2024/08/29, 2024 네트워크 세미나

TCP/IP 프로토콜

-4장 네트워크 계층 소개, 5장 IPv4 주소, 6장 IP 패킷의 전달-

김 혜 정(hyejeong@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 보충
- 네트워크 계층 소개
- IPv4 주소
- IP 패킷의 전달과 포워딩

보충

- 대칭키와 비대칭키
  - 활용 예시 (1/2)
    - 대칭키 암호화 방식
      - 실시간 데이터 처리
        - 처리 속도가 빠르며 대량의 데이터를 암호화할 때 유리하여 대량의 파일이나 데이터 베이스 처리할 때 유리함
      - 컴퓨팅 자원 환경
        - 연산량이 다소 적어 모바일 기기, IoT 기기 등 제한된 컴퓨팅 자원을 활용하는 시스템에 적합함
    - 비대칭키 암호화 방식
      - 전자서명
        - 송신자의 개인키로 서명하면 송신자의 공개키로 검증할 수 있어 데이터 무결성 검증과 출처가 인증 가능함
      - 키 교환
        - SSL/TLS, IPsec 등 프로토콜에서 대칭키를 교환하는 데에 쓰임

보충

- 대칭키와 비대칭키
  - 활용 예시 (2/2)
    - 비대칭키
      - 공개키로 암호화하는 경우
        - 수신자의 공개키로 암호화함
        - 장점
          - 수신자만이 자신의 개인키로 복호화 할 수 있어 데이터 기밀성이 보장됨
        - 단점
          - 공개키를 사전에 알아야 하므로 데이터 무결성 검증과 출처 인증이 어려움
      - 개인키로 암호화하는 경우
        - 송신자의 개인키로 암호화함
        - 장점
          - 데이터 무결성 검증과 출처 인증이 가능
        - 단점
          - 송신자의 공개키를 알고 있다면 누구나 복호화 할 수 있어 비밀성 보장이 어려움

보충

- RSA-공격
  - 암호화 지수에 대한 공격 (1/2)
    - 관련된 메시지 공격(Franklin-Reiter Related Message Attack)
      - Franklin-Reiter가 발견한 방법으로, RSA 암호화된 메시지들을 선형 관계를 이용해 특정 값을 빠르게 계산하는 방법
        - e.g., 공격자가 $c_i = (x + b_i)^e \; mod \; N = f(x)$를 알고 있을 때, 비밀값 x를 알고자 함
          - 모든 암호문이 x와 직선 관계에 있음
          - 각 암호문 간의 상수 b_i들 간의 차이를 사용하여 특정 값 p_k를 계산
          - $p_k = b_i - b_k$의 역수가 존재하지 않으면 N을 쉽게 인수분해할 수 있음
            - 역수가 존재하려면 $\gcd(p_k, N) = 1$인데 이는 두 수가 공통된 약수를 갖지 않는다는 것을 의미함
          - $p_k = \prod_{i=0}^{e-1} b_k - b_i$
          - $v = \sum_{k=0}^{e-1} (b_k)^e (p_k)^{-1} mod \; N$
          - $w(x) = \sum_{k=0}^{e-1} c_k p_k^{-1} mod \; N$
          - $x \equiv e^{-1}(w(x) - v) mod \; N$

출처: Yacobi, Oded, and Yacov Yacobi, "A new related message attack on RSA." *Public Key Cryptography-PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23-26, 2005. Proceedings 8.*, 2005.

보 충

- RSA-공격
  - 암호화 지수에 대한 공격 (2/2)
    - 짧은 패드 공격(Coppersmith's Short Pad Attack)
      - 메시지에 패딩을 추가할 때, 랜덤 비트를 추가하는 방법을 적용하거나 짧은 패딩을 적용할 경우 발생하는 공격
        - e.g., 사용자가 메시지 m을 암호화한다고 가정
          - 메시지 m에 짧은 패딩 p를 추가하여 암호화할 메시지 c를 생성
            - $c = (m + p)^e mod\, n$
          - 공격자가 암호문 c와 공개키 e를 알고 있다면, 가능한 모든 메시지 m'을 시도하면서 $c = (m + p)^e mod\, n$를 만족시키는 m'을 찾음

보 충

- ElGamal-동작
  - 키 구성
    - 공개키(p, g, h), 개인키 (x)
      - g: p의 원시근
      - h: $g^x \ mod \ p$
      - x: $1 \leq x \leq p-2$의 정수
  - 키 생성
    1. 큰 소수 p와 원시근 g를 선택
    2. 1이상 $p-2$이하의 정수 x 선택
       - $p-1$일 경우, 수학적 결함이 발생
         - $p-1$은 p의 약수와 관련된 성질을 가져, 모듈로 연산에 영향을 미침
         - 페르마 소정리에 의해, $h = g^{p-1} \ mod \ p = 1$이 되어 공개키가 1이 됨
    3. $h = g^x \ mod \ p$ 계산

목 차

- 보충
- 네트워크 계층 소개
- IPv4 주소
- IP 패킷의 전달과 포워딩

네트워크 계층 소개

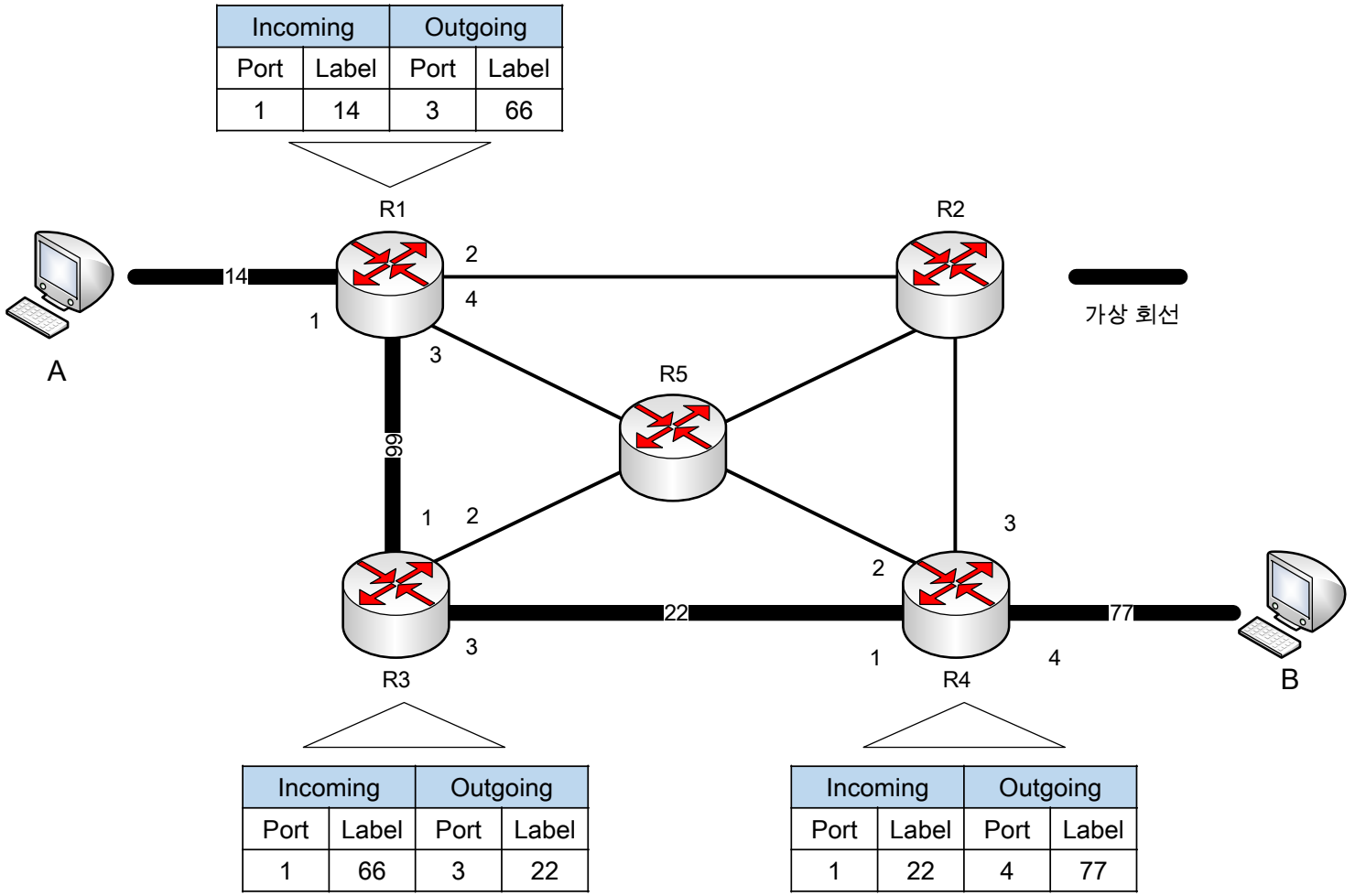
- 정의
  - OSI의 3계층으로, 데이터 전송을 위한 경로 선택 및 패킷 전달을 담당하는 계층

네트워크 계층 소개

- 주요 개념
  - 라우팅(Routing)
    - 라우팅 알고리즘을 사용하여 출발지부터 목적지까지의 경로를 설정해주는 것
      - 라우팅 테이블
        - 정의: 네트워크에서 패킷이 목적지에 도달하기 위해 어떤 경로를 따라야 하는지 결정하는 데 사용되는 데이터 구조
        - 구성: 목적지 주소, 서브넷 마스크, 게이트웨이, 인터페이스, 매트릭
  - 포워딩(Forwarding)
    - 패킷이 라우터의 입력 링크에 도달했을 때, 라우터는 그 패킷을 적절한 출력 링크로 이동시켜는 것
      - 포워딩 테이블
        - 정의: 입력 링크에서 적절한 출력 링크에 도달하기 위한 정보가 있는 데이터 구조
        - 구성: 목적지 주소, 인터페이스

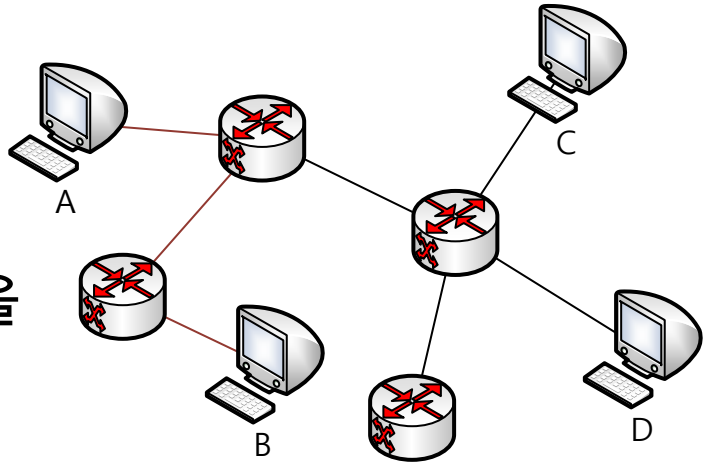
네트워크 계층 소개

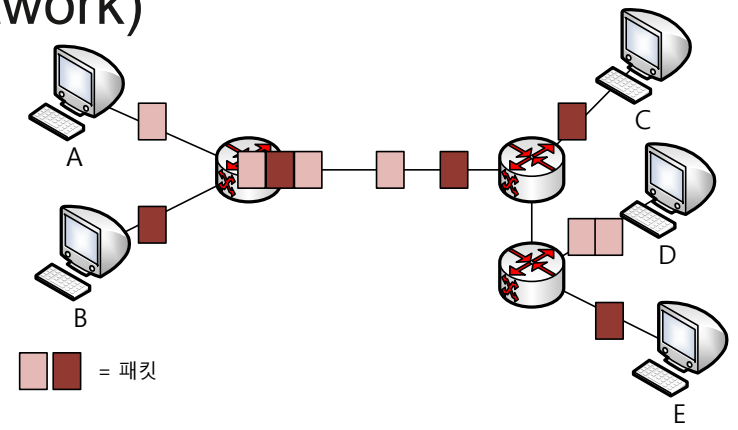
- 주요 개념
  - e.g., 라우팅과 포워딩 활용 예시



네트워크 계층 소개

- 교환(Switching)
  - 정의
    - 호스트 간에 데이터를 전달하는 방식을 설정하는 기능
  - 종류
    - 회선 교환(Circuit Switching)
      - 송수신자 사이에 물리 회선(또는 채널)을 이용하여 메시지를 주고받는 교환 방식
      - e.g., 아날로그 신호 전화 시스템, ISDN(Integrated Services Digital Network)
    - 패킷 교환(Packet Switching)
      - 송신자가 전송하고자 하는 메시지를 일정한 크기로 분할한 후, 전달하여 수신자가 이를 병합하는 교환 방식
      - e.g., IP



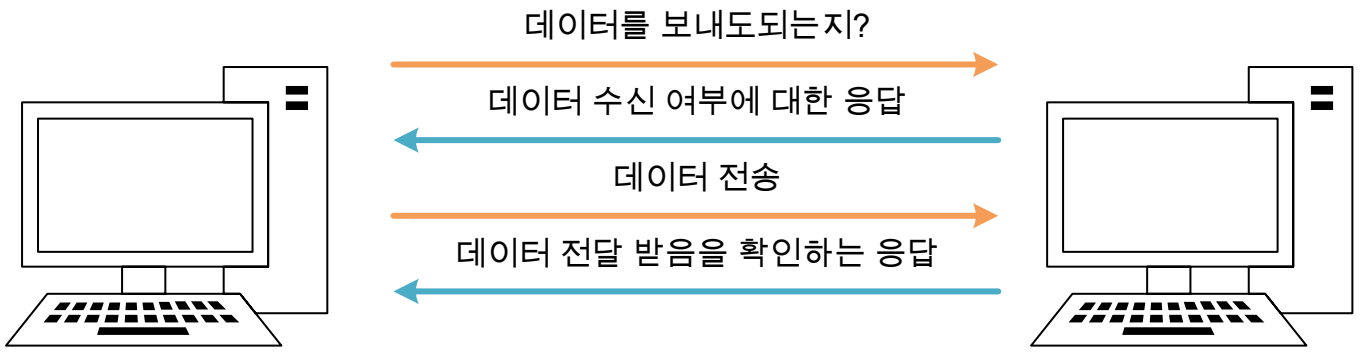


네트워크 계층 소개

- 교환(Switching)
  - 비교
    - 경로
      - 회선 교환
        - 통신이 시작되기 전에 고정된 경로를 설정
        - 경로가 고정되어 있기 때문에 지여, 속도 등의 성능을 예측 가능
      - 패킷 교환
        - 각 패킷이 네트워크를 통해 독립적으로 전달됨
    - 자원
      - 회선 교환
        - 설정된 경로의 모든 자원이 해당 통신에 전용이 됨
        - 경로가 설정된 동안 다른 사용자는 이 자원을 사용할 수 없음
      - 패킷 교환
        - 네트워크 자원이 모든 사용자에게 공유되어 필요한 순간에만 자원을 사용함
        - 패킷이 경로를 동적으로 선택하기 때문에 지연이나 속도가 네트워크 상황에 따라 변할 수 있음

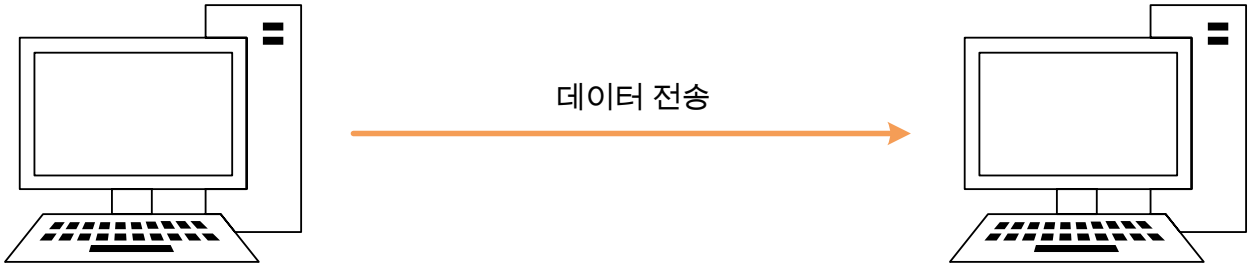
네트워크 계층 소개

- 패킷 교환 방법
  - 연결형 서비스(Connection-Oriented Service)
    - 정의
      - 패킷을 전송하기 전에 송수신 호스트 간에 연결을 수립하는 서비스
    - 장점
      - 수신자가 데이터 순서를 정리할 필요없이 순차적 수신이 가능함
    - 단점
      - 비연결형 서비스에 비해 상대적으로 지연이 많음



네트워크 계층 소개

- 패킷 교환 방법
  - 비연결형 서비스(Connectionless Service)
    - 정의
      - 패킷을 전송하기 전에 송수신 호스트 간 연결 설정 없이 데이터를 패킷 단위로 전송하는 서비스
    - 장점
      - 연결형 서비스에 비해 지연이 적음
    - 단점
      - 패킷이 독립적으로 전달되어 수신자가 별도로 패킷의 순서를 정리해야 함
      - 오류 검출, 재전송 메커니즘 결여

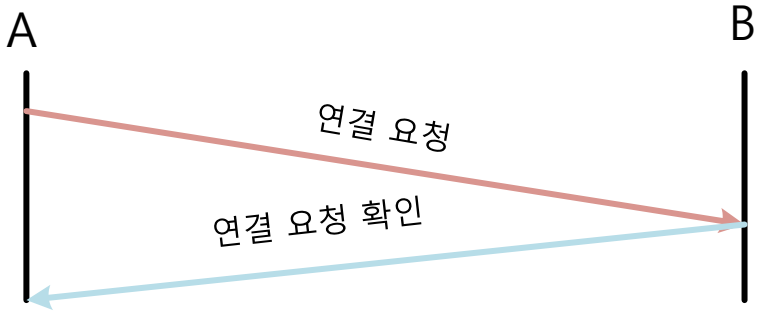


네트워크 계층 소개

- 연결형 서비스 – 통신 과정(1/2)

*엔트리
라우터나 스위치의 테이블에 저장된 정보
- 출발지 및 도착지 IP 또는 MAC 주소
- 포트 번호
- 프로토콜 정보
- 상태 정보 및 우선순위 등

1. 설정 단계
   - 정의
     - 라우터가 가상 회선을 위한 엔트리를 생성하고 요청 및 응답하는 단계
   - 특징
     - A에서 B로 가상 회선을 수립할 때, 송수신자 사이에 요청 패킷과 응답 패킷이 교환됨
       - 요청 패킷
         - A가 B에게 연결을 요청하는 패킷
       - 응답 패킷
         - B가 A의 요청을 수신하고, 요청에 대한 응답을 패킷을 전송

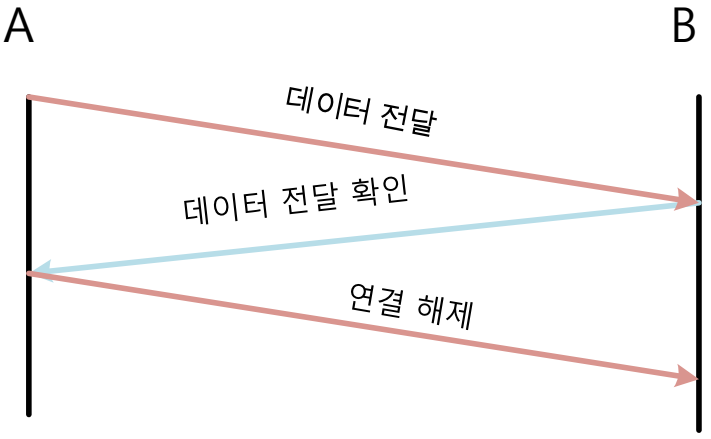


네트워크 계층 소개

- 연결형 서비스 – 통신 과정(2/2)

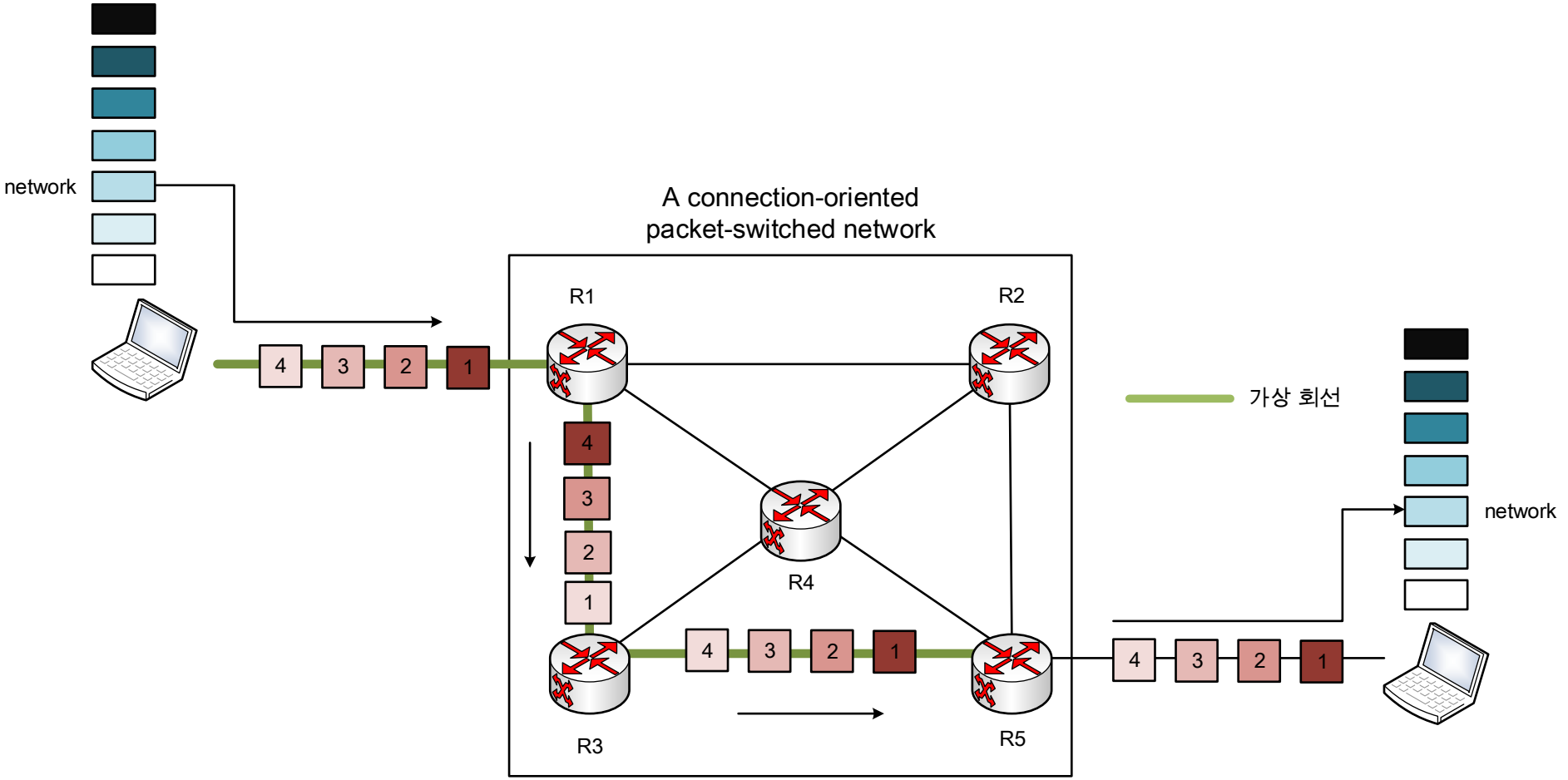
2. 전송 단계
   - 설정된 연결을 통해 데이터가 전송되는 단계

3. 연결 해제 단계
   - 호스트 간의 연결을 종료하고 기록된 엔트리를 삭제하는 단계
     - e.g., TCP 4way handshake



네트워크 계층 소개

- 연결형 서비스 – 패킷 전송 과정

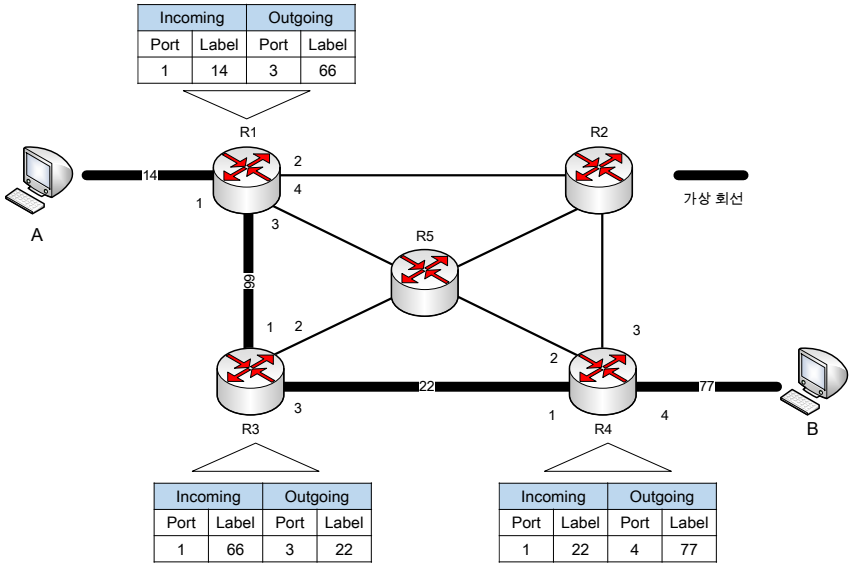


네트워크 계층 소개

- 연결형 서비스 – 패킷 전송 과정
  - 발신지 A가 R1에게 요청 패킷을 보냄
  - R1이 요청 수신
    - R1은 A에서 B로 가는 패킷이 포트 3으로 가야 함을 알고 있음
    - 출력 포트 3을 통해 R3로 패킷 전달
  - R3이 R1의 설정 요청 수신
    - 입력포트 (1), 출력포트(3)
  - R4이 R3의 설정 요청 수신
    - 입력포트(1), 출력포트(4)
  - 목적지 B가 설정 패킷 수신 후 A로부터 패킷을 수신할 준비가 되어있으면 A로부터 온 패킷에 레이블을 할당함
    - 레이블(77) 할당

네트워크 계층 소개

- 연결형 서비스 – 패킷 전송 과정
  - 목적지 B가 발신지 A와 통신하기 위해 할당한 레이블(77)이 R4에서 출력 레이블이 됨
  - R4는 설정 단계에서 선정된 자신의 입력 레이블이 포함된 확인응답을 R3에 보냄
    - R3과 R1 동일
  - 해당 테이블에 적힌 입력 및 출력 레이블을 기반으로 A가 B에게 패킷을 전달

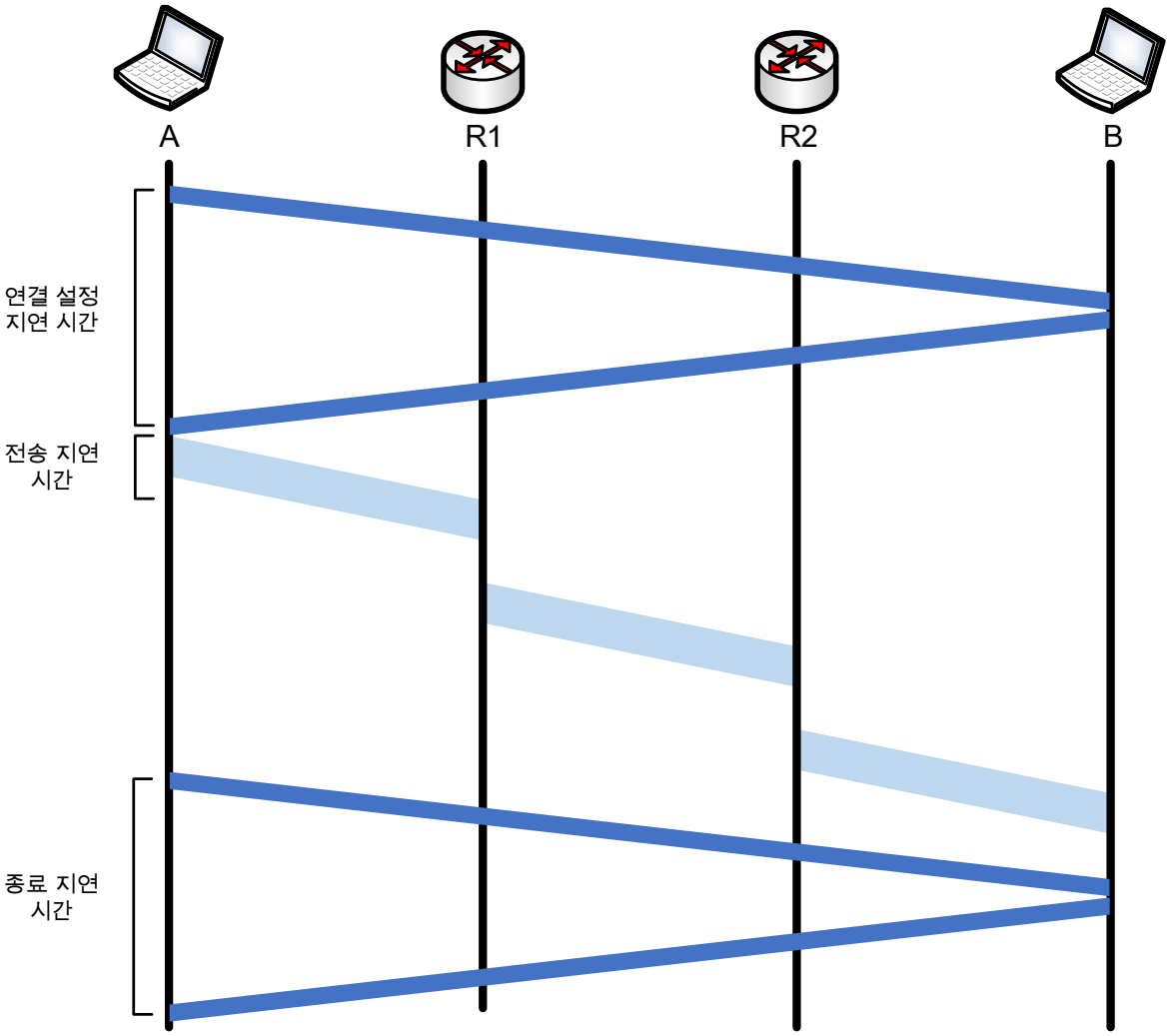


네트워크 계층 소개

- 연결형 서비스 - 문제점(1/2)
  - 지연
    - 연결 설정 지연
      - 데이터 전송을 시작하기 전 연결을 설정하는 과정에서 지연이 발생
    - 전송 지연
      - 데이터가 전송되는 동안 발생하는 지연으로 패킷의 크기와 혼잡도에 따라 달라짐
    - 종료 지연
      - 데이터 전송이 완료된 후, 연결을 종료하는 과정에서 발생하는 지연

네트워크 계층 소개

- 연결형 서비스 – 문제점(2/2)



네트워크 계층 소개

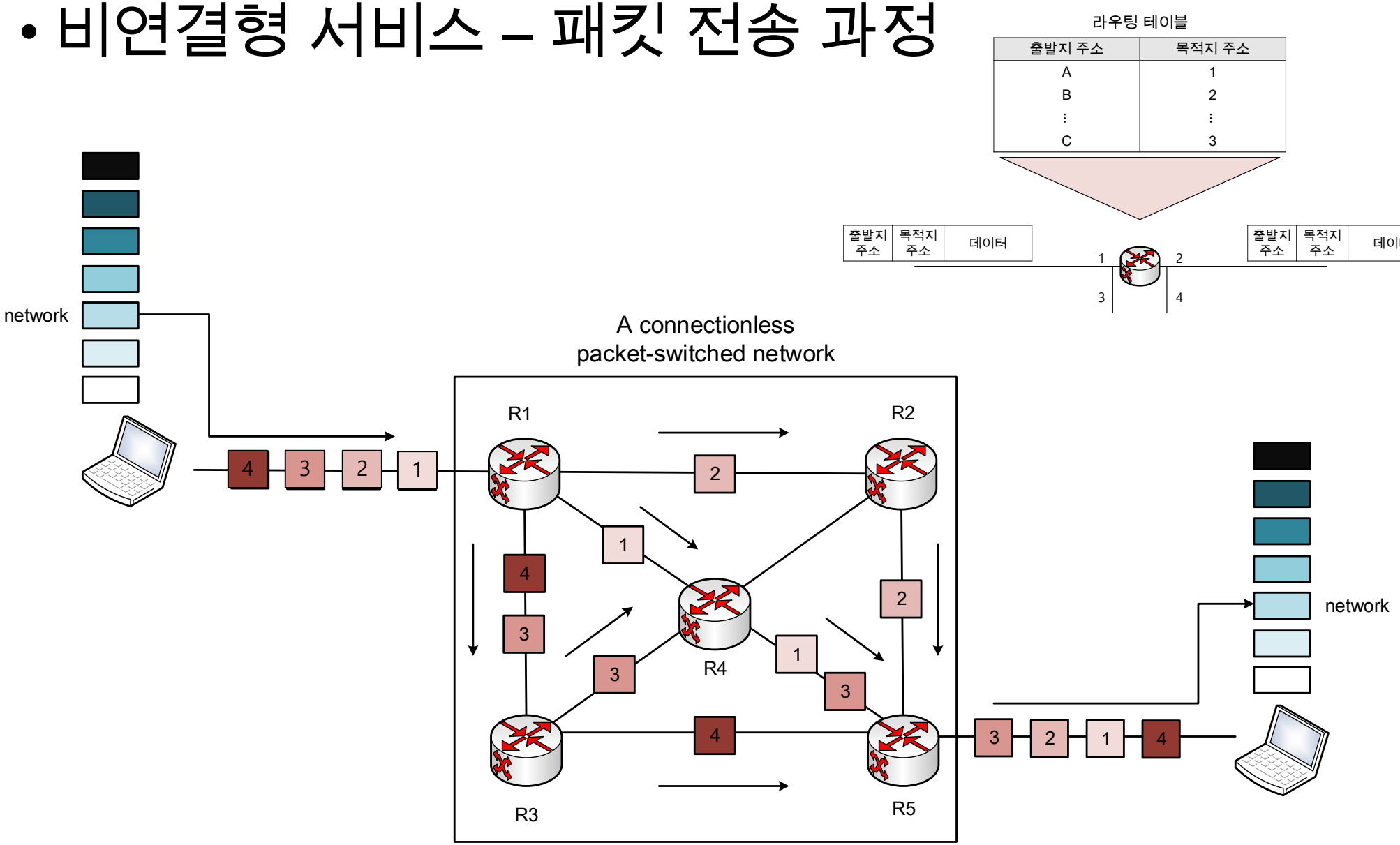
- 비연결형 서비스 - 포워딩 과정

1. 전송 단계
   - 패킷 생성
     - 송신자가 수신자에게 전송할 패킷을 생성
     - 패킷에는 수신자의 주소가 포함되어 있음
   - 패킷 전송
     - 송신자가 생성한 패킷을 네트워크를 통해 수신자에게 전송
     - 연결 설정이 따로 필요 없으므로 각 패킷은 독립적으로 전송됨

2. 응답 단계
   - 응답 패킷을 생성하고 이를 전송함
     - 연결 설정이 따로 필요 없어 패킷은 독립적으로 전송됨

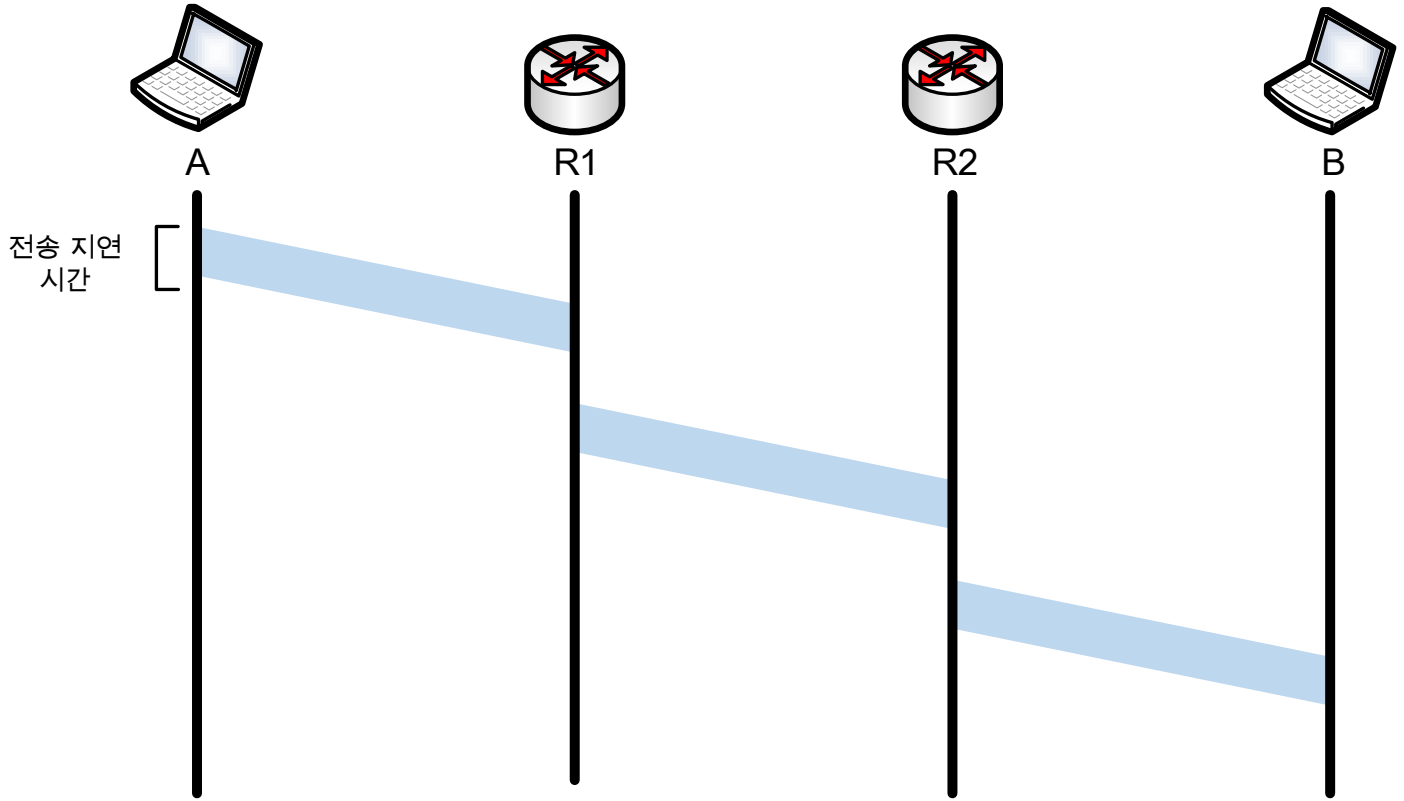
네트워크 계층 소개

- 비연결형 서비스 – 패킷 전송 과정



네트워크 계층 소개

- 비연결형 서비스 - 지연
  - 처리 지연
    - 패킷 전송 과정에서 걸리는 시간으로 인해 지연 발생

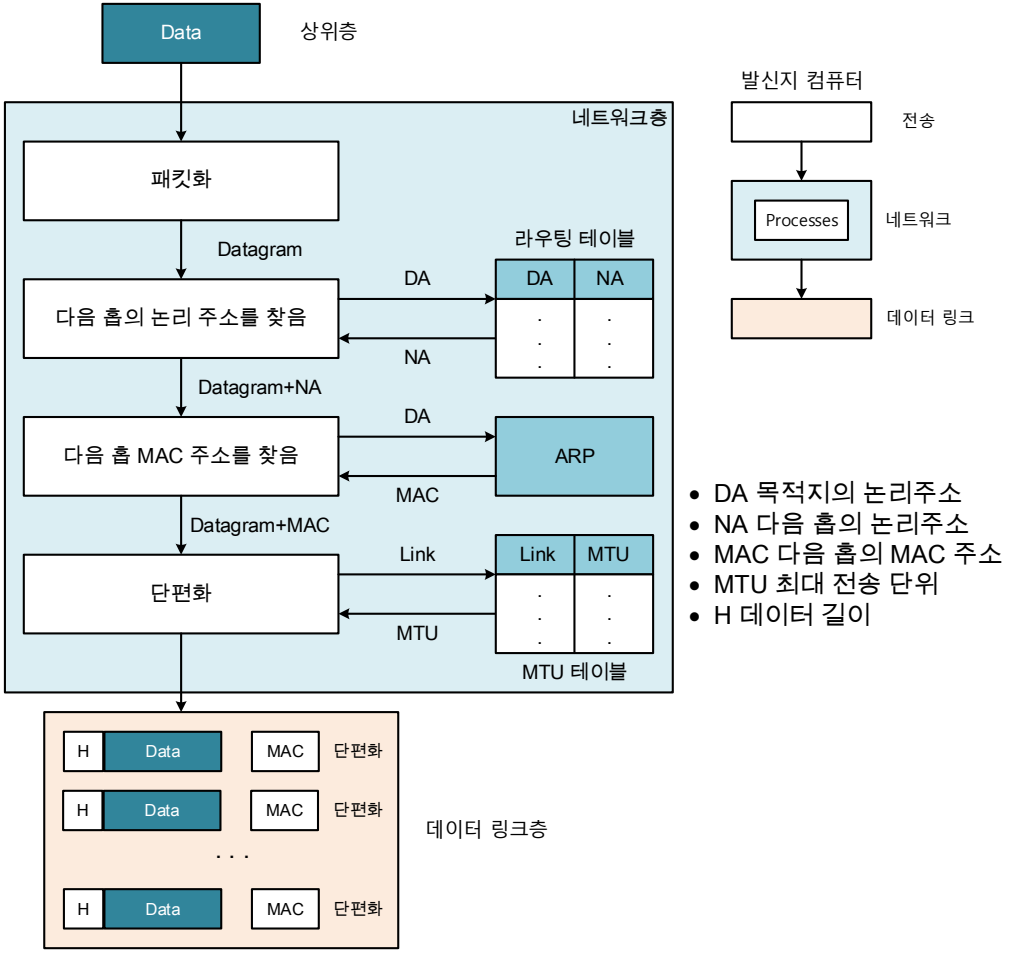


네트워크 계층 소개

- 네트워크 계층 서비스
  - 논리적 주소(Logical address)
    - 정의
      - 네트워크 장치나 호스트를 식별하기 위해 사용되는 주소
    - 특징
      - 동일한 네트워크에 있는 장치들은 서로 다른 논리 주소를 가짐
      - e.g., IP 주소

네트워크 계층 소개

- 네트워크 계층 서비스 - 발신지(1/5)
  - 역할
    - 데이터 전송을 시작하는 컴퓨터로, 전송할 데이터를 패킷으로 분할하고, 목적지 IP주소를 포함하여 수신자의 네트워크 계층에 전달

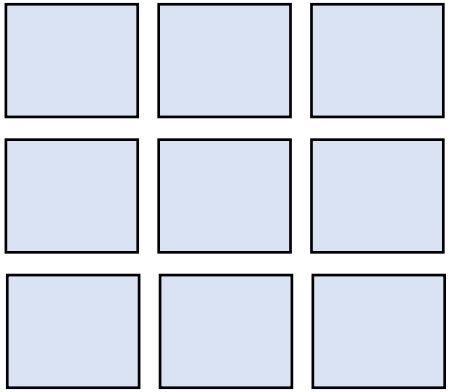


네트워크 계층 소개

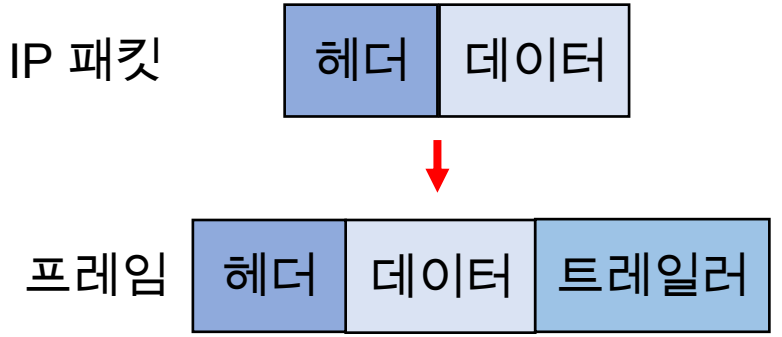
- 네트워크 계층 서비스 - 발신지(2/5)
  - 패킷화(Packetizing)
    - 정의
      - 상위층으로부터 받은 데이터를 데이터그램으로 캡슐화 하는 것
    - 특징
      - 데이터 분할, 패킷 전송, 패킷 재조립으로 과정이 이루어짐
      - 전송할 데이터가 생성되면 이를 헤더와 페이로드로 구성된 패킷으로 나누어 전송

* 헤더
패킷의 출발지 및 목적지 주소, 순서 번호, 프로토콜 등의 정보

원본 데이터



패킷화



네트워크 계층 소개

- 네트워크 계층 서비스 - 발신지(3/5)
  - 다음 홉의 논리 주소 찾기
    - 데이터 그램의 목적지 주소를 보고 라우팅 테이블을 참고하여 다음 라우터의 논리 주소를 찾는 것

목적지	서브넷 마스크	다음 홉	매트릭	인터페이스
192.168.1.0	255.255.255.0	192.168.1.1	35	Eth0
192.168.2.0	255.255.255.0	192.168.1.2	1	Eth1
59.3.135.0	255.255.255.0	-	23	Eth2
0.0.0.0	0.0.0.0	59.3.135.254	15	Eth3

네트워크 계층 소개

- 네트워크 계층 서비스 - 발신지(4/5)
  - 다음 홉 MAC(Media Access Control) 주소 찾기
    - 다음 홉의 논리 주소를 ARP(Address Resolution Protocol)를 통해 MAC(Media Access Control)주소로 변환하는 것
      - ARP: 논리주소를 물리주소로 변환시켜주는 프로토콜
      - MAC: 물리적 네트워크 세그먼트의 통신을 위해 네트워크 인터페이스에 할당된 고유 식별자

```
무선 LAN 어댑터 로컬 영역 연결* 1:

   미디어 상태 . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :
   설명. . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   물리적 주소 . . . . . : C8-09-A8-8E-59-8C
   DHCP 사용 . . . . . : 예
   자동 구성 사용. . . . . : 예
```

```
C:\Users\Owner>arp -a

인터페이스: 192.168.183.1 --- 0x5
  인터넷 주소           물리적 주소           유형
  192.168.183.255       ff-ff-ff-ff-ff-ff     정적
  224.0.0.22            01-00-5e-00-00-16     정적
  224.0.0.251           01-00-5e-00-00-fb     정적
  224.0.0.252           01-00-5e-00-00-fc     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적
```

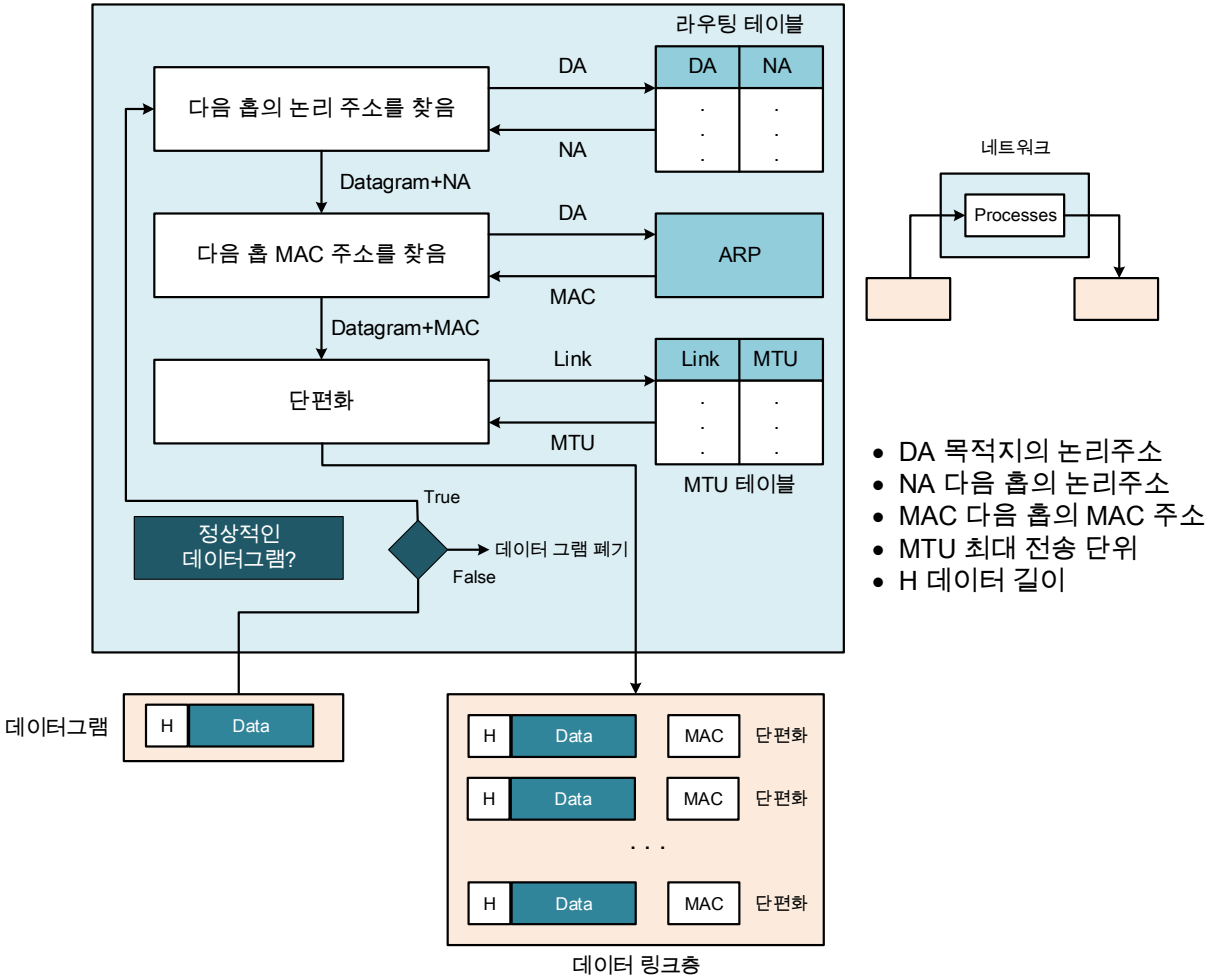
네트워크 계층 소개

- 네트워크 계층 서비스 - 발신지(5/5)
  - 단편화(Fragmentation)
    - 정의
      - 데이터그램의 크기가 전송가능한 MTU(Maximum Transfer Unit) 크기보다 클 경우, 분할하여 전송하는 것
    - e.g., 데이터그램의 크기 4000byte, MTU는 1500byte
      - 4000byte = 20byte(IP 헤더값)+3980byte(페이로드값)
      - 1500byte = 20byte(헤더값)+1480byte(페이로드값)
      - 즉, 데이터 필드 부분인 3980byte를 1480byte로 쪼갠다면, 1480, 1480, 1020 byte의 페이로드 크기를 가진 데이터그램 세조각으로 나눌 수 있음

*MTU
물리 네트워크로 전달될 수 있는 최대 IP 데이터그램의 크기

네트워크 계층 소개

- 네트워크 계층 서비스 – 라우터(1/2)



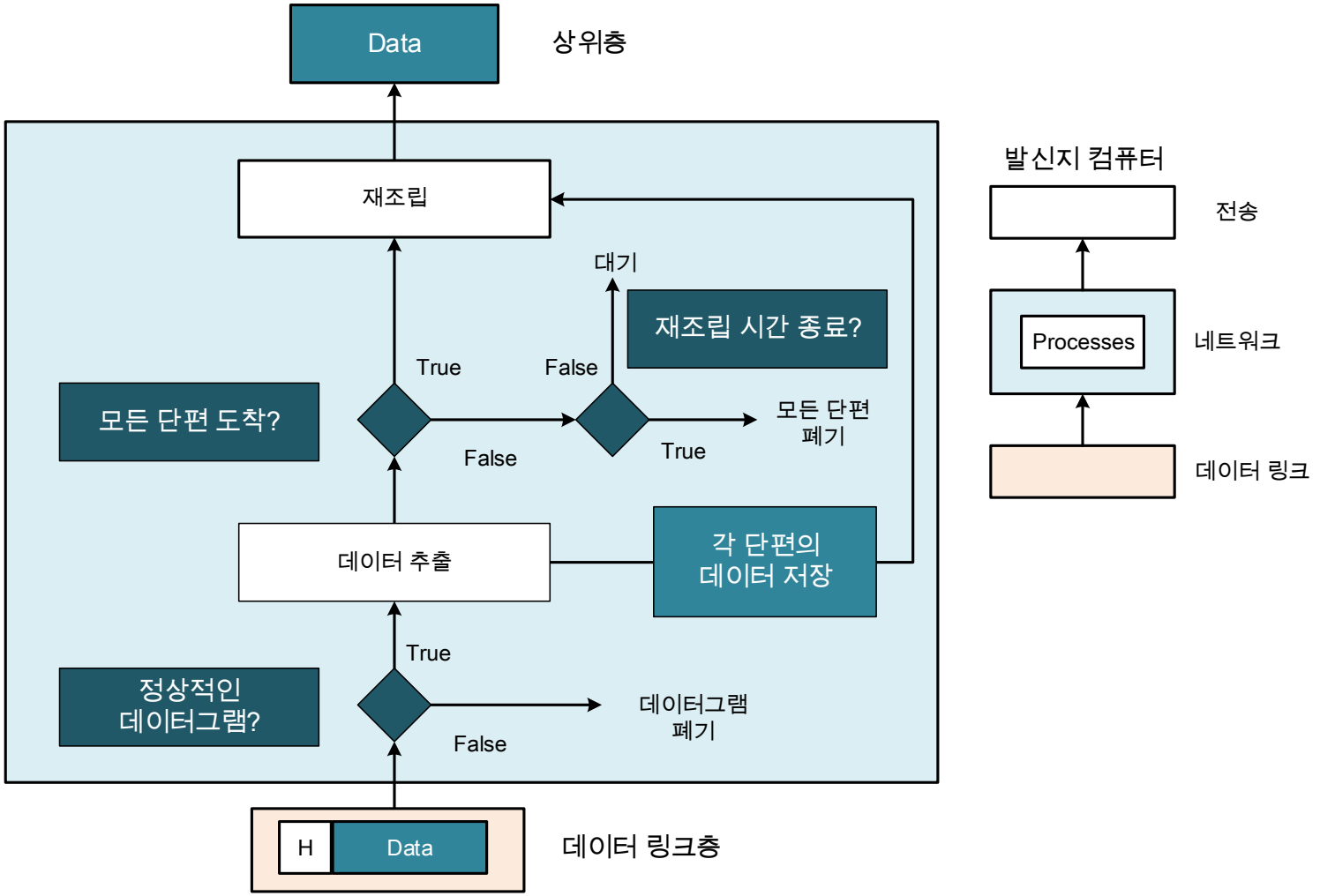
- DA 목적지의 논리주소
- NA 다음 홉의 논리주소
- MAC 다음 홉의 MAC 주소
- MTU 최대 전송 단위
- H 데이터 길이

네트워크 계층 소개

- 네트워크 계층 서비스 – 라우터(2/2)
  - 데이터그램 유효성 검사
    - 데이터그램 헤더의 무결성을 검사하여 데이터그램이 올바르게 전달되었는지 확인
    - e.g., 체크섬(Checksum)
      - 송신 측에서 데이터그램을 생성할 때, 헤더와 페이로드를 모두 더한 후, 그 합의 보수를 계산하고 수신 측에서 수신한 데이터그램의 헤더와 페이로드에 대해 동일한 체크섬 계산을 수행하고 위의 값과 비교
        - 일치하면 데이터그램이 손상되지 않았다고 간주

네트워크 계층 소개

- 네트워크 계층 서비스 – 목적지(1/2)



네트워크 계층 소개

- 네트워크 계층 서비스 – 목적지(2/2)
  - 역할
    - 데이터를 수신하는 컴퓨터로, 발신지에서 전송된 패킷을 받아 처리
    - 데이터그램 유효성 검사
      - 수신한 데이터그램에 체크섬을 계산하여 데이터 무결성을 확인
    - 전달
      - 수신한 데이터그램의 목적지 포트 번호를 확인하고 상위계층으로 전달
    - 패킷 재조립
      - 데이터그램이 여러 개의 조각으로 나누어 전송된 경우, 원래의 데이터로 복원

네트워크 계층 소개

- 문제점
  - 오류 수정
    - 체크섬으로 오류를 감지할 수는 있으나, 이를 수정하는 기능이 없음
      - 패킷이 손상된 상태로 전송될 수 있으며 이로 인해 잘못된 데이터가 애플리케이션에 전달될 수 있음
  - 흐름 제어
    - 발신지 컴퓨터가 목적지 컴퓨터가 처리할 수 있는 양보다 많은 데이터를 생성하여 전송할 수 있음
      - 수신 측의 버퍼가 포화 상태가 되면 패킷 손실이 발생할 수 있음

네트워크 계층 소개

- 문제점
  - 서비스 품질 (QoS, Quality of Service)
    - 서비스마다 최적화된 품질 제공에 어려움이 있음
      - 대역폭 부족, 지연, 혼잡 및 오류 등으로 인해 서비스가 불안정해지거나 품질의 저하가 존재함
  - 라우팅
    - 네트워크 규모 확대, 라우팅 경로 변경 등에 따른 성능 저하에 대한 문제를 고려해야 함
      - e.g., 네트워크 규모가 커질수록 라우팅 테이블의 크기도 증가하게 되며 라우팅 정보를 관리하는 데 필요한 메모리와 처리 능력이 부족할 경우, 성능 저하 발생

네트워크 계층 소개

- 문제점
  - 보안 위협
    - 데이터 무결성 및 기밀성 부족
      - 암호화하지 않거나 신뢰할 수 없는 공개 네트워크를 사용할 경우, 데이터그램이 전송되는 동안 변조되거나 도청될 수 있음
    - 패킷 분할 및 재조립 취약성
      - 패킷을 분할하고 재조립하는 과정에서 공격자가 패킷을 변형하거나 삭제할 수 있는 위험이 있음

목 차

- 보충
- 네트워크 계층 소개
- IPv4 주소
- IP 패킷의 전달과 포워딩

IPv4 주소

- 정의
  - 네트워크에서 호스트 간의 데이터 전송을 위해 경로 및 목적지를 지정하는 32비트 체계의 고유한 주소
- 특징(1/2)
  - 주소 공간을 지님
    - 모든 네트워크에서 주소 지정을 통해 장비를 유일하게 식별
  - 비연결형
    - 송수신 장비 간 논리적 연결을 수립하지 않고 데이터 전송
    - 데이터그램 전달 기능만을 담당하여 기타 다른 기능을 제공하지 않음
      - e.g., 흐름 제어, 재전송

IPv4 주소

- 특징(2/2)
  - 한정된 주소범위를 가짐
    - 주소 고갈 문제를 야기시켜 새로운 주소를 할당하기 어려움
  - 2진수, 10진수, 16진수 표기법도 가능

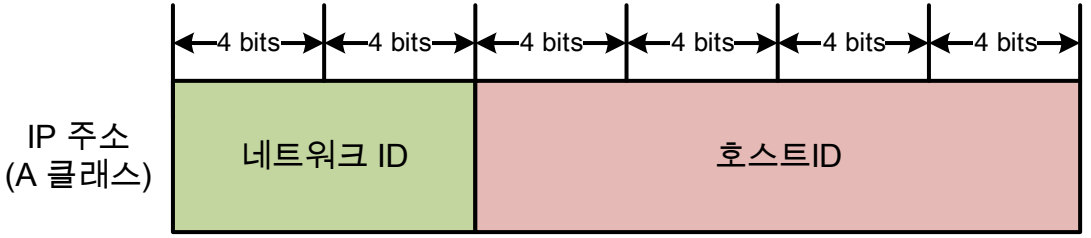
16 HEX	CA	67	00	44
10 DEC	**202**	**103**	**0**	**68**
2 BIN	11001010	01100111	00000000	01000100

IPv4 주소

- 기능
  - 인터페이스 식별
    - 데이터그램이 올바른 수신자에게 전달되는 것을 보장하기 위해 장비와 네트워크의 인터페이스를 식별할 수 있도록 함
  - 라우팅 지원
    - 송수신 측이 다른 네트워크에 있는 경우, 라우터를 통해 데이터를 간접 전달

IPv4 주소

- 구조
  - 네트워크 ID
    - 정의
      - 맨 왼쪽에서 시작하는 특정 수의 비트
    - 역할
      - 호스트가 위치한 네트워크를 식별하는 데 사용됨
  - 호스트 ID
    - 정의
      - 네트워크 ID를 제외한 나머지 비트
    - 역할
      - 네트워크의 호스트를 식별하는 데 사용됨



IP 주소: 127.10.2.56
네트워크 주소: 127.0.0.0
호스트 주소: 0.10.2.56

IPv4 주소

- 구조
  - 예제 5.1

다음 2진(또는 10진) 표기법 IPv4 주소를 점 10진(또는 2진) 표기법으로 변환하라.

1. 10000001 00001011 00001011 11101111
2. 111.56.45.78

- 1번 풀이
  - $(10000001)_2 \rightarrow (129)_{10}$
  - $(00001011)_2 \rightarrow (11)_{10}$
  - $(00001011)_2 \rightarrow (11)_{10}$
  - $(11101111)_2 \rightarrow (239)_{10}$

  ∴ 129.11.11.239

- 2번 풀이
  - $(111)_{10} \rightarrow (01101111)_2$
  - $(56)_{10} \rightarrow (00111000)_2$
  - $(45)_{10} \rightarrow (00101101)_2$
  - $(78)_{10} \rightarrow (01001110)_2$

  ∴ 01101111 00111000 00101101 01001110

IPv4 주소

- 구조
  - 예제 5.2

다음 IPv4 주소 표기법에서 잘못된 점을 찾아보라.

1. 111.56.045.78
2. 221.34.7.8.20
3. 75.45.301.14
4. 11100010.23.14.67

- 풀이
  1. 점 10진 표기법에서 0이 맨 앞에 나와서는 안 됨
  2. IPv4 주소는 4바이트보다 많으면 안 됨
  3. 점 10진 표기법에서 각 숫자는 255보다 작거나 같아야 함
  4. 2진 표기법과 점 10진 표기법을 혼합해서 사용해선 안 됨

IPv4 주소

- 구조
  - 예제 5.3

다음 2진 표기법 IPv4 주소를 16진 표기법으로 변환하라.
10000001 00001011 00001011 11101111

- 풀이
  - $(10000001)_2 \rightarrow (81)_{16}$
  - $(00001011)_2 \rightarrow (0B)_{16}$
  - $(00001011)_2 \rightarrow (0B)_{16}$
  - $(11101111)_2 \rightarrow (EF)_{16}$

  $\therefore$ 0x810B0BEF

IPv4 주소

- 주소 범위
  - 정의
    - 네트워크에서 사용할 수 있는 주소의 집합
  - 특징
    - 한 개의 주소 대신에 주소의 범위가 필요한 경우가 있음
      - e.g., 네트워크 모니터링, 서브넷 설계
    - 0.0.0.0부터 255.255.255.255까지 정의되어 있음
      - e.g., 처음 주소와 마지막 주소가 주어졌을 경우
        - 처음 주소: 146.102.29.0
        - 마지막 주소: 146.102.32.225
        - 마지막 주소 – 처음 주소 = 0.0.3.225
        - 주소의 수 $(0 \times 256^3 + 0 \times 256^2 + 3 \times 256^1 + 255 \times 256^0) + 1 = 1024$
    - 각 연산은 NOT, AND, OR 사용

IPv4 주소

- 연산(NOT)
  - 예제 5.4

다음 수로 이진법으로 표기된 32비트 수에 NOT 연산을 적용하는 방법을 설명하라.
00010001 01111001 00001110 00100011

- 풀이
  - 00010001 01111001 00001110 00100011
  - 원래의 수: 17.121.14.35
  - 11101110 10000110 11110001 11011100
  - 보수: 238.134.241.220

IPv4 주소

- 연산(AND)
  - 예제 5.5(1/3)

다음 수로 이진법으로 표기된 두 개의 32비트에 AND 연산을 적용하는 방법을 보여라.

1. 00010001 01111001 00001110 00100011
2. 11111111 11111111 10001100 00000000

- 첫 번째 풀이
  - 오른쪽 자리의 수부터 차례대로 AND 연산을 수행함
  - 00010001 01111001 00001100 00000000

IPv4 주소

- 연산(AND)
  - 예제 5.5(2/3)

다음 수로 이진법으로 표기된 두 개의 32비트에 AND 연산을 적용하는 방법을 보여라.

1. 00010001 01111001 00001110 00100011
2. 11111111 11111111 10001100 00000000

- 두 번째 풀이(1/2)
  - 숫자들이 점 10진법으로 표기된 경우
    - (A)수 중 적어도 하나가 0 또는 255라면 AND 연산은 작은 바이트 값을 취함
    - (B) 두 바이트가 0도 아니고 255도 아니라면 각 바이트를 8항의 합으로 쓸 수 있음
  - 00010001 01111001 00001110 00100011 = 17.121.14.35
  - 11111111 11111111 10001100 00000000 = 255.255.140.0
  - 첫 번째, 두 번째, 네 번째 바이트는 (A) 방법을, 세 번째 바이트는 (B) 방법을 적용

IPv4 주소

- 연산(AND)
  - 예제 5.5(3/3)

다음 수로 이진법으로 표기된 두 개의 32비트에 AND 연산을 적용하는 방법을 보여라.

1. 00010001 01111001 00001110 00100011
2. 11111111 11111111 10001100 00000000

- 두 번째 풀이(2/2)
  - 세 번째 바이트 계산 과정
    - 14 = $2^3 + 2^2 + 2^1$
    - 140 = $2^7 + 2^3 + 2^2$
    - 같은 거듭 제곱 쌍이 있는 경우, 이 수들을 더 함
      - $2^3 + 2^2 = 8 + 4 = 12$

∴ 17.121.12.0

IPv4 주소

- 연산(OR)
  - 예제 5.6(1/2)

다음 수로 이진법으로 표기된 두 개의 32비트에 OR 연산을 적용하는 방법을 보여라.
1. 00010001 01111001 00001110 00100011
2. 11111111 11111111 10001100 00000000

- 첫 번째 풀이
  ∴ 11111111 11111111 10001110 00100011

- 두 번째 풀이(1/2)
  - 숫자들이 점 10진법으로 표기된 경우
    - (A)수 중 적어도 하나가 0 또는 225라면 OR 연산은 큰 바이트 값을 취함
    - (B) 두 바이트가 0도 아니고 255도 아니라면 각 바이트를 8항의 합으로 쓸 수 있음
  - 첫 번째 수: 17.121.14.35
  - 두 번째 수: 255.255.140.0

IPv4 주소

- 연산(OR)
  - 예제 5.6(2/2)

다음 수로 이진법으로 표기된 두 개의 32비트에 OR 연산을 적용하는 방법을 보여라.

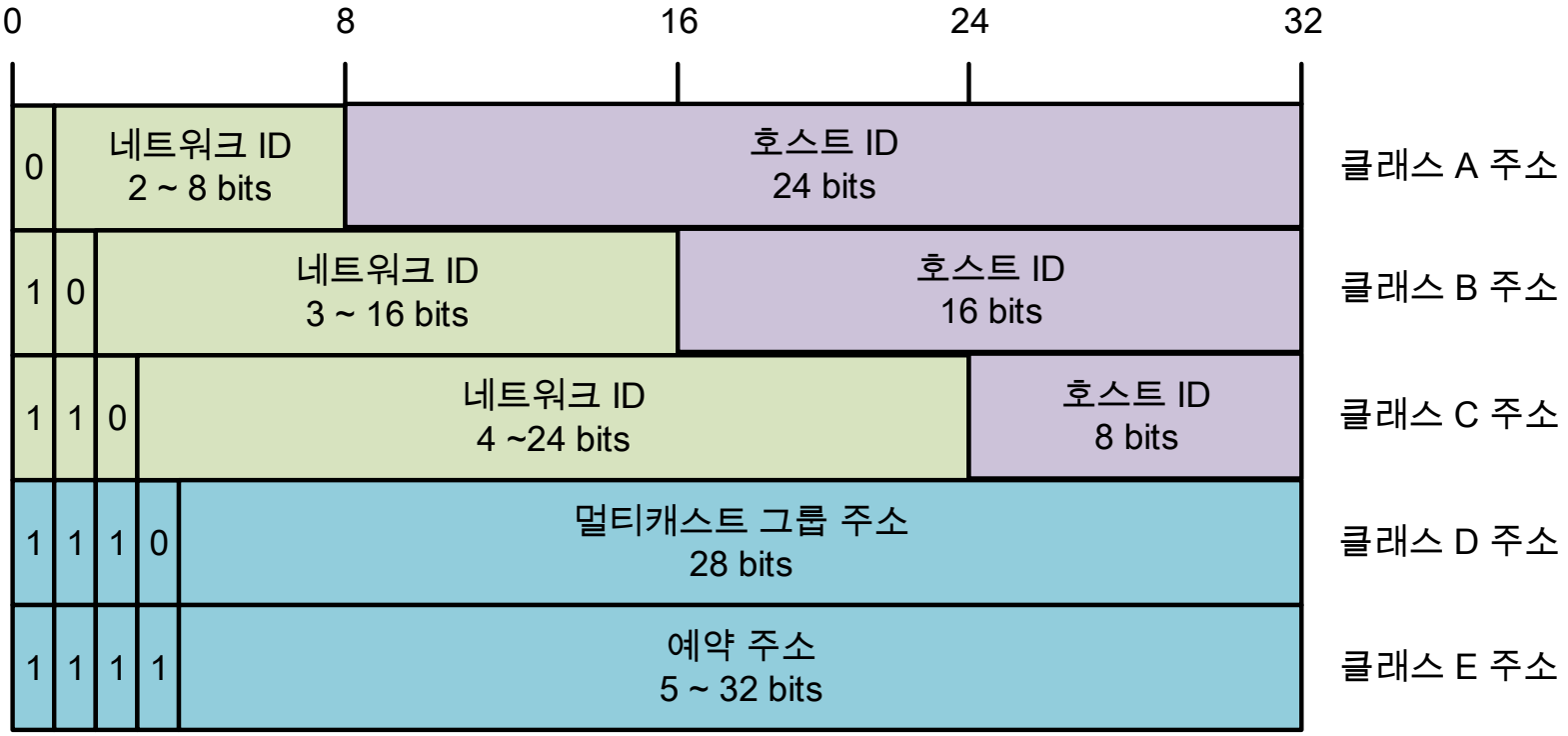
1. 00010001 01111001 00001110 00100011
2. 11111111 11111111 10001100 00000000

- 두 번째 풀이(2/2)
  - 세 번째 바이트 계산 과정
    - 14 = $2^3 + 2^2 + 2^1$
    - 140 = $2^7 + 2^3 + 2^2$
    - 같은 거듭 제곱 쌍이 있는 경우, 이 쌍 중 하나를 남기고 지운 후 모든 수와 더함
      - $2^7 + (2^3 + 2^2) + 2^1 = 128 + 8 + 4 + 1$

∴ 255.255.142.35

IPv4 주소

- 클래스 기반 주소 지정
  - 정의
    - IPv4 주소 체계에서 사용되는 방식으로, IP 주소 공간을 다섯 개의 클래스(A, B, C, D, E)로 나누어 배분하는 방법



IPv4 주소

- 클래스 기반 주소 지정
  - 예제 5.7

각 주소의 클래스를 나타내어라.
1. 00000001 00001011 00001011 11101111
2. 10100111 11011011 10001011 01101111

- 첫 번째 풀이
  - 첫 번째 비트가 0이므로 클래스 A

- 두 번째 풀이
  - 첫 번째 비트가 1이고, 두 번째 비트가 0이므로 클래스 B

IPv4 주소

- 클래스 기반 주소 지정
  - 특징(1/3)
    - 각 클래스마다 네트워크 ID와 호스트 ID의 비율이 다르며 사용하는 분야도 다름
      - e.g., IP 주소 클래스 종류별 특성 표

IP 주소 클래스	전체 IP 주소 공간에서 차지하는 비율	네트워크 ID 비트 수	호스트 ID 비트 수	용도
클래스 A	$^1/_3$	8	24	• 대규모 네트워크에 적합함 • 많은 수의 호스트를 지원($2^{24}-2$)
클래스 B	1	16	16	• 중규모의 네트워크에 적합함 • 클래스 A보다는 적은 수의 호스트를 지원($2^{16}-2$)
클래스 C	$^3/_1$	24	8	• 소규모의 네트워크에 적합함(2^8-2)
클래스 D	-	X	X	• IP 멀티 캐스팅
클래스 E	-	X	X	• 연구용으로 예약됨

IPv4 주소

- 클래스 기반 주소 지정
  - 특징(2/3)
    - 클래스의 상위 비트의 주소 시작 값을 보고 판별할 수 있음
      - e.g., IP 클래스 단위 비트패턴, 주소 범위 표

IP 주소 클래스	비트 패턴	최소값	최대값	첫 번째 옥텟 값의 범위 (10진수)	네트워크 ID 및 호스트 ID에 속한 옥텟 수	이론적 IP 주소 범위
클래스 A	0xxx xxxx	0000 0001	0111 1110	1~127	$^{1}/_{3}$	1.0.0.0 ~ 127.255.255.255
클래스 B	10xx xxxx	1000 0000	1011 1111	128~191	$^{2}/_{2}$	128.0.0.0 ~ 191.255.255.255
클래스 C	110x xxxx	1100 0000	1101 1111	192~223	$^{3}/_{1}$	192.0.0.0 ~ 223.255.255.255
클래스 D	1110 xxxx	1110 0000	1110 1111	224~239	-	224.0.0.0 ~ 239.255.255.255
클래스 E	1111 xxxx	1111 0000	1111 1111	240~255	-	224.0.0.0 ~ 239.255.255.255

IPv4 주소

- 클래스 기반 주소 지정
  - 특징(3/3)
    - 단순성과 명확성
      - 선택할 수 있는 클래스가 적고 클래스 간 구분이 명확
    - 라우팅 용이성
      - 라우터는 특정 주소의 네트워크 ID 와 호스트 ID 를 쉽게 파악할 수 있음
    - 예약 주소
      - 일부 클래스(D, E)는 향후에 필요할 수 있는 일부 주소 영역을 예약

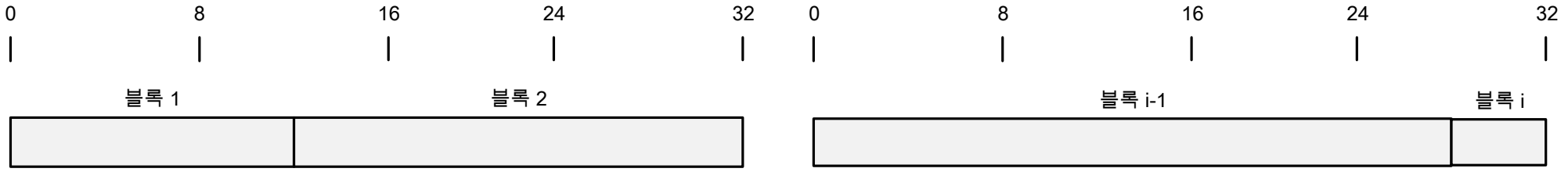
IPv4 주소

- 클래스 기반 주소 지정
  - 문제점
    - 주소 낭비
      - 클래스의 고정된 크기 때문에 작은 네트워크가 클래스 A를 사용하면 많은 IP주소가 낭비
      - 오직 세 가지 블록 크기(클래스 A, B, C) 밖에 없기 때문에 한정된 IP 주소 공간 낭비
    - 유연성 부족
      - 주소 공간을 정적으로 분할하므로, 네트워크 요구사항이 변화할 때 유연하게 대처하기 어려움
    - 라우팅 테이블의 비대화
      - 각 클래스가 특정 범위의 주소를 할당받고 고정된 수의 네트워크가 생성되어 결과적으로 과도하게 많은 네트워크를 생성함
        - 각 네트워크를 구분하는 라우팅 테이블이 이와 비슷하게 많이 생성되어야 함

*블록
연속된 IP 주소의 집합

IPv4주소

- 클래스 없는 주소 지정
  - 정의
    - 기존 클래스 단위 주소 지정 방법에서 클래스를 제외한 주소 지정 방법
  - 특징 (1/2)
    - 공식적으론 클래스 없는 도메인 간 라우팅(CIDR, Classless Interdomain Routing)이라고 함
    - 전체 주소 공간을 가변 길이 블록으로 나눔
      - 블록에 포함되는 주소의 개수는 2의 거듭 제곱이어야 함



IPv4주소

- 클래스 없는 주소 지정
  - 특징 (2/2)
    - 2단계 주소 체계
      - 클래스 기반 주소 지정 방식과 동일한 개념으로, 블록을 프리픽스(prefix)와 서픽스(suffix)로 나눔
      - 프리픽스는 netid, 서픽스는 hostid 기능을 함
      - n은 블록 크기에 따라 0부터 32까지의 수가 될 수 있음

IPv4주소

- 클래스 없는 주소 지정
  - 슬래시(/n) 표기법
    - 주소의 블록을 알기 위해서 프리픽스의 길이(n) 정보가 주소 정보에 포함됨

Byte . Byte . Byte . Byte / n

      - e.g., 192.168.10.1/24

IPv4 주소

- 클래스 없는 주소 지정
  - 예제 5.8

전체 인터넷을 4,294,967,296개의 주소를 갖는 하나의 단일 블록이라고 하면, 인터넷의 프리 픽스 길이와 서픽스 길이는 얼마인가?

- 풀이
  - $2^{32} = 4{,}294{,}967{,}296$ 로, 2의 제곱에 해당하며 32개의 모든 비트가 단일 블록에서 호스트를 구분하기 위해 사용됨
  - 즉, 전체 주소 공간이 하나의 블록으로 간주되고 모든 주소가 하나의 네트워크에 속한다는 것을 의미함

  ∴ 프리픽스 0bit, 서픽스 32bits

IPv4주소

- 클래스 없는 주소 지정
  - 블록 할당
    - (조건 1) 블록에 속하는 주소의 개수로부터 프리픽스 길이의 값을 알 수 있어야 함
      - $N = 2^{32-n}$이므로, $n = 32 - \log_2 N$
    - (조건 2) 요구 주소의 수인 N은 2의 거듭제곱이야 함
      - 프리픽스 길이 n이 정수가 되기 위해 필요함
    - (조건 2) 블록에 속하는 주소의 개수로부터 프리픽스 길이의 값을 알 수 있어야 함
      - $N = 2^{32-n}$이므로, $n = 32 - \log_2 N$
    - (조건 3) 첫 번째 주소는 N으로 나누어질 수 있어야 함
      - 프리픽스를 십진수로 표현한 값을 X라고 하였을 때, $X \times 2^{n-32}$

IPv4 주소

- 클래스 없는 주소 지정
  - 블록 할당
    - 예제 5.9

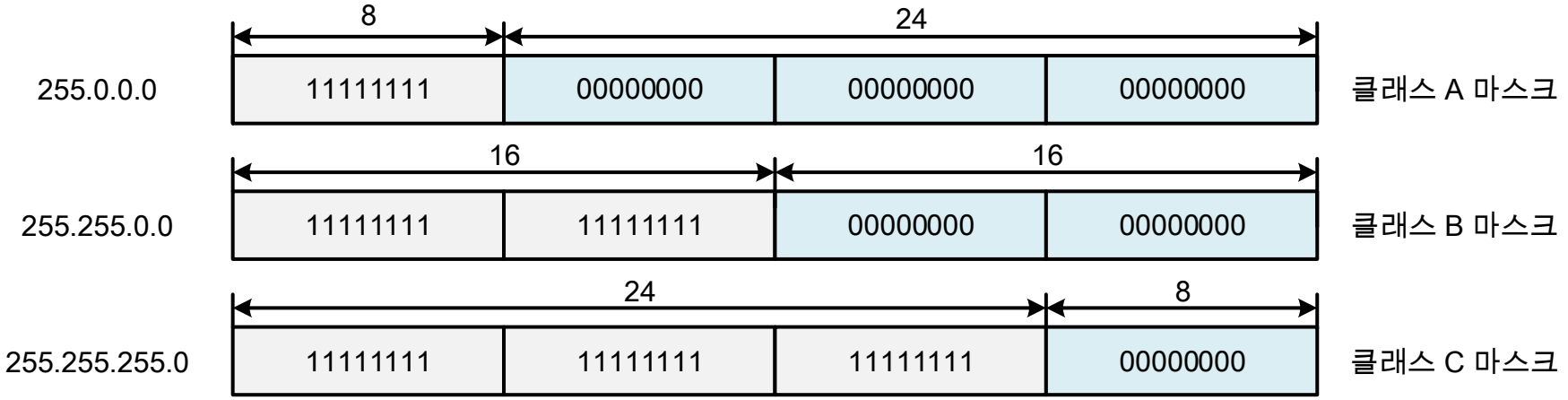
ISP는 1000개의 주소를 갖는 블록을 요청하였을 때 블록 할당 조건을 살펴보라.

- 풀이
  - 1000이 2의 거듭제곱이 아니기 때문에 1024개의 주소가 할당됨
  - 블록의 프리픽스 길이는 $n = 32 - \log_2 N = 32 - 10 = 22$
  - 첫 번째 주소는 1024로 나누어지는 값인 18.14.12.0/22로 선택하였을 때, 마지막 주소는 18.14.15.255/22가 됨

	프리픽스	서픽스
첫 번째 주소	00010010000011110000011	0000000000
마지막 주소	00010010.00001110.000011	1111111111

IPv4주소

- 네트워크 마스크(디폴트 마스크)
  - 정의
    - 네트워크 주소를 추출하기 위해 n개의 왼쪽 비트들을 1로, $32-n$개의 오른쪽 비트들을 0으로 만든 블록
  - 특징
    - 네트워크 마스크와 목적지 주소(또는 임의의 주소)를 AND 연산하면 네트워크 주소임



IPv4 주소

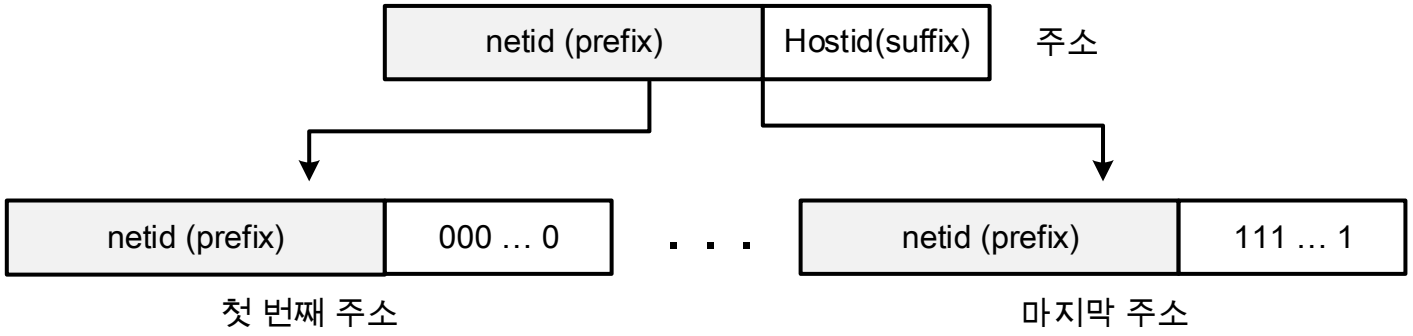
- 네트워크 마스크
  - 예제 5.10

라우터가 목적지 주소가 201.24.67.32인 패킷을 받는다. 라우터가 패킷의 네트워크 주소를 찾는 방법을 보여라.

- 풀이
  - 상위 바이트가 201이므로 클래스 B
  - 클래스 B의 네트워크 마스크인 255.255.0.0을 적용하여 AND 연산함
  - 201.24.67.32(목적지 주소) + 255.255.0.0(네트워크 마스크) = 201.24.0.0
    ∴ 201.24.0.0

IPv4주소

- 블록 정보 추출
  - 표시된 주소는 첫 번째 주소(네트워크 주소), 주소의 개수, 마지막 주소 등과 같은 블록에 포함된 정보를 포함함



주소의 수: $N = 2^{32-n}$

- 블록에 속하는 주소의 개수
  - N은 블록에 속하는 주소 개수, n은 네트워크 ID 또는 프리픽스 길이

$$N = 2^{32-n}$$

IPv4주소

- 블록 정보 추출
  - 첫 번째 주소
    - 첫 번째 방식

      첫 번째 주소 = (임의의 주소) AND (네트워크 마스크)

    - 두 번째 방식
      - 주소의 왼쪽 비트를 n으로 유지하고 오른쪽 $32-n$를 0으로 설정
  - 마지막 주소
    - 첫 번째 방식

      마지막 주소 = (임의의 주소) OR [NOT (네트워크 마스크)]

    - 두 번째 방식
      - 주소의 왼쪽비트는 n으로 유지하고 오른쪽 $32-n$을 1로 설정

IPv4 주소

- 블록 정보 추출
  - 예제 5.11

블록 내의 주소가 73.22.17.25일 때 블록 내의 주소 수, 첫 번째 주소와 마지막 주소를 구하라.

- 풀이
  - 73은 0과 127 사이의 값이므로 클래스 A임
  - 클래스 A의 n값은 8
  - 주소 수: $2^{32-8} = 2^{24}$
  - 왼쪽과 오른쪽 비트를 미루고 각 0과 1로 만듦
    - 1001000 00000000 00000000 00000000
    - 1001000 11111111 11111111 11111111
  - 첫 번째 주소 ~ 마지막 주소 = 73.0.0.0 ~ 73.255.255.255

IPv4 주소

- 블록 정보 추출
  - 예제 5.12

블록 내의 주소가 200.11.8.45일 때 블록 내의 주소 수, 첫 번째 주소와 마지막 주소를 구하라.

- 풀이
  - 200은 192와 223 사이의 값이므로 클래스 C임
  - 클래스 C의 n값은 24
  - 주소 수: $2^{32-24} = 2^8$
  - 왼쪽과 오른쪽 비트를 미루고 각 0과 1로 만듦
  - 첫 번째 주소와 마지막 주소 = 200.11.8.0 ~ 200.11.8.255

IPv4 주소

- 블록 정보 추출
  - 예제 5.13(1/2)

167.199.170.82/27은 블록에 속하는 하나의 주소이다. 이 네트워크에 속하는 주소의 개수와 첫 번째 주소, 마지막 주소를 구하라.

- 풀이(1/2)
  - 주소의 개수
    - $N = 2^{32-27} = 2^5$
  - 네트워크 마스크
    - n 값이 27이므로, 네트워크 마스크는 27개의 1과 5개의 0으로 구성
      - 11111111 11111111 11111111 11100000
  - 첫 번째 주소
    - 임의의 주소 $\oplus$ 네트워크 마스크
      = 10100111 11000111 10101010 01010010
      $\oplus$ 11111111 11111111 11111111 11100000
      = 10100111 11000111 10101010 01000000

IPv4 주소

- 블록 정보 추출
  - 예제 5.13(2/2)

167.199.170.82/27은 블록에 속하는 하나의 주소이다. 이 네트워크에 속하는 주소의 개수와 첫 번째 주소, 마지막 주소를 구하라.

- 풀이(2/2)
  - 마지막 주소
    - 네트워크 마스크: 11111111 11111111 11111111 11100000
    - 네트워크 마스크 보수: 00000000 00000000 00000000 00011111
    - 임의의 주소 | ~네트워크 마스크
      = 10100111 11000111 10101010 01011111

IPv4 주소

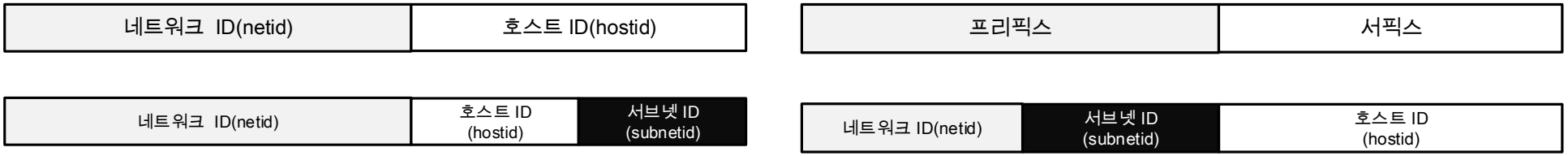
- 블록 정보 추출
  - 예제 5.14

110.23.120.14/20은 블록에 속하는 하나의 주소이다. 이 네트워크에 속하는 주소의 개수와 첫 번째 주소, 마지막 주소를 구하라.

- 풀이
  - 주소의 개수
    - $N = 2^{32-20} = 2^{12}$
  - 네트워크 마스크
    - 255.255.240.0
  - 첫 번째 주소
    - 주소: 110.23.120.14
    - 네트워크 마스크: 255.255.240.0
    - 앞 장에서 설명했던 (A) 방식을 첫 번째, 두 번째, 네 번째 바이트에 적용하고 (B) 방식을 세 번째 바이트에 적용
  - 첫 번째 주소: 110.23.112.0
  - 마지막 주소: 110.23.127.255

IPv4주소

- 서브네팅(Subnetting)
  - 정의
    - 네트워크가 몇 개의 작은 서브 네트워크(서브넷)으로 나뉘는 개념
  - 구성
    - 네트워크 ID, 서브넷 ID, 호스트 ID
  - 특징(1/2)
    - 클래스 기반 주소 지정은 기존 호스트 ID가 서브넷 ID와 호스트 ID로 나뉘고, CIDR은 기존 프리픽스가 네트워크 ID와 서브넷 ID로 나뉨



IPv4주소

- 서브네팅(Subnetting)
  - 특징(2/2)
    - 슬래시 표기법(/n) 사용
    - 브로드캐스트 영역의 크기를 줄이고 IP 주소 공간을 효율적으로 사용할 수 있음
    - 첫 번째 주소(임의의 주소) ⊕ 서브넷 마스크 = 서브넷 주소

IPv4주소

- 서브네팅(Subnetting)
  - 서브넷 마스크
    - 정의
      - 서브넷 ID를 할당하기 위한 32bits 2진수 값

기호	설명
N	기존 네트워크에 할당된 주소의 개수
N_{sub}	서브 네트워크에게 할당된 주소의 개수
n	기존 프리픽스나 네트워크 ID 길이
n_{sub}	서브넷 주소의 subnetid 길이
s	네트워크가 분할된 갯수

    - 설계
      - 네트워크 ID와 서브넷 ID의 모든 비트를 1, 호스트 ID의 모든 비트를 0으로 설정
      - 같은 수의 호스트를 가지는 s개의 서브 네트워크로 나뉘면 subnetid 길이를 구할 수 있음($s = N/N_{sub}$)

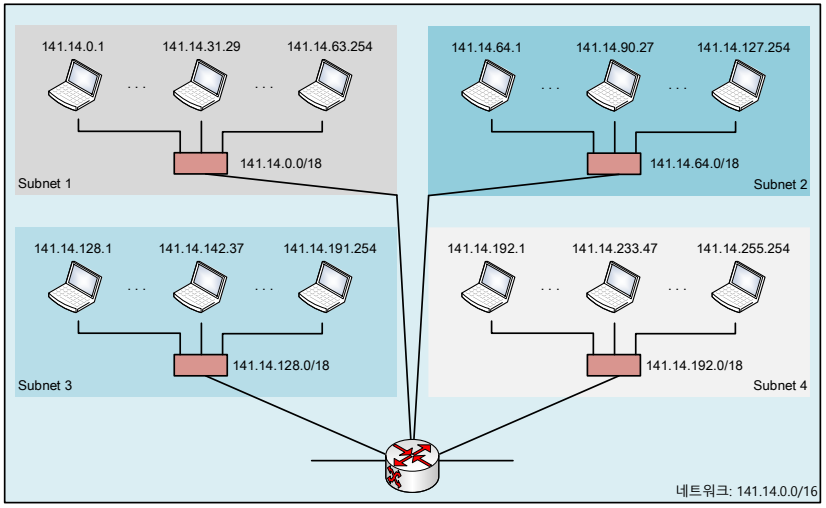
$$n_{sub} = n + \log_2 s$$

IPv4 주소

- 서브네팅(Subnetting)
  - 예제 5.15

그림의 subnetid 길이를 구하라.

- 풀이
  - 141.14.0.0 ~ 141.14.255.255 대역의 네트워크를 네 개의 서브넷으로 나뉘면
    $n = 16, n_1 = n_2 = n_3 = n_4 = 16 + \log_2 4 = 18$
  - 서브넷 마스크가 18개의 1을 가지고
    14개의 0을 가짐을 의미함
    - 255.255.192.0



IPv4 주소

- 서브네팅(Subnetting)
  - 예제 5.16

130.34.12.64/26 블록이 기관에게 할당되었다. 기관은 각각이 동일한 개수의 호스트를 갖는 4개의 서브 네트워크를 구성하고자 한다. 서브 네트워크를 설계하고 각각의 서브 네트워크에 대한 정보를 구하라.

- 풀이
  - 총 주소의 개수 $N = 2^6$
  - 네트워크의 첫 번째 주소는 130.34.12.64/26
  - 네트워크의 마지막 주소는 130.34.12.127/26
  - 서브 네트워크를 위한 서브 네트워크 프리픽스 길이는 $n_{sub} = n_1 = n_2 = n_3 = n_4 = n + \log_2(N/N_{sub}) = 26 + 2 = 28$
  - 총 4개의 서브네트워크로 나뉘므로 한 서브 네트워크당 $N_{\mathrm{sub}} = 2^4$개의 주소를 할당함(조건 2)
  - 각 서브 네트워크의 시작 주소는
    - 130.34.12.64/28 130.34.12.80/28
    - 130.34.12.96/28 130.34.12.112/28

IPv4 주소

- 서브네팅(Subnetting)
  - 예제 5.17 (1/2)

어떤 회사가 중앙, 동쪽, 서쪽의 3개의 사무실을 가지고 있다. 중앙 사무실 사설 WAN 선로를 이용하여 동쪽과 서쪽 사무실과 연결되어 있다. 이 회사가 할당받은 블록은 70.12.100.128/26의 시작주소를 갖는 64개의 주소로 이루어진다. 중앙 사무실에 32개의 주소를 할당하고 나머지 주소를 다른 두 개의 사무실에 할당하라

- 풀이
  - 중앙 사무실 $N_c = 32$, 동쪽 사무실 $N_e = 16$, 서쪽 사무실 $N_w = 16$
  - 각각의 서브 네트워크를 위한 프리픽스 길이는

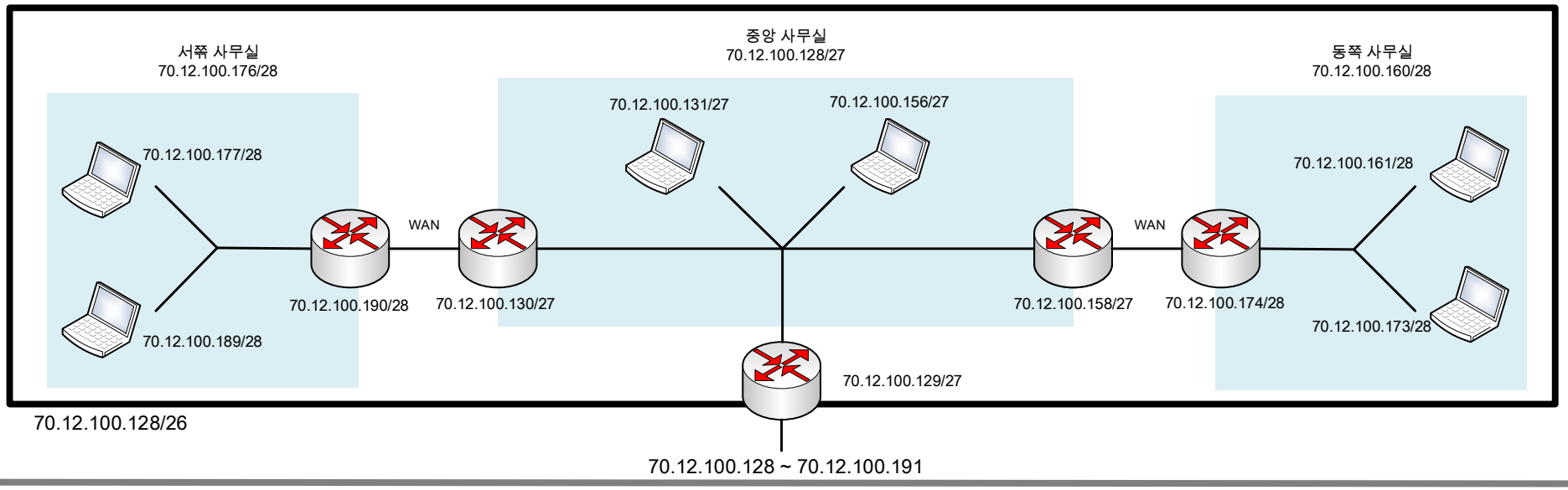
$$n_c = n + \log_2\left(\frac{64}{32}\right) = 27,\ n_e, n_w = n + \log_2\left(\frac{64}{16}\right) = 28$$

기호	설명
N_c	중앙 사무실에 할당된 주소의 개수
N_e	동쪽 사무실에 할당된 주소의 개수
N_w	서쪽 사무실에 할당된 주소

기호	설명
n_c	중앙 사무실의 subnetid 길이
n_e	동쪽 사무실의 subnetid 길이
n_w	중앙 사무실의 subnetid 길이
n	기존 프리픽스의 길이

IPv4 주소

- 서브네팅(Subnetting)
  - 예제 5.17 (2/2)
- 풀이
  - 중앙 사무실이 사용한 주소는 70.12.100.128/27부터 70.12.100.159/27까지 (32개 사용)
    - 이 주소 중 3개는 라우터에 하나는 마지막 주소는 다른 목적을 위해 사용
  - 동쪽 사무실은 70.12.100.160/28부터 70.12.100.175/28까지 사용(16개 사용)
  - 서쪽 사무실은 70.12.100.176/28부터 70.12.100 191/28까지 사용(16개 사용)
    - 이중 마지막 주소는 다른 목적을 위해 사용

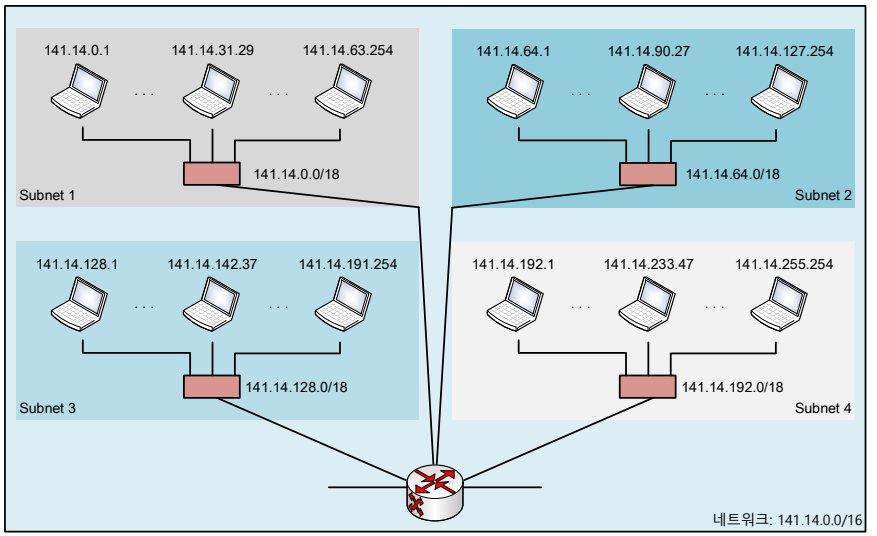


IPv4 주소

- 서브네팅(Subnetting)
  - 예제 5.18

그림의 서브넷 2의 주소 중의 하나가 141.14.120.77일 때 서브넷 주소를 찾아라.

- 풀이
  - Subnet2 주소: 141.14.120.77
  - 마스크: 255.255.192.0
  - 첫 번째, 두 번째, 네 번째는 (A) 방법 사용
    - 141, 14, 0 선택
  - 세 번째는 (B) 방법 사용
    - 바이트(120): $2^6 + 2^5 + 2^4 + 2^3$
    - 바이트(192): $2^7 + 2^6$
  - 서브넷 주소: 141.14.64.0



IPv4주소

- 슈퍼네팅(Supernetting)
  - 정의
    - 여러 개의 작은 네트워크를 하나의 큰 네트워크로 통합하는 기술
  - 특징
    - 여러 개의 작은 네트워크를 통합하여 IP 주소를 효율적으로 관리
    - 라우팅 테이블이 다루어야 할 정보량을 줄여 컴퓨팅 자원 낭비를 막음
    - 주소 고갈 문제를 간접적으로 해결함
      - 대부분의 조직은 할당 받은 블록을 다른 조직과 공유되는 것을 원치 않기 때문에 서브네팅은 주소 고갈 문제를 완전히 해결할 수 없음

IPv4주소

- 슈퍼네팅(Supernetting)
  - 슈퍼넷 마스크
    - 정의
      - 서브네팅을 수행하기 위한 32bits 2진수 값
    - 계산
      - 결합된 클래스 C의 블록 개수를 c라고 할 때, supernetid 길이를 구할 수 있음

$$n_{super} = n - \log_2 c$$

$n_{super} = 24 - \log_2(2^3)$	11111111	11111111	11111000	00000000	슈퍼넷 마스크
$n = 24$	11111111	11111111	11111111	00000000	네트워크 마스크
$n_{sub} = 24 + \log_2(2^3)$	11111111	11111111	11111111	11100000	서브넷 마스크

IPv4주소

- 특수 주소(1/3)
  - 모두 0인 주소(0.0.0.0/32)
    - IPv4 패킷을 전송하고자 하는 호스트가 자신의 IPv4 주소를 모르는 경우 사용
  - 호스트 ID가 0인 주소
    - 해당 네트워크를 가르킴
    - e.g., 192.168.1.0
  - 네트워크 ID가 0인 주소
    - 현재 네트워크에 지정된 호스트를 가르킴
    - e.g., 0.0.72.8

IPv4주소

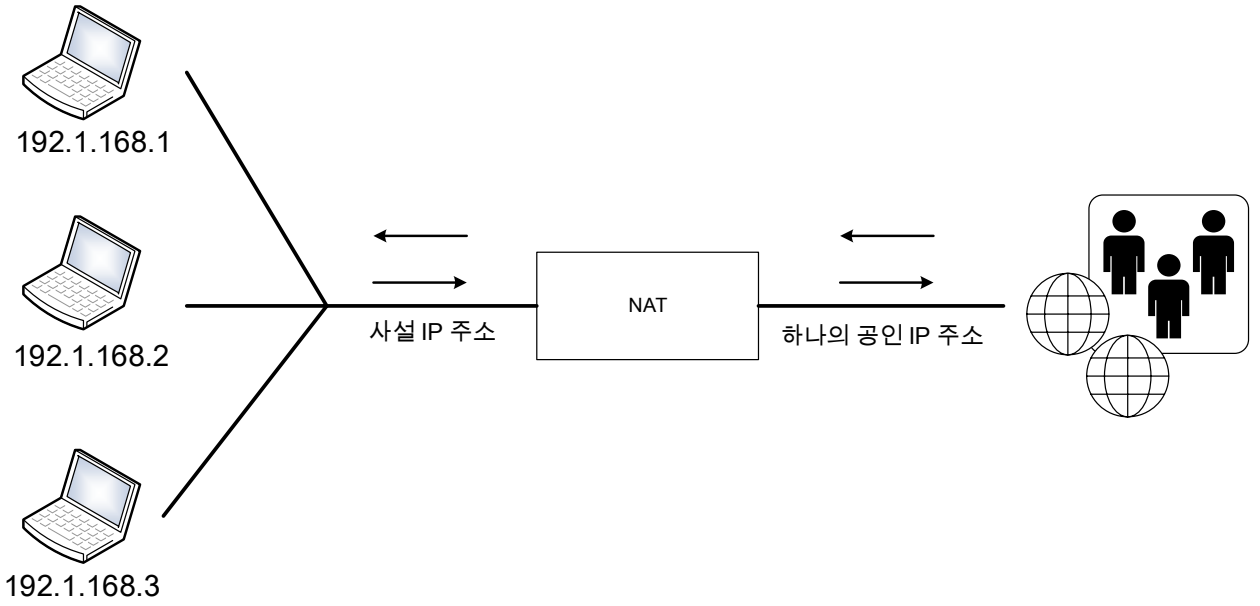
- 특수 주소(2/3)
  - 모두 1인 주소(255.255.255.255/32)
    - 네트워크 내의 모든 다른 호스트들에게 메시지를 전송하고자 할 때, 해당 주소를 목적지 주소로 사용
  - 호스트 ID가 1인 주소
    - 지정된 네트워크의 모든 호스트를 가르킴
    - e.g., 192.168.255.255

IPv4주소

- 특수 주소(3/3)
  - 루프백 주소(127.0.0.0~127.255.255.255)
    - 컴퓨터에 설치된 소프트웨어를 시험하기 위해 사용
    - e.g., 웹서버와 웹페이지들이 제대로 설정되어 있는지 확인 (http://127.0.0.1/index.html)
  - 사설 주소
    - 인터넷에 인식되지 않는 IP 주소 공간으로, 내부 네트워크에서 사용되며 NAT(Network Address Translation)를 통해 인터넷 연결을 가능하게 함
    - e.g., 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
  - 멀티캐스트 주소(225.0.0.0)
    - 하나의 출발지 장비에서 여러 장비로 구성된 그룹으로 데이터를 전송하기 위한 주소

IPv4주소

- NAT (Network Address Translation)
  - 정의
    - 하나의 공인 IP 주소를 사용하여 여러 개의 사설 IP 주소를 인터넷에 연결할 수 있도록 해주는 기술



IPv4주소

- NAT (Network Address Translation)
  - 변환 테이블
    - 정의
      - 인터넷으로부터 들어오는 패킷의 목적지 주소를 식별하기 위해 NAT 라우터가 가지고 있는 테이블
  - 방식
    - Static NAT
      - 정의
        - 내부 네트워크의 모든 사설 IP 주소를 하나의 공인 IP 주소로 변환
      - 특징
        - 사설 IP와 공인 IP가 고정되어 있어 항상 동일한 공인 IP로 변환됨
        - 외부에서 내부 네트워크의 특정 호스트에 접근하기 어려움
        - 주로 가정용 공유기에 사용되는 방식

사설 IP	공인 IP
192.168.1.10	203.0.113.10
192.168.1.11	
192.169.2.34	203.0.113.11

IPv4주소

- NAT (Network Address Translation)
  - 방식
    - Dynamic NAT
      - 정의
        - 사설 IP 주소를 공인 IP 주소 풀에서 선택하여 변환
      - 특징
        - 사설 IP가 인터넷에 접근할 때마다 사용가능한 공인 IP 주소 중 하나로 변환됨
          - 다른 사설 IP에 재사용이 가능함
        - 공인 IP 풀의 크기에 따라 변환할 수 있는 내부 호스트의 수가 제한됨
        - 주로 중소 기업이나 ISP에서 사용하는 방식

사설 IP	공인 IP	상태
192.169.1.10	203.0.113.1	할당됨
192.168.1.11	203.0.113.2	할당됨
192.168.1.12	203.0.113.3	재사용 중
192.168.1.13	-	대기 중

IPv4주소

- NAT (Network Address Translation)
  - 방식
    - PAT(Port Address Translation)
      - 정의
        - IP와 포트번호를 함께 변환
          - e.g., 내부 IP(192.168.1.10:5000) ↔ 공인IP(203.0.113.5:10000)
      - 특징
        - 하나의 공인 IP 주소로 여러 내부 호스트를 지원할 수 있음
        - 대규모 기업이나 ISP에서 사용하는 방식

사설 IP	포트	공인 IP	포트
192.168.1.10	1000	203.0.113.1	60001
192.168.1.11	2000		60002
192.168.1.12	3000	203.0.113.2	60003

목 차

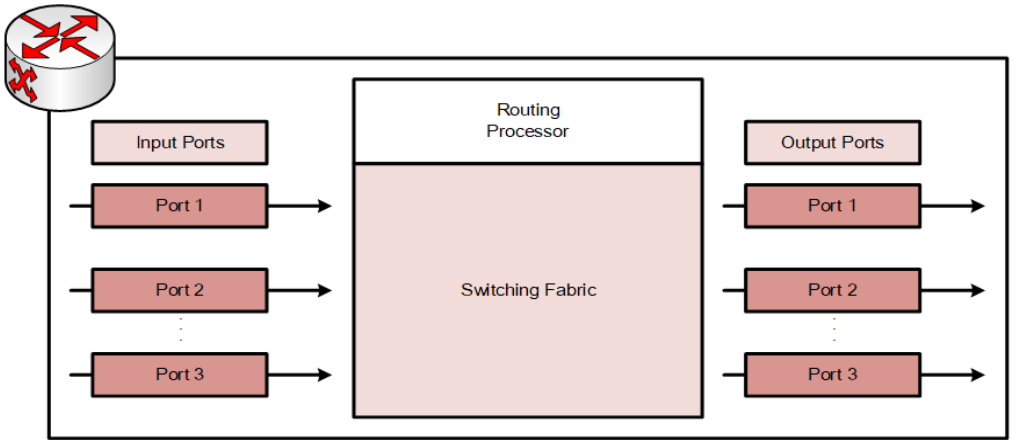
- 보충
- 네트워크 계층 소개
- IPv4 주소
- IP 패킷의 전달과 포워딩

IP 패킷의 전달과 포워딩

- IP 데이터그램 전달 방식
  - 직접 전달
    - 송수신자의 목적지 주소에서 네트워크 ID를 추출하여 비교한 결과가 같은 경우, 데이터그램을 전달되는 방식
      - 라우터를 거치지 않음
  - 간접 전달
    - 송수신자의 최종 목적지가 같은 네트워크에 있지 않을 경우 데이터그램을 전달하는 방식
      - 라우터를 포함한 여러 장치 및 네트워크를 통해 이루어짐

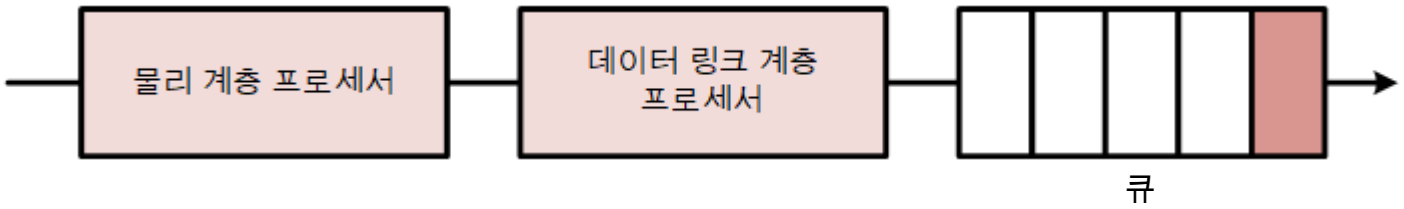
IP 패킷의 전달과 포워딩

- 라우터의 구조
  - 라우터
    - 정의
      - 패킷을 목적지까지 전달하기 위해 다음 홉을 결정하는 장치나 컴퓨터 내 소프트웨어
    - 기능
      - 라우팅 알고리즘이나 프로토콜을 작동 시킴
      - 입력포트로부터 들어오는 데이터그램을 출력포트를 사용해 나가는 링크로 포워딩 시킴
    - 구조
      - 입력포트
      - 출력포트
      - 라우팅 프로세서
      - 스위칭 구조



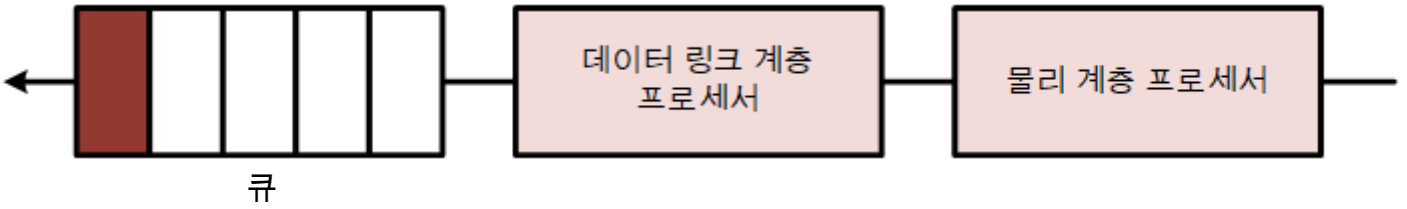
IP 패킷의 전달과 포워딩

- 라우터의 구조
  - 입력 포트(Input Port)
    - 역할
      - 수신된 신호로부터 비트를 생성하는 기능을 담당
    - 기능
      - 생성된 비트를 프레임을 역캡슐화(Decapsulation)해서 패킷을 추출함
        - 역캡슐화: TCP/IP 모델, OSI 모델에서 상위 계층으로 메세지를 변환하는 것
    - 구성
      - 물리 계층 프로세서, 데이터 링크 계층 프로세서, 큐로 구성됨



IP 패킷의 전달과 포워딩

- 라우터의 구조
  - 출력 포트(Output Port)
    - 역할
      - 입력 포트와 같은 기능을 역순으로 수행하는 담당
    - 기능
      - 출력될 패킷이 큐에 들어오면 각 패킷을 프레임으로 캡슐화(Encapstulation)하여 전기 신호로 송신함
    - 구성
      - 큐, 데이터 링크 계층 프로세서, 물리 계층 프로세서로 구성



IP 패킷의 전달과 포워딩

- 라우터의 구조
  - 라우팅 프로세서(Routing Processor)
    - 역할
      - 패킷의 최적의 경로를 결정하고, 이를 기반으로 패킷을 적절한 인터페이스로 포워딩하는 담당
    - 기능
      - 패킷을 전송할 출력 포트번호와 다음 홉 주소를 찾기 위해 패킷의 주소를 참조함

IP 패킷의 전달과 포워딩

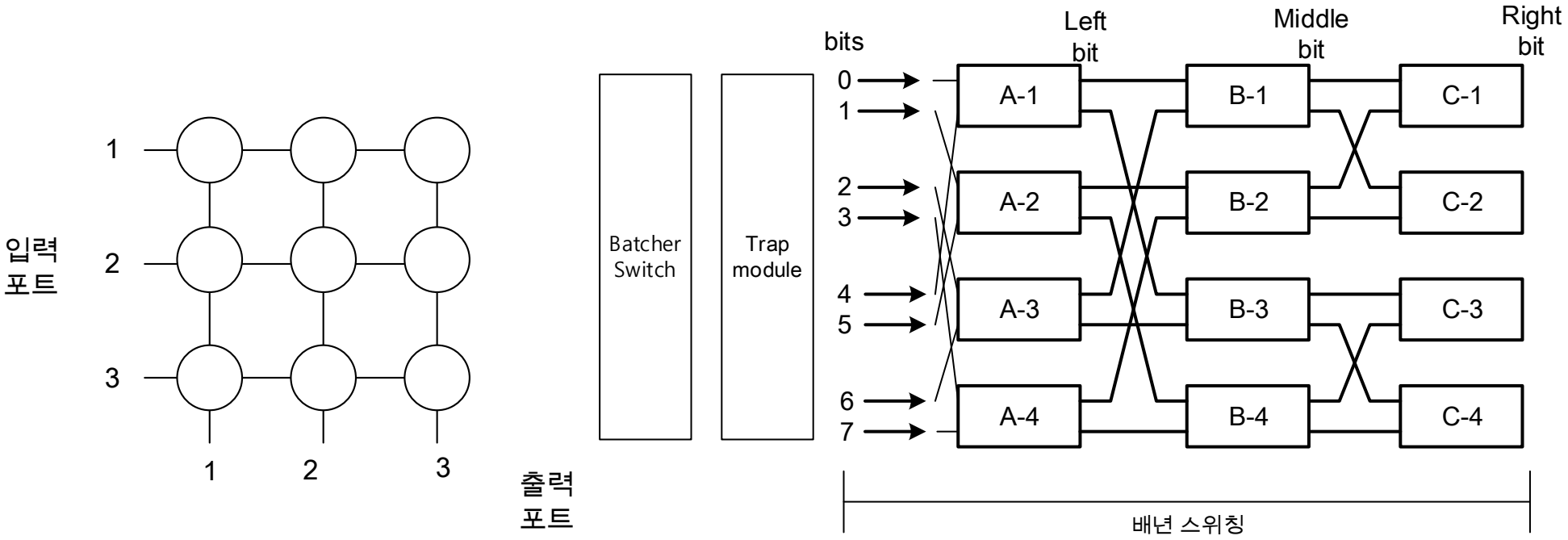
- 라우터의 구조
  - 스위칭 구조(Switching Fabrics)(1/3)
    - 역할
      - 패킷을 입력 포트에서 출력 포트로 전달하는 담당
    - 특징
      - 작업 수행 속도가 패킷 전달 전체 지연에 영향을 줌
      - 여러 유형에 따라 다른 방식으로 패킷을 전송함

IP 패킷의 전달과 포워딩

- 라우터의 구조
  - 스위칭 구조(Switching Fabrics)(2/3)
    - 유형
      - 교차점 스위칭(Crossbar Switching)
        - 입력 포트와 출력 포트가 교차점으로 구성된 스위치 매트릭스를 사용하여 입력 포트에서 출력 포트로 전송
      - 배년 스위칭(Banyan Switching)
        - 다단계 스위치를 통해 데이터 패킷을 입력 포트에서 출력 포트로 전송
      - 배년-배치 스위칭(Banyan-Batching Switching)
        - 다단계 구조와 패킷을 일정 크기로 묶어서 동시에 처리하여 입력 포트에서 출력 포트로 전송

IP 패킷의 전달과 포워딩

- 라우터의 구조
  - 스위칭 구조(Switching Fabrics)(3/3)

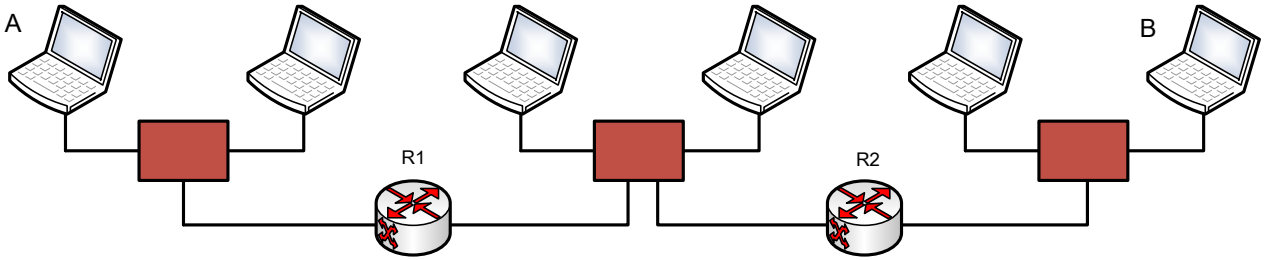


IP 패킷의 전달과 포워딩

- 포워딩
  - 정의
    - 패킷이 라우터의 입력 링크에 도달했을 때, 라우터는 그 패킷을 적절한 출력 링크로 이동시켜는 것
  - 목적지 주소 기반 포워딩(1/6)
    - 정의
      - 네트워크에서 패킷을 전송할 때, 패킷의 목적지 주소를 기준으로 경로를 설정하는 포워딩
    - 기능
      - 비연결형 프로토콜로 사용되었을 때 데이터그램의 목적지 주소를 기반으로 수행됨
      - 송수신자는 라우팅 테이블을 참조하여 최종 목적지로 가는 경로를 찾음

IP 패킷의 전달과 포워딩

- 포워딩
  - 목적지 주소 기반 포워딩(2/6)
    - 다음 홉 방법
      - 정의
        - 라우팅 테이블 경로에 전체 경로가 아닌 다음 홉의 주소만 저장하는 방법
      - 특징
        - 전체 경로에 대한 주소를 알아야 할 필요가 없으므로, 주소 길이가 단축
        - 네트워크 변화가 발생하더라도 다음 홉의 고정된 주소만 다루어 라우팅 테이블을 수정할 필요 적음



경로기반 라우팅 테이블

A

목적지 주소	경로
호스트 B	R1, R2, 호스트 B

R1

목적지 주소	경로
호스트 B	R2, 호스트 B

R2

목적지 주소	경로
호스트 B	호스트 B

A

목적지 주소	다음 홉
호스트 B	R1

R1

목적지 주소	다음 홉
호스트 B	R2

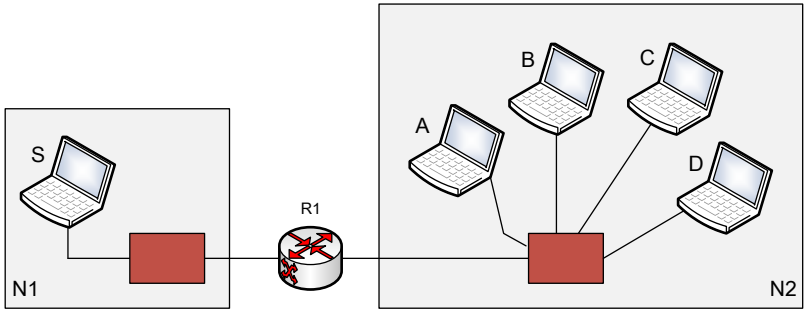
R2

목적지 주소	다음 홉
호스트 B	-

다음 홉 기반 라우팅 테이블

IP 패킷의 전달과 포워딩

- 포워딩
  - 목적지 주소 기반 포워딩(3/6)
    - 네트워크 지정 방법
      - 정의
        - 같은 네트워크에 연결된 모든 호스트를 하나의 엔트리로 간주하는 방법
          - 즉, 네트워크 하나를 정의하는 엔트리만 가지고 있음
      - 특징
        - 모든 호스트에 대한 개별적인 인터페이스와 고정 길이 주소를 저장
        - 라우팅 테이블의 크기를 줄이고 관리의 복잡성을 낮춤



호스트 S의 호스트에 대한 라우팅 테이블

목적지 주소	다음 홉
A	R1
B	R1
C	R1
D	R1

호스트 S의 네트워크에 대한 라우팅 테이블

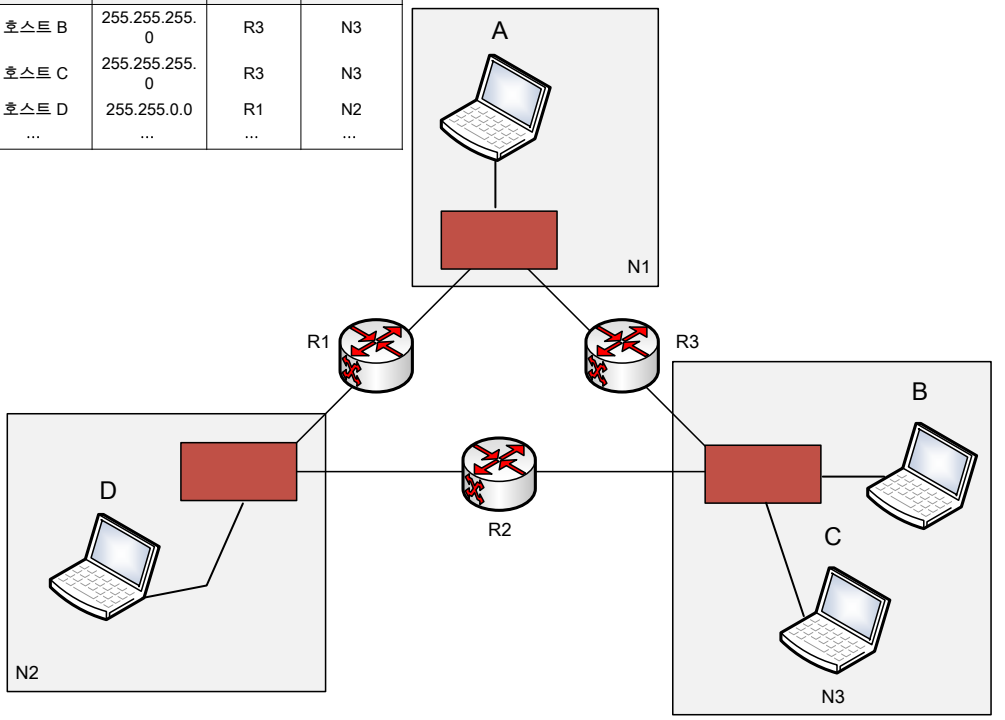
목적지 주소	다음 홉
N2	R1

IP 패킷의 전달과 포워딩

- 포워딩
  - 목적지 주소 기반 포워딩(4/6)
    - 호스트 지정 방법
      - 정의
        - 네트워크에 연결된 모든 호스트에 대한 인터페이스와 IP 주소 등을 저장하는 방식
      - 특징
        - 특정 호스트에 대한 세부 정보를 제공하므로 라우팅 테이블이 복잡해지고 처리 속도가 느려짐

호스트 A의 라우팅 테이블

목적지 주소	넷 마스크	다음 홉	인터페이스
호스트 B	255.255.255.0	R3	N3
호스트 C	255.255.255.0	R3	N3
호스트 D	255.255.0.0	R1	N2
...	...	...	...



IP 패킷의 전달과 포워딩

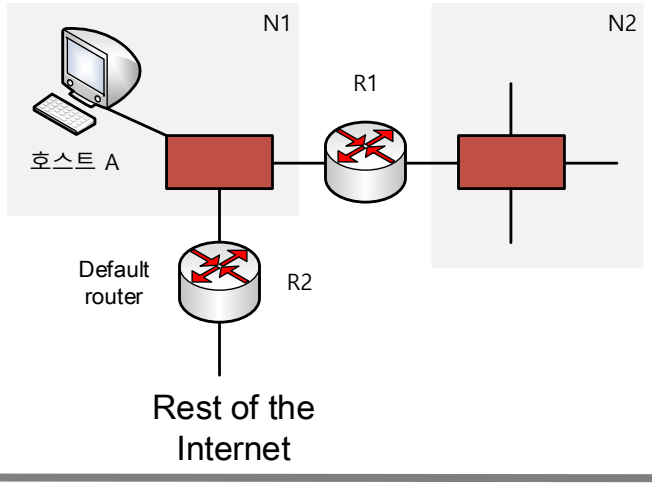
- 포워딩
  - 목적지 주소 기반 포워딩(5/6)
    - 계층적 라우팅
      - 정의
        - 네트워크를 여러 계층으로 나누어 각 계층에서 라우팅 정보를 관리하는 방식
      - 특징
        - 상위 계층은 더 큰 네트워크를 관리하며 라우터 간의 연결을 담당하고, 하위 계층은 다소 작은 네트워크를 관리함
        - 라우팅 테이블의 크기를 줄여 메모리 사용을 최적화함
    - 지리적 라우팅
      - 정의
        - 패킷의 목적지 주소를 지리적 위치에 기반하여 결정하는 방식
      - 특징
        - 각 라우터는 자신의 위치 정보를 알고 있으므로 이동성이 높은 환경에 효과적
        - 국가간 라우팅을 위한 라우팅 테이블에 한 개의 엔트리 사용

IP 패킷의 전달과 포워딩

- 포워딩
  - 목적지 주소 기반 포워딩(6/6)
    - 디폴트 방법
      - 정의
        - 일부 목적지에 대한 엔트리를 따로 정의하고, 자주 보내지는 경로를 디폴트로 지정하는 방식
      - 특징
        - 디폴트 엔트리의 네트워크 주소와 마스크는 0.0.0.0으로 설정되어 있음
          - e.g.,호스트 A는 네트워크 주소가 0.0.0.0으로 되어 있음

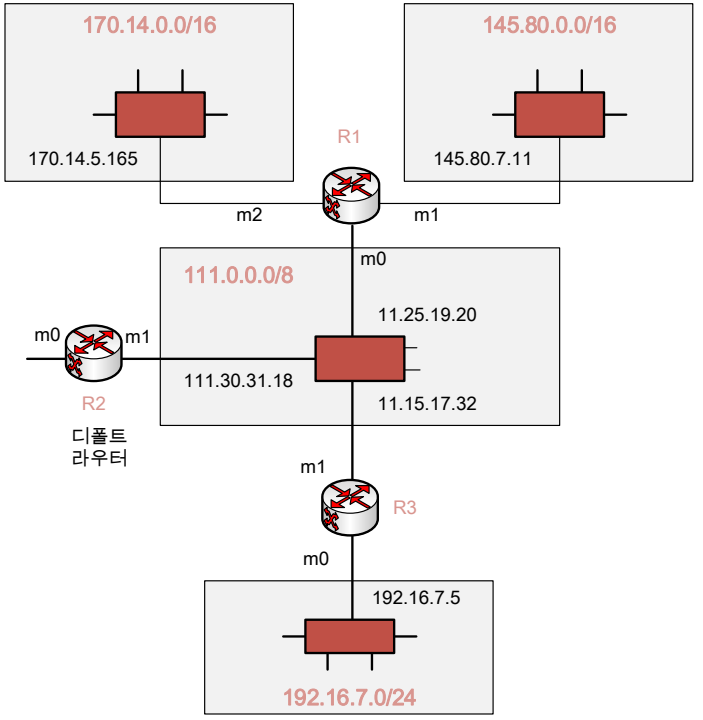
목적지 주소	다음 홉
N2 …. Default	R1 …. R2

호스트 A의 라우터



IP 패킷의 전달과 포워딩

- 포워딩 – 목적지 주소 기반
  - 클래스 기반 주소 체계 기반 포워딩
    - 서브네팅이 이루어지지 않은 경우
      1. 패킷의 목적지 주소를 추출
      2. 목적지 주소의 오른쪽으로 28비트 이동한 결과로 클래스 정보를 얻음
      3. 네트워크 주소를 찾음
      4. 클래스와 네트워크 주소를 사용하여 다음 홉 주소를 찾음
      5. 다음 홉 주소와 인터페이스번호를 ARP에게 전달



IP 패킷의 전달과 포워딩

- 포워딩 – 목적지 주소 기반
  - 예제 6.1

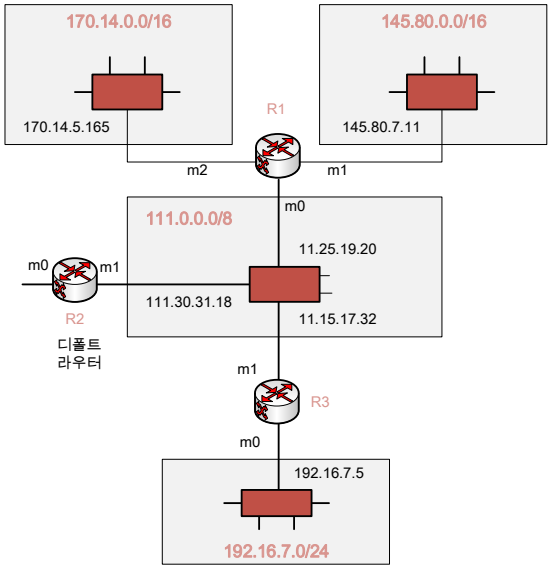
그림은 가상 네트워크를 보여준다. 라우터 R1의 라우팅 테이블을 보여라

- 풀이
  - 라우터에 연결된 네트워크와 같은 네트워크일 경우, 다음 홉이 비어있음

네트워크 주소	다음 홉 주소	인터페이스
111.0.0.0	-	M0

네트워크 주소	다음 홉 주소	인터페이스
192.16.7.0	111.15.17.32	M0

네트워크 주소	다음 홉 주소	인터페이스
145.80.0.0	-	M1
170.14.0.0	-	M2

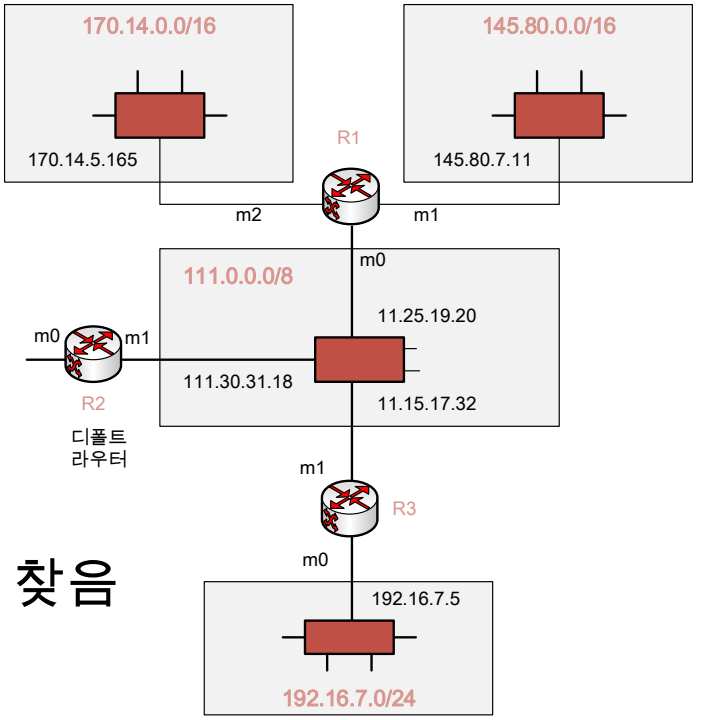


IP 패킷의 전달과 포워딩

- 포워딩 – 목적지 주소 기반
  - 예제 6.2

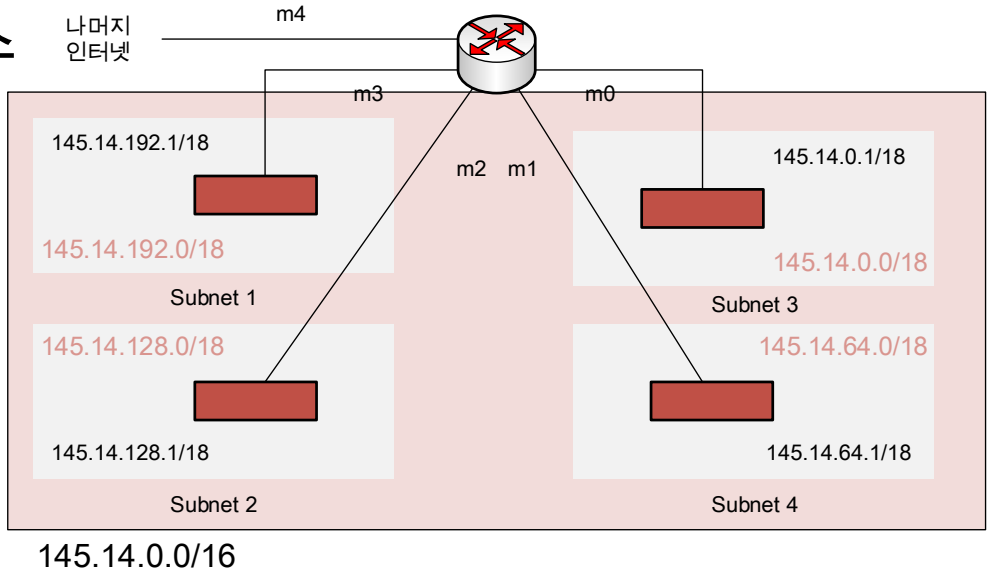
그림은 가상 네트워크를 보여준다. 라우터 R1은 목적지 주소가 192.16.7.14인 패킷을 받았다. 패킷이 어떻게 포워딩 되는지 보여라.

- 풀이
  - 목적지 주소
    - 11000000 00010000 00000111 00001110
  - 오른쪽으로 28비트 쉬프트 하면 결과는
    000 … 00001100 또는 12이므로
    목적지 네트워크는 클래스 C가 됨
  - 목적지 주소에서 왼쪽 24비트만 뽑아내면
    네트워크 주소는 192.16.7.0
  - 클래스 C의 테이블에서 192.16.7.0의 엔트리를 찾음
  - 다음 홉 주소 111.15.17.32와
    인터페이스 번호 m0를 ARP에게 전달



IP 패킷의 전달과 포워딩

- 포워딩 – 목적지 주소 기반
  - 클래스 주소 기반 포워딩
    - 서브넷팅이 이루어진 경우
      1. 목적지 주소를 추출
      2. 목적지 주소와 서브넷 마스크를 사용하여 서브넷 주소를 추출
      3. 테이블에서 서브넷 주소를 탐색하여 다음 홉 주소와 인터페이스 번호를 찾음
      4. 다음 홉 주소와 인터페이스 번호를 ARP에게 전달

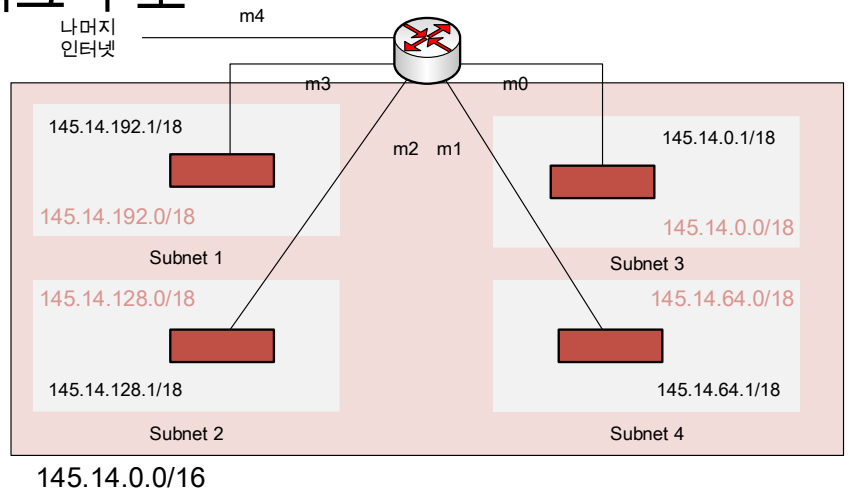


IP 패킷의 전달과 포워딩

- 포워딩 – 목적지 주소 기반
  - 예제 6.3

그림의 라우터가 목적지 주소 145.14.32.78인 패킷을 수신했다. 패킷이 어떻게 포워딩 되는가?

- 풀이
  - 프리픽스 길이가 18이므로, 서브넷 마스크는 255.255.192.0
  - 145.14.32.78 ⊕ 255.255.192.0 = 145.14.0.0
    - 임의의 주소 ⊕ 서브넷 마스크 = 네트워크 주소
    - 서브넷 1에 해당함
  - 다음 홉 주소 145.14.32.28과 출력 인터페이스 m0와 함께 ARP로 전달



IP 패킷의 전달과 포워딩

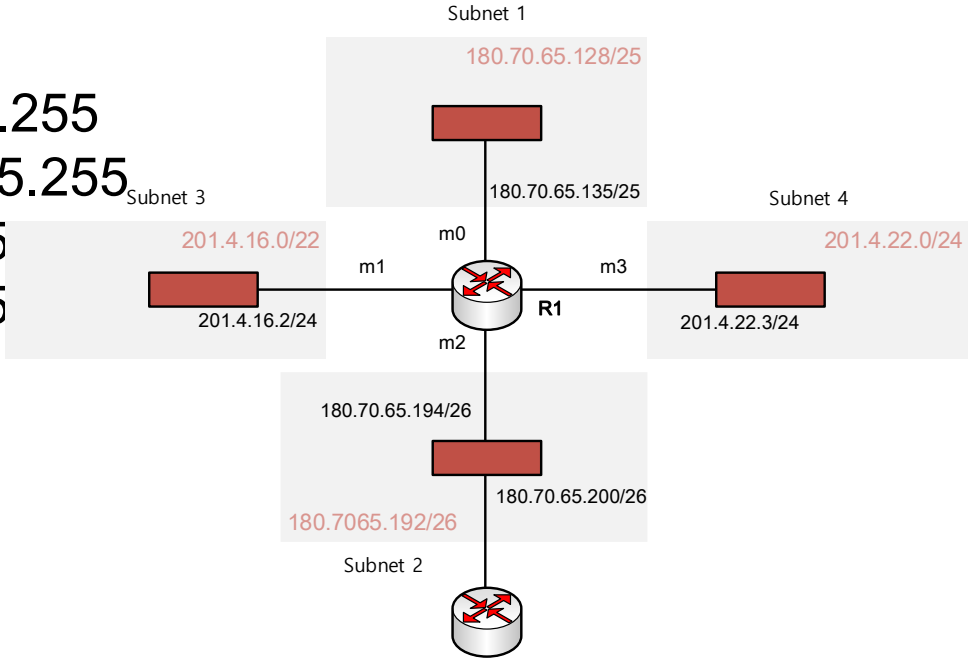
- 포워딩 – 목적지 주소 기반
  - 클래스 없는 주소 체계에서 포워딩
    - 정의
      - 클래스에 얽매이지 않고, 가변 길이의 서브넷 마스크를 사용하여 네트워크를 구분함
    - 특징
      - 클래스 기반 주소 체계와 달리 목적지 주소가 네트워크 주소에 대한 힌트를 주지 못함
      - 주소 표현 시, 슬래시(/n) 포함해주어야 함

IP 패킷의 전달과 포워딩

- 포워딩 – 목적지 주소 기반
  - 예제 6.4

그림에서 R1에서 목적지 주소가 180.70.65.140인 패킷을 보내라

- 풀이(1/2)
  - 각 서브넷의 범위를 구하면
  - 서브넷 1: 180.70.65.128 ~ 80.70.65.255
  - 서브넷 2: 180.70.65.192 ~ 180.70.65.255
  - 서브넷 3: 201.4.16.0 ~ 201.4.19.255
  - 서브넷 4: 201.4.22.0 ~ 201.4.22.255
  - 서브넷 1의 범위에 들어감

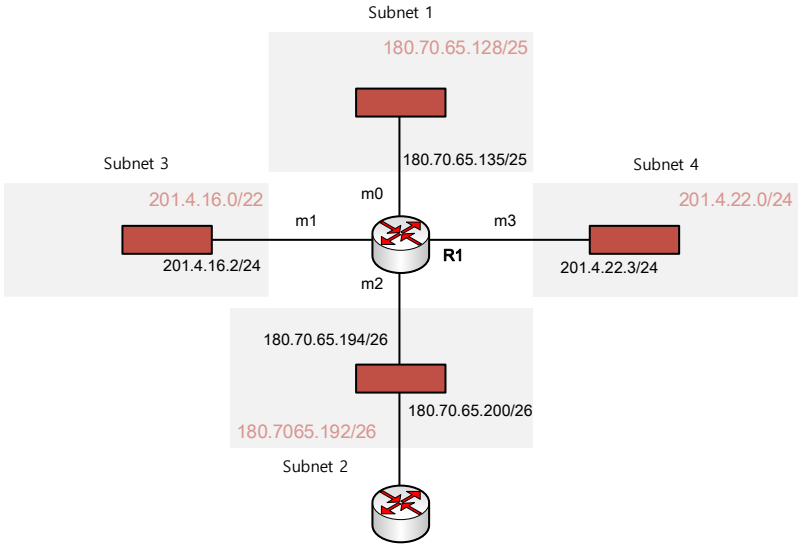


IP 패킷의 전달과 포워딩

- 포워딩 – 목적지 주소 기반
  - 예제 6.5

그림에서 R1에서 목적지 주소가 180.70.65.140인 패킷을 보내라

- 풀이(2/2)
  - R1은 패킷을 m0 인터페이스를 통해 180.70.65.140로 포워딩 함



마스크	네트워크 주소	인터페이스
/26	180.70.65.192	M2
/25	180.70.65.128	M0
/24	201.4.22.0	M3
/23	201.4.16.0	M1

IP 패킷의 전달과 포워딩

- 포워딩 테이블 검색 - 목적지 주소 기반
  - 클래스 기반 주소 체계에서의 라우팅 테이블 탐색
    - 클래스에 따라 세 개의 버킷으로 나뉨

	클래스 A	클래스 B	클래스 C
주소 범위	0.0.0.0~ 127.255.255.255	128.0.0.0 ~ 191.255.255.255	192.0.0.0 ~ 223.255.255.255
디폴트 마스크	255.0.0.0(또는 /8)	255.255.0.0 (또는 /16)	255.255.255.0 (또는 /24)
비트 패턴	첫 번째 비트가 0인 주소	첫 번째 비트가 1이고 두 번째 비트가 0인 주소	첫 번째 비트가 1이고 두 번째 비트가 1이며 세 번째 비트가 0인 주소
버킷	대규모 네트워크에 적합하며 첫 번째 버킷에 저장됨	중간 규모의 네트워크에 적합하며 두 번째 버킷에 저장	소규모 네트워크에 적합하며 세 번째 버킷에 저장

IP 패킷의 전달과 포워딩

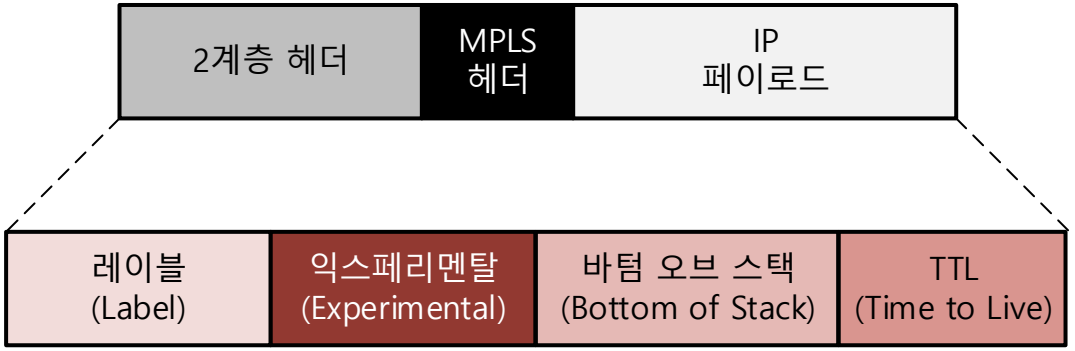
- 포워딩 테이블 검색 – 목적지 주소 기반
  - 클래스 없는 주소 체계에서의 라우팅 테이블 탐색
    - 특징
      - 목적지 주소 내에 네트워크 정보가 없음
      - 프리픽스 길이에 따라 여러 개의 버킷으로 나눌 수 있음
        - 각 버킷은 특정 프리픽스에 대한 경로 정보를 포함
        - e.g., 192.168.1.0/24, 192.168.2.0/24, 10.0.0.0/8
      - 긴 프리픽스 부합 방법 사용
        - 라우터가 패킷을 수신하면, 패킷의 목적지 IP 주소를 확인하고 해당 주소와 일치하는 프리픽스를 찾아 가장 많은 비트를 공유하는 프리픽스를 우선적으로 선택함
        - e.g., IP 목적지 주소가 192.168.1.130일 경우
          - 192.168.1.0/24
          - 192.168.1.128/25
          - 192.168.0.0/16
          - 192.168.1.128/25가 가장 긴 프리픽스로 이 경로로 패킷이 포워딩 됨

IP 패킷의 전달과 포워딩

- 레이블 기반의 포워딩
  - MPLS(Multi Protocol Label Switching)
    - 정의
      - IP 헤더 와 IP 페이로드로 구성된 기존 패킷에 MPLS 헤더를 추가로 붙이는 포워딩 방식
    - 특징
      - 패킷이 출력될 때 MPLS 헤더가 있는 레이블 기반으로 포워딩을 실행
        - 라우팅 테이블 크기가 줄어듦
      - 고정 경로가 아닌 동적 경로로 설정할 수 있어 효율적임
        - 네트워크 트래픽을 관리하고 고속 데이터 전송을 위함
        - IP헤더를 분석하고 최적의 경로를 찾는 과정에서 발생하는 성능 저하를 위함
      - 다양한 네트워크 환경에 사용할 수 있음

IP 패킷의 전달과 포워딩

- 레이블 기반의 포워딩
  - MPLS(Multi Protocol Label Switching)
    - 헤더 구조



헤더 항목	크기(bits)	설명
레이블	20	경로를 식별하기 위한 값(16~1,048,561)
익스페리멘탈	3	서비스 품질 처리를 위한 우선순위 지정
바텀오브 스택	1	0과 1을 사용하여 1일 경우, 마지막 레이블임을 알림
TTL	8	최대 홉 수를 지정

Thanks!

김 혜 정(hyejeong@pel.sejong.ac.kr)

부 록

- 루프백 주소(127.0.0.0~127.255.255.255)
  - 127.0.0.0
    - 네트워크 ID를 나타내는 주소로 직접 사용할 수 없음
  - 127.0.0.1
    - 일반적 루프백 주소로 지정될 때 가장 많이 사용되는 주소로, 대부분이 시스템에서 로컬 호스트로 지정되어있음
  - 127.0.0.2~255.255.255.254
    - 컴퓨터가 자신과 통신할 때 사용
    - 외부 네트워크나 인터넷을 거치지 않고 내부적으로 네트워크 통신을 테스트할 수 있음
    - 운영체제에 따라 구현이 다름
  - 127.255.255.255
    - 루프백 네트워크 내의 모든 가능한 호스트를 대상으로 함