

OSI 7계층과 IP 주소

정승호

Contents

01 OSI 7계층과 캡슐화

05 ARP

02 전송 계층 (Transport Layer)

06 IPv4와 NAT

03 네트워크 계층 (Network Layer)

07 IPv6

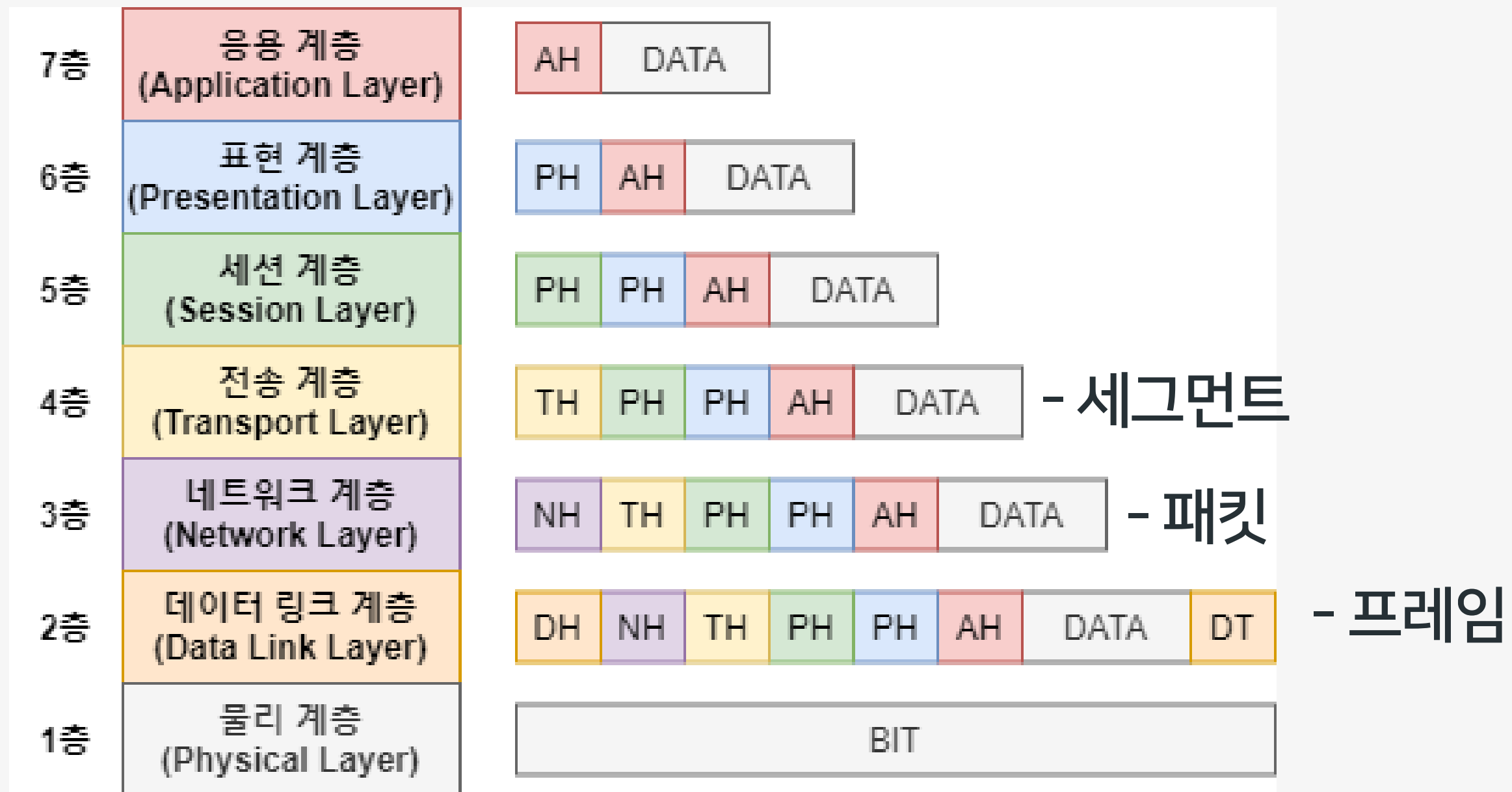
04 데이터 계층 (Data Link Layer)

08 요약

01 OSI 7계층과 캡슐화

- OSI 7 계층이란?
 - 복잡한 네트워크 통신을 7개의 역할로 분리한 설계도
- 캡슐화 (Encapsulation)
 - 상위 계층 데이터에 하위 계층의 정보(헤더)를 덧붙이는 과정
- 데이터 단위(PDU)의 변화
 - 데이터(Application) → 세그먼트(Transport) → 패킷(Network) → 프레임(Data Link) → 비트(Physical)

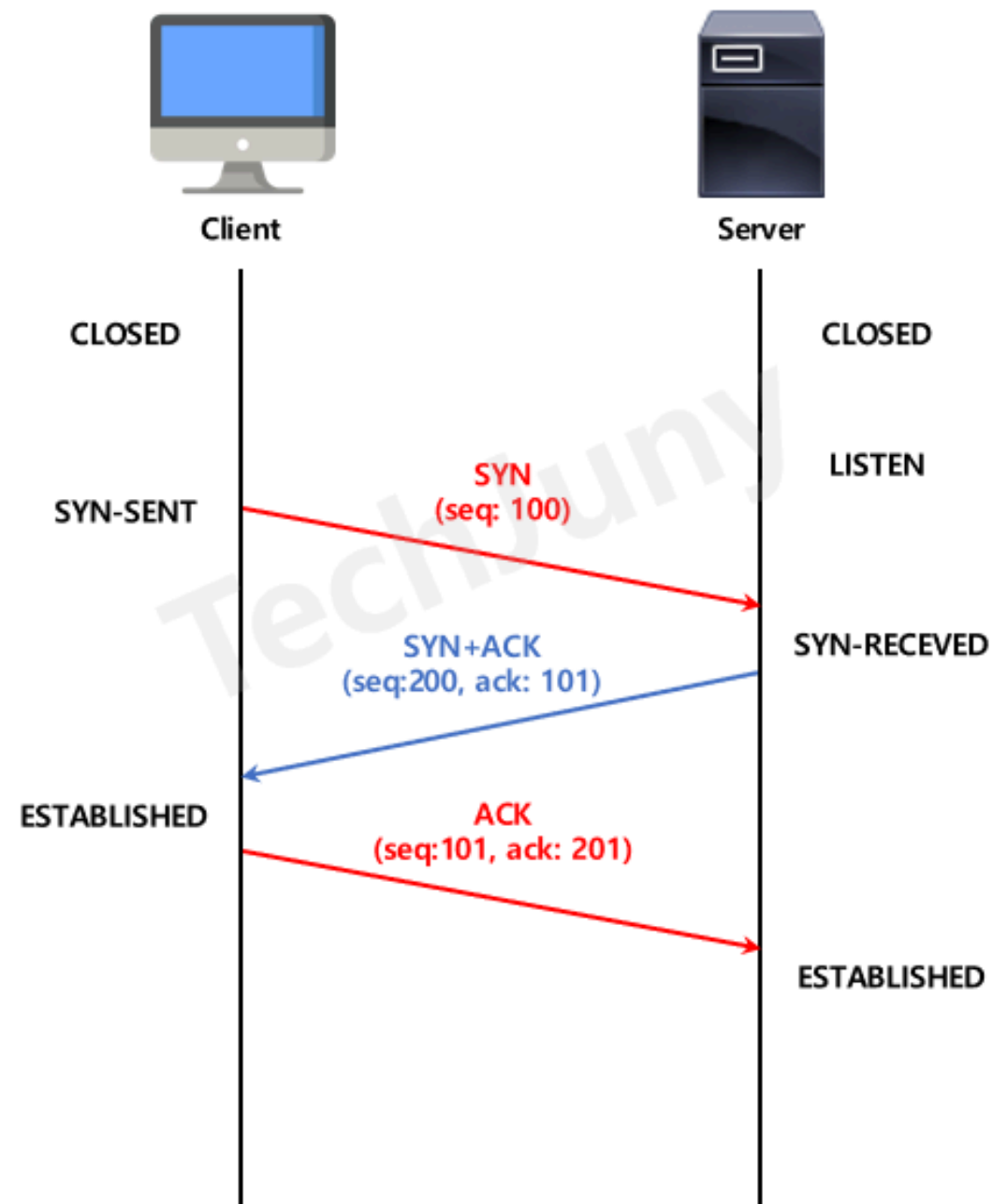
01 OSI 7계층과 캡슐화



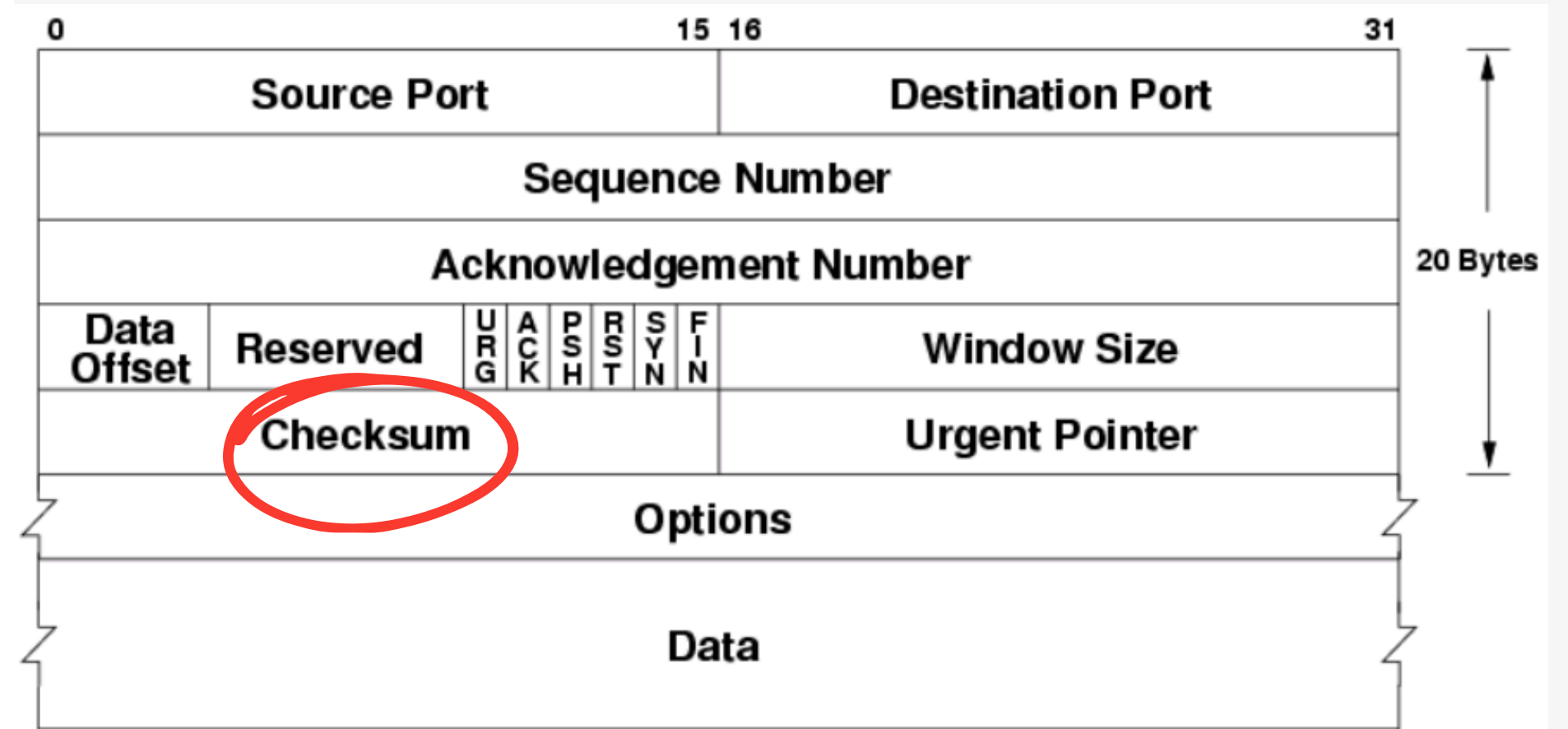
02 전송 계층 (Transport Layer)

- 전송 계층 (Transport Layer)
 - 프로세스 간의 신뢰성 있는 통신
 - 대표 프로토콜: TCP
- TCP의 신뢰성을 위한 3가지 요소:
 - 연결 지향: 3-Way Handshake를 통한 사전 연결
 - 데이터 무결성: Sequence/Ack 번호, Checksum
 - 흐름/혼잡 제어: Sliding Window

02 전송 계층 (Transport Layer)



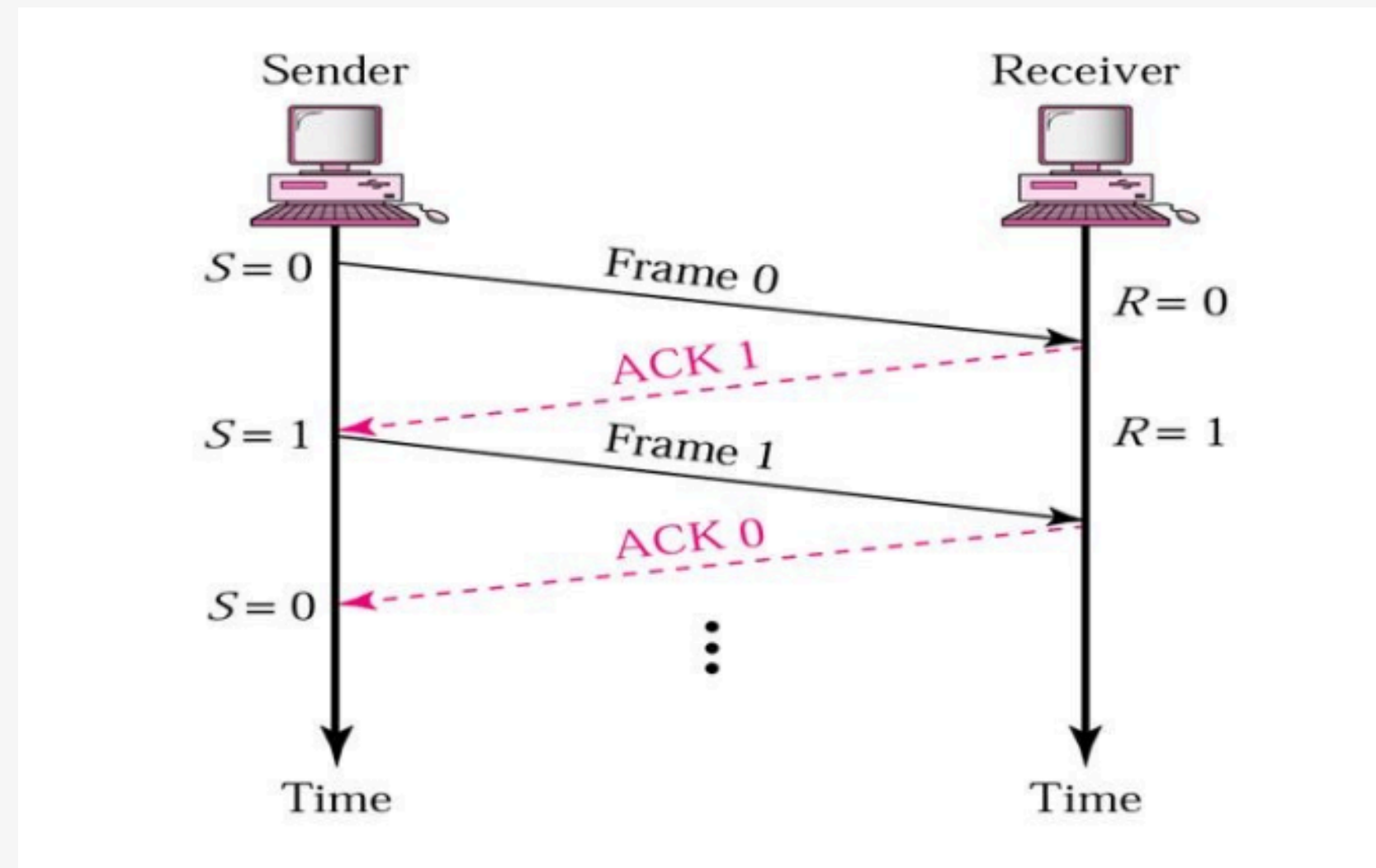
3-Way Handshake



TCP Header

02 흐름 제어

- 기존 Stop-and-Wait 방식의 비효율
 - 데이터 하나 보내고, ACK 올 때까지 대기 (Idle)
 - 확실한 오버플로우 방지 → But, 네트워크 자원 낭비



02 흐름 제어 - 슬라이딩 윈도우

- 슬라이딩 윈도우
 - 윈도우 크기(Window Size)만큼 ACK 없이 연속으로 데이터 전송
 - 수신자는 받은 데이터 기준 누적 ACK(Cumulative ACK)로 응답
- 동작 원리
 - 송신자는 윈도우 크기만큼 데이터를 연속 전송
 - 수신자는 "N번까지 잘 받았으니, N+1번 줘" 라는 의미의 누적 ACK 전송
 - 송신자는 ACK를 받은 만큼 윈도우를 오른쪽으로 밀어서(Slide) 다음 데이터 전송
- 손실 기반 TCP Cubic 알고리즘의 한계
 - TCP Cubic (3차 함수 그래프) 형태로 패킷 손실 기반으로 전송 속도를 조절
 - 버퍼블로트 (Bufferbloat): 패킷 손실이 날 때까지 버퍼를 꽉 채우기 때문에, 평상시에도 라우터의 버퍼가 꽉 차서 지연 시간(Latency) 발생
 - 비효율성: 실제로는 길이 더 넓어질 수 있는데도, 굳이 벽에 부딪혀 보고 나서야 속도를 줄이는 방식

02 흐름 제어

window size = 7



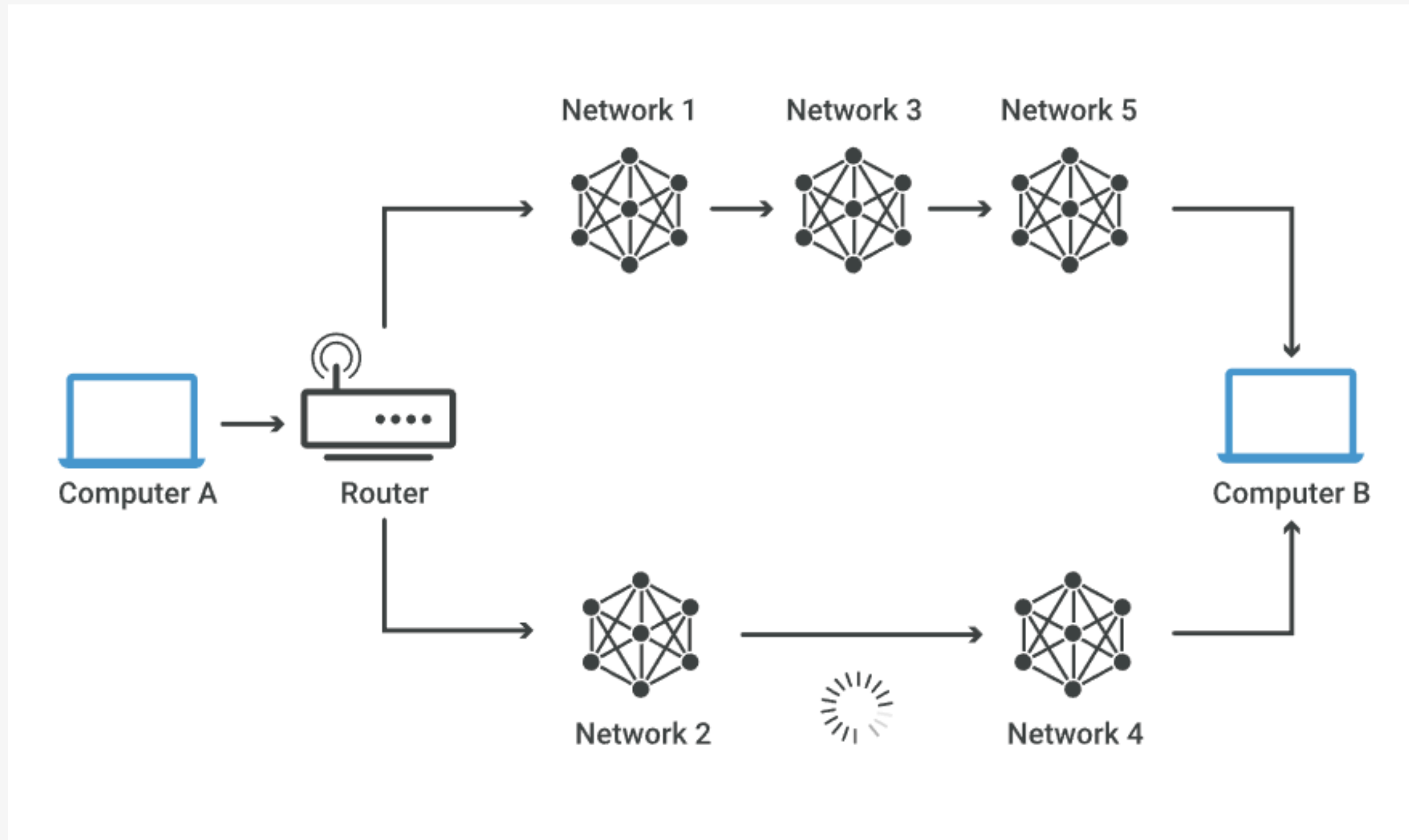
02 흐름 제어 - Rate Control (Measurement-based)

- 예시 - 구글의 BBR
- 데이터를 보낸 시간과 데이터에 대한 ACK가 도착한 시간을 측정
 - RTT (Round-Trip Time): 신호가 왕복하는 데 걸리는 시간.
 - 병목 대역폭 (Bottleneck Bandwidth): 내 컴퓨터에서 서버까지 가는 길목 중에서 가장 좁은 길(병목)이 초당 얼마나 많은 데이터를 처리할 수 있는가
- 지속적으로 ACK 도착 시간을 분석해서 '가장 효율적인 전송률'을 스스로 모델링하고, 그 속도에 맞춰서 데이터를 전송
- 장점: 저지연 & 높은 처리량

03 네트워크 계층 (Network Layer)

- 3계층 (Network): 최적 경로 탐색 및 패킷 포워딩
- 대표 프로토콜: IP (Internet Protocol)
- 특징
 - 비신뢰성 (Unreliable): 최선 노력형 전송, 도착 보장 X
 - 비연결성 (Connectionless): 사전 연결 절차 X
- 주요 헤더
 - Source/Destination IP 주소
 - TTL (Time To Live): 무한 루프 방지를 위한 패킷 수명
 - Header Checksum: IP 헤더 자체의 오류 검증

03 네트워크 계층 (Network Layer)



04 데이터 계층 (Data Link Layer)

- 2계층 (Data Link): 같은 같은 네트워크 세그먼트 간 데이터 전달 장비 대 장비로 데이터를 전달
- IP 주소 (논리적 주소): 네트워크 상의 위치, 네트워크 환경이 바뀌면 변경
- MAC 주소 (물리적 주소): 네트워크 인터페이스 카드(NIC)에 고정된 유일한 하드웨어 주소, 절대 바뀌지 않음

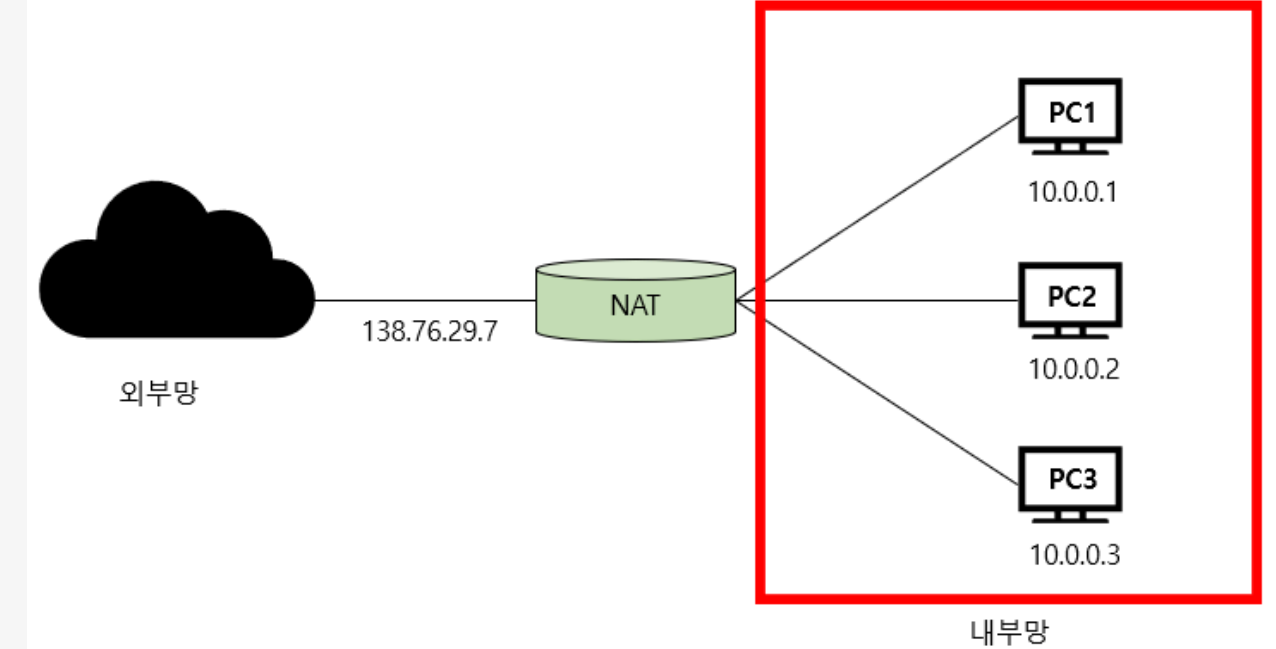
	IP 주소	MAC 주소
역할	최종 목적지	바로 다음 홉(Hop)
주소	논리적 주소	물리적 주소 (HW)
변경	O (네트워크 변경 시)	X (고유값)

04 데이터 계층 (Data Link Layer) - ARP

- 다음 홉인 공유기의 IP 주소는 알지만, 프레임을 만들기 위해 필요한 공유기의 MAC 주소는 어떻게 알 수 있을까?
- ARP(Address Resolution Protocol)
 - ARP Request (Broadcast): "192.168.0.1의 MAC 주소는?" (네트워크 전체에 질문)
 - ARP Reply (Unicast): "그건 나야. 내 MAC은 A1-B2-..." (해당 장비만 1:1로 응답)
 - ARP Cache: 알아낸 MAC 주소를 캐시에 저장하여 재사용

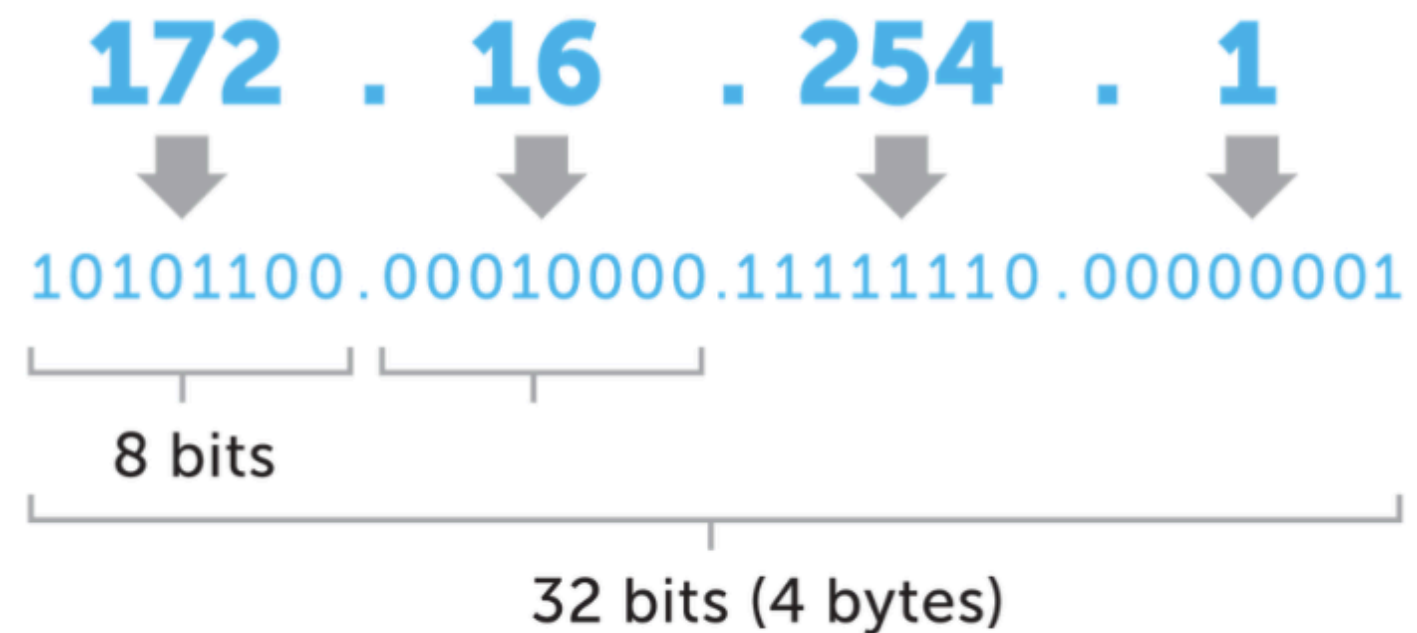
05 IPv4 주소 고갈 문제와 NAT

- NAT/PAT: IPv4의 생명을 연장시킨 주소 변환 기술
- 문제: 전 세계 IPv4 주소 (약 43억 개) 고갈
- 원리: 하나의 공인 IP와 포트 번호를 이용해, 여러 사설 IP 기기가 외부와 통신
- 동작 예시:
 - 내부 192.168.0.5:12345 → [NAT] → 외부 210.10.20.30:50001
 - 공유기의 NAT 테이블이 이 변환 정보를 기억하고 양방향 통신을 중개



06 IPv6

IPv4 address in dotted-decimal notation



IPv4

An IPv6 address (in hexadecimal)

2001:0DB8:AC10:FE01:0000:0000:0000:0000

↓ ↓ ↓ ↓

2001:0DB8:AC10:FE01:: Zeroes can be omitted

IPv6

06 IPv6

IPv4와 IPv6 비교

구분	IPv4	IPv6
주소 길이	32비트	128비트
주소 할당	DHCP로 동적 할당	SLAAC로 자동 할당
헤더 구조	복잡하고 가변 길이	단순하고 고정 길이
Checksum	헤더에 포함됨	헤더에서 제거됨
데이터 처리	라우터에서 매번 체크섬 계산	라우터에서 계산 없이 처리 (속도 향상)
보안 기능	IPsec을 선택적으로 사용	IPsec이 기본 탑재

- 전환 기술: 듀얼 스택 (Dual Stack), 터널링 (Tunneling)

07 요약

- 데이터의 흐름: 캡슐화(세그먼트→패킷→프레임)
- 신뢰성 vs 경로탐색: TCP(4계층) vs IP(3계층)
- 주소의 두 종류: IP(최종 목적지) vs MAC(다음 홉), 그리고 ARP
- 주소 고갈과 해결책: NAT vs IPv6

Thank you

감사합니다.