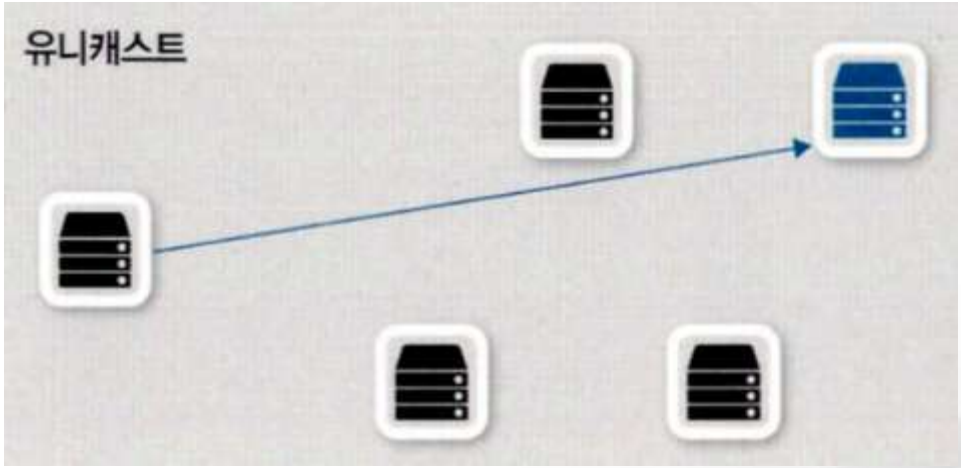
네트워크 통신

수신자의 범위에 따른 분류

- 종류 – 목적지 Address를 기준으로 구분
  - ✓ Unicast
  - ✓ Multicast
  - ✓ Broadcast
  - ✓ Anycast

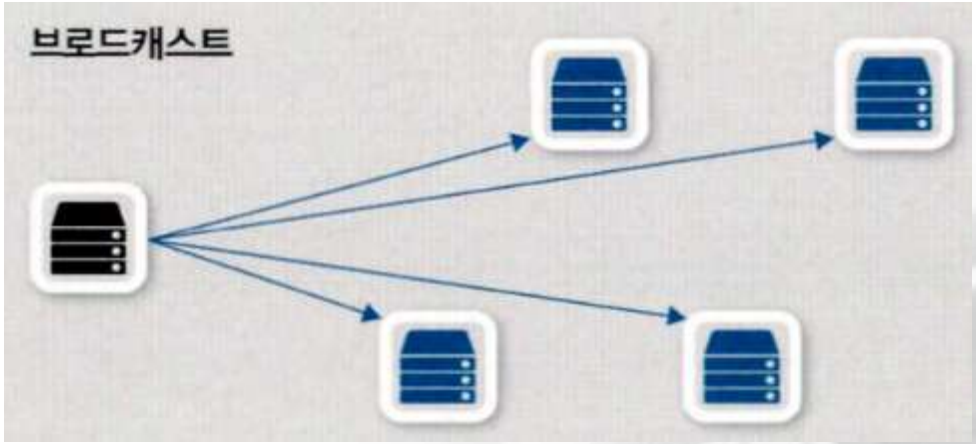
수신자의 범위에 따른 분류

- ❖ Unicast
  - ✓ 개요
    - 1:1 통신
    - 출발지 와 목적지가 1:1로 통신
    - 실제 사용하는 대다수의 통신



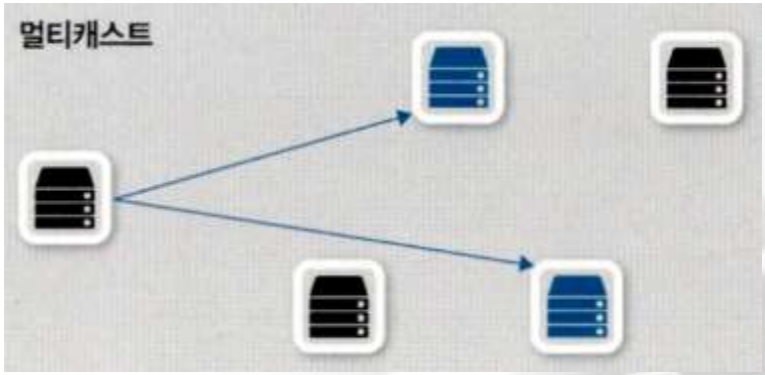
수신자의 범위에 따른 분류

- ❖ Broadcast
  - ✓ 개요
    - 동일 네트워크에 존재하는 모든 호스트가 목적지
  - ✓ 용도
    - Broadcast는 목적지 Address가 네트워크 전체로 표기되어 있는 통신 방식으로 Unicast로 통신하기 전에 상대방의 정확한 위치를 알기 위해 사용
    - Address 체계에 따라 Broadcast를 다양하게 분류할 수 있지만 기본 동작은 로컬 네트워크 내에서 모든 호스트에 패킷을 전달해야 할 때 사용



수신자의 범위에 따른 분류

- ❖ Multicast
  - ✓ 개요
    - 1:그룹(Multicast 구독 호스트) 통신
    - 하나의 출발지에서 다수의 특정 목적지로 데이터 전송
  - ✓ 용도
    - Multicast는 Multicast 그룹 Address를 이용해 해당 그룹에 속한 다수의 호스트로 패킷을 전송하기 위한 통신 방식으로 IPTV와 같은 실시간 방송을 구현할 때 이 Multicast 통신 방식을 사용
    - 사내 방송이나 증권 시세 전송과 같이 단방향으로 다수에게 동시에 같은 내용을 전달할 때 사용



수신자의 범위에 따른 분류

- Anycast
  - ✓ 개요
    - 1:1통신(목적지는 동일 그룹 내의 1개 호스트)
    - 다수의 동일 그룹 중 가장 가까운 호스트에서 응답
    - IPv4에서는 일부 기능 구현
    - IPv6은 모두 구현 가능
  - ✓ 용도
    - Anycast는 Anycast Address가 같은 호스트들 중에서 가장 가깝거나 가장 효율적으로 서비스 할 수 있는 호스트와 통신하는 방식
    - 가장 가까운 DNS 서버를 찾을 때 사용

수신자의 범위에 따른 분류

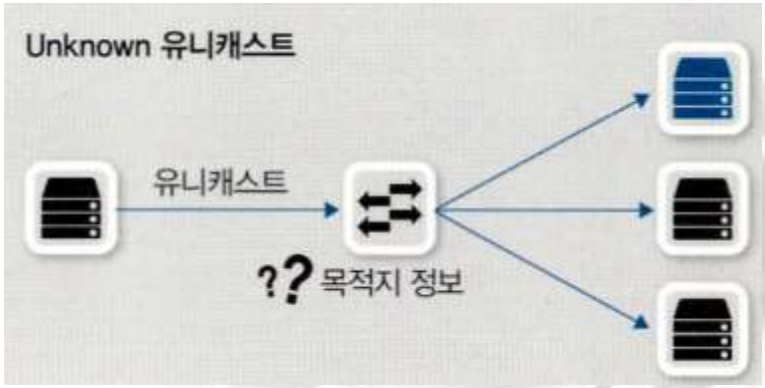
- 결론
  - ✓ 최종 통신은 1:1로 Unicast와 Anycast가 동일하지만 통신할 수 있는 후보자는 서로 다른데 Unicast는 출발지와 목적지가 모두 한 대 씩이지만 Anycast는 같은 목적지 Address를 가진 서버가 여러 대 여서 통신 가능한 다수의 후보군이 있음
  - ✓ 현재 주로 사용되는 네트워크 Address 체계는 IPv4 기반이며 일부 모바일 네트워크와 데이터 센터 위주로 새로운 IPv6 기반 Address 체계가 사용되고 있는데 IPv6에서는 Broadcast가 존재하지 않고 링크 로컬 Multicast로 대체되어 사용됨

수신자의 범위에 따른 분류

- ❖ 링크 로컬 Multicast
  - ✓ 개요
    - 같은 물리적 네트워크(링크) 안에 있는 특정 그룹의 장치들에만 데이터를 전송
    - 링크 로컬은 라우터를 넘어가지 않는 동일한 로컬 네트워크 내로 범위를 한정된다는 의미
  - ✓ 특징
    - 범위 제한: 데이터가 라우터를 통과하지 못하므로 오직 같은 스위치나 허브에 연결된 장치들끼리만 통신이 가능
    - 네트워크상의 장치들이 서로를 찾거나(Neighbor Discovery) 자동으로 설정을 주고받을 때 사용
    - Broadcast와 달리 해당 데이터를 받기로 설정된 그룹 멤버들에게만 전달되어 불필요한 트래픽을 줄임
  - ✓ 사용 사례
    - IPv6 환경에서는 Broadcast가 아예 존재하지 않기 때문에 링크 로컬 Multicast가 그 역할을 대신 수행
    - Address 자동 설정(SLAAC): 장치가 네트워크에 접속하자마자 자신의 존재를 알리거나 중복 Address를 확인할 때 사용
    - 인근 장치 탐색(Neighbor Discovery): IPv4의 ARP 대신 상대방의 MAC Address를 알아내기 위해 Multicast 메시지를 보냄

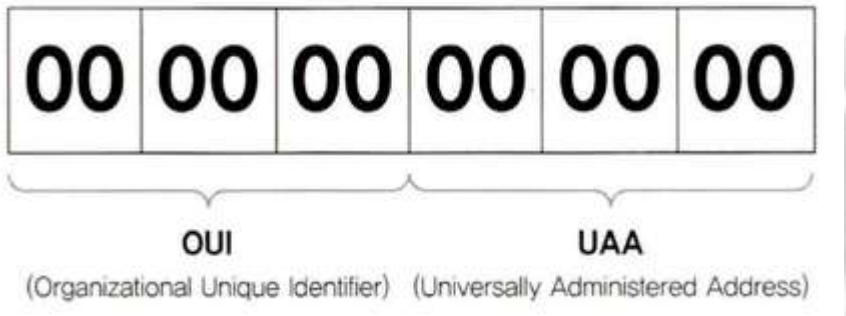
수신자의 범위에 따른 분류

- BUM 트래픽
  - ✓ BUM 트래픽이라는 용어를 쓰는 경우가 있는데 BUM은 B(Broadcast), U(Unknown Unicast), M(Multicast)을 지칭하는데 B, U, M은 서로 다른 종류의 트래픽이지만 네트워크에서의 동작은 유사
  - ✓ Unknown Unicast는 목적지 Address가 명확히 명시되어 있지만 네트워크에서의 동작은 Broadcast와 유사한데 스위치가 목적지에 대한 Address를 학습하지 못했기 때문에 스위치 입장에서 패킷을 모든 포트로 플러딩(전송)하는데 이런 Unicast가 Unknown Unicast
  - ✓ Unicast이지만 실제로 겉으로 보이는 동작 방식은 Broadcast에 가까움
  - ✓ Unicast이므로 전달받는 모든 단말 NIC에서 도착지 Address를 확인하고 자신이 목적지가 아니면 패킷을 버리지만 네트워크 입장에서는 네트워크 자원을 쓸데없이 사용하므로 네트워크상에서 불필요한 BUM 트래픽이 많아지면 네트워크 성능이 저하될 수 있음
  - ✓ 이더넷 환경에서는 ARP Broadcast를 먼저 보내고 이후 통신을 시작하므로 BUM 트래픽이 많이 발생하지 않음



MAC Address

- MAC Address 체계
  - ✓ 개요
    - MAC Address는 변경할 수 없도록 하드웨어에 고정되어 출하되므로 네트워크 구성 요소마다 다른 Address를 가지고 있음
    - 모든 네트워크 장비 제조업체에서 장비가 출하될 때마다 MAC Address를 할당하게 되는데 매번 이 Address의 할당 여부를 확인할 수 없으므로 한 제조업체에 하나 이상의 Address 풀을 주고 그 풀 안에서 각 제조업체가 자체적으로 MAC Address를 할당
    - 네트워크 장비 제조업체에 Address 풀을 할당하는 것을 제조사 코드(Vendor Code)라고 부르며 이 Address는 국제기구인 IEEE가 관리
    - MAC Address는 48비트의 16진수 12자리로 표현
    - 앞의 24비트와 뒤의 24비트로 나누어 구분하는데 MAC Address 앞의 24비트 인 OUI는 제조사 코드이고 뒤의 24비트의 값인 UAA는 각 제조사에서 자체적으로 할당하여 네트워크에서 각 장비를 구분하게 함



MAC Address

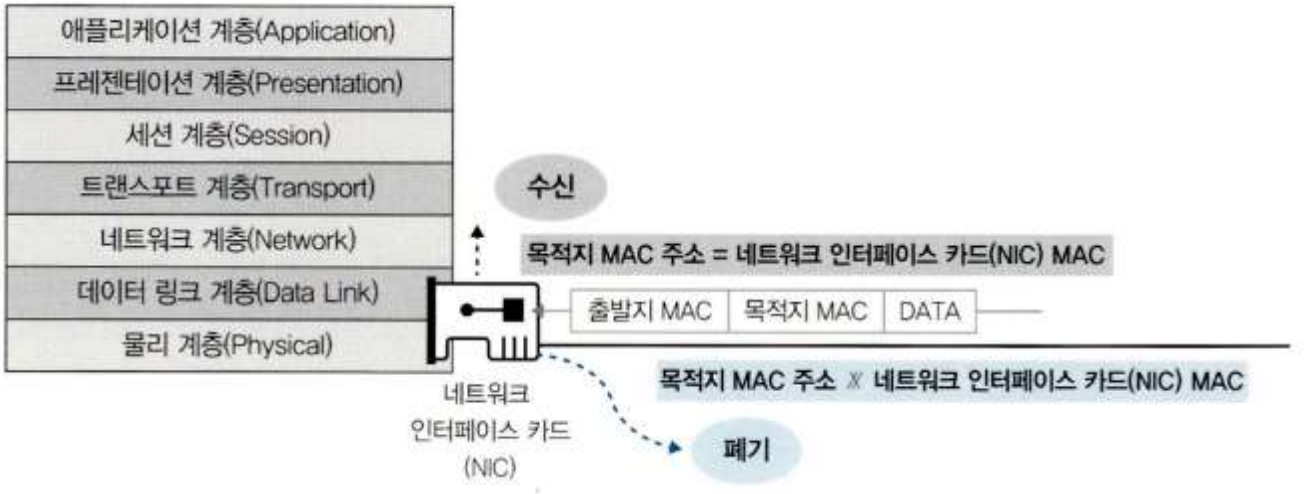
- MAC Address 체계
  - ✓ 유일하지 않은 MAC Address
    - MAC Address는 유일한 값이라고 생각하지만 유일하지 않을 수도 있음
    - 네트워크 장비 제조업체는 자신의 제조업체 코드 내에서 뒤의 24비트의 UAA 값을 할당하는데 실수나 의도적으로 MAC Address가 중복될 수도 있음
    - MAC Address는 동일 네트워크에서만 중복되지 않으면 동작하는 데 문제가 없음
    - 네트워크 통신을 할 때 네트워크가 달라 라우터의 도움을 받아야 할 경우 라우터에서 다른 네트워크로 넘겨줄 때 출발지와 도착지의 MAC Address가 변경되므로 네트워크를 넘어가면 기존 출발지와 도착지 MAC Address를 유지하지 않음

MAC Address

- MAC Address 체계
  - ✓ MAC Address 변경
    - MAC Address는 BIA(Burned-In Address) 상태로 NIC에 할당
    - 일반적으로 ROM 형태로 고정되어 출하되므로 NIC에 고정된 MAC Address를 변경하기는 어렵지만 결국 이 MAC Address도 메모리에 적재되어 구동되므로 여러 가지 방법을 이용해 변경된 MAC Address로 NIC을 동작시킬 수 있음
    - 보안상의 이유로 MAC Address 변경을 막아놓은 운영체제도 있지만 손쉽게 명령어나 설정 파일 변경만으로도 MAC Address를 변경할 수 있는 운영체제도 있음
    - 윈도의 경우 Driver 상세 정보에서 MAC Address 변경을 제공하기 때문에 쉽게 변경이 가능
    - 리눅스는 GNU MacChanger나 각 리눅스 배포판의 네트워크 설정 파일에 MAC Address를 입력하면 Address 변경이 가능

MAC Address

- MAC Address 동작
  - ✓ 기본 동작 방식
    - NIC는 자신의 MAC Address를 가지고 있고 전기 신호가 들어오면 2계층에서 데이터 형태(패킷)로 변환하여 내용을 구분한 후 도착지 MAC Address를 확인
    - 도착지 MAC Address가 자신이 갖고 있는 MAC Address와 다르면 그 패킷을 폐기
    - 패킷의 목적지 Address가 자기 자신이거나 Broadcast, Multicast와 같은 그룹 Address 이면 처리해야 할 Address로 인지해 패킷 정보를 상위 계층으로 넘겨줌
    - 도착지 Address가 일치하지 않아 NIC에서 자체적으로 패킷을 폐기하는 경우와 달리 본인의 Address, Broadcast Address는 NIC 자체적으로 패킷을 처리하는 것이 아니라 OS나 애플리케이션에서 처리해야 하므로 시스템에 부하가 작용



MAC Address

- MAC Address 동작
  - ✓ 무차별 모드(Promiscuous Mode)
    - 기본 NIC 동작 방식은 패킷이 자신의 MAC Address와 일치하지 않는 도착지 Address를 가졌을 경우 자체적으로 폐기됨
    - 네트워크 상태를 모니터링하거나 디버그 또는 분석 용도로 네트워크 전체 패킷을 수집해 분석해야 할 경우 NIC가 정상적으로 동작하면 다른 목적지를 가진 패킷을 분석할 수 없음
    - 다른 목적지를 가진 패킷을 분석하거나 수집해야 할 경우 무차별 모드로 NIC를 구성
    - 무차별 모드는 자신의 MAC Address와 상관없는 패킷이 들어와도 이를 분석할 수 있도록 메모리에 올려 처리할 수 있게 함
    - 무차별 모드를 사용하는 가장 대표적인 애플리케이션은 네트워크 패킷 분석 애플리케이션인 Wireshark

MAC Address

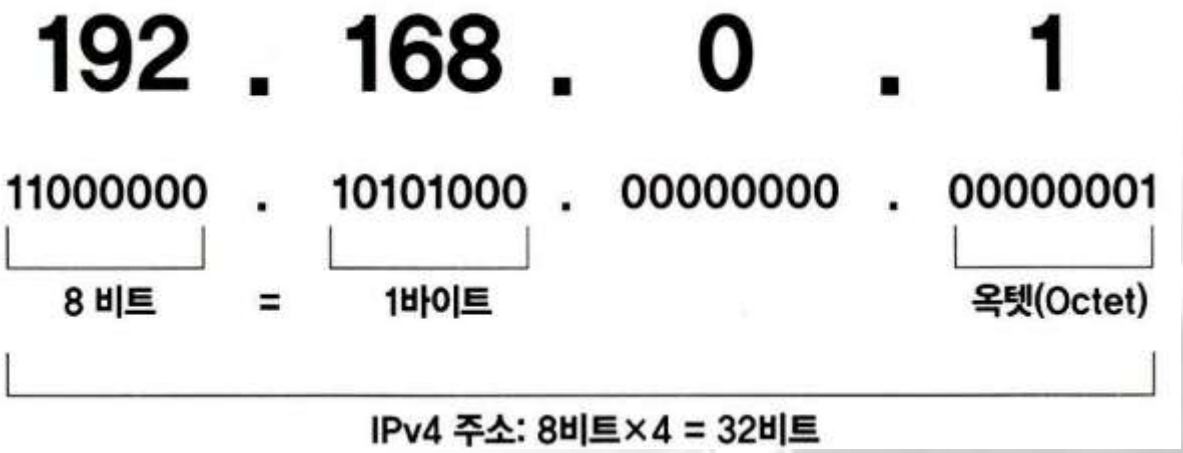
- MAC Address 동작
  - ✓ MAC Address를 여러 개 갖는 경우
    - MAC Address는 단말에 종속되지 않고 NIC에 종속되는데 단말은 NIC를 여러 개 가질 수 있으므로 MAC Address도 여러 개 가질 수 있음
    - 멀티 레이어 스위치, 라우터와 같은 복잡한 네트워크 장비는 NIC가 여러 개이고 MAC Address도 여러 개가 할당됨
  - ✓ MAC Address로 제조업체 찾기
    - MAC Address에 제조업체를 나타내는 OUI 값이 있으므로 MAC Address를 알면 해당 장비를 생산한 업체를 확인할 수 있음
    - 구글과 같은 검색 사이트에서 MAC Address만 검색해도 되지만 MAC Address를 관리하는IEEE 홈페이지에서 더 정확한 최신 정보를 확인할 수도 있음
    - URL: https://regauth.standards.ieee.org/standards-ra-web/pub/view.html#registries
    - 단축 URL: bit.ly/ieee_list

IP Address

- 개요
  - ✓ OSI 7계층에서 Address를 갖는 계층은 2계층과 3계층
  - ✓ 2계층은 물리 Address인 MAC Address를 사용하고 3계층은 논리 Address인 IP Address를 사용
  - ✓ 특징
    - 사용자가 변경 가능한 논리 Address
    - Address에 레벨이 있는데 그룹을 의미하는 네트워크 Address와 호스트 Address로 나눔

IP Address

- IP Address 체계
  - ✓ IP는 v4, v6 두 체계가 사용되며 IPv6 Address는 128비트 체계임
  - ✓ 우리가 많이 사용하는 IP Address는 32비트인 IPv4 Address
  - ✓ IPv4 Address를 표기할 때는 4개의 옥텟(Octet)이라고 부르는 8비트 단위로 나누고 각 옥텟은 "."으로 구분
  - ✓ 2계층의 MAC Address가 16진수로 표기된 것과 달리 IP Address는 10진수로 표기하므로 8비트 옥텟은 0~255의 값을 사용할 수 있음



IP Address

- IP Address 체계
  - ✓ 2계층 Address인 MAC Address가 제조업체 코드인 OUI와 제조업체별 일련 번호인 UAA의 두 부분으로 나뉘는 것과 목적이 다르지만 3계층 Address인 IP Address도 네트워크 Address와 호스트 Address 두 부분으로 나눔
    - 네트워크 Address
      - 호스트들을 모은 네트워크를 지칭하는 Address
      - 네트워크 Address가 동일한 네트워크를 로컬 네트워크라고 함
    - 호스트 Address: 하나의 네트워크 내에 존재하는 호스트를 구분하기 위한 Address
  - ✓ MAC Address는 24비트씩 절반으로 나뉘지만 IP Address의 네트워크 Address와 호스트 Address는 이 둘을 구분하는 경계점이 고정되어 있지 않은데 이것이 다른 Address 체계와 IP Address 체계를 구분하는 가장 큰 특징

IP Address

- IP Address 체계
  - ✓ Classful
    - IP Address 체계는 필요한 호스트 IP 개수에 따라 네트워크의 크기를 다르게 할당할 수 있는 클래스(Class) 개념을 도입
    - A 클래스는 가장 큰 Address를 갖는데 약 1.600만 개의 IP Address를 가질 수 있으며 B 클래스는 약 6만 5천 개이며 C 클래스는 약 250 개의 IP Address를 가질 수 있음
    - A 클래스는 첫 번째 옥텟에 네트워크 Address와 호스트 Address를 나누는 구분자가 있고 B 클래스는 두 번째 옥텟 C 클래스는 세 번째 옥텟에 구분자가 있음
    - 이 구분자가 서브넷 마스크

A 클래스	네트워크	호스트	호스트	호스트
기본 서브넷	255	0	0	0
B 클래스	네트워크	네트워크	호스트	호스트
기본 서브넷	255	255	0	0
C 클래스	네트워크	네트워크	네트워크	호스트
기본 서브넷	255	255	255	0

D 클래스 멀티캐스트(Multicast)

E 클래스 예약(Reserved for futures)

IP Address

- IP Address 체계
  - ✓ Classful
    - 네트워크 Address와 호스트 Address를 나누는 구분자가 고정되어 있다면 네트워크가 가질 수 있는 호스트 IP 숫자가 같기 때문에 모두 같은 크기의 네트워크가 되지만 구분자가 이동할 수 있어 네트워크의 크기가 달라질 수 있음
    - IP Address가 도입한 클래스 개념은 다른 고정된 네트워크 Address 체계에 비해 Address를 절약할 수 있다는 장점이 있음
    - 네트워크의 크기가 모두 같은 경우 큰 네트워크를 필요로 하는 조직은 네트워크를 여러 개 확보해야 하는 어려움이 있고 연속된 네트워크를 할당받기 어려움
    - 작은 네트워크가 필요한 조직의 입장에서는 너무 많은 IP를 가져가므로 IP가 낭비됨
    - A 클래스는 네트워크 Address를 표현하는 부분이 1개의 옥텟이고 호스트 Address를 나타낼 수 있는 부분이 3개의 옥텟이기 때문에 2^8(256)개의 네트워크와 한 네트워크 당 2^{24}(16,777,216)개의 호스트 Address를 갖게 됨
    - B 클래스는 네트워크 Address 부분이 2개의 옥텟이고 호스트 Address가 2개의 옥텟이기 때문에 2^{16}(65,536)개의 네트워크 와 1개의 네트워크에 2^{16}(65,536) 개의 IP를 가질 수 있음
    - C 클래스는 세 번째 옥텟에 네트워크 Address와 호스트 Address를 나누는 구분자가 있고 2^{24} (16,777,216)개의 네트워크 와 1개의 네트워크 당 2^8(256)개의 호스트를 가질 수 있음

IP Address

❖ IP Address 체계

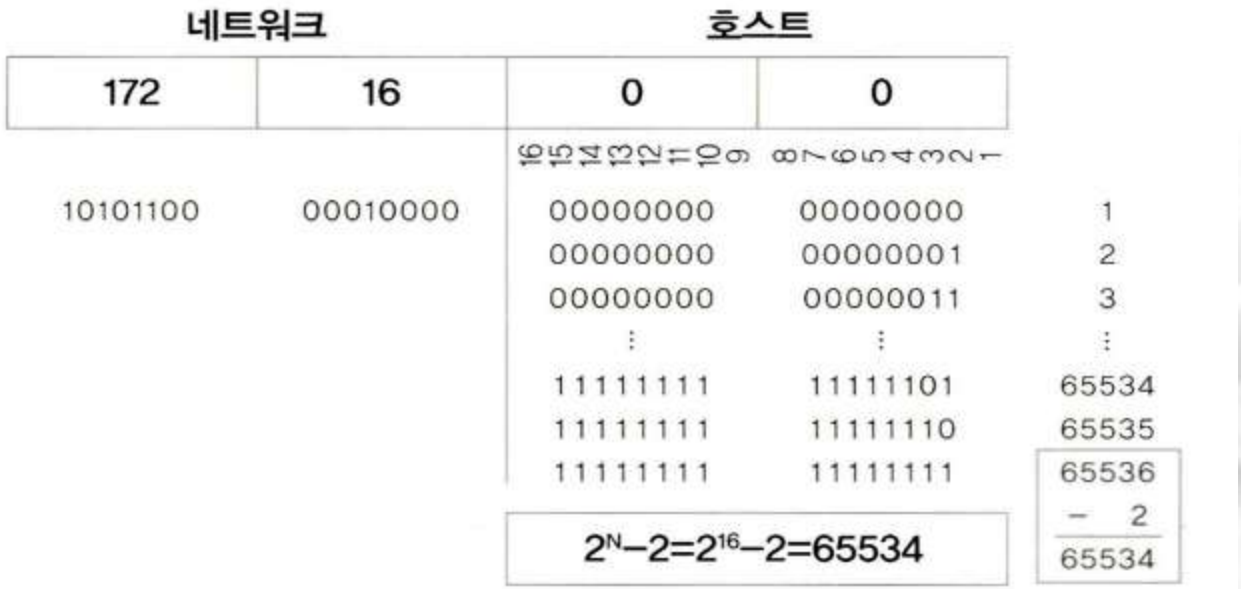
✓ Classful

IP Address

- IP Address 체계
  - ✓ Classful
    - A. B. C 클래스는 맨 앞 옥텟의 Address만 보고 구분할 수 있는데 앞 옥텟의 Address가 0 ~127의 범위이면 이 Address는 A 클래스인데 첫 옥텟을 이진수로 표기했을 때 맨 앞 자리가 1인 Address가 A 클래스인데 2진수로 첫 옥텟이 00000000-01111111 인데 127만 예외로 자신을 의미하는 루프백(Loopback) Address로 사용되므로 실제로 A 클래스로 사용할 수 있는 Address는 1.0.0.0~ 126.255.255.255 까지
    - B 클래스는 첫 옥텟을 2진수로 표기했을 때 첫 번째 자리가 1이고 두 번째 자리가 0인 Address로 2진수로 첫 옥텟이 1000000-10111111 인 수를 10진수로 표현하면 128 ~ 191까지 이고 이 수를 갖는 IP는 B 클래스
    - C 클래스는 첫 옥텟을 2진수로 표기했을 때 8 자리 중 첫 번째 두 번째 자리가 1이고 세 번째 자리가 0인 Address로 첫 옥텟이 11000000-11011111로 표기되며 10진수로는 192~223 까지의 IP가 C 클래스
    - 클래스 기반의 네트워크 분할 기법은 과거에 사용했던 개념으로 현재는 위에서 설명한 것처럼 클래스 기반으로 네트워크를 분할하지 않음
    - 보다 네트워크 Address를 세밀하게 분할하고 할당하기 위해 필요한 네트워크의 크기에 맞추어 1비트 단위로 네트워크를 상세히 분할하는 방법을 사용

IP Address

- IP Address 체계
  - ✓ Classful
    - 사용 가능한 호스트 개수 파악
      - 네트워크 Address: 172.16.0.0
      - Broadcast Address: 172.16.255.255
      - 유효 IP 범위: 172.16.0.1 ~172.16.255.254



IP Address

- IP Address 체계
  - ✓ Classful
    - 사용 가능한 호스트 개수 파악
      - IP Address 체계에서 설명한 클래스(Class) 기반의 IP Address 체계를 클래스풀(Classful)이라고 부르는데 IP Address 체계를 처음 만들었을 때는 클래스 개념을 도입한 것이 확장성이 있고 Address 낭비가 적은 최적의 조건을 만들 수 있었던 좋은 선택이었는데 이 Address 체계에서는 네트워크 Address와 호스트 Address를 구분짓는 구분자(서브넷 마스크)가 필요없음
      - 맨 앞자리 숫자만 보면 자연스럽게 이 Address가 어느 클래스에 속해 있는지 구분할 수 있고 Address 구분자를 적용할 수 있음

IP Address

- IP 주소 체계
  - ✓ IP 주소 부족
    - 인터넷이 상용화되면서 인터넷에 연결되는 호스트 숫자가 폭발적으로 증가함
    - 기존 클래스풀 기반의 Address 체계는 확장성과 효율성을 모두 잡는 좋은 Address 체계였지만 기하급수적으로 늘어나는 IP Address 요구를 감당하기에는 너무 부족
    - 이론적으로 사용할 수 있는 IP 개수는 43억여 개이지만 실제로 사용할 수 있는 IP 숫자는 이보다 훨씬 적음
    - 현재 전 세계 인구가 하나의 IP만 갖더라도 IP 할당이 불가능한 크기이고 이 외에도 네트워크 Address를 계층화하고 분할하기 위해 낭비되는 IP가 매우 많았음
    - 하나의 네트워크에서 IP가 사용되지 않더라도 그 IP를 다른 네트워크에서 사용하지 못함
    - IP Address 부족과 낭비 문제를 해결하기 위한 3가지 전략
      - CIDR((Classless inter-Domain Routing) 기반의 Address 체계
      - NAT와 Private IP Address
      - IPv6

IP Address

- ❖ 클래스리스
  - ✓ 등장
    - IPv4의 가장 큰 문제는 Address 자체의 부족도 있지만 상위 클래스(A Class)를 할당받은 조직에서 이 Address들을 제대로 사용하지 못하면서 낭비하는 것
    - 인터넷 초창기에 여러 회사에서 미래를 위해 IP를 많이 확보할 수 있는 A 클래스를 할당받았지만 실제로는 수천 수만 개만 사용하는 곳이 대부분이었고 나머지 수천만 개의 IP는 사용되지 못함
    - 클래스풀에서는 한 개의 클래스 네트워크가 한 조직에 할당되면 아무리 비어 있는 Address라도 IP를 분할해 다른 기관이 사용하도록 할 수 없음
    - 이 문제를 해결하기 위해 클래스 개념 자체를 버리는데 이를 클래스리스라고 부름
    - 현재 우리가 사용하는 Address 체계는 클래스 개념을 적용하지 않는 클래스리스 기반 Address 체계

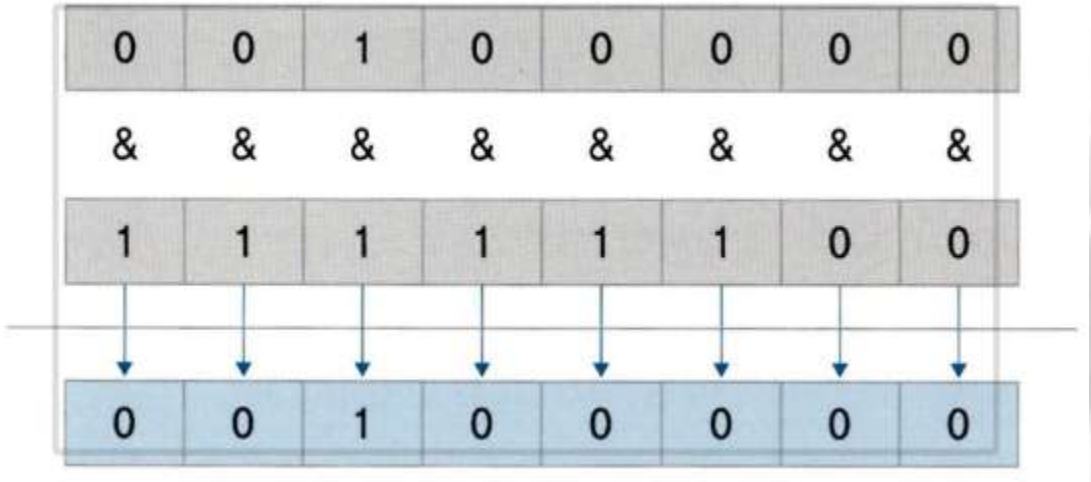
IP Address

- 클래스리스
  - ✓ 기존 방식

	8비트	8비트	8비트	8비트
A 클래스	네트워크	호스트	호스트	호스트
기본 서브넷 마스크	255	0	0	0
B 클래스	네트워크	네트워크	호스트	호스트
기본 서브넷 마스크	255	255	0	0
C 클래스	네트워크	네트워크	네트워크	호스트
기본 서브넷 마스크	255	255	255	0

IP Address

- ❖ 클래스리스
  - ✓ 서브넷 마스크 표현 방법
    - 서브넷 마스크는 IP Address와 네트워크 Address를 구분할 때 사용하는데 2진수 숫자 1은 네트워크 Address로 표시하고 0은 호스트 Address로 표시
    - 우리가 편하게 받아들일 수 있는 10진수를 사용해 255.0.0.0 255.255.0.0 255.255.255.0와 같이 표현
    - 2진수 11111111을 10진수로 표현하면 255가 되어 255는 네트워크 Address 부분이고 0은 호스트 Address 부분으로 구분
    - 2진수의 and 연산 - IP Address에서 네트워크 Address만 뽑아낼 때 사용



IP Address

- ❖ 클래스리스
  - ✓ 서브넷 마스크 표현 방법
    - 103.9.32.146 Address에 255.255.255.0 서브넷 마스크를 사용하는 IP는 네트워크 Address가 103.9.32.0이고 호스트 Address는 0.0.0.146
    - 서브넷 마스크가 2진수 1인 부분(10진수 255인 부분)은 IP 숫자가 그대로 연산 결과가 되고 서브넷 마스크가 0인 부분은 모두 0으로 변경
    - 클래스리스 기반의 IP 네트워크에서는 네트워크를 표현할 때 반드시 서브넷 마스크가 필요하고 서버나 PC에 IP Address를 부여할 때도 사용되어야 함
    - 네트워크 Address 구하기

	네트워크		서브넷	호스트
호스트 주소 103.9.32.146	01100111	00001001	00100000	10010010
255.255.255.0	11111111	11111111	11111111	00000000
네트워크 주소 103.9.32.0	01100111	00001001	00100000	00000000

IP Address

- 클래스리스
  - ✓ 서브넷 마스크 표현 방법
    - 서브넷 마스크를 표현하는 방법은 비트 단위로 표현하는 방법 과 10진수로 표현하는 방법을 사용
    - 일부 네트워크 장비는 8진수나 16진수로 나타내기도 함
    - 비트 단위로 표현하는 방법은 서브넷 마스크에서 1 부분이 연속된 자릿수를 표현주는 것인데 A 클래스를 서브넷 마스크로 나타내면 첫 번째 옥텟이 1 나머지 옥텟이 0이므로 /8로 표현하고 B 클래스는 /16 C 클래스는 /24로 표기
    - 10진수로 서브넷 마스크를 표현할 때 A 클래스는 255.0.0.0, B 클래스는 255.255.0.0. C 클래스는 255.255.255.0으로 표기

IP Address

- ❖ 클래스리스
  - ✓ 서브네팅
    - 원래 부여된 클래스의 기준을 무시하고 새로운 네트워크-호스트 구분 기준을 사용자가 정해 원래 클래스풀 단위의 네트워크보다 더 쪼개 사용하는 것이 서브네팅(Subnetting)
    - 부여된 Address를 다시 잘라 사용해 서브네팅이라고 부르는데 현대 클래스리스 네트워크의 가장 큰 특징
    - 옥텟 단위로 구분되는 서브네팅은 이해와 운영이 쉽지만 옥텟 단위보다 더 잘게 네트워크를 쪼개 2진수의 1비트 단위로 네트워크를 분할
    - 다양한 서브네팅

IP 주소	103.9.32.146
서브넷	255.255.255.192
네트워크 주소	103.9.32.128

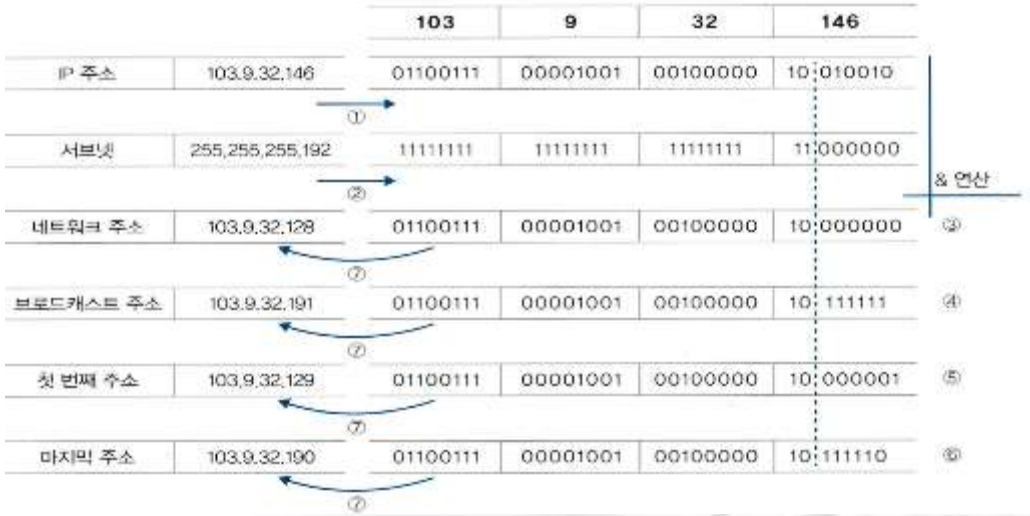
네트워크			호스트
01100111	00001001	00100000	10 010010
11111111	11111111	11111111	11 000000
01100111	00001001	00100000	10 000000

IP Address

- ❖ 클래스리스
  - ✓ 서브네팅
    - ● 실무에서 고려해야 할 부분
      - ◆ 네트워크 사용자 입장
        - ▪ 네트워크에서 사용할 수 있는 IP 범위 파악
        - ▪ 기본 게이트웨이와 서브넷 마스크 설정이 제대로 되어 있는지 확인
      - ◆ 네트워크 설계자 입장: 네트워크 설계 시 네트워크 내에 필요한 단말을 고려한 네트워크 범위 설계

IP Address

- ❖ 클래스리스
  - ✓ 네트워크 사용자의 서브네팅
    - 네트워크 사용자는 이미 설계되어 있는 네트워크에서 사용할 수 있는 IP Address 범위를 파악해야 하는데 주어진 네트워크 범위 밖의 IP를 할당하거나 서브넷 마스크를 잘못 입력하면 로컬 네트워크의 특정 범위에 속해있는 단말과 통신에 문제가 생기거나 외부 네트워크 전체에 통신하지 못하는 상황이 발생
    - 기존 클래스 단위처럼 옥텟 단위의 네트워크를 사용할 경우 모든 수가 10진수 0과 255로 표현되므로 관리자나 사용자가 이해하기 쉽지만 대부분의 서브네팅은 비트 단위로 분할되므로 이런 환경에 속해 있을 경우 어떤 IP 범위가 내가 속한 네트워크이고 어떤 IP 범위가 원격지 네트워크인지 판단하기 어려울 수 있음

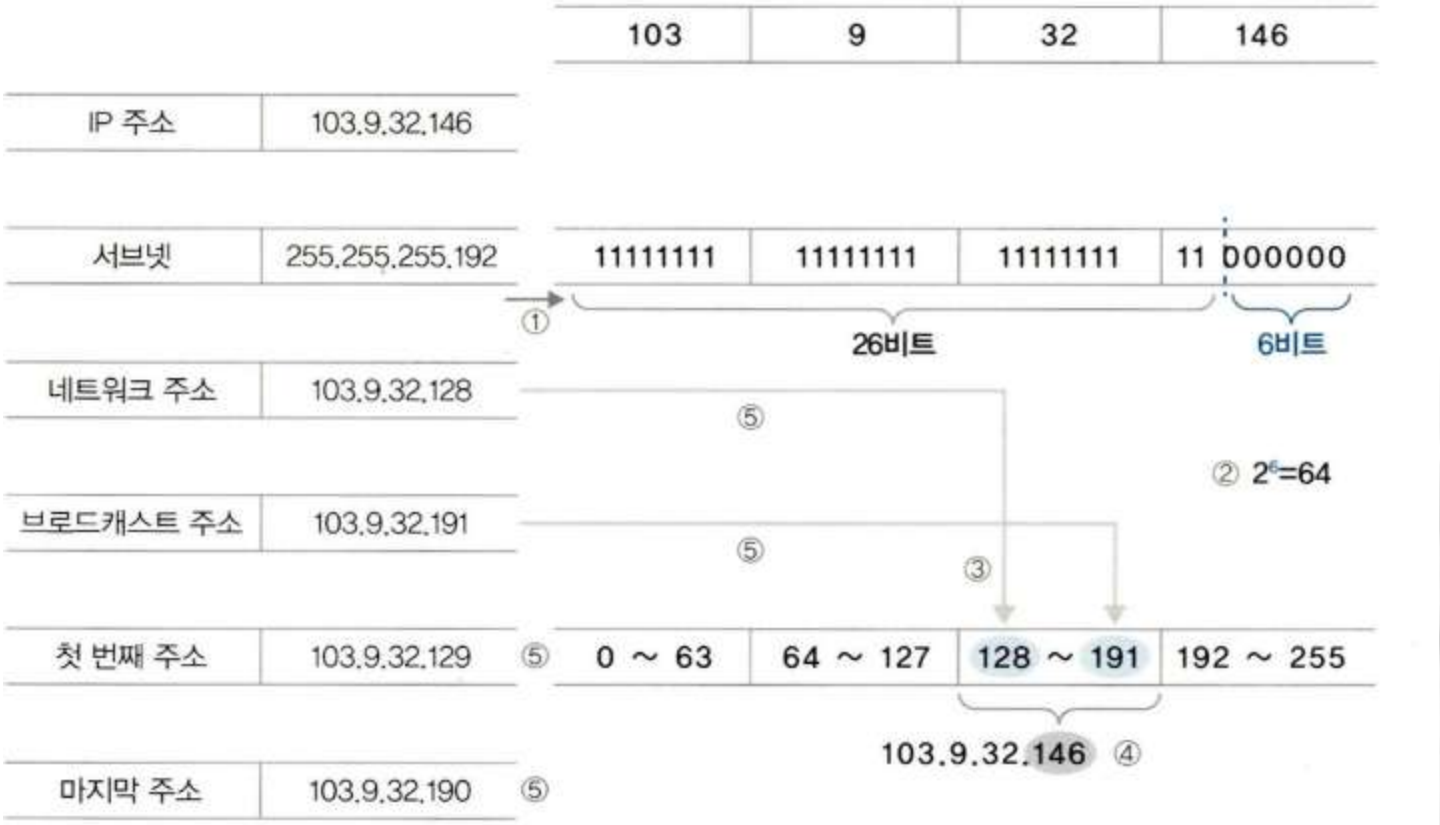


IP Address

- ❖ 클래스리스
  - ✓ 네트워크 사용자의 서브네팅
    - IP Address 체계는 컴퓨터가 처리하므로 2진수로 되어 있는데 2진수에 익숙하거나 서브네팅이 옥텟 단위로 되어 있는 경우 암산으로 내가 속한 네트워크 크기와 IP 범위를 쉽게 알아낼 수 있지만 1비트 단위로 서브네팅된 경우 유효한 네트워크 범위를 알아내기 어려움
    - 일반적으로 자신이 속한 네트워크의 유효 범위를 파악하는 방법은 다음과 같음
      1. 내 IP를 2진수로 표현
      2. 서브넷 마스크를 2진수로 표현
      3. 2진수 AND 연산으로 서브네팅된 네트워크 Address를 알아냄
      4. 호스트 Address 부분을 2진수 1로 모두 변경해 Broadcast Address를 알아냄
      5. 유효 IP 범위를 파악하는데 서브네팅된 네트워크 Address +1은 유효 IP 중 가장 작은 IP
      6. BroadcastAddress -1은 유효 IP 중 가장 큰 IP
      7. 2진수로 연산되어 있는 결괏값을 10진수로 변환

IP Address

- ❖ 클래스리스
  - ✓ 네트워크 사용자의 서브네팅
    - ● 서브네팅 방법



IP Address

- 클래스리스
  - ✓ 네트워크 설계자의 서브네팅
    - 네트워크를 새로 구축하는 경우 네트워크 사용자와 반대로 설계자는 서브넷 마스크가 지정되어 주어지는 것이 아니라 네트워크의 크기를 고민해 서브넷 마스크를 결정하고 설계에 반영
    - 네트워크 설계자가 IP를 설계할 때 고민해야 할 부분
      - 서브넷된 하나의 네트워크에 IP를 몇 개나 할당해야 하는가?(PC는 몇 대나 있는가)
      - 서브넷된 네트워크가 몇 개나 필요한가?

IP Address

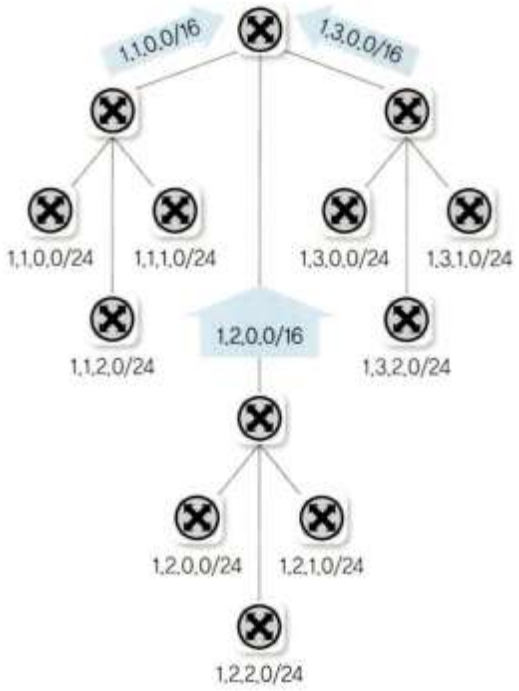
- ❖ 클래스리스
  - ✓ 네트워크 설계자의 서브네팅
    - ● 회사에 총 12곳의 지사가 있고 이 지사들에는 최대 12대의 IP가 필요한 PC와 복합기, IP 카메라가 운영될 예정이고 현재 가진 네트워크는 103.9.32.0/24 네트워크 인 경우
      1. 서브넷된 하나의 네트워크에 12개 IP를 할당해야 함
      2. 네트워크는 2진수의 배수로 커지므로 4, 8. 16, 32, 64. 128, 256개 단위로 네트워크를 할당할 수 있음
         - ◆ 12개 IP를 수용할 수 있는 가장 작은 네트워크는 16개이므로 16개짜리 네트워크를 할당
         - ◆ 16개짜리 네트워크는 네트워크 Address와 Broadcast Address로 사용할 2개 IP를 제외해야 하므로 실제로 사용할 수 있는 IP는 14개이므로 이 유효 IP 개수는 필요한 12개에 포함되므로 사용 가능
      3. 16개짜리 네트워크 12개를 확보하는데 16의 배수를 0부터 나열해 네트워크 Address를 확인

IP Address

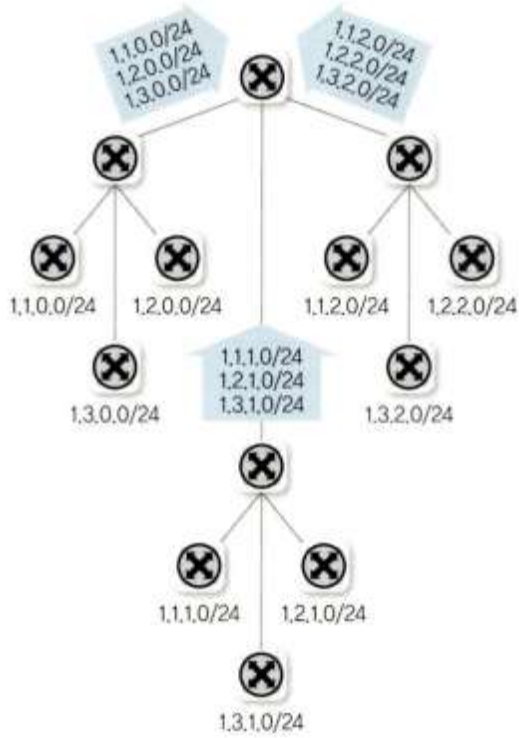
- 클래스리스
  - ✓ 네트워크 설계자의 서브네팅
    - 네트워크를 설계할 때 가능하면 Private IP 대역을 사용해 충분한 IP 대역을 사용하는 것이 좋음
    - Public IP는 인터넷에서 유일하게 사용되므로 사용할 수 있는 IP 수가 제한되어 있고 할당받은 IP를 사용하지 않는 경우 IP 할당 기관이 회수함
    - Private IP는 회사 내부에서만 사용하므로 제한없이 큰 네트워크를 사용할 수 있음
    - Private IP를 사용해 여유없이 네트워크를 할당하면 크기가 다른 네트워크가 많아지는데 네트워크 관리자 입장에서도 관리가 힘들어지고 일반 사용자도 IP를 쉽게 구분하거나 알아볼 수 없게 되므로 최대한 같은 크기의 네트워크를 할당하고 10진수로 표현해도 쉽게 이해할 수 있는 C 클래스 단위인 24비트로 쪼개 할당하는 것이 바람직
    - 네트워크를 단계적으로 잘 설계하면 관리하기 쉽고 네트워크 장비 성능도 향상되는데 잘 설계된 네트워크는 라우터가 관리하는 경로가 적고 관리하기 쉬움
    - 잘못 설계된 네트워크에서는 실제로 존재하는 네트워크 수만큼 Core 라우터가 모든 경로를 알고 있어야 함

IP Address

- 클래스리스
  - ✓ 네트워크 설계자의 서브네팅



잘 설계된 네트워크
(네트워크 전파가 단순)



잘못 설계된 네트워크
(네트워크 전파가 복잡)

IP Address

- Public IP 와 Private IP
  - ✓ 인터넷에 접속하려면 IP Address가 있어야 하고 이 IP Address는 전 세계에서 유일해야 하는 식별자인데 이러한 IP Address가 Public IP
  - ✓ 인터넷에 연결하지 않고 개인적으로 네트워크를 구성한다면 Public IP Address를 할당받지 않고도 네트워크를 구축할 수 있는데 이 때 사용하는 IP Address가 Private IP
  - ✓ 인터넷에 접속하려면 통신사업자로부터 IP Address를 할당받거나 IP 할당기관(한국의 경우는 KISA)에서 인터넷 독립기관 Address(Autonomous System Number – ASN)를 할당받은 후 독립 IP를 할당받아야 하므로 절차가 복잡
  - ✓ 인터넷에 접속하지 않거나 NAT(Network Address Translation - 네트워크 Address 변환) 기술을 사용할 경우(공유기나 회사 방화벽을 사용하는 경우)에는 Private IP Address를 사용할 수 있는데 이 Address들은 인터넷 표준 문서인 RFC에 명시되어 있음습니다
  - ✓ Private IP를 사용하면 인터넷에 직접 접속하지 못하지만 IP를 변환해주는 NAT 장비에 Public IP로 변경한 후에는 인터넷 접속이 가능
  - ✓ 클래스 별 Private IP

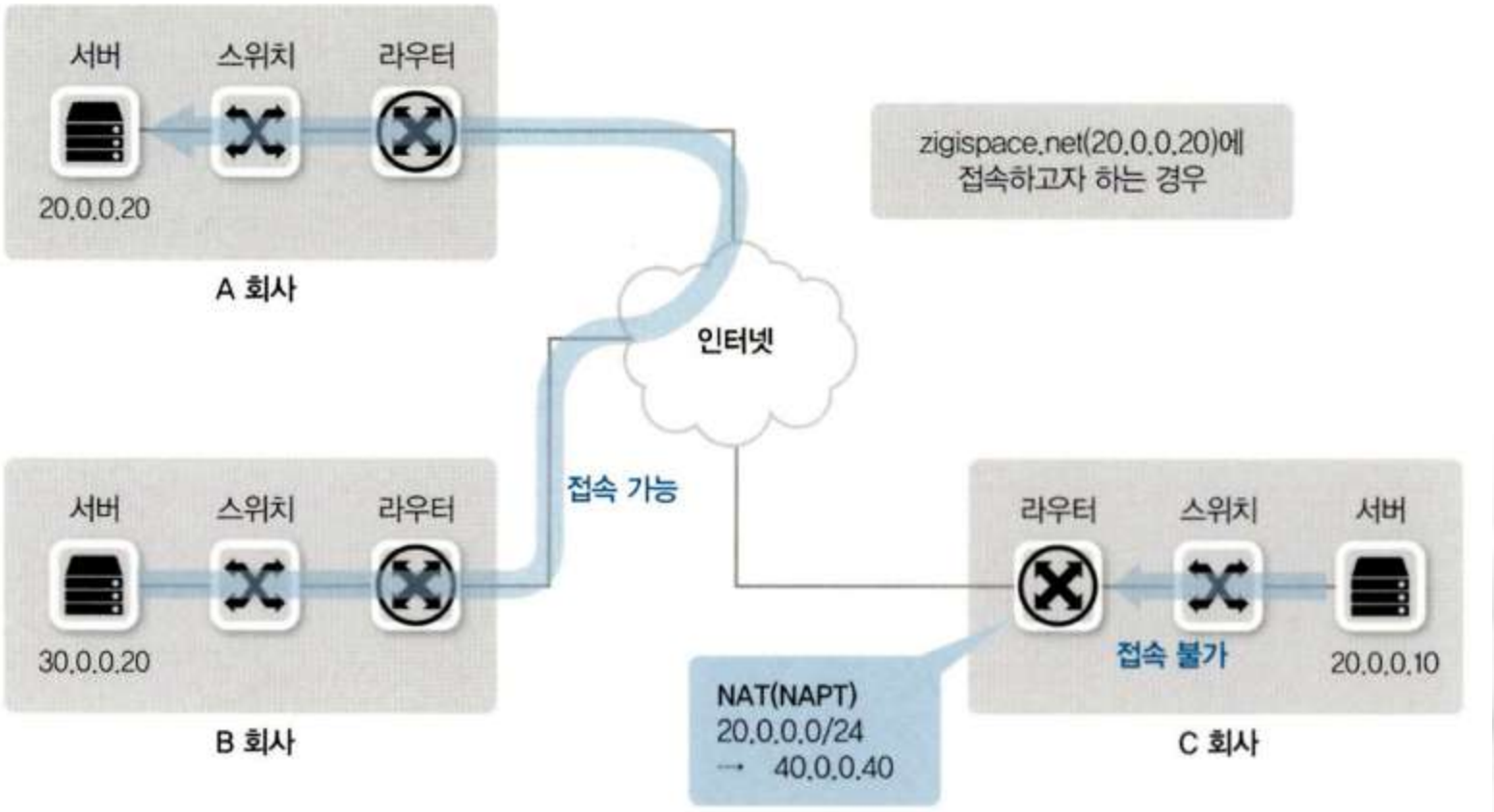
네트워크 주소	IP 범위	클래스 크기
10.0.0.0/8	10.0.0.0~10.255.255.255	A 클래스 1개
172.16.0.0/12	172.16.0.0~172.31.255.255	B 클래스 16개
192.168.0.0/16	192.168.0.0~192.168.255.255	C 클래스 256개

IP Address

- Public IP 와 Private IP
  - ✓ 회사 내부에서 사설 네트워크를 구축할 때 NAT를 사용하여 인터넷에 연결하더라도 다른 사용자에게 할당된 IP를 사설 네트워크 Address로 사용하면 안되는데 인터넷 연결에는 문제가 없지만 내부 네트워크에 할당된 IP를 공식으로 사용하는 네트워크로 접속할 수 없으므로 인터넷 어느 구간에서도 사용되지 않는 RFC에 명시된 사설 IP 대역을 사용할 것을 추천
  - ✓ C 회사처럼 20.0.0.0/24 네트워크로 회사 내부의 사설 IP를 할당한 경우 NAT를 거쳐 공인 IP로 아무리 변환되더라도 실제 인터넷 구간에 존재하는 A 회사의 20.0.0.0/24 네트워크로는 통신이 불가능
  - ✓ C 회사의 20.0.0.10 서버에서 인터넷에서 서비스하는 A 회사의 20.0.0.20 서버로 접근을 시도할 경우 C 회사의 서버는 A 회사의 서버가 같은 네트워크에 존재하는 것으로 인식하고 A 회사 서버인 20.0.0.20과 통신하기 위해 브로드캐스팅 함
  - ✓ 실제로 A 회사와 C 회사는 인터넷 구간을 거쳐 통신해야 하므로 원격지와 통신 시도를 서로 인식해야만 정상적인 통신이 가능하지만 현재 네트워크 구조에서는 같은 네트워크로 인식하므로 정상적인 통신이 불가능

IP Address

❖ Public IP 와 Private IP



IP Address

- Public IP 와 Private IP
  - ✓ Private IP는 A 클래스 1개 B 클래스 16개 C 클래스 256개를 사용할 수 있음
  - ✓ 규모가 큰 엔터프라이즈 네트워크에서는 대부분 A 클래스 크기인 10.0.0.0/8 네트워크를 사용하고 규모가 작은 네트워크를 위해서는 C 클래스 192.168.X.0/24을 사용
  - ✓ 공유기에서 가장 많이 사용되는 기본 IP가 192.168.0.1 인 이유
  - ✓ 이외에도 아이폰, 갤럭시와 같은 모바일 디바이스에서는 10.x.x.x 인 A 클래스 와 192.168.x.0/24 인 C 클래스와 겹치지 않도록 B 클래스인 172.x.x.x 네트워크를 이용해 테더링 기능을 제공

IP Address

- Public IP 와 Private IP
  - ✓ Bogon IP
    - 모든 IP가 인터넷에서 사용되는 것은 아님
    - IP Address를 할당하는 최상위 기구인 IANA가 여러가지 목적으로 예약해 놓아 Public IP로 할당하지 않는 Address를 Bogon IP라고 함
    - Bogon IP는 인터넷에서 동작하는 라우터에 해당 IP 정보가 존재하지 않음
    - Bogon IP 대역의 Address를 사용한 통신 시도가 있었다면 해킹을 목적으로 IP Spoofing(IP Address 변조)했거나 실수로 할당된 IP를 사용한 경우이므로 적절히 필터링하는 것이 좋음
    - 새로 할당되는 경우도 있으니 Bogon IP Address 대역이 변경되는 것을 확인해야 함

IP Address

- Public IP 와 Private IP
  - ✓ Bogon IP

네트워크 주소	설명
0.0.0.0/8	"This" 네트워크
10.0.0.0/8	사설 네트워크
100.64.0.0/10	캐리어 그레이드 NAT(통신사업자에서 사설 IP 주소를 할당하기 위해 사용)
127.0.0.0/8	루프백(Loopback) 주소
127.0.53.53	네임 콜리전 발생
169.254.0.0/16	링크 로컬(Link Local)
172.16.0.0/12	사설 네트워크
192.0.0.0/24	IETF 프로토콜 할당
192.0.2.0/24	테스트용
192.168.0.0/16	사설 네트워크
198.18.0.0/15	네트워크 인터커넥트 디바이드 벤치마크 테스팅
198.51.100.0/24	테스트용
203.0.113.0/24	테스트용
224.0.0.0/4	멀티캐스트용
240.0.0.0/4	예약
255.255.255.255/32	브로드캐스트

IP Address

- Public IP 와 Private IP
  - ✓ IP Address 발신지 확인 방법
    - IP Address를 보고 발신자의 신원이나 소속된 조직 이름을 알 수 있음
    - 후이즈(Whois)나 IP 검색 서비스를 제공하는 사이트가 많은데 한국에서는 한국인터넷진흥원(KISA)에서 이 서비스를 제공
    - 사이트에서 IP를 조회하면 해당 IP가 어떤 회선 사업자를 사용하는지 알 수 있거나 어떤 조직 소유의 IP인지 확인할 수 있음

TCP & UDP

- 4계층 프로토콜과 서비스 포트
  - ✓ 개요
    - 각 계층을 정의하는 정보는 수신 측의 동일 계층에서 사용하기 위한 정보인데 송신 측에서 추가한 2계층 헤더의 MAC 주소 정보는 수신 측의 2계층에서 확인하고 사용하고 송신 측에서 추가한 3계층 IP 주소는 수신 측 3계층에서 사용
    - 4계층에서는 이런 정보로 시퀀스 번호, ACK 번호가 있음
    - 상위 프로토콜 지시자는 디캡슐레이션 과정에서 상위 계층의 프로토콜이나 프로세스를 정확히 찾 아가기 위한 목적으로 사용됨
    - 2계층은 이더 타입 3계층은 프로토콜 번호 4계층은 포트 번호가 상위 프로토콜 지시자
    - TCP/IP 프로토콜 스택에서 4계층은 TCP와 UDP가 담당
    - 4계층의 목적은 목적지를 찾아가는 주소가 아니라 애플리케이션에서 사용하는 프로세스를 정확히 찾아가고 데이터를 분할한 패킷을 잘 쪼개 보내고 잘 조립하는 것
    - 패킷을 분할하고 조합하기 위해 TCP 프로토콜에서는 시퀀스 번호와 ACK 번호를 사용

TCP & UDP

- 4계층 프로토콜과 서비스 포트
  - ✓ 개요
    - 헤더 포맷

TCP 세그먼트 헤더 포맷

비트#	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Sequence Number							
64	Acknowledgement Number							
96	Data Offset	Res	Flags		Windows Size			
128	Header and Data Checksum				Urgent Pointer			
160...	Options							

UDP 데이터그램 헤더 포맷

비트#	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Length				Header and Data Checksum			

TCP & UDP

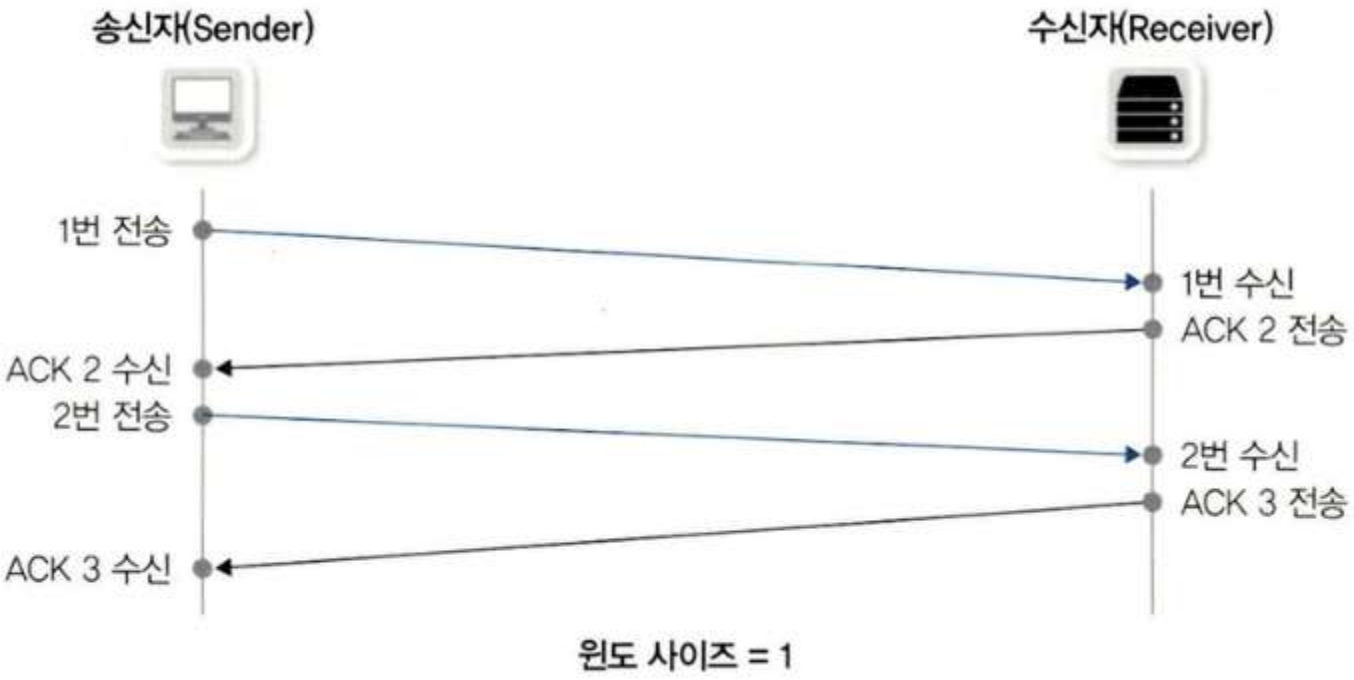
- 4계층 프로토콜과 서비스 포트
  - ✓ 개요
    - TCP/IP 프로토콜 스택에서 4계층의 상위 프로토콜 지시자는 포트 번호
    - 일반적으로 TCP/IP에서는 클라이언트 - 서버 방식으로 서비스를 제공하고 클라이언트용 프로그램과 서버용 프로그램을 구분해 개발
    - 3계층의 프로토콜 번호나 2계층의 이더 타입과 같은 상위 프로토콜 지시자는 출발지와 도착지를 구분해 사용하지 않고 한 개만 사용하지만 4계층 프로토콜 지시자인 포트 번호는 출발지와 목적지를 구분해 처리
    - 평소 우리가 표현하는 포트 번호의 기준은 서버의 포트
    - 이 포트 번호 중 HTTP TCP 80, HTTPS TCP 443, SMTP TCP 25 와 같이 잘 알려진 포트를 Well Known 포트라고 하는데 이 포트들은 이미 인터넷 주소 할당기구인 IANA(Internet Assigned Numbers Authority)에 등록되고 1023번 이하의 포트 번호를 사용
    - 다양한 애플리케이션에 포트 번호를 할당하기 위해 Registered Port 범위를 사용하는데 1024 ~ 49151의 범위이며 포트 번호를 할당받기 위해 신청하면 IANA에 등록되어 관리되지만 공식(Official) 번호와 비공식(Unofficial) 번호가 혼재되어 있고 사설 포트 번호로 사용되기도 함
    - 동적 사설 임시 포트의 범위는 49152~65535이며 이 범위의 포트 번호는 IANA에 등록되어 사용되지 않음
    - 이 포트 번호는 자동 할당되거나 사설 용도로 할당되고 클라이언트의 임시 포트 번호로 사용

TCP & UDP

- 4계층 프로토콜과 서비스 포트
  - ✓ 서비스 요청 시와 응답 시에는 출발지 IP와 목적지 IP가 반대가 되듯이 출발지와 도착지 포트 번호도 요청 패킷과 응답 패킷이 반대가 됨
  - ✓ 서버 측에서 클라이언트 측의 요청에 대한 응답을 할 때는 출발지 포트가 서버의 포트, 도착지 포트가 클라이언트의 포트가 되어 전송됨
  - ✓ 이런 포트의 방향 변화는 문제를 해결할 때 서비스 흐름을 이해하는 데 매우 중요

TCP & UDP

- TCP
  - ✓ 패킷 순서, 응답 번호
    - ACK

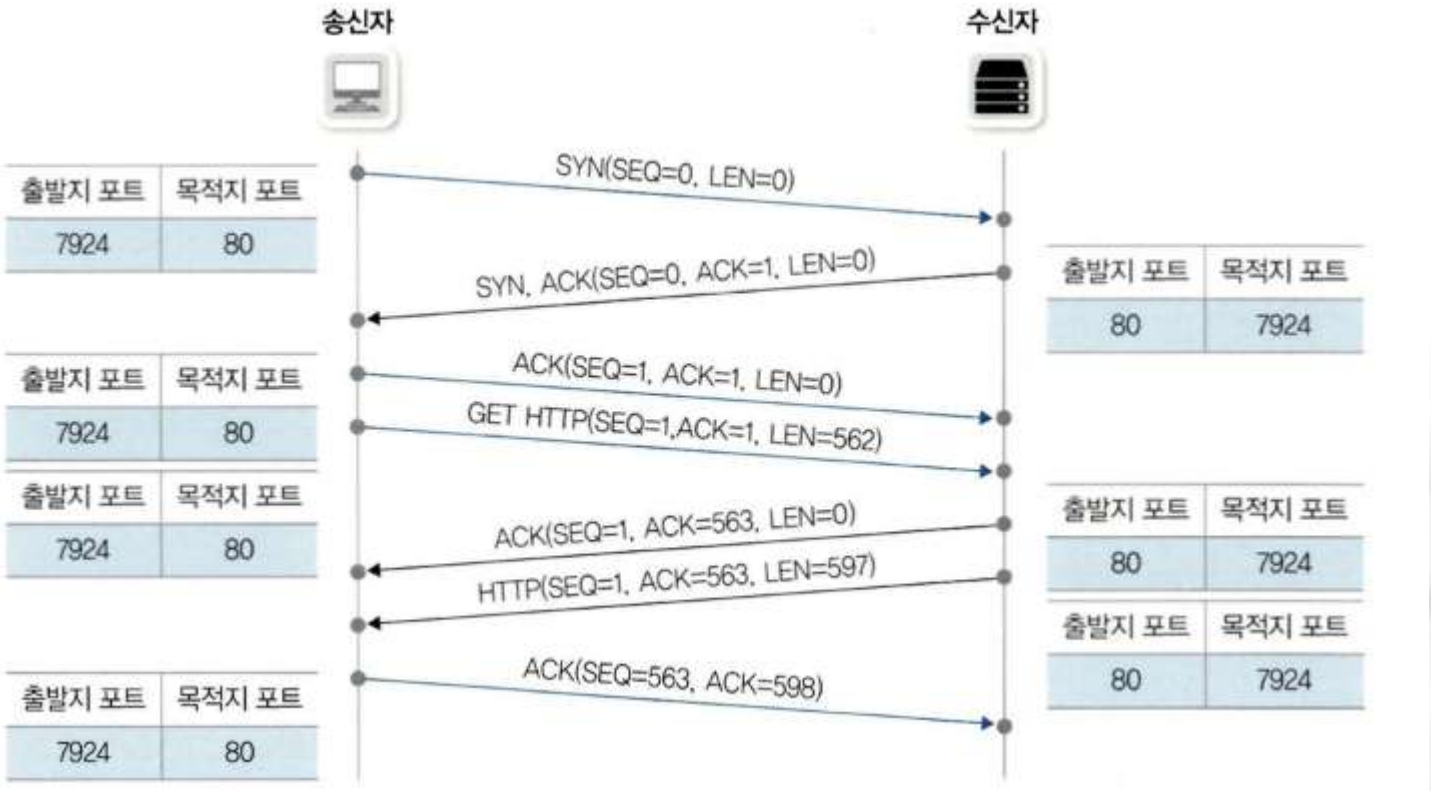


TCP & UDP

- TCP
  - ✓ 패킷 순서, 응답 번호
    - ACK
      - TCP에서는 분할된 패킷을 잘 분할하고 수신 측이 잘 조합하도록 패킷에 순서를 주고 응답 번호를 부여
      - 패킷에 순서를 부여하는 것이 시퀀스 번호이고 응답 번호를 부여하는 것이 ACK 번호
      - 두 번호가 상호작용해 순서가 바뀌거나 중간에 패킷이 손실된 것을 파악할 수 있음
      - 보내는 쪽에서 패킷에 번호를 부여하고 받는 쪽은 이 번호의 순서가 맞는지 확인
      - 받은 패킷 번호가 맞으면 응답을 주는데 이때 다음 번호의 패킷을 요청
      - 송신 측이 1번 패킷을 보냈는데 수신 측이 이 패킷을 잘 받는다면 이 숫자를 ACK 번호라고 부르고 1번을 잘 받았으니 다음에는 2번을 달라는 표시로 ACK 번호 2를 보냄

TCP & UDP

- TCP
  - ✓ 패킷 순서, 응답 번호
    - Sequence Number 와 ACK 동작



TCP & UDP

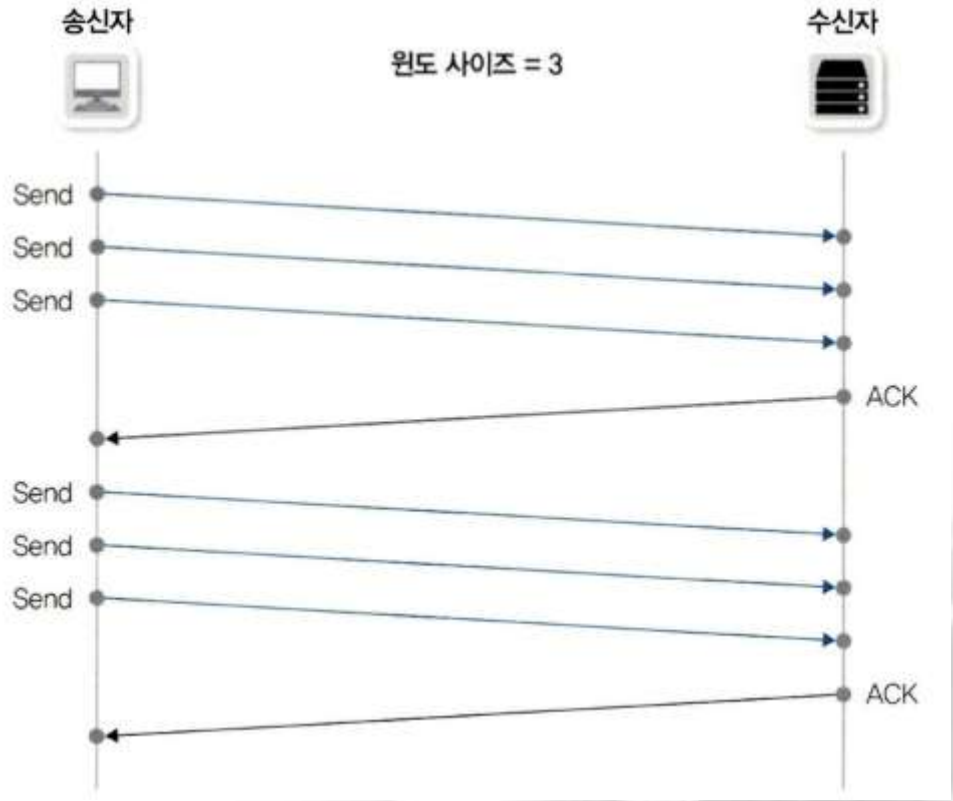
- TCP
  - ✓ 패킷 순서, 응답 번호
    - Sequence Number 와 ACK 동작
      - 출발지에서 시퀀스 번호를 0으로 보냄 (SEQ = 0).
      - 수신 측에서는 0번 패킷을 잘 받았다는 표시로 응답 번호(ACK)에 1을 적어 응답하고 수신 측에서는 자신이 처음 보내는 패킷이므로 자신의 패킷에 시퀀스 번호 0을 부여
      - 패킷을 받은 송신 측은 시퀀스 번호를 1로(수신 측이 ACK 번호로 1번 패킷을 달라고 요청) ACK 번호는 상대방의 0번 시퀀스를 잘 받았다는 의미로 1로 부여해 다시 송신

TCP & UDP

- TCP
  - ✓ 윈도 사이즈와 슬라이딩 윈도
    - TCP는 일방적으로 패킷을 보내는 것이 아니라 상대방이 얼마나 잘 받았는지 확인하기 위해 ACK 번호를 확인하고 다음 패킷을 전송
    - 패킷이 잘 전송되었는지 확인하기 위해 별도 패킷을 받는 것 자체가 통신 시간을 늘리지만 송신자와 수신자가 먼 거리에 떨어져 있으면 왕복 지연시간(Round Trip Time. RTT)이 늘어나므로 응답을 기다리는 시간이 더 길어짐
    - 작은 패킷을 하나 보내고 응답을 받아야만 하나를 더 보낼 수 있다면 모든 데이터를 전송하는 데 긴 시간이 걸릴 것이므로 데이터를 보낼 때 패킷을 하나만 보내는 것이 아니라 많은 패킷을 한꺼번에 보내고 응답을 하나만 받음
    - 가능하면 최대한 많은 패킷을 한꺼번에 보내는 것이 효율적이지만 네트워크 상태가 안 좋으면 패킷 유실 가능성이 커지므로 적절한 송신량을 결정해야 하는데 한 번에 데이터를 받을 수 있는 데이터 크기를 윈도 사이즈라고 하고 네트워크 상황에 따라 이 윈도 사이즈를 조절하는 것을 슬라이딩 윈도라고 함

TCP & UDP

- TCP
  - ✓ 윈도 사이즈와 슬라이딩 윈도

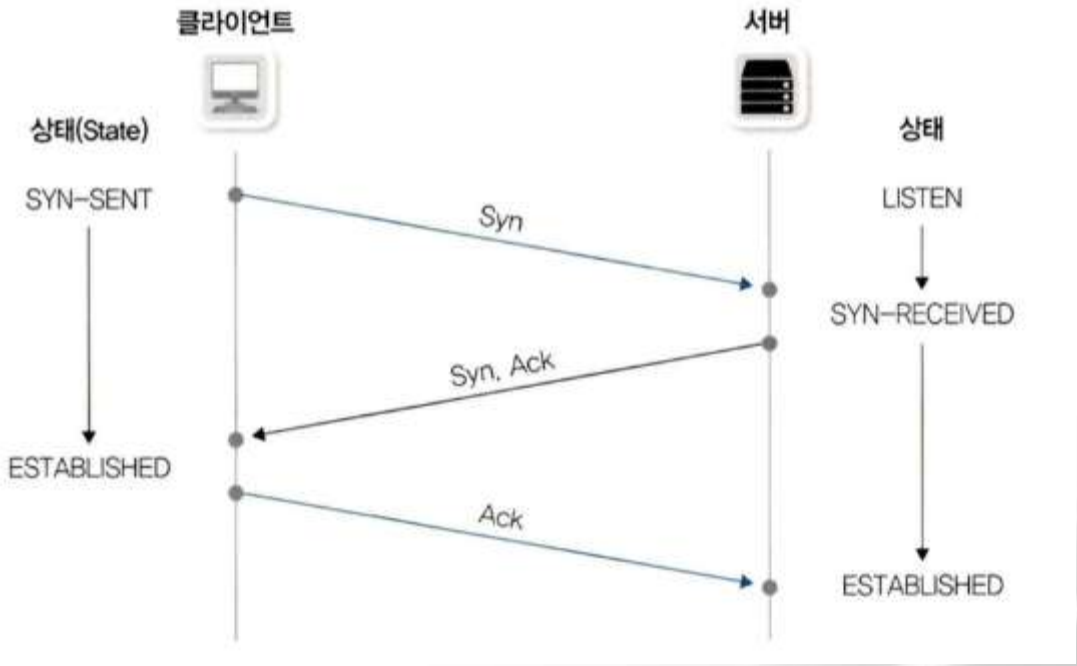


TCP & UDP

- TCP
  - ✓ 윈도 사이즈와 슬라이딩 윈도
    - TCP 헤더에서 윈도 사이즈로 표현할 수 있는 최대 크기는 2^{16}
    - 실제로 64K만큼 윈도 사이즈를 가질 수 있지만 이 사이즈는 회선의 안정성이 높아지고 고속화되는 현대 네트워크에서는 너무 작은 숫자인데 점점 고속화 및 안정화되는 환경에 적응하기 위해 윈도 사이즈를 64K보다 대폭 늘려 통신하는데 TCP 헤더는 변경이 불가능하므로 헤더 사이즈를 늘리지 않고 뒤의 숫자를 무시하는 방법으로 윈도 사이즈를 증가시켜 통신을 수행하는데 이런 방법을 사용하면 기존 숫자에 10배, 100배로 윈도 사이즈가 커짐
    - TCP는 데이터에 유실이 발생하면 윈도 사이즈를 절반으로 떨어뜨리고 정상적인 통신이 되는 경우 서서히 하나씩 늘림
    - 네트워크에 경합이 발생해 패킷 드롭이 생기면 작아진 윈도 사이즈로 인해 데이터 통신 속도가 느려져 회선을 제대로 사용하지 못하는 상황이 발생할 수 있는데 이런 경합을 피하기 위해 회선 속도를 증가시키거나 경합을 임시로 피하게 할 수 있는 버퍼가 큰 네트워크 장비를 사용하거나 TCP 최적화 솔루션을 사용해 이런 문제들을 해결할 수 있음

TCP & UDP

- TCP
  - ✓ 3방향 핸드셰이크

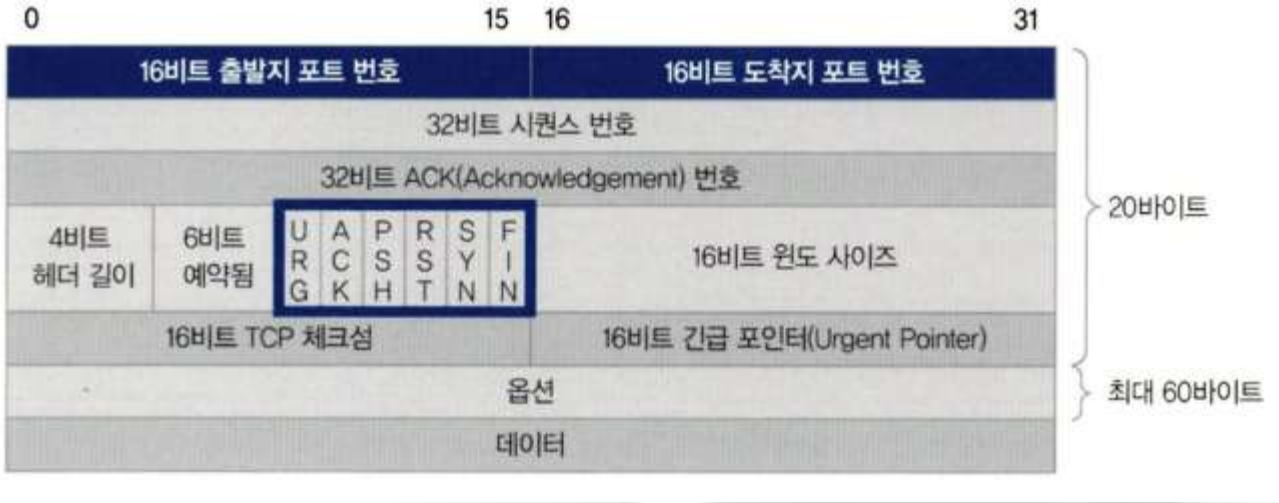


TCP & UDP

- TCP
  - ✓ 3방향 핸드셰이크
    - TCP에서는 유실없는 안전한 통신을 위해 통신 시작 전 사전 연결작업을 진행
    - 목적지가 데이터를 받을 준비가 안 된 상황에서 데이터를 일방적으로 전송하면 목적지에서는 데이터를 정상적으로 처리할 수 없어 데이터가 버려지는데 TCP 프로토콜은 이런 상황을 만들지 않기 위해 통신 전 데이터를 안전하게 보내고 받을 수 있는지 미리 확인하는 작업을 거침
    - 패킷 네트워크에서는 동시에 많은 상대방과 통신하므로 정확한 통신을 위해서는 통신 전 각 통신에 필요한 준비하므로 리소스를 미리 확보하는 작업이 중요
    - TCP에서는 3번의 패킷을 주고받으면서 통신을 서로 준비하므로 3방향 핸드셰이크하고 부름
    - TCP는 이런 3방향 핸드쉐이크 진행 상황에 따라 상태(state) 정보를 부르는 이름이 다름
    - 서버에서는 서비스를 제공하기 위해 클라이언트의 접속을 받아들일 수 있는 LISTEN 상태로 대기하고 있다가 클라이언트에서 통신을 시도할 때 Syn 패킷을 보내는데 클라이언트에서는 이 상태를 SYN-SENT라고 부르며 클라이언트의 Syn을 받은 서버는 SYN-RECEIVE 상태로 변경되고 Syn, Ack로 응답하는데 이 응답을 받은 클라이언트는 ESTABLISHED 상태로 변경하고 그에 대한 응답을 서버로 다시 보내면 서버에서도 클라이언트의 이 응답을 받고 ESTABLISHED 상태로 변경되는데 ESTABLISHED 상태는 서버와 클라이언트 간의 연결이 성공적으로 완료됨

TCP & UDP

- TCP
  - ✓ 3방향 핸드셰이크
    - TCP 플래그
      - ◆ 3방향 핸드셰이크 과정이 생기다보니 기존 통신과 새로운 통신을 구분해야 하는데 어떤 패킷이 새로운 연결 시도이고 기존 통신에 대한 응답인지 구분하기 위해 헤더에 플래그(Flag)라는 값을 넣어 통신을 수행
      - ◆ TCP 헤더 내용



TCP & UDP

- ❖ TCP
  - ✓ 3방향 핸드셰이크
    - ● TCP 플래그
      - ◆ TCP 플래그 내용
        - ▪ SYN: 연결 시작 용도로 사용되는데 연결이 시작될 때 SYN 플래그에 1로 표시
        - ▪ ACK: ACK 번호가 유효할 경우 1로 표시해 보내고 초기 SYN이 아닌 모든 패킷은 기존 메시지에 대한 응답이므로 ACK 플래그가 1로 표기
        - ▪ FIN: 연결 종료 시 1로 표시되는데 데이터 전송을 마친 후 정상적으로 양방향 종료 시 사용
        - ▪ RST: 연결 종료 시 1로 표시되는데 연결 강제 종료를 위해 연결을 일방적으로 끊을 때 사용
        - ▪ URG: 긴급 데이터인 경우 1로 표시
        - ▪ PSH: 서버 측에서 전송할 데이터가 없거나 데이터를 버퍼링 없이 응용 프로그램으로 즉시 전달할 것을 지시할 때 사용

TCP & UDP

- UDP
  - ✓ UDP는 4계층 프로토콜이 가져야 할 특징이 거의 없음
  - ✓ TCP에서는 신뢰성 있는 통신을 위해 연결을 미리 확립(3방향 핸드셰이크)했고 데이터를 잘 분할하고 조립하기 위해 패킷 번호를 부여하고 수신된 데이터에 대해 응답하는 작업을 수행하는데 데이터를 특정 단위(윈도 사이즈)로 보내고 메모리에 유지하다가 ACK 번호를 받은 후 통신이 잘 된 상황을 파악하고 나서야 메모리에서 이 데이터들을 제거하는 방식으로 중간에 유실이 있으면 시퀀스 번호와 ACK 번호를 비교해가며 이를 파악하고 메모리에 유지해 놓은 데이터를 이용해 재전송하는 기능이 있는데 이 기능을 이용해 데이터 유실이 발생하거나 순서가 바뀌더라도 바로잡을 수 있지만 UDP에는 이런 기능이 전혀 없음
  - ✓ UDP의 헤더에는 신뢰 통신을 위한 내용(시퀀스 번호, ACK 번호, 플래그, 윈도 사이즈)이 없음

TCP & UDP

- UDP
  - ✓ 데이터 통신은 데이터 전송의 신뢰성이 핵심
  - ✓ 애플리케이션에서 걱정하지 않고 데이터를 만들고 사용하게 하는 것이 데이터 통신의 목적이지만 UDP는 데이터 전송을 보장하지 않는 프로
  - ✓ 토콜이므로 제한된 용도로만 사용
  - ✓ UDP는 음성 데이터나 실시간 스트리밍과 같이 시간에 민감한 프로토콜이나 애플리케이션을 사용하는 경우나 사내 방송이나 증권 시세 데이터 전송에 사용되는 멀티캐스트처럼 단방향으로 다수의 단말과 통신해 응답을 받기 어려운 환경에서 주로 사용
  - ✓ 음성 및 동영상은 디지털 환경에서는 연속된 데이터가 아닌 매우 짧은 시간 단위로 잘게 분할한 데이터가 전송되는 형태
  - ✓ 동영상은 1초 동안 30~60회의 정지 영상이 빠르게 바뀌면서 우리 눈에는 마치 움직이는 영상처럼 보이는 원리를 이용하며 반대로 실 세계 데이터를 디지털화 할 때는 시간을 잘게 쪼개 데이터를 샘플링하는데 이런 데이터들은 다른 일반 데이터처럼 취급하면 시간 지연에 따른 어려움이 있음
  - ✓ 중간에 데이터가 몇 개쯤 유실되는 것보다 재전송하기 위해 잠시 동영상이나 음성이 멈추는 것을 더 이상하게 느낄 수 있으므로 데이터를 전송하는 데 신뢰성보다 일부 데이터가 유실되더라도 시간에 맞추어 계속 전송하는 것이 중요한 화상회의 시스템과 같은 서비스인 경우 UDP를 사용

TCP & UDP

- UDP
  - ✓ UDP는 중간에 데이터가 일부 유실되더라도 그냥 유실된 상대로 데이터를 처리
  - ✓ 30 프레임 크기의 동영상에서 1 프레임이 잘린 29프레임만으로도 사람들의 눈에는 이질감이 별로 없지만 실시간 동영상을 전송하는 경우 잘린 1개 프레임을 재전송하기 위해 일시적으로 화면이 멈추면 음성과 영상이 뒤늦게 전달되어 사용자는 네트워크 품질이 떨어진다고 생각하게 됨
  - ✓ UDP는 TCP와 달리 통신 시작 전 3방향 핸드셰이크와 같이 사전에 연결을 확립하는 절차가 없는데 그 대신 UDP에서 첫 데이터는 리소스 확보를 위해 Interrupt를 거는 용도로 사용되고 유실되는데 UDP 프로토콜을 사용하는 애플리케이션이 대부분 이런 상황을 인지하고 동작하거나 연결 확립은 TCP 프로토콜을 사용하고 애플리케이션끼리 모든 준비를 마친 후 실제 데이터만 UDP를 이용하는 경우가 대부분
  - ✓ 동영상 전송의 다른 방식
    - 같은 동영상 스트리밍이더라도 넷플릭스나 유튜브와 같이 시간에 민감하지 않은 단일 시청자를 위한 연결은 TCP를 사용
    - 원활한 시청을 위해 수 초~수 분의 동영상 데이터를 미리 받아놓고 네트워크에 잠시 문제가 발생하더라도 동영상이 끊기지 않도록 캐시에 저장

TCP & UDP

❖ TCP & UDP

TCP	UDP
연결 지향(Connection Oriented)	비연결형(Connectionless)
오류 제어 수행	오류 제어 수행 안 함
흐름 제어 수행	흐름 제어 수행 안 함
유니캐스트	유니캐스트, 멀티캐스트, 브로드캐스트
Full Duplex	Half Duplex
데이터 전송	실시간 트래픽 전송

ARP

❖ 개요

- ✓ OSI 7계층 중 2, 3 계층이 주소를 가지고 있고 통신할 때 목적지를 찾아갈 수 있도록 하지만 사실 2계층 MAC 주소와 3계층 IP 주소 간에는 아무 관계도 없음
- ✓ MAC 주소는 하드웨어 생산업체가 임의적으로 할당한 주소이고 NIC에 종속된 주소
- ✓ 3계층 IP 주소는 우리가 직접 할당하거나 DHCP를 이용해 자동으로 할당
- ✓ 실제로 통신은 IP 주소 기반으로 일어나고 MAC 주소는 상대방의 주소를 자동으로 알아내 통신하는데 상대방의 MAC 주소를 알아내기 위해 사용되는 프로토콜이 ARP(Address Resolution Protocol)
- ✓ 데이터 통신을 위해 2계층 물리적 주소인 MAC 주소와 3계층 논리적 IP 주소 두 개가 사용되는데 IP 주소 체계는 물리적 MAC 주소와 전혀 연관성이 없으므로 두 개의 주소를 연계시켜 주기 위한 메커니즘이 필요한데 이때 사용되는 프토토콜이 ARP
- ✓ ARP 프로토콜 필드

<table>
<tr><td colspan="2">하드웨어 타입(2계층)</td><td>프로토콜 타입(3계층)</td></tr>
<tr><td>하드웨어 주소 길이
(2계층)</td><td>프로토콜 주소 길이
(3계층)</td><td>오퍼레이션 코드</td></tr>
<tr><td colspan="3">송신자 하드웨어 주소(2계층) – MAC 주소</td></tr>
<tr><td colspan="3">송신자 프로토콜 주소(3계층) – IP 주소</td></tr>
<tr><td colspan="3">대상자 하드웨어 주소(2계층)</td></tr>
<tr><td colspan="3">대상자 프로토콜 주소(3계층) – IP 주소</td></tr>
</table>

ARP

- 개요
  - ✓ ARP 프로토콜은 TCP/IP 프로토콜 스택을 위해서만 동작하는 것은 아님
  - ✓ 호스트에서 아무 통신이 없다가 처음 통신을 시도하면 패킷을 바로 캡슐화 할 수 없음
  - ✓ 통신을 시도할 때 출발지와 목적지 IP 주소는 미리 알고 있어 캡슐화하는 데 문제가 없지만 상대방의 MAC 주소를 알 수 없어 2계층 캡슐화를 수행할 수 없음
  - ✓ 상대방의 MAC 주소를 알아내려면 ARP 브로드캐스트를 이용해 네트워크 전체에 상대방의 MAC 주소를 질의해야 함
  - ✓ ARP 브로드캐스트를 받은 목적지는 ARP 프로토콜을 이용해 자신의 MAC 주소를 응답하고 이 작업이 완료되면 출발지 와 목적지 둘 다 상대방에 대한 MAC 주소를 학습하고 이후 패킷이 정상적으로 인캡슐레이션되어 상대방에게 전달될 수 있음
  - ✓ 패킷 네트워크에서는 큰 데이터를 잘라 전송하므로 여러 개의 패킷을 전송해야 하는데 패킷을 보낼 때마다 ARP 브로드캐스트를 수행하면 네트워크 통신의 효율성이 크게 저하되므로 메모리에 이 정보를 테이블 형태로 저장해두고 재사용
  - ✓ arp -a 명령을 입력하면 PC의 ARP 테이블 정보를 확인할 수 있음
  - ✓ 성능 유지를 위해서는 ARP 테이블을 오래 유지하는 것이 좋지만 논리 주소는 언제든지 바뀔 수 있으므로 일정 시간 동안 통신이 없으면 이 테이블은 삭제
  - ✓ 패킷으로 데이터를 전송하는 네트워크에서는 데이터를 보내기 위해 여러 개의 패킷을 사용해야 하므로 반복 작업을 줄이기 위해 참조할 수 있는 여러 가지 테이블을 사용하는데 DNS Cache 나 Routing Cache 같은 테이블 등을 사용

ARP

- 개요
  - ✓ 네트워크 장비에서의 ARP 작업은 하드웨어 가속으로 처리되지 않고 CPU에서 직접 수행하므로 짧은 시간에 많은 ARP 요청이 들어오면 네트워크 장비에서는 큰 부하로 작용
  - ✓ 그동안 해커들이 네트워크 장비를 무력화하는 데 다량의 ARP를 이용한 공격을 많이 수행했는데 이런 공격에 대응하기 위해 네트워크 장비는 ARP 테이블 저장 기간을 일반 PC보다 길게 설정하고 많은 ARP 요청이 들어오면 이를 필터링하거나 천천히 처리하는 방식으로 네트워크 장비를 보호
  - ✓ 이외에도 일부 장비는 ARP 테이블을 수동으로만 갱신하도록 설정해 운영하기도 함(특히 회선 사업자가 임대하는 네트워크 장비는 MAC 테이블이나 ARP 테이블을 정적으로 유지하기 때문에 외부 회선에 연결된 네트워크 장비가 변경된 경우 이 정보를 회선 사업자에게 알려주어야 함)

ARP

- 구성 필드
  - ✓ ARP 패킷은 여러 가지 필드 중 ARP 데이터에 사용되는 송신자 하드웨어 MAC 주소, 송신자 IP 프로토콜 주소, 대상자 MAC 주소, 대상자 IP 프로토콜 주소 4개의 필드가 중요하게 사용
  - ✓ 패킷

하드웨어 타입(2계층)		프로토콜 타입(3계층)
하드웨어 주소 길이 (2계층)	프로토콜 주소 길이 (3계층)	오퍼레이션 코드
송신자 하드웨어 주소(2계층) – MAC 주소		
송신자 프로토콜 주소(3계층) – IP 주소		
대상자 하드웨어 주소(2계층)		
대상자 프로토콜 주소(3계층) – IP 주소		

ARP

- GARP
  - ✓ 일반적인 ARP 외에도 ARP 프로토콜 필드를 그대로 사용하지만 내용을 변경해 원래 ARP 프로토콜의 목적과 다른 용도로 사용하는 GARP. RARP와 같은 프로토콜이 있음
  - ✓ Gratuitous ARP의 약자인 GARP는 대상자 IP 필드에 자신의 IP 주소를 채워 ARP 요청을 보냄
  - ✓ ARP가 상대방의 MAC 주소를 알아내기 위해 사용되는 반면 GARP는 자신의 IP와 MAC 주소를 알릴 목적으로 사용
  - ✓ GARP는 로컬 네트워크에 자신의 IP와 MAC 주소를 알릴 목적으로 사용되므로 GARP의 목적지 MAC 주소(2계층 Destination MAC)는 브로드캐스트 MAC 주소를 사용
  - ✓ GARP의 패킷을 살펴보면 송신자 MAC(Sendcr MAC)은 자신의 MAC 주소,
  - ✓ 송신자 IP(Sender IP) 주소는 자신의 IP 주소, 대상자 MAC(Target MAC) 주소는 모두 0으로 표기해 대상자 IP(Target IP) 주소도 자신의 IP 주소로 넣어 네트워크에 브로드캐스팅
  - ✓ 용도
    - IP 주소충돌감지
    - 상대방(동일 서브넷 상의 다른)의 ARP 테이블 갱신
    - HA(고가용성) 용도의 클러스터링

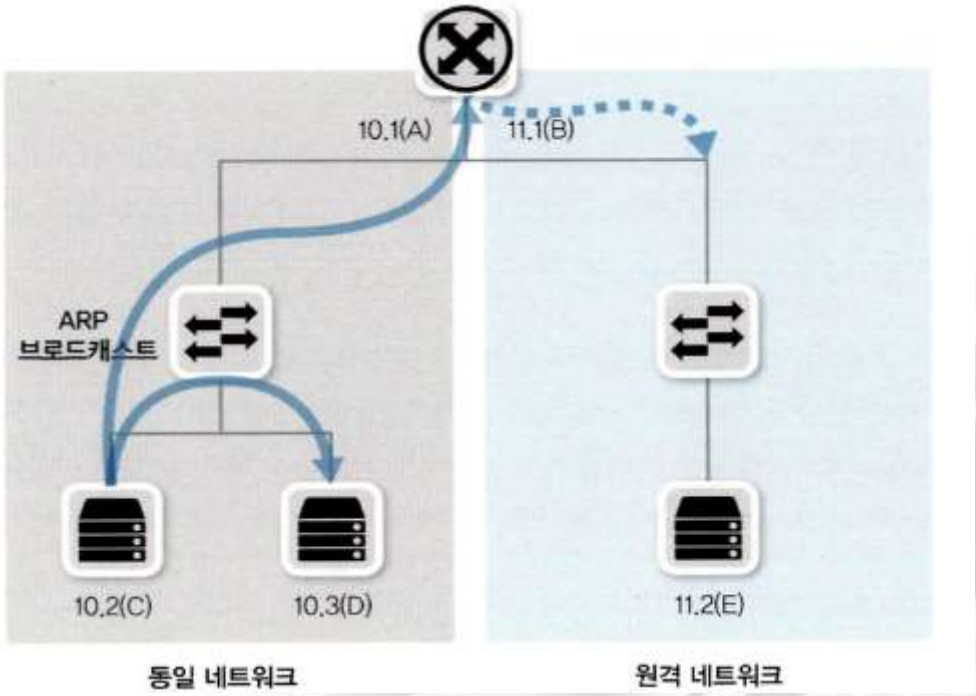
ARP

- RARP
  - ✓ RARP는 Reverse ARP의 줄임말로 반대로 동작하는 ARP인데 GARP처럼 ARP 프로토콜 구조는 같지만 필드에 들어가는 내용이 다르고 원래 목적과 반대로 사용

  - ✓ RARP는 IP 주소가 정해져 있지 않은 단말이 IP 할당을 요청할 때 사용
  - ✓ ARP는 내가 통신해야 할 상대방의 MAC 주소를 모를 때 상대방의 IP 주소로 MAC 주소를 물어볼 목적으로 만들어진 프로토콜
  - ✓ RARP는 반대로 나 자신의 MAC 주소는 알지만 IP가 아직 할당되지 않아 IP를 할당해주는 서버에 어떤 IP 주소를 써야 하는지 물어볼 때 사용
  - ✓ RARP는 과거에 네트워크 호스트의 주소 할당에 사용되었지만 제한된 기능으로 인해 BOOTP와 DHCP로 대체되어 사용되지 않음

Subnet & Gateway

- 게이트웨이
  - ✓ 로컬 네트워크에서는 ARP 브로드캐스트를 이용해 도착지 MAC 주소를 학습할 수 있고 이 MAC 주소를 이용해 직접 통신할 수 있지만 원격 네트워크 통신은 네트워크를 넘어 전달되지 못하는 브로드캐스트의 성질 때문에 네트워크 장비의 도움이 필요한데 이 장비를 게이트웨이(Gatway)라고 하고 게이트웨이에 대한 정보를 PC나 네트워크 장비에 설정하는 항목이 기본 게이트웨이(Default Gateway)



Subnet & Gateway

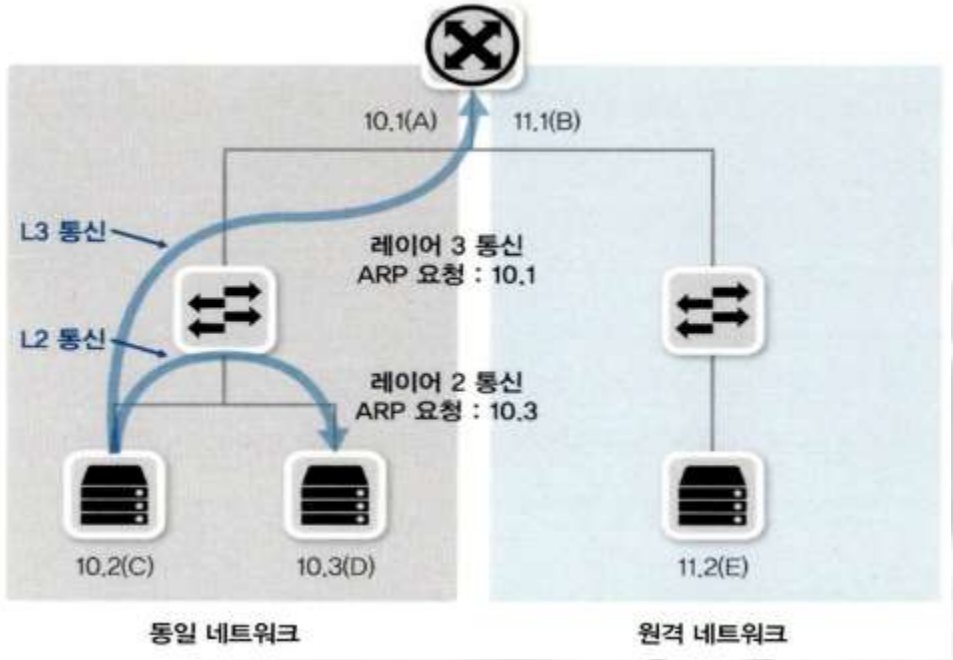
- 프록시 ARP
  - ✓ 프록시 ARP는 ARP를 대행해주는 기능
  - ✓ 원격지 통신은 기본 게이트웨이를 찾아 ARP 요청을 보내고 패킷을 기본 게이트웨이 쪽으로 보내야만 통신할 수 있지만 기본 게이트웨이에 프록시 ARP가 활성화된 경우 원격지 통신이더라도 로컬에 ARP 브로드캐스트를 보내 통신할 수 있음
  - ✓ 프록시 ARP가 활성화된 기본 게이트웨이는 ARP 브로드캐스트가 들어오면 자신이 대행해 ARP 응답을 해주는데 이 경우 기본 게이트웨이 쪽으로 보내지므로 원격지 경로로 전달될 수 있음
  - ✓ 프록시 ARP 기능은 라우터에 기본으로 활성화되어 있어 사용자 몰래 동작하는 경우가 많음
  - ✓ 프록시 ARP는 사용자 설정 없이 자동으로 동작해 편리한 것처럼 느껴지지만 네트워크에 설정 오류가 있거나 꼭 입력해야 할 설정이 되어 있지 않아도 동작하는 경우가 많아 장애가 발생했을 때 쉽게 해결할 수 없게 만드는 장애 요소가 되기도 함
  - ✓ 원격지 통신인 경우 프록시 ARP가 활성화된 라우터가 운영되고 있다면 엉뚱한 ARP 요청도 받아 처리해주므로 PC에서 기본 게이트웨이가 잘못 입력되어 있어도 통신이 되는 경우가 발생할 수 있음
  - ✓ 평소에는 프록시 ARP 기능으로 문제 없이 통신되다가 라우터를 교체하거나 라우터의 구성을 변경할 때 장애가 발생하는데 문제의 근본 원인은 PC에서 기본 게이트웨이를 잘못 설정한 것인데도 불구하고 라우터의 문제로 인식되어 근본적인 문제 해결에 방해가 됨

Subnet & Gateway

- 2계층 통신 & 3계층 통신
  - ✓ 2계층 통신 이나 3계층 통신은 원래 정확한 표현은 아니지만 실무에서 많이 쓰는 표현인데 정확한 표현은 로컬 네트워크 통신, 원격지 네트워크 통신
  - ✓ 단말 간의 통신은 애플리케이션 계층부터 시작해 캡슐화, 디캡슐화를 거쳐 통신하는데 로컬 네트워크에서 직접 통신할 경우(출발지와 목적지가 같은 네트워크에 존재할 경우) 라우터와 같은 3계층 네트워크 장비의 도움 없이 통신이 가능해서 단말 간을 연결해주는 네트워크 장비에서 2계층까지만 정보를 확인해 통신하고 ARP 요청을 보낼 때 직접 브로드캐스트를 이용하므로 이를 L2 통신이라 하고 원격지 네트워크와 통신해야 할 경우(출발지와 목적지가 다른 네트워크에 존재할 경우) 라우터와 같은 3계층 장비의 도움이 없으면 통신할 수 없는데 해당 패킷을 전송하는 네트워크 장비에서 3계층 정보까지 확인해야 하며 이것을 L3 통신이라고 함

Subnet & Gateway

- 2계층 통신 & 3계층 통신
  - ✓ 통신하는 출발지와 도착지의 네트워크가 같다 와 다르다라는 조건에 따라 통신 방식이 달라지는데 이런 차이는 로컬과 리모트 통신을 위한 ARP 동작 방식이 다른 것으로 인해 발생
  - ✓ 같은 네트워크에 있는 단말 간 통신은 직접적으로 이루어지는데 상대방의 MAC 주소를 알아내기위해 ARP 브로드캐스트를 이용하고 상대방의 MAC 주소를 알아내자마자 패킷이 캡슐화되어 통신이 시작되고 외부 네트워크와 통신이 필요할 때는 단말이 자신이 직접 보낼 수 없는 위치에 목적지가 있다고 판단하고 ARP 요청을 기본 게이트웨이의 IP 주소로 요청



Subnet & Gateway

- 2계층 통신 & 3계층 통신
  - ✓ 게이트웨이에서 ARP 응답을 받은 단말은 도착지 MAC 주소에 응답받은 기본 게이트웨이의 MAC 주소를 적어넣고 통신을 시작
  - ✓ 로컬 통신(L2 통신)은 도착지 MAC 주소와 도착지 IP 주소가 같은 반면 원격지 통신은 도착지 MAC 주소와 도착지 IP 주소가 다름
  - ✓ 도착지 IP 주소는 통신의 실제 도착지이고 도착지 MAC 주소는 디폴트 게이트웨이의 MAC 주소가 사용됨
  - ✓ 실제 패킷