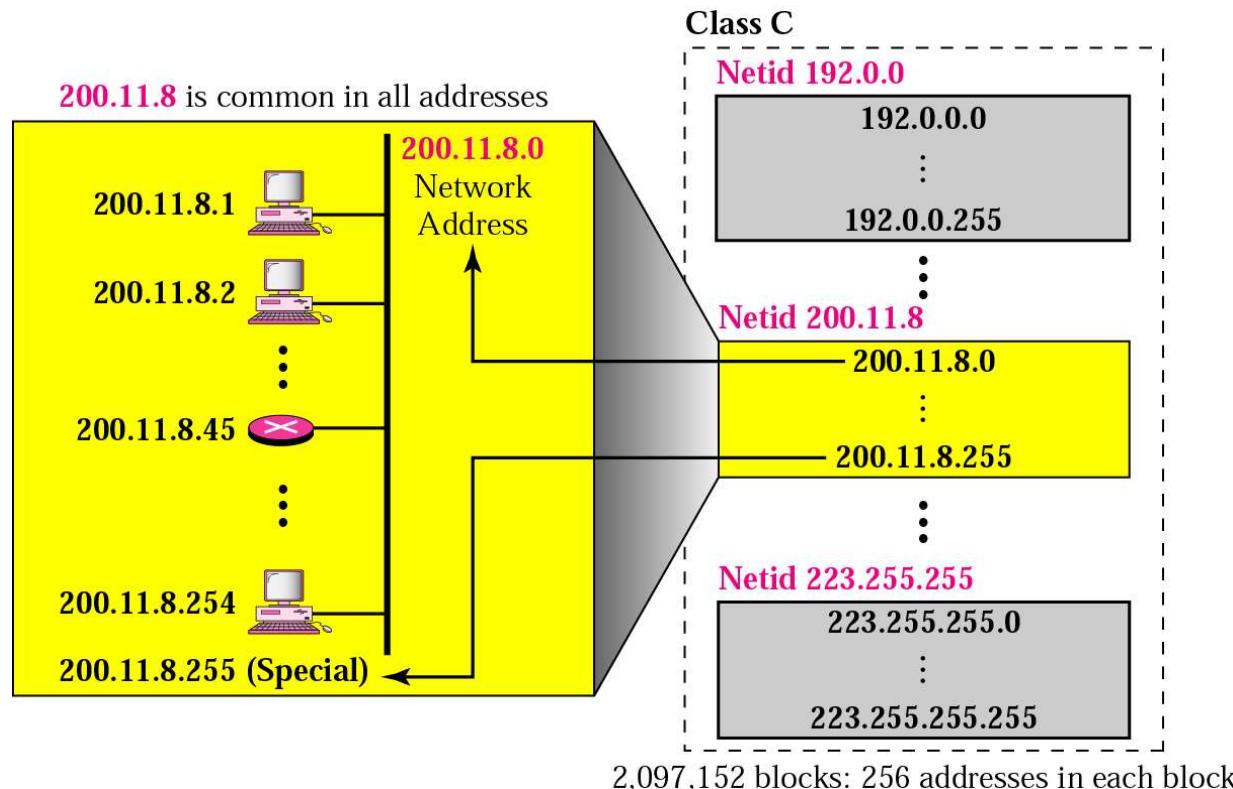
- ▶ 128 개의 블록 (구성 가능한 네트워크 수)
- ▶ 블록당 16,777,216개의 주소 (대체적으로 낭비가 심하다)



주소 지정 (5/11)

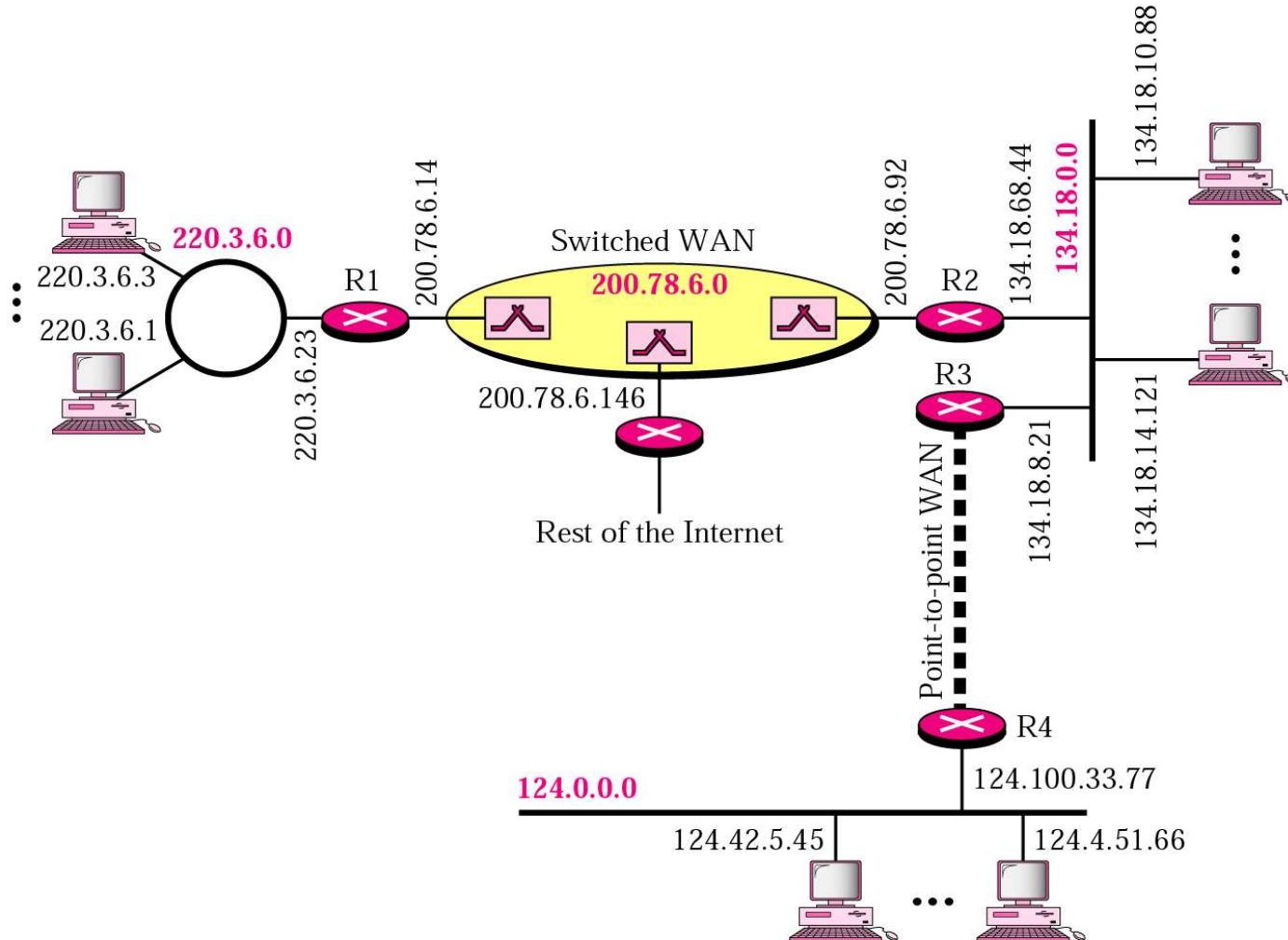
❖ 클래스 C

- ▶ 2,097,152 블록, 256개 사설 블록
- ▶ 2,096,896개 블록 할당, 블록당 256개의 주소



주소 지정 (6/11)

❖ 네트워크 주소 지정의 예





주소 지정 (7/11)

❖ 마스크 (mask)

- ▶ 라우터가 네트워크 주소와 서브네트워크 주소를 찾아내는가?
- ▶ 네트워크 주소를 기반의 컬럼
- ▶ 외부 라우터 - 기본 마스크 사용
- ▶ 내부 라우터 - 서브넷 마스크 사용

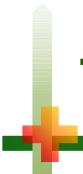
Class	In Binary	In Dotted-Decimal	Using Slash
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24



주소 지정 (8/11)

❖ 동적 주소 설정

- ▶ 인터넷에 연결시 정보
 - ↳ IP 주소, 서브넷 마스크, 라우터의 주소, 네임 서버의 주소
- ▶ 동적 호스트 설정 프로토콜
(Dynamic Host Configuration Protocol, DHCP)
 - ↳ 요구기반의 동적 정보 제공을 위한 프로토콜
 - ↳ DHCP 클라이언트가 서버에게 요청 신호
 - ↳ 요청된 물리주소의 항목을 정적 데이터베이스 찾음
 - ↳ 사용 가능한 대기장소(pool)의 주소 할당하고 동적 데이터베이스에 추가



주소 지정 (9/11)

❖ 사설 주소

- ▶ RFC 1918
- ▶ 3개 주소 집합(사설 주소) 예약

Range	Total
10.0.0.0 to 10.255.255.255	2^{24}
172.16.0.0 to 172.31.255.255	2^{20}
192.168.0.0 to 192.168.255.255	2^{16}

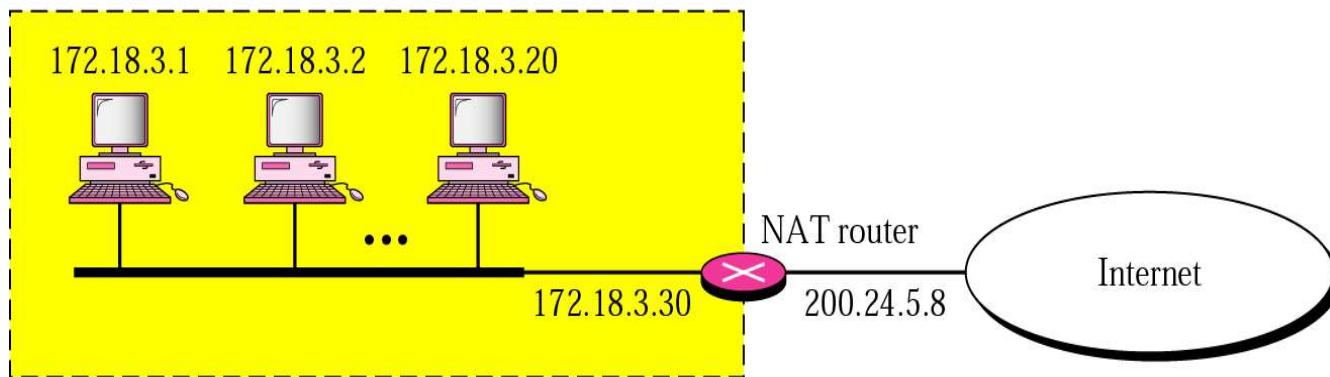
❖ 루프백 주소

- ▶ 보통 127.0.0.1으로 할당
- ▶ INADDR_LOOPBACK

주소 지정 (10/11)

❖ 주소 변환(NAT)

Site using private addresses



▶ 변환 테이블

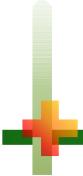
- ↳ IP 주소 한 개 사용하기
- ↳ IP 주소의 집단(pool) 사용하기
- ↳ IP 주소와 포트 번호 동시 사용하기

주소 지정 (11/11)

❖ 주소 변환(NAT) (Cont.)

- ▶ 주소 집단(pool) 사용하기
 - ↳ 포괄적 주소의 집단 사용
 - ↳ 사설 네트워크 호스트는 동일한 시간에 2개의 외부 서버 프로그램(HTTP, FTP)에 접속 불가
- ▶ IP 주소와 포트 번호 동시 사용하기
 - ↳ 다 대 다(many-to-many) 연결 허용
 - ↳ 발신지와 목적지의 전송층 포트번호를 포함한 5개 컬럼 사용

Private Address	Private Port	External Address	External Port	Transport Protocol
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
...



이름과 주소변환 API

- ❖ `struct hostent gethostbyname(const char *hostname);`
 - ▶ 호스트의 이름을 찾아보는 기본 함수
 - ▶ 십진 표기법으로 인한 검색이 가능하다.
 - ▶ 성공하면 호스트에 대한 모든 IPv4 주소를 가지고 있는 `hostent` 구조를 가리키는 지시자를 돌려준다.

- ❖ `struct hostent gethostbyaddr(const char *addr, socklen_t len, int family);`
 - ▶ 이진 IPv4의 주소를 해당하는 호스트 이름을 찾으려 할 때

Routing (1/4)

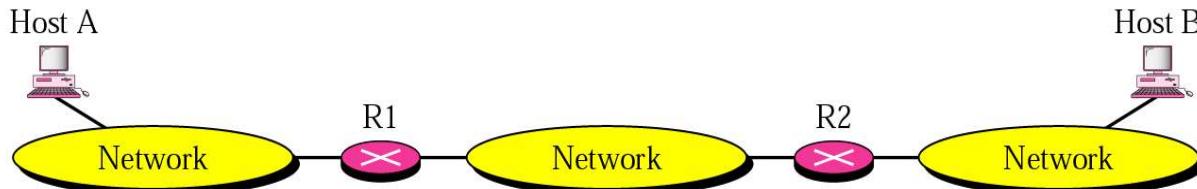
❖ 라우팅 기술

- ▶ 라우팅 테이블(routing table)을 소유하기 위한 호스트 또는 라우터 필요하다.
- ▶ 라우팅 테이블을 통한 최종 목적지에 대한 경로 탐색

❖ 다음-홀 라우팅(next-hop routing)

Routing table for host A		Routing table for R1		Routing table for R2	
Destination	Route	Destination	Route	Destination	Route
Host B	R1, R2, Host B	Host B	R2, Host B	Host B	Host B

a. Routing tables based on route



Routing table for host A		Routing table for R1		Routing table for R2	
Destination	Next Hop	Destination	Next Hop	Destination	Next Hop
Host B	R1	Host B	R2	Host B	—

b. Routing tables based on next hop

Routing (2/4)

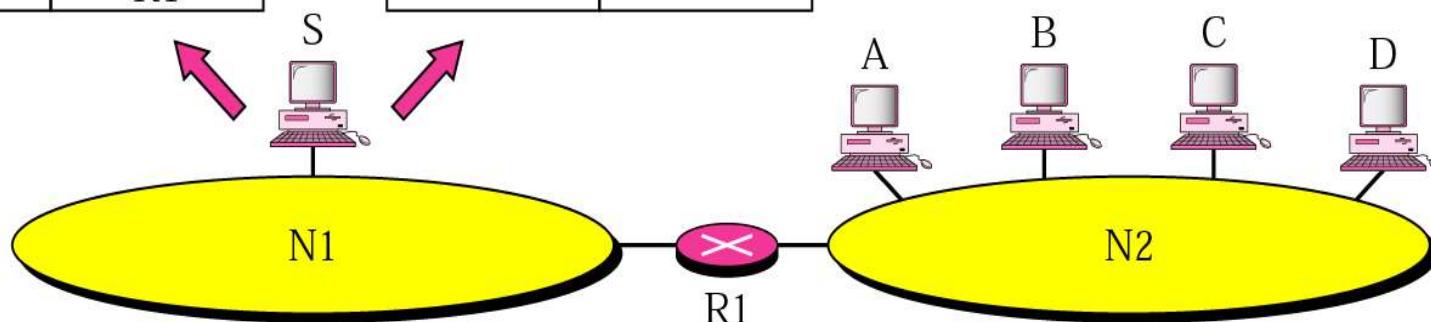
❖ 네트워크 특정 라우팅

Routing table for host S based
on host-specific routing

Destination	Next Hop
A	R1
B	R1
C	R1
D	R1

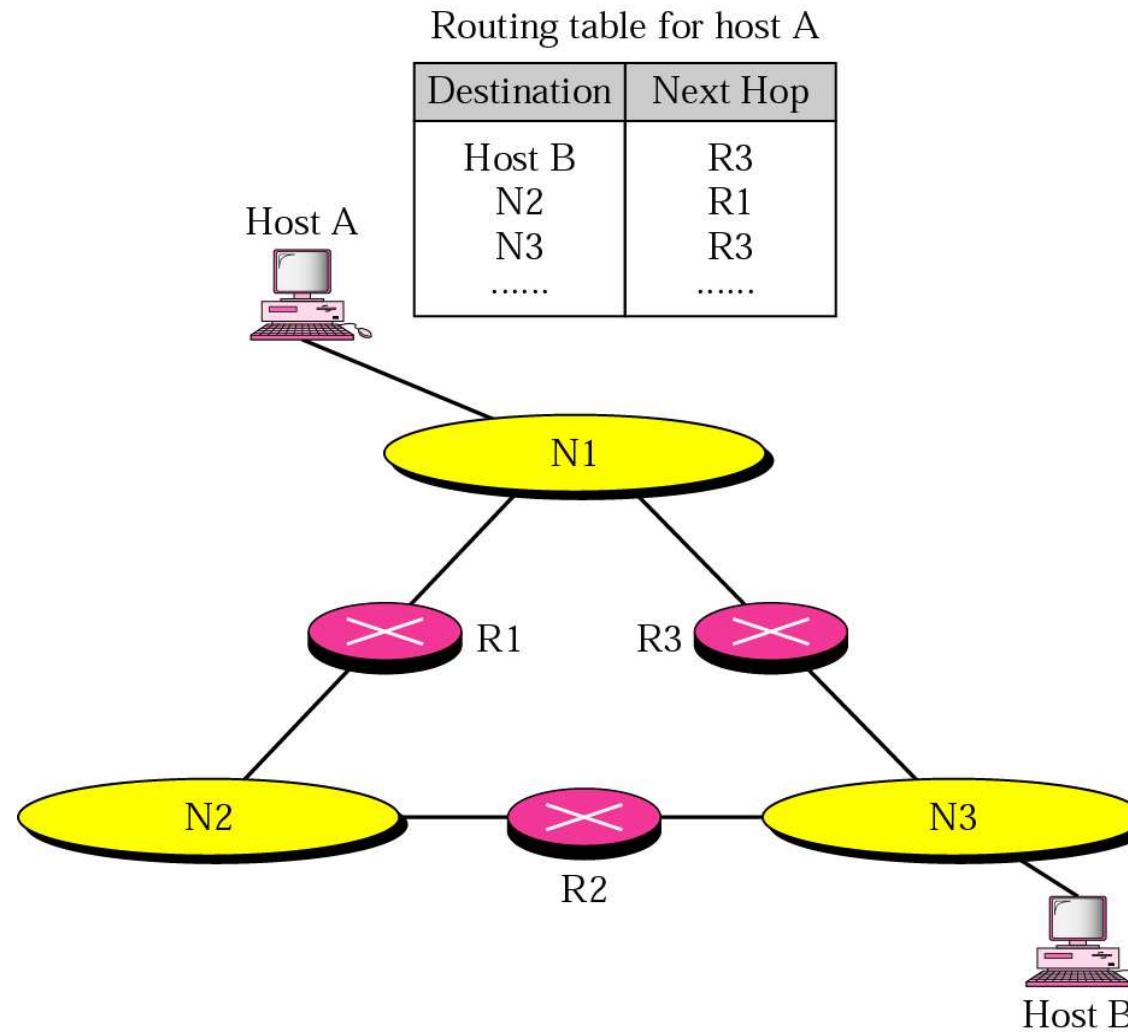
Routing table for host S based
on network-specific routing

Destination	Next Hop
N2	R1



Routing (3/4)

❖ 호스트-특정 라우팅



Routing (4/4)

❖ 정적 대 동적 라우팅

- ▶ 정적 라우팅 테이블(static routing table)
 - ↳ 수동으로 입력된 정보
- ▶ 동적 라우팅 테이블(dynamic routing table)
 - ↳ RIP, OSPF, BGP의 동적 라우팅 프로토콜을 사용
 - ↳ 주기적인 갱신

❖ 클래스형 주소 지정을 위한 라우팅 테이블

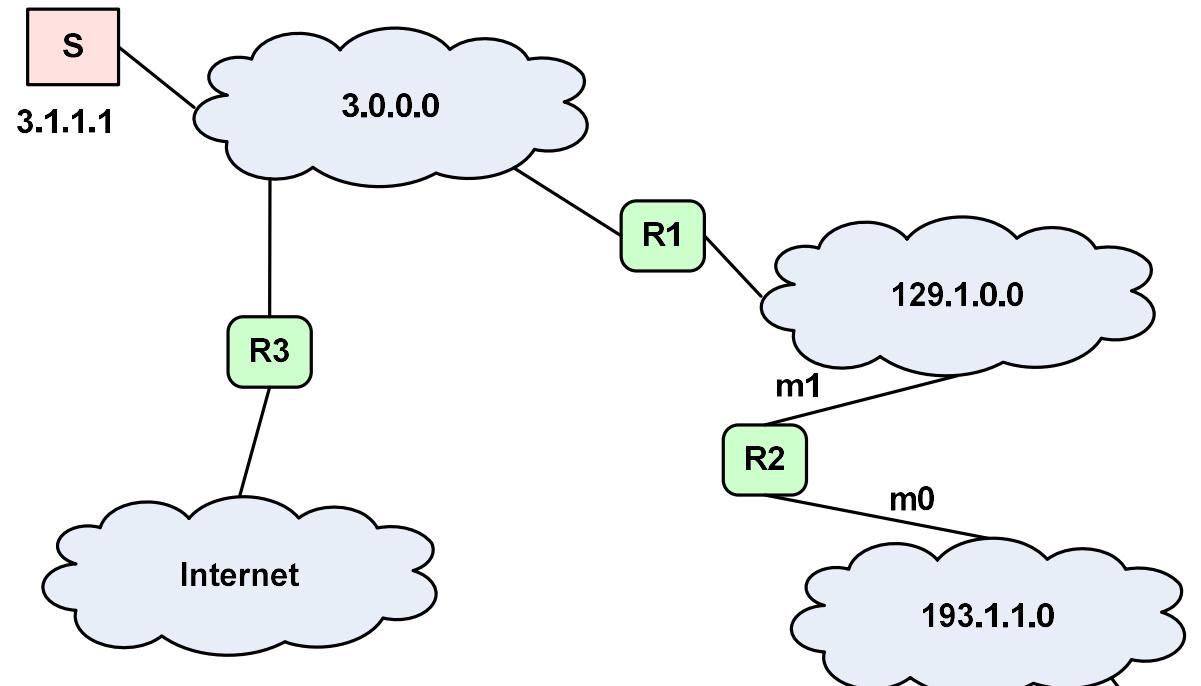
- ▶ 마스크, 목적지 네트워크 주소, 다음-홉 주소, 인터페이스로 구성
- ▶ 목적지 네트워크 주소를 찾지 못하면 기본 인터페이스로 전달

Mask	Destination address	Next-hop address	Interface
Host-specific	/8	14.0.0.0	118.45.23.8
	/32	192.16.7.1	m0
	/24	193.14.5.0	m2
Default	/0	/0	145.11.10.6

실습: 라우팅 프로토콜 테이블 그리기

❖ R2에 대한 라우팅 프로토콜을 작성하라.

- ▶ H(Host-specific), G(Gateway), U(Up)



예)

mask	D.A	N.H	Flags	Ref.cnt	Use.P	연결점

193.1.1.4



IP와 함께 사용되는 프로토콜

❖ ARP, ICMP, IGMP의 관계

- ▶ IP는 다음 흡의 MAC 주소를 알아내기 위해 ARP 프로토콜 사용
- ▶ IP는 메시지를 제어하고 오류 제어를 위해 ICMP 프로토콜 사용
- ▶ IP는 멀티캐스팅을 위해 IGMP 사용



ARP 개요

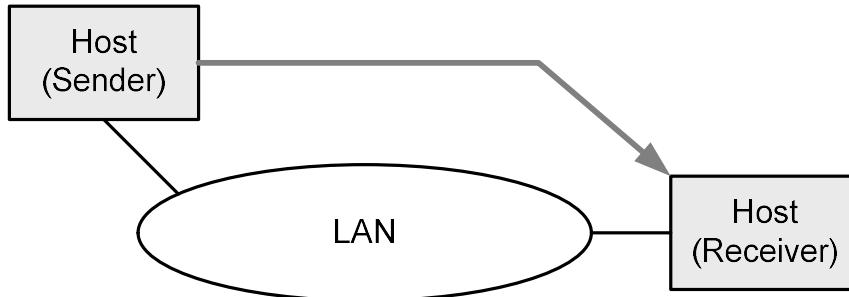
- ❖ ARP 맵핑을 위한 요구사항

- ▶ 호스트와 라우터는 **IP** 주소에 의해 네트워크 층에서 인식
- ▶ **IP** 주소(논리적)는 전세계적으로 유일함
- ▶ 호스트와 라우터는 **MAC** 주소에 의해 물리 층에서 인식
- ▶ **MAC** 주소(물리적)는 지역적으로 유일함
- ▶ ARP는 IP와 MAC 주소를 맵핑 함

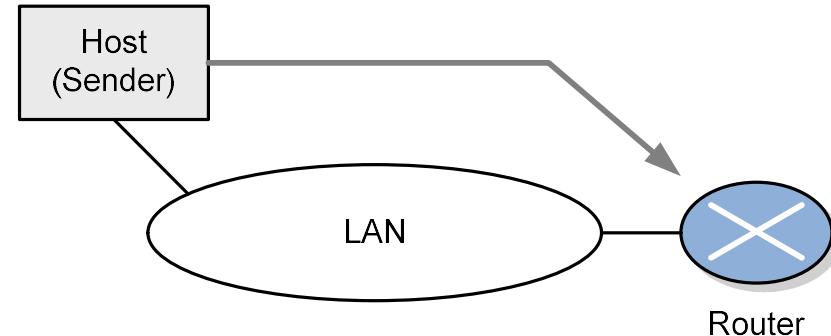
- ❖ ARP가 사용되는 4가지 경우

- ▶ 송신자 호스트에서 같은 네트워크 상에 있는 다른 호스트에게로 패킷을 전송하고자 하는 경우
- ▶ 송신자 호스트에서 다른 네트워크 상에 있는 호스트에게 패킷을 전송하고자 하는 경우
- ▶ 다른 네트워크상에 있는 데이터를 수신한 라우터가 송신자인 경우
- ▶ 같은 네트워크 상에 있는 호스트로 가는 데이터 그램을 수신한 라우터가 송신자인 경우

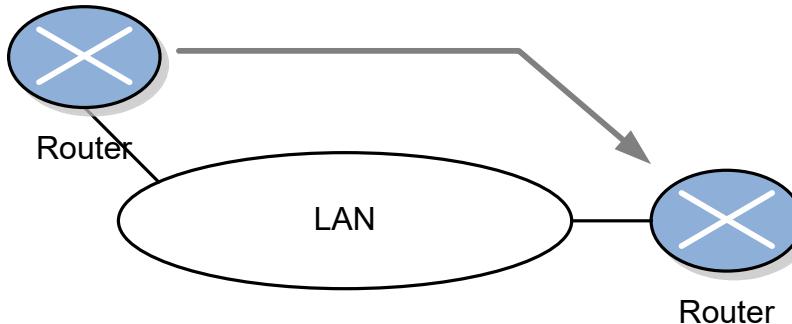
ARP가 사용되는 4가지 경우



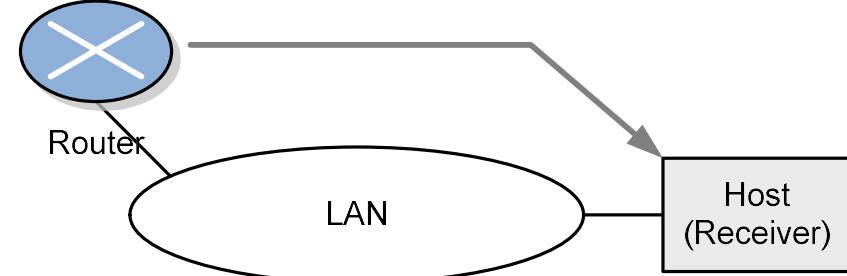
**<Case 1. 호스트에서
같은 네트워크 내에 있는 경우>**



<Case 2. 다른 네트워크에 있는 경우>



<Case 3. 네트워크에서 네트워크로>



**<Case 4. 라우터에서
같은 네트워크에 있는 경우>**



ARP 맵핑 유형

❖ 맵핑 유형

▶ 정적 맵핑(Static Mapping)

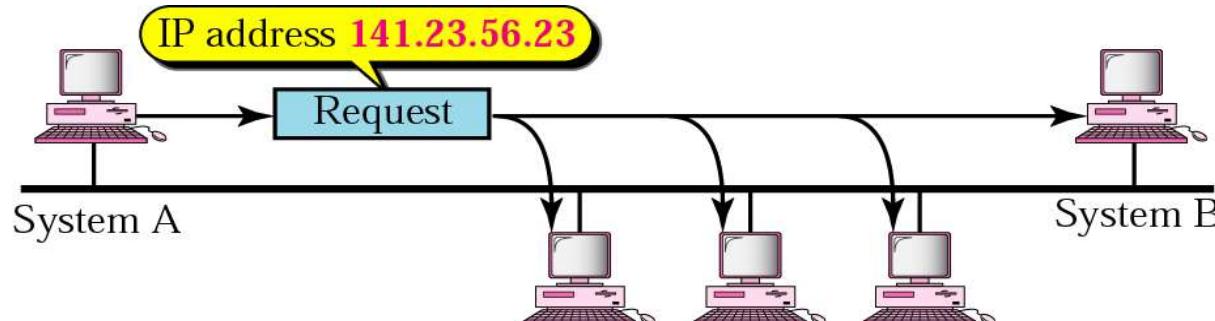
- ↳ IP 주소와 MAC 주소의 연관 테이블 생성
- ↳ 네트워크 상의 각 시스템에 저장
- ↳ 필요 시 테이블 검색
- ↳ 물리주소가 변경될 경우 정적 테이블의 주기적인 갱신으로 오버헤드 (NIC 변경, 이동 컴퓨터의 네트워크 이동 등)

▶ 동적 맵핑(Dynamic Mapping)

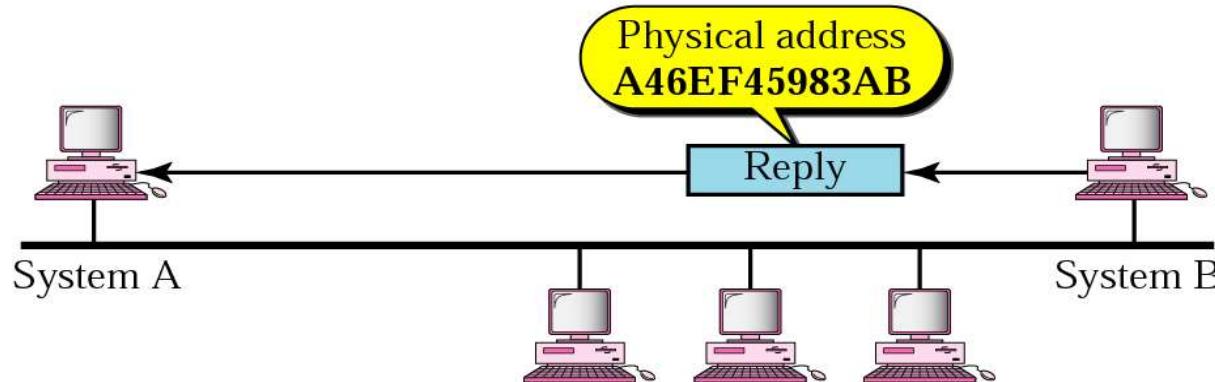
- ↳ 물리 주소와 논리 주소 쌍 중 하나만 알면 프로토콜을 이용하여 다른 하나를 알아냄
- ↳ ARP : 논리 주소를 물리 주소로 변환
- ↳ RARP : 물리 주소를 논리 주소로 변환

ARP request or reply

❖ ARP 요청과 응답



a. ARP request is broadcast



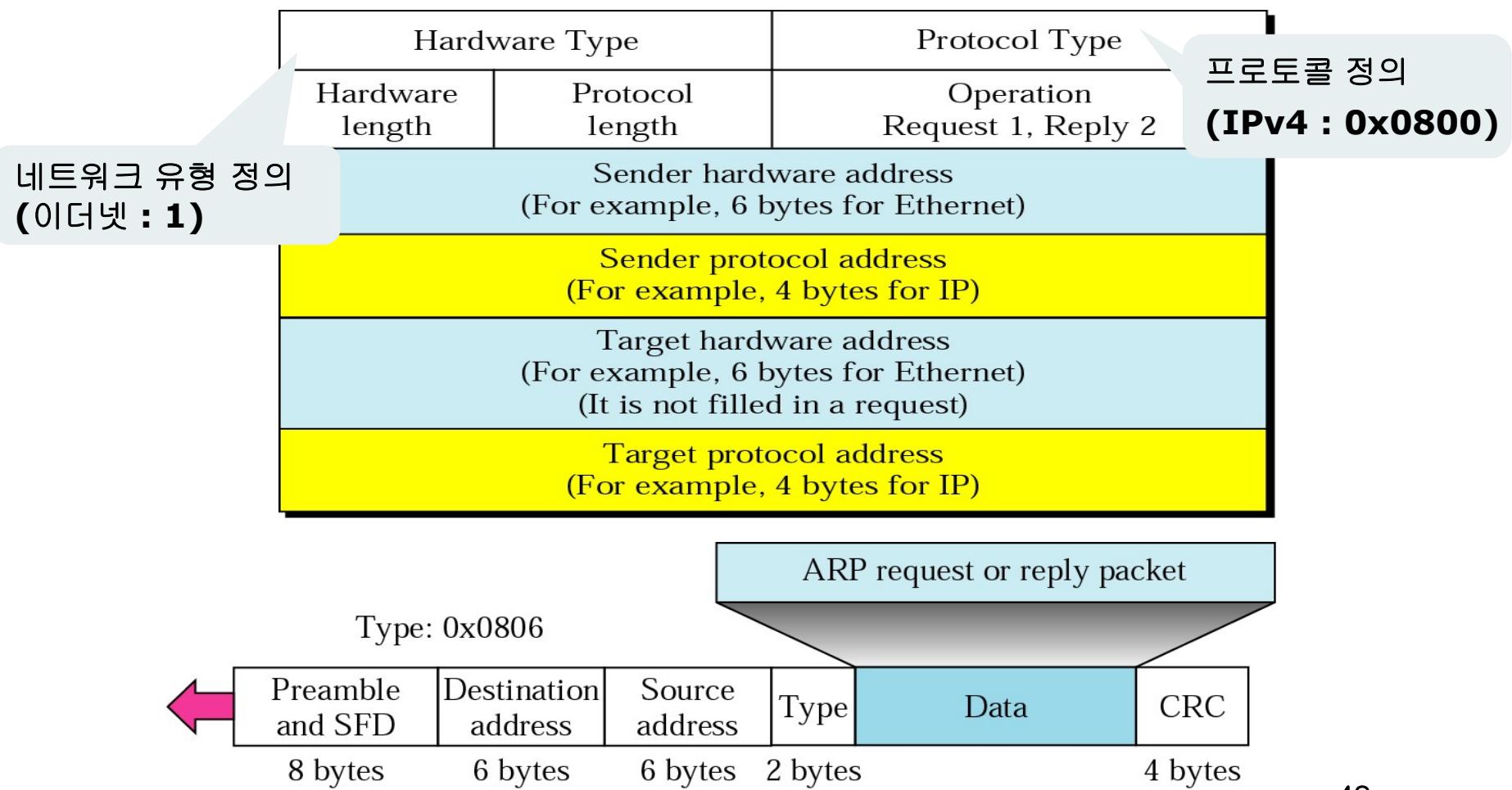
b. ARP reply is unicast

<Fig. ARP의 동작>

ARP 패킷 형식

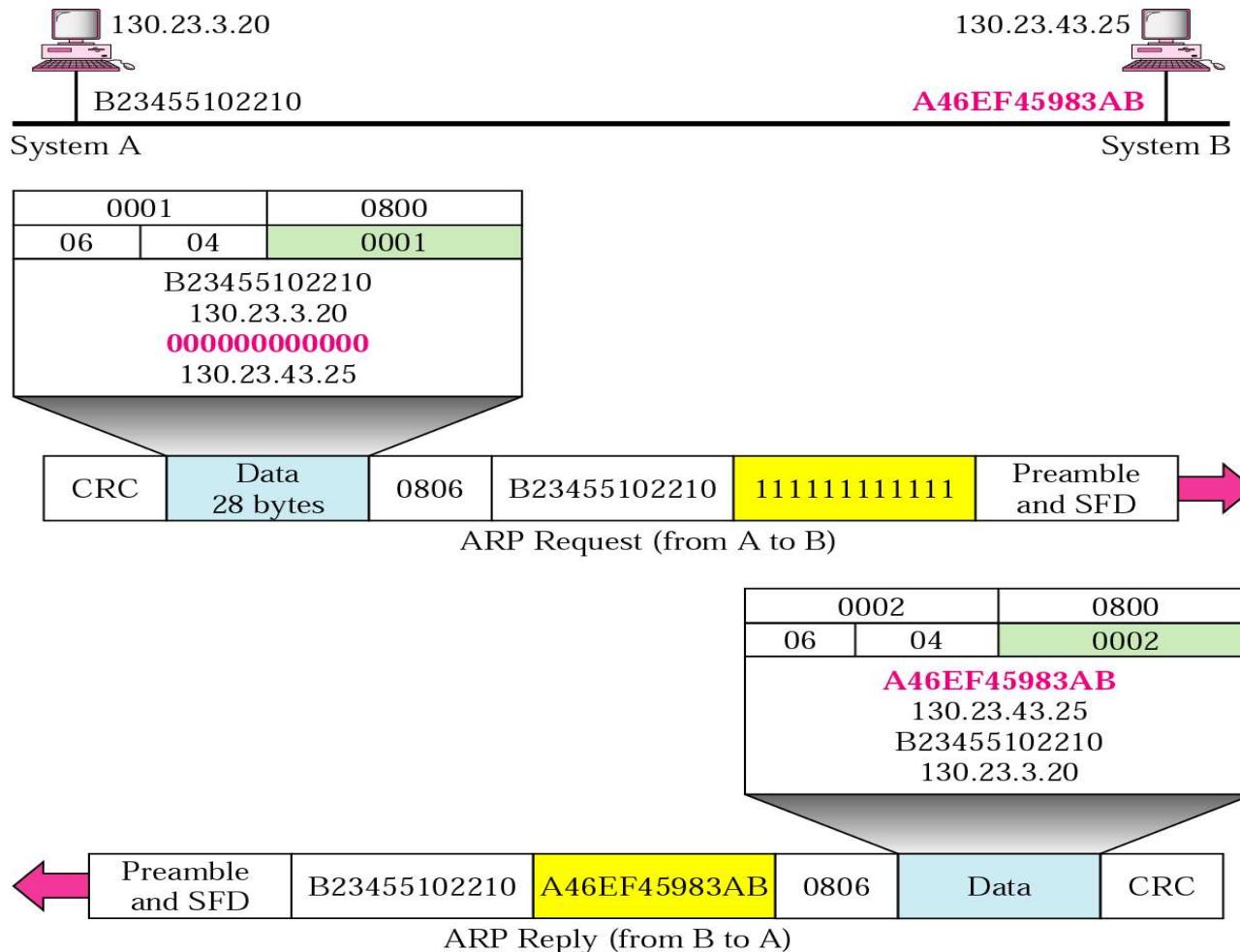
❖ ARP 패킷 형식

- ▶ 9개의 필드로 구성, 데이터링크 프레임에 직접 캡슐화



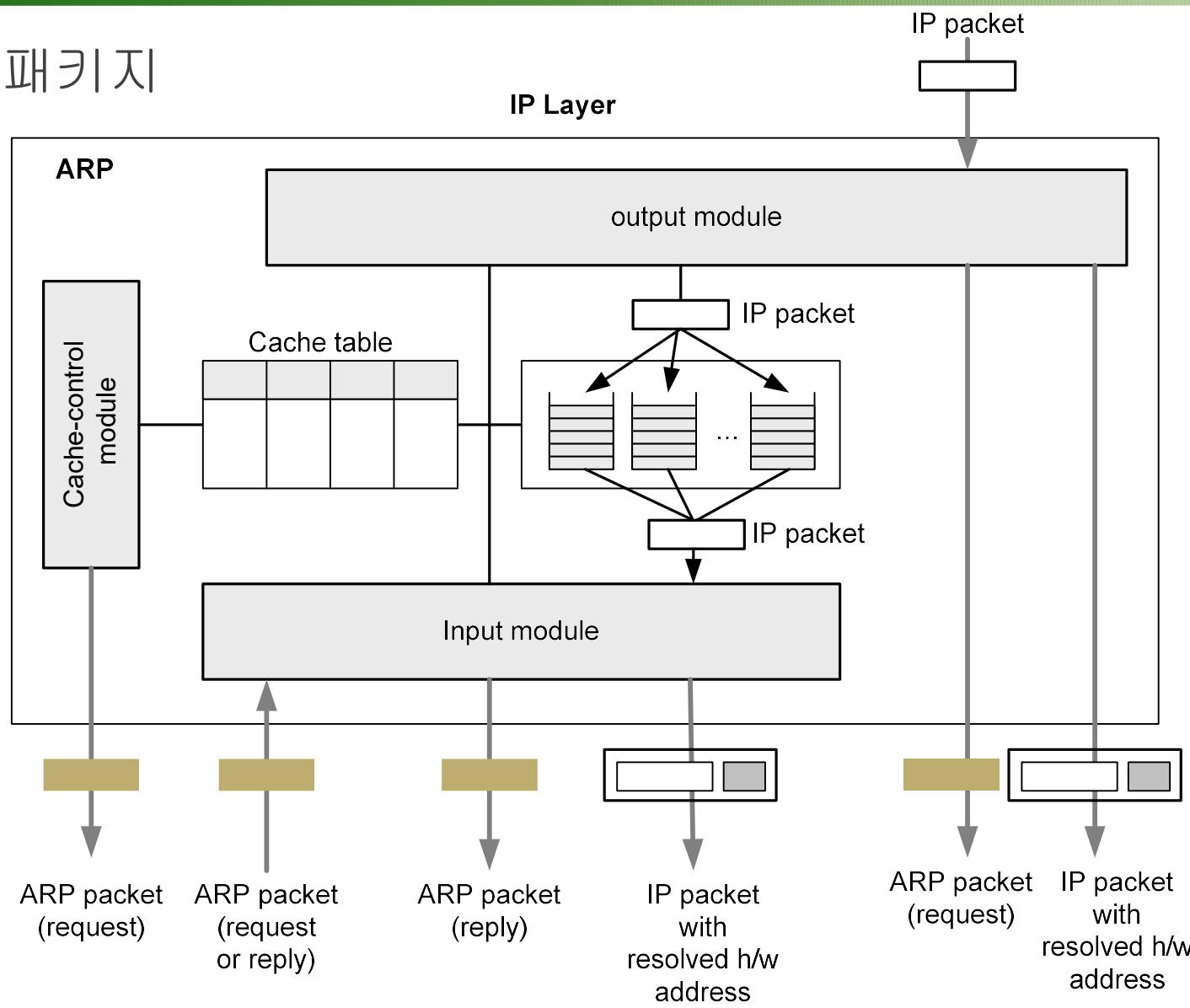
ARP 예제

❖ ARP 동작 예제



ARP 구현

❖ ARP 패키지



<Fig. ARP 구성요소>



ARP 구현



출력 모듈

Output Module

- 1. Sleep until an IP packet is received from IP software**
- 2. Check cache table for an entry corresponding to the destination of this IP packet.**
- 3. If(found)**
 - 1. If(the state is RESOLVED)**
 - 1. Extract the value of the hardware address from the entry.**
 - 2. Send the packet and the hardware address to data link layer.**
 - 3. Return**
 - 2. If(the state is PENDING)**
 - 1. Enqueue the packet to the corresponding queue.**
 - 2. Return.**
- 4. If(not found)**
 - 1. Create a cache entry with state set to PENDING and ATTEMPTS set to 1.**
 - 2. Create a queue.**
 - 3. Enqueue the packet.**
 - 4. Send an ARP request.**
- 5. Return.**



RARP

❖ 개요

- ▶ 물리주소만을 알고 있는 기계의 논리주소를 찾기 위해 사용
- ▶ 디스크가 없는 시스템은 IP주소를 저장할 공간이 없으므로 RARP를 이용해 논리주소를 구할 수 있다.
- ▶ RARP 요청 패킷은 브로드캐스트 되고 응답 패킷은 유니캐스트 된다.

❖ 패킷 형식

- ▶ Operation 필드의 값이 3, 4를 제외하고는 나머지는 ARP와 동일
 - ↳ RARP 요청 : 3
 - ↳ RARP 응답 : 4

❖ RARP 서버

- ▶ RARP는 데이터 링크층에서 구현되지만, UNIX와 같은 운영체제의 도움을 받아 응용계층의 서버로 구현되어야 한다.
- ▶ RARP 외에 다른 해결책: BOOTP, DHCP

실습: ARP 요청패킷 분석

- ❖ 텔넷 등의 응용 프로그램을 이용해 ARP 패킷을 분석해 보자.
 - ▶ Ex) date; telnet 140.252.13.36; date
- ❖ Ethereal을 사용하여 패킷 분석

No.	Time	Source	Destination	Protocol	Info
1	0.000000	00:0c:29:83:5f:ac	Broadcast	ARP	Who has 192.168.123.254? Tell 192.168.123.102
2	0.001972	Advanced_4d:f0:4c	00:0c:29:83:5f:ac	ARP	192.168.123.254 is at 00:50:18:4d:f0:4c
3	0.002037	192.168.123.102	140.252.13.23	TCP	42103 > telnet [SYN] Seq=1596301764 Ack=0 Win=5840 Len=0
4	2.991246	192.168.123.102	140.252.13.23	TCP	42103 > telnet [SYN] Seq=1596301764 Ack=0 Win=5840 Len=0
5	8.991444	192.168.123.102	140.252.13.23	TCP	42103 > telnet [SYN] Seq=1596301764 Ack=0 Win=5840 Len=0

- ❖ ARP 캐시의 타임아웃
 - ▶ 각 호스트는 매번 ARP를 실행하지 않고 ARP 캐시에 물리적인 주소를 임시저장
 - ↳ 완전 엔트리 : 20분 후 종료
 - ↳ 불완전 엔트리: 3분 후 종료
 - ▶ 엔트리가 참조될 때마다 타임아웃 재 시작



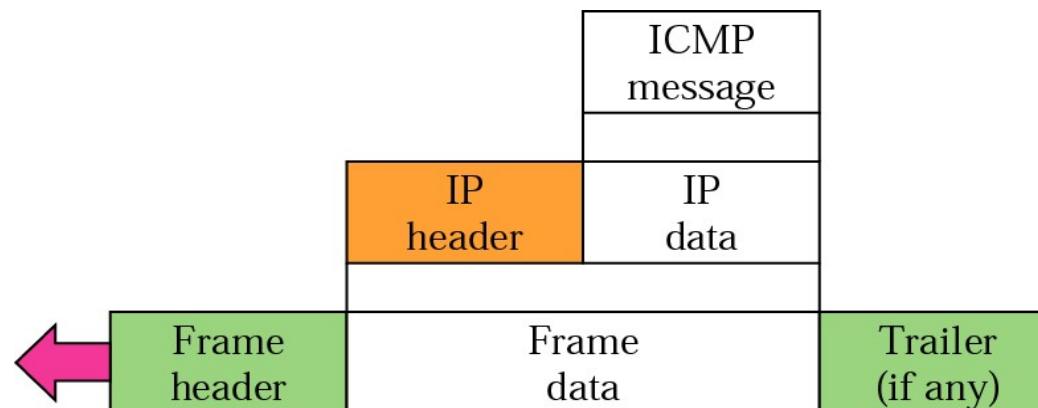
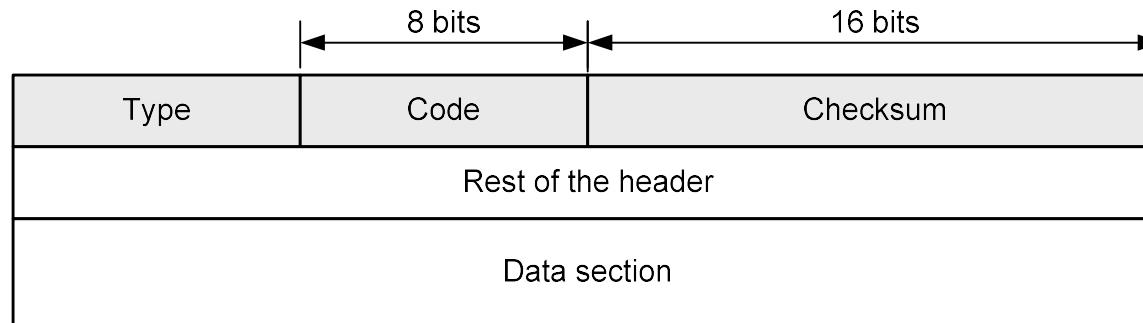
ICMP (1/2)

- ❖ 인터넷 프로토콜의 취약점
 - ▶ IP는 오류 보고와 오류 정정 메커니즘이 없음
 - ▶ IP는 호스트와 관리 질의를 위한 메커니즘이 부족
- ❖ ICMP(Internet Control Message Protocol)
 - ▶ 인터넷 제어 메시지 프로토콜
 - ▶ IP의 약점을 보완
 - ▶ IP 데이터그램으로 캡슐화

ICMP

◆ ICMP의 캡슐화 위치

- ▶ IP데이터가 ICMP 메시지임을 알리기 위해 IP 데이터그램의 프로토콜 필드의 값은 1이 된다.



<Fig. ICMP 메시지의 일반 형식과 캡슐화 위치>

ICMP의 메시지 유형

❖ ICMP 메시지 유형

▶ 오류 보고 메시지(error reporting)

↳ ICMP는 오류를 수정하지 않고, 단지 보고만 수행

↳ IP 주소를 이용하여 발신지에 오류 전송

▶ 질의 메시지(query)

↳ 일부 네트워크의 문제를 진단

<Table. ICMP 메시지>

Category	Type	Message
Error-reporting message	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query message	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router solicitation or advertisement



ICMP 메시지 유형

- ❖ 오류 보고(error-reporting) 메시지
 - ▶ ICMP는 언제나 최초의 발신지로 오류 메시지를 보고한다.
 - ▶ Type 3: 목적지 도달 불가(Destination Unreachable)
 - ↳ Code: 0 하드웨어 고장 등의 이유로 네트워크에 도달할 수 없다.
(라우터에 의해 생성)
 - ↳ Code: 1 호스트에 도달할 수 없다.(라우터에 의해 생성)
 - ↳ Code: 2 프로토콜에 도달할 수 없다. (목적지 호스트에서만)
 - ↳ Code: 3 포트에 도달할 수 없다.(목적지 호스트에서만)
 - ↳ Code: 4 단편화가 필요하나 데이터그램의 DF필드가 설정되어 있다.
 - ↳ Code: 5 발신지 라우팅이 수행될 수 없다.
 - ↳ Code: 6 목적지 네트워크가 알려져 있지 않다.
 - ↳ Code: 7 목적지 호스트가 알려져 있지 않다.
 - ↳ Code: 8 발신지 호스트가 고립되어 있다.
 - ↳ Code: 9 목적지 네트워크와 통신이 관리상의 이유로 금지 되었다.
 - ↳ Code: 10 목적지 호스트로의 통신이 관리상의 이유로 금지되어 있다.
 - ↳ ...



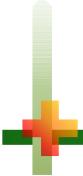
ICMP 메시지 유형

▶ Type 4: 발신지 억제(Source quench)

- ↳ IP의 흐름제어 부재로 인해 생기는 혼잡(congestion)을 야기하는데 혼잡으로 인해 데이터그램이 폐기되면 송신자에게 발신지 억제 메시지를 보낸다.
- ↳ 데이터그램이 폐기되었음을 알리고, 송신 과정을 천천히 하여야 한다는 것을 발신지에게 경고한다.

▶ Type 11: 시간 경과(Time Exceeded)

- ↳ Code 0: 데이터그램의 TTL이 감소하여 폐기될 때 라우터는 시간 경과 메시지를 원 발신지에 송신하여야 한다.
- ↳ Code 1: 단편들이 정해진 시간 내에 목적지 호스트에 전부 도착하지 않은 경우에도 시간 경과 메시지가 생성된다.



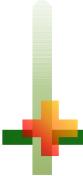
ICMP 메시지 유형

▶ Type 12: 매개변수 문제(Parameter Problem)

- ↳ Code 0: 헤더 필드 중에서 불명료하거나 빠진 것이 있다.
- ↳ Code 1: 옵션의 요구되는 부분이 빠졌다.

▶ Type 5: 재지정(Redirection)

- ↳ 다른 네트워크로 가는 데이터그램이 틀린 라우터로 보내는 경우 호스트에게 라우팅 테이블을 갱신 하라는 재지정 메시지를 보낼 수 있다.
- ↳ Code 0: 네트워크 지정 경로를 위한 재지정
- ↳ Code 1: 호스트 지정 경로를 위한 재지정
- ↳ Code 2: 특정한서비스 유형에 기초한 네트워크 지정 경로를 위한 재지정
- ↳ Code 3: 특정한서비스 유형에 기초한 호스트 지정 경로를 위한 재지정



ICMP 메시지 유형

❖ 질의(Query) 메시지

- ▶ 에코 요청 및 응답(echo request and reply)
 - ↳ 네트워크 진단을 목적으로 고안
 - ↳ 이 메시지 쌍의 조합은 두 시스템(호스트들과 라우터들)이 서로간에 통신할 수 있는지를 결정하는데 사용
- ▶ 타임스탬프 요청과 응답(time-stamp request and reply)
 - ↳ 두 시스템 간에 IP 데이터그램이 오고 가는데 필요한 왕복 (round-trip) 시간을 결정하는데 사용
 - ↳ 두 시스템간의 동기화에도 사용
- ▶ 주소 마스크 요청 및 응답(address mask request and reply)
 - ↳ IP 주소의 일부를 정의하는 네트워크 주소, 서브넷 주소, 호스트 식별자를 모르는 경우에 사용
- ▶ 라우터 간청 및 광고(router solicitation and advertisement)
 - ↳ 호스트는 인근 라우터가 정상적인 기능을 수행하는지 식별하기 위해 라우터 간청 메시지 전송
 - ↳ 라우터는 라우터 광고 메시지로 라우터 간청 메시지에 응답



실습: ping 프로그램을 통한 패킷 분석

❖ 목 표

- ▶ ping 프로그램을 사용하여 ICMP 메시지를 분석해 본다.
 - ↳ Ex) ping 192.168.1.55
- ▶ ethereal을 통해 패킷을 캡쳐 하여 살펴보도록 한다.
- ▶ traceroute 프로그램을 사용하여 발신지에서 목적지까지 전달되는 경로를 추적해 본다.
 - ↳ Ex) traceroute xerox.com



❖ IGMP (Internet Group Management Protocol)

- ▶ 동시에 많은 수의 수신자에게 보낼 필요가 있을 때 멀티캐스팅의 통신유형을 가져야 하는데 IGMP가 그룹관리를 통해 관리하게 한다.
- ▶ 예를 들면 주식 가격의 변동에 대해 동시에 정보를 전달 받거나 여행사들이 최소 된 여행에 대해 정보를 받을 수 있고, VoD등에도 응용 될 수 있다.

❖ 그룹 관리

- ▶ IGMP는 그룹관리 프로토콜이다. 이것은 멀티캐스트 라우터가 인터페이스에 관련된 멤버의 리스트를 생성하고 갱신하는 것을 돋는다.

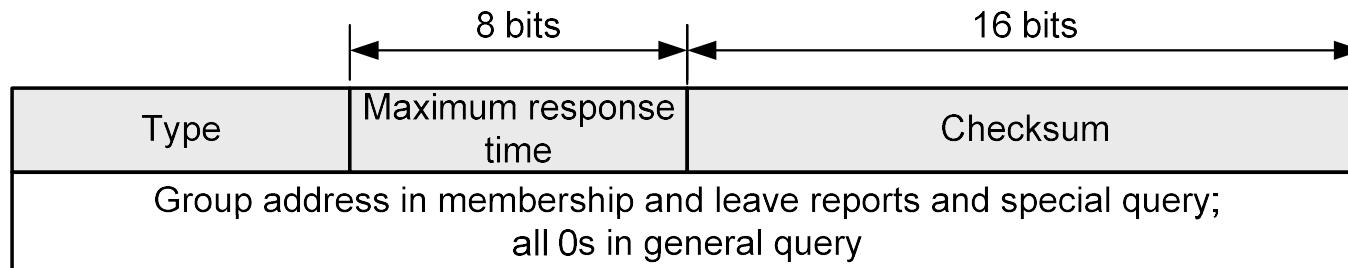
IGMP 메시지

❖ IGMP의 메시지 유형

- ▶ 질의(query)
- ▶ 멤버십 보고(membership report)
- ▶ 탈퇴 보고(leave report)

❖ IGMP 메시지 형식

- ▶ 유형(Type): 표 참조
- ▶ 최대 응답 시간: 질의가 응답되어야 하는 시간을 정의(100=10초)
- ▶ 그룹주소: 일반 질의 메시지에서는 0, 그 이외에 **groupid**를 정의



<Fig. IGMP의 메시지 형식>



IGMP의 동작

- ❖ 그룹 가입
 - ▶ Type 0x16(Membership Report)
 - ▶ 호스트나 라우터는 그룹에 가입할 수 있다.
 - ▶ IGMP에서 멤버십 보고는 두 번 보내진다. 첫 번째 보고가 손실되거나 훼손되더라도 두 번째 보고가 대체할 수 있다.
- ❖ 그룹 탈퇴
 - ▶ Type 0x17(Leave Report), 11(Special Query)
 - ▶ 특정 그룹에 관심이 있는 프로세스가 없다면 탈퇴 메시지를 보낸다.
- ❖ 멤버십 모니터링
 - ▶ 라우터는 주기적(기본적으로 125초 간격)으로 일반 질의 메시지를 보낸다.

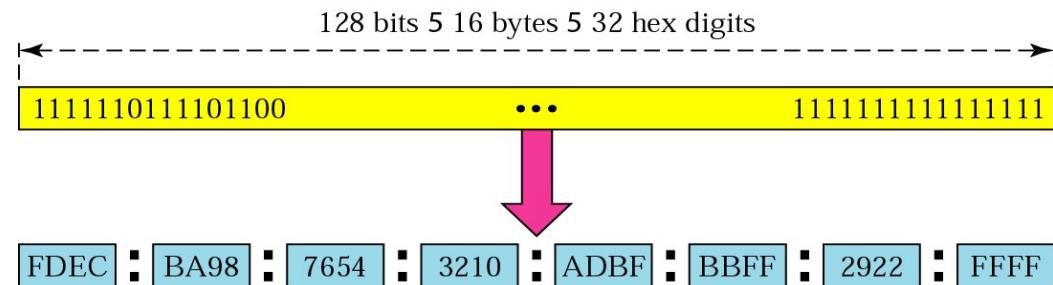
IPv6 (1 / 7)

❖ IPv4의 문제점

- ▶ IPv4 주소 공간의 한계
- ▶ IPv4는 최소 지연과 자원의 예약 불가
- ▶ IPv4에서는 보안 메커니즘(암호화와 인증)을 제공하지 않음

❖ IPv6 장점

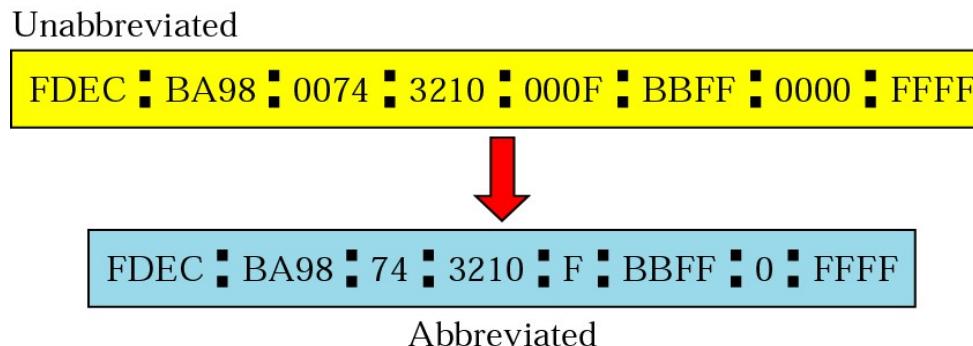
- ▶ 확장된 주소 공간
- ▶ 개선된 헤더 형식
- ▶ 새로운 옵션
- ▶ 확장 허용
- ▶ 자원 할당에 대한 지원
- ▶ 향상된 보안성 제공



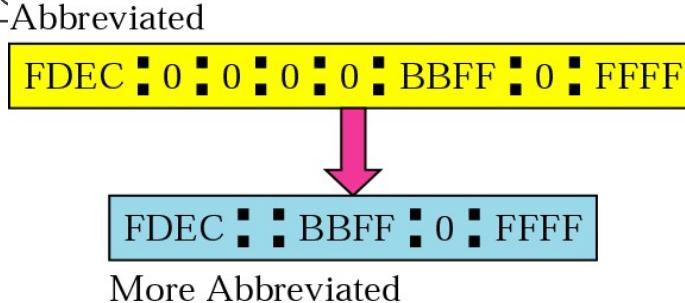
IPv6 (2/7)

❖ 생략형(abbreviation)

- ▶ 섹션(두 개의 콜론 사이에 있는 4개의 숫자)의 앞에 있는 0들은 생략 가능



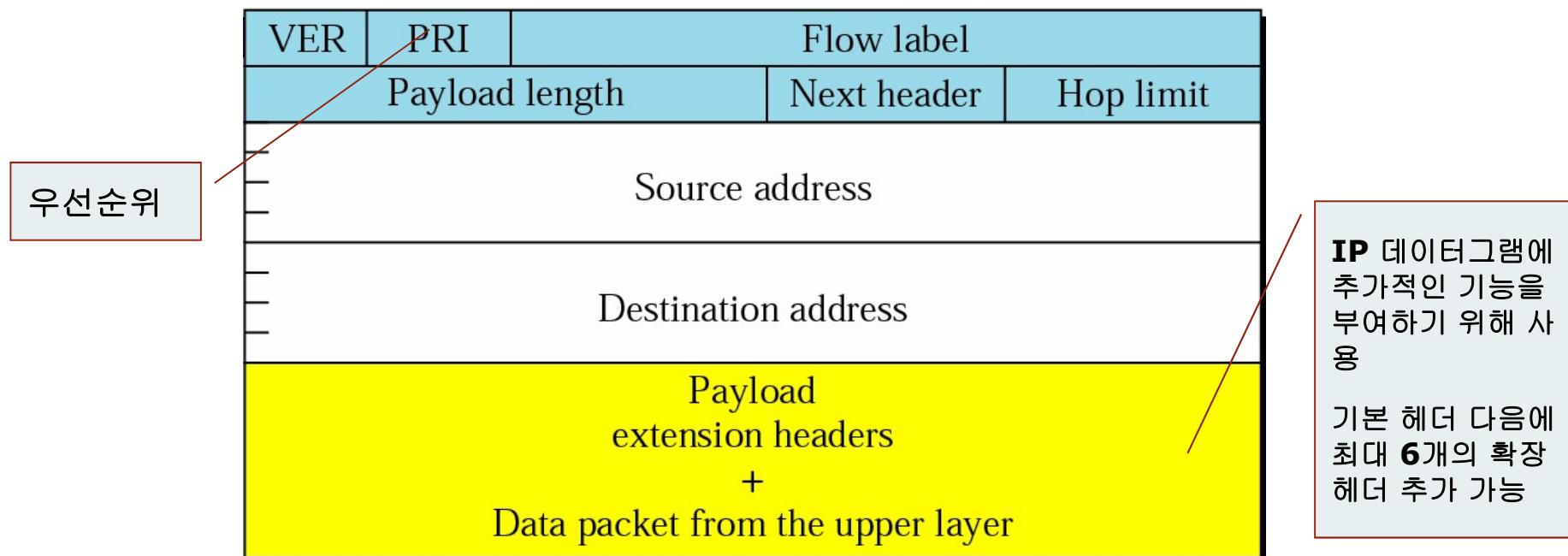
- ▶ 연속된 0이 생략된 주소



IPv6 (3/7)

❖ IPv6 패킷 형식

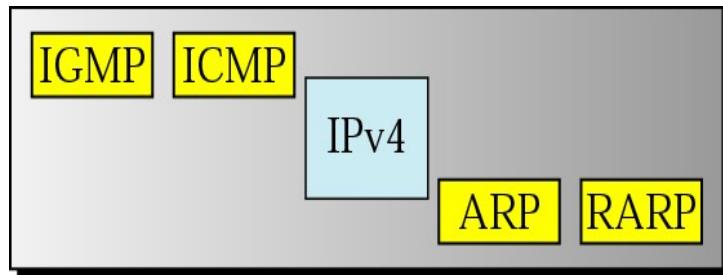
- ▶ 각 패킷은 두 부분으로 구성
 - ↳ 기본 헤더(base header) 40바이트
 - ↳ 페이로드(payload) 65,535 바이트
 - ▶ 선택적인 확장 헤더들과 상위 계층의 데이터로 구성



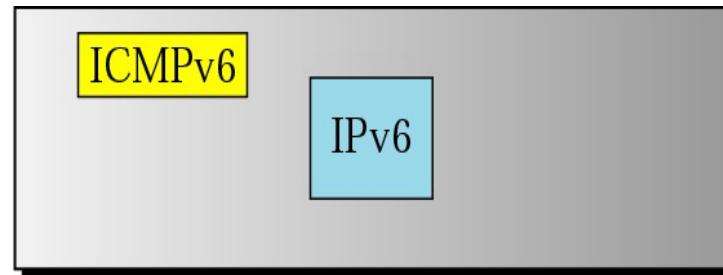
IPv6 (4/7)

❖ ICMPv6

- ▶ IPv4와 동일한 방법과 목적을 따름
- ▶ 일부 프로토콜은 IPv4에서는 독립적이지만, 현재는 ICMPv6의 일부가 됨
- ▶ IPv4와 IPv6의 네트워크 층 비교



Network layer in version 4

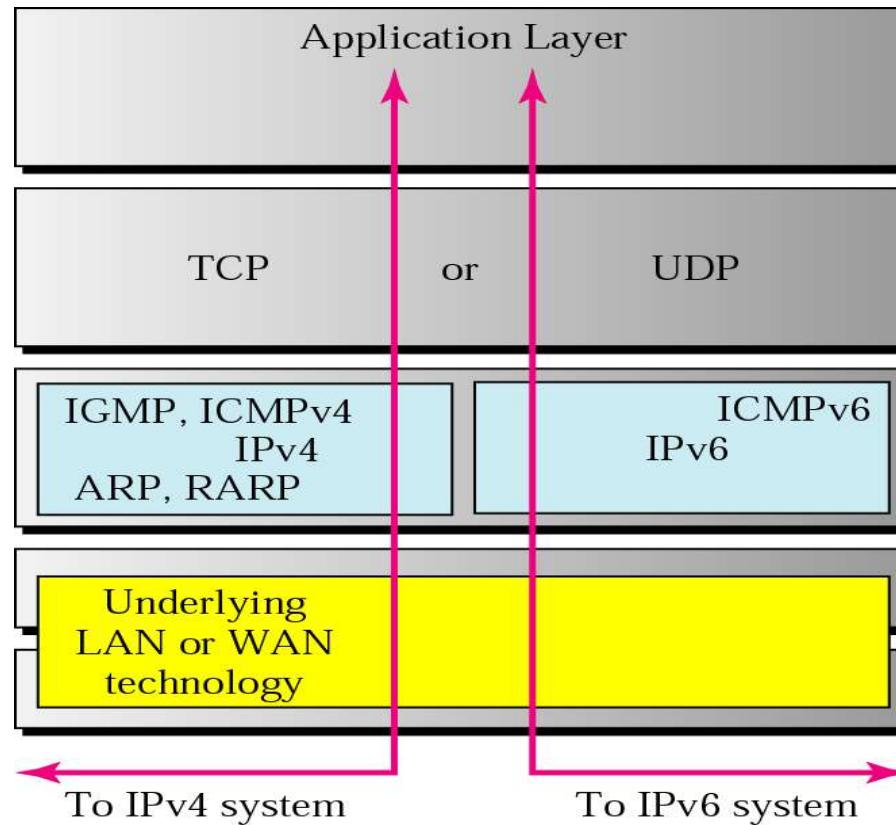


Network layer in version 6

IPv6 (5/7)

❖ 이중 스택(dual stack)

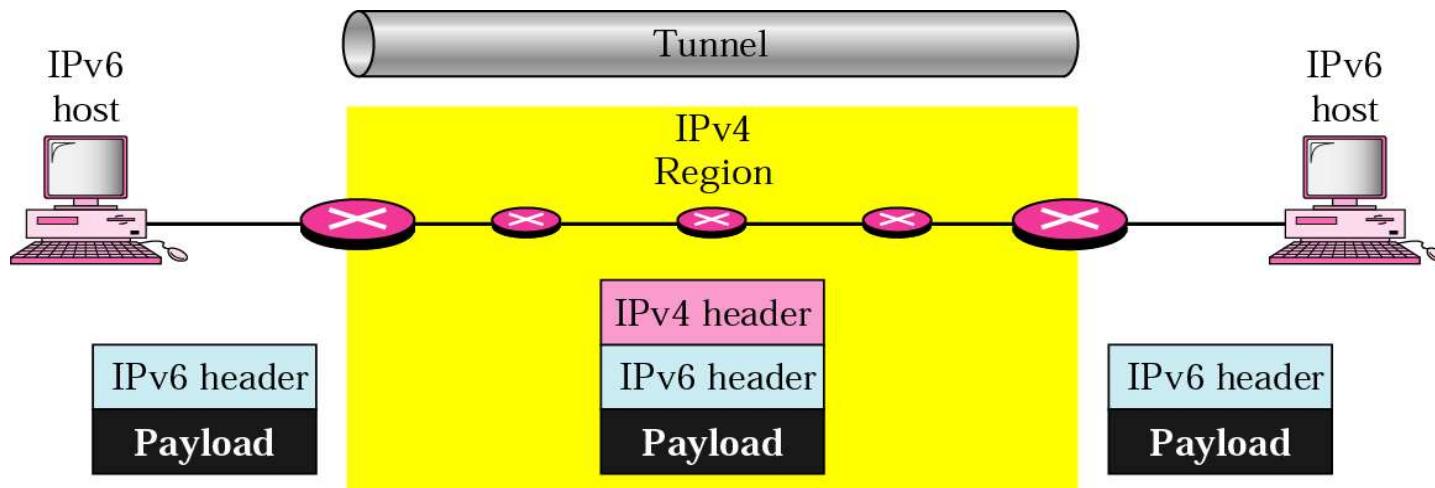
- ▶ 지금은 IPv4와 IPv6가 동시에 동작하여야 함
- ▶ 발신지 호스트는 어떤 버전을 사용할 것인가를 결정하기 위해 DNS에게 질의



IPv6 (6/7)

❖ 터널링(tunneling)

- ▶ IPv6를 사용하여 상호 통신을 원하는 두 컴퓨터가 IPv4를 사용하는 지역에 패킷을 통과시켜야 할 때 사용하는 방법
- ▶ IPv6 패킷을 IPv4 패킷으로 캡슐화하여 전달하고, 지역을 벗어나면 IPv6 패킷으로 변환



IPv6 (7/7)

❖ 헤더 변환(header translation)

- ▶ 송신자는 수신자가 오직 IPv4 패킷만을 이해하는 경우를 제외하고, IPv6 패킷을 전송
- ▶ IPv6 패킷의 헤더를 IPv4 헤더로 변환

