네트워크 (Network)

1. 네트워크 소개

네트워크: 분산되어 있는 컴퓨터들이 자원을 공유할 수 있게 통신망으로 연결한 것

1) 네트워크 형태
- LAN(Local Area Network): 근거리 통신망. 사무실 등 가까운 지역을 묶은 네트워크
- WAN(Wide-): 장거리 통신망. 각각의 떨어진 LAN망을 ISP로 엮은 것
  ㄴ ISP(Internet Service Provider): 인터넷 서비스 제공자. SKT, KT, LG 등
- VPN(Virtual Private Network): 공중망을 사설망처럼 사용하는 기술, 암호화되어 있다.
  ㄴ 물리적으로 구현된 네트워크가 아닌 논리적인 구분

2) 네트워크 표준
- 네트워크 표준 기구: ISO, IEEE (주로 LAN), ITU-T (주로 WAN)
  ㄴ 후술할 OSI 7단계는 ISO 표준, 이더넷은 IEEE 표준에 따른 것
- 인터넷 표준 기구: IETF, RFC (프로토콜 정의 문서)
  ㄴ TCP/IP, HTTP 등이 RFC에 정의되어 있다.

3) 네트워크 구조
- 토폴로지: 네트워크의 다양한 형태를 나타낸다.
  ㄴ Star, Ring, Mesh, Bus, Tree, Redundancy형 토폴로지 등이 있다.
- 홈 네트워크: 인터넷 - ISP - 모뎀 - 공유기 - 컴퓨터
- 기업용 네트워크: 인터넷 - ISP(전용선) - 라우터 - 방화벽 - 컴퓨터 및 서버
  ㄴ 컴퓨터: [L3백본 - L2스위치(팀별) - 개인 PC]
  ㄴ 서버: [L4 로드밸런서(과부하 방지를 위한 서비스 분산용) - 서버 - DMZ]
- 클라우드 네트워크(AWS 기준): 인터넷 - Route53(URL -> IP로 변환) - IGW - VPC
  ㄴ VPC: ELB - SG(EC2), RDS 등. EC2는 동적으로 서버 크기를 조절한다는 특징이 있다.
- 무선랜: 인터넷 - ISP - 라우터 - WIPS - AP - 컴퓨터 (상세설명은 2-8 항목 참고)

4) OSI 7계층
- 네트워크 프로토콜과 통신을 7개의 계층(Layer)으로 표현한 것
 (1) Physical: 네트워크 하드웨어 전송 기술
 - 장치와 통신 매체 사이의 비정형 데이터 전송
 - 디지털 bit를 전기, 무선, 광 신호 등으로 변환
 - 전송 방법, 제어 신호, 기계적 속성 등을 정의하는 물리적 계층
   ㄴ 케이블, 인터페이스, 허브, 리피터 등이 여기에 속한다.
 (2) Data Link: 이더넷, 랜카드, 랜카드 간 통신(Mac 통신), 에러검출, 재전송
 - 동일 네트워크 내에서의 링크를 통한 연결, 데이터 전송, 물리계층의 오류 감지 및 수정
 - MAC(Media Access Control): 장비 고유의 일련번호로 식별하는 것
   ㄴ 모뎀, 스위치 등이 여기에 속한다.

(3) Network: IP 통신, 라우팅
- 다른 네트워크 사이의 데이터 전송, IP주소로 통신하며 라우팅 처리도 포함
  ㄴ 데이터가 큰 경우 분할 전송 후 수신하는 쪽에서 재조립한다.
  ㄴ L3 스위치, 라우터 등이 여기에 속한다.
(4) Transport: IP통신 이후 각 서비스의 포트 정의. TCP/UDP 로 구분된다.
- 호스트(PC) 간의 데이터 전송, TCP/UDP 프로토콜 구분
- 하위 단계의 오류 복구와 흐름 제어를 통해 완벽한 데이터 전송을 보장
  ㄴ L4 로드 밸런서가 여기에 속한다. 단, 4계층부터는 하드웨어로 구분하기는 어렵다.
(5) Session: TCP/IP 통신 연결의 수립/유지/중단
- 로컬 및 원격 애플리케이션 간의 IP / Port 연결(3~4단계의 통신)을 관리
(6) Presentation: 데이터를 사람이 이해 가능한 형태로 인코딩, 암호화, 압축 등
- 사용자(프로그램)-네트워크 간 데이터 변환을 통해 표현과 독립성을 제공
- 인코딩, 디코딩, 암호화, 압축
  ㄴ ex) ASCII 코드, JPG 형식 등
(7) Application: 응용 프로그램, 웹 서비스, 메일 등
- 사용자에게 제공되는 서비스

5) TCP/IP
- 네트워크 프로토콜 모음. 패킷 통신 방식의 IP와 전송 조절 프로토콜인 TCP로 구성된다.
- 인터페이스, 네트워크, 트랜스포트, 애플리케이션으로 구분된다. 네트워크 인터페이스를 둘로 나누어 5계층으로 보기도 한다.
(1) Network Interface: 물리 계층. 네트워크 노드들을 상호 연결
(2) Network: 패킷 처리, 라우팅, 다른 네트워크로 연결
(3) Transport: TCP/UDP
(4) Application: 응용 프로그램 간 표준화 데이터 교환

6) 통신 모델 비교와 캡슐화

<table>
<tr><th>TCP/IP</th><th>Service & Protocol</th><th>OSI 7 Layer</th><th>데이터 캡슐화 과정</th></tr>
<tr><td rowspan="3">Application</td><td rowspan="3">HTTP, SMTP, DNS</td><td>Application</td><td rowspan="2">사용자 데이터</td></tr>
<tr><td>Presentation</td></tr>
<tr><td>Session</td><td>표준화된 데이터</td></tr>
<tr><td>Transport</td><td>TCP, UDP</td><td>Transport</td><td>TCP 헤더 + Data
(Data Unit: Segment)</td></tr>
<tr><td>Network</td><td>IP, 라우팅(ICMP, OSPF)</td><td>Network</td><td>IP헤더 + Data
(Data Unit: Packet)</td></tr>
<tr><td rowspan="2">Network
Interface</td><td rowspan="2">이더넷(Ethernet)</td><td>Data Link</td><td>헤더와 오류검출 추가
(Data Unit: frame)
Mac LLC + Data + FCS</td></tr>
<tr><td>Physical</td><td>1과 0의 기계 신호 (bit)</td></tr>
</table>

2. 물리 계층

1) 역할
- 네트워크 장치의 전기적, 기계적 속성(장치 스펙) 및 전송 수단을 정의한다.
- 2계층(Data Link)에서 Frame(PDU, Protocol Data Unit) 형태로 데이터가 전달되면, bit 데이터를 신호(전기, 빛, 전파 등)로 인코딩하여 네트워크 장치로 전송한다.
 ㄴ 시그널링(Signaling): bit 데이터를 신호로 바꾼 형태
 ㄴ 2계층(Data Link)에서 Frame(PDU, Protocol Data Unit) 형태로 데이터가 전달되면,
- 통신 장치, 커넥터, 인코딩(bit -> signal), 송수신용 회로 등이 포함된다.

2) 시그널 종류
- 전기: 구리 케이블을 사용하며 전화선, UTP 케이블, 동축 케이블 등이 속한다.
- 빛(광): 빛의 패턴을 신호로 사용한다. Optical Fiber 케이블이 여기 속한다.
 ㄴ IEEE 802.3: 이더넷에서 물리 계층과 데이터 링크 계층의 매체 접근 제어를 정의하는 부분. 케이블도 여기에 속하며, 전기와 빛을 전송하는 케이블 규격은 여기에 정의되어 있다.
- 전파: 마이크로파 패턴을 신호로 사용한다. Wi-Fi 등의 무선 통신이 여기 속한다.
 ㄴ IEEE 802.11: 무선랜 규격이 정의되어 있다.

3) 전송 방식
- Simplex: 단방향 통신. 수신측은 송신측에 응답 불가.
- Half Duplex: 반이중 전송방식. 양방향 통신이지만 송수신 시간이 정해져 있다. (무전기)
- Full Duplex: 전이중 전송방식. 동시 양방향 통신이 가능하다. (전화기)

4) 물리계층 장비
 (1) 허브: 전기신호를 증폭하여 연결된 PC들끼리 통신이 가능하게 하는 장비
 - 동작 방식: 대표적으로 브로드캐스팅 방식이 있다.
  ㄴ 브로드캐스팅 통신(1대 All) : 허브에 연결된 PC 하나가 데이터를 전송하면, 허브에 연결된 모든 PC에게 데이터가 전달된다. 불필요한 PC는 거부하고, 데이터를 받을 PC는 수신한다.
  ㄴ 그 외에 유니캐스팅(1대 1), 멀티캐스트(1대 n) 등의 동작방식도 있다.
 - CSMA/CD(Carrier Sense Multiple Access/Collision Detection): 허브 구동방식으로 데이터 충돌을 방지하기 위한 통신 규약
  ㄴ Carrier Sensing: 데이터 전송 전 다른 노드에서 데이터를 보내는 중인지 확인한다.
  ㄴ Multiple Access: 버스에 연결된 송신 노드들은 데이터를 전송한 후 다른 노드의 충돌 발생을 감지한다.
  ㄴ Collision Detection: 동시간대에 데이터를 전송하여 충돌이 나면 모든 노드에게 통지하고 재전송을 시도한다.
  ㄴ 반이중 전송방식(Half Duplex)을 사용한다.
 (2) 리피터: 신호의 세기를 증폭하여 연결된 허브들이 좀 더 먼 거리까지 통신 가능하게 하는 장비. 현재는 거의 쓰이지 않는다.

5) 케이블과 커넥터
- 케이블: 전송 장치에 신호를 전달하는 통로. TP, 동축, Fiber 등이 있다.
- 커넥터: 케이블 양 끝단에 붙어 다른 기기와 연결시키는 부분
(1) TP (Twisted Pair): 두 개의 선을 꼬아놓은 8가닥의 선으로 구성
- 선을 꼬아놓은 건 자기장 간섭을 최소화하여 속도와 거리를 향상시키기 위함
- STP(Shield TP)는 꼬아놓은 선마다 은박같은 실드를 씌운 것이며, 그렇지 않은 것은 UTP(Unshield-)라 부른다. UTP가 좀더 광범위하게 쓰인다.
ㄴ 커넥터로는 RJ-45를 사용한다.
(2) 동축 (Coaxial): 선 중앙에 심선이 있으며 그 주위를 절연물과 외부 도체로 감싼 형태
(3) 광 (Fiber): 전기신호의 자기장이 없는 빛으로 통신하여 장거리 고속 통신이 가능
- 2개의 모드(Single, Multi)와 주요 커넥터 타입으로 LC, SC(대부분 LC 사용)가 있다.
- 광 트랜시버: 광통신에 사용되는 네트워크 인터페이스 모듈. SFP, GBIC가 있다.

6) 단위와 성능
- bit 단위는 회선 속도를 나타낼 때, btye는 데이터 크기를 나타낼 때 사용된다.
ㄴ 공통적으로 1000개 단위마다 K, M, G, T 등이 붙는다.
- 장비의 Capacity는 대역폭, 처리량, 백플레인으로 설명된다.
ㄴ 대역폭(bandwidth): 네트워크를 통해 이동할 수 있는 정보의 양
ㄴ 처리량(throughput): 실제 단위 시간당 데이터 전송을 처리하는 양
ㄴ BackPlane: 네트워크 장비가 최대로 처리할 수 있는 데이터 용량
ㄴ 대역폭이 n차선 도로라면, 처리량은 도로의 자동차 수와 같다. 장비의 대역폭은 고정되어 있으나, 처리량은 작은 데이터 여러 개보다 큰 데이터 하나일 경우 더 많아진다. (방화벽의 처리시간 등을 고려하면 큰 데이터 하나가 더 효율적이기 때문)
- 장비 계측에 사용되는 3가지 지표로 CPS, CC, TPS가 있다.
ㄴ CPS(Connections Pre Second): 초당 커넥션 연결 수 (Layer 4)
ㄴ CC(Concurent Connections): 최대 수용 가능한 커넥션
ㄴ TPS(Transactions Per Seconds): 초당 트랜잭션 연결 수, 주로 HTTP 성능 (Layer 7)

7) UTP
- UTP 케이블: 주로 근거리 통신망(LAN)에서 사용되는 케이블. 이더넷 망 구성 시 사용된다.
ㄴ 커넥터는 RJ-45를 사용하며, 코드 배열은 TIA-568A와 TIA-568B의 두 가지가 있다.
ㄴ TIA-568A: 색상 배열 [초초주파파주주갈]
ㄴ TIA-568B: 색상 배열 [주주초파파초초갈]
- 코드 배열: 8P8C - 8개의 선 배열에 따라 다이렉트 또는 크로스 케이블로 구성
ㄴ Direct Cable(568B-568B): PC-허브 연결, DTE(단말 장치)-DCE(통신 장치) 연결
ㄴ DTE(Data Terminal Equipment): 데이터가 끝나거나 변경되는 장비(PC 등)
ㄴ DCE(- Communication -): 데이터를 받아서 다시 내보내는 장비(허브 등)
ㄴ Cross Cable(568A-568B): 동일한 장치 연결에 사용(PC끼리, 허브끼리, DTE끼리 등)
ㄴ Auto MDI-X(Automatic Medium Dependent Interface Crossover): 케이블 타입에 무관하게 노드 상호간 자동으로 통신 가능하게 하는 기술. 다이렉트와 크로스 케이블을 선택하는 불편함을 해소하기 위해 등장하였다.

- UTP 카테고리: UTP 케이블의 전송 가능 대역폭을 기준으로 분류

이름	최대속도	최대길이	주 용도
Cat 3	10Mbps	100m	전화선
Cat 4	16Mbps		거의 안 쓰임
Cat 5	100Mbps		주로 예전의 100M LAN 환경
Cat 5e	1Gbps		100M(최근) ~ 1G LAN 환경
Cat 6	10Gbps		10G 통신 시 55m까지만 가능

* ISO/IEC 11801에서 구리/Fiber 케이블 등의 코드 배열에 대해 정의하고 있다.

8) 무선랜과 Wi-Fi
- 전자기기들이 무선랜에 연결할 수 있는 기술
 ㄴ 1999년 무선 네트워킹 기술 발전을 위한 협회인 비영리 기구 Wi-Fi Alliance 창립, 이후 수십 개 국가의 수백 기업이 참여하면서 Wi-Fi 용어를 채택했다.
- 무선랜: 인터넷 - ISP - 라우터 - WIPS(Wireless IPS) - AP(Access Point) - 컴퓨터
 ㄴ IPS(Intrusion Prevention System)는 침입 방지 시스템으로 일종의 보안장치
 ㄴ AP는 공유기를 의미하며, 무선랜 구축 시 AP반경과 동시접속 단말기 개수 및 802.11 규격을 고려해야 한다.
- 무선랜의 규격은 아래와 같다.

	802.11b	802.11a	802.11g	802.11n	802.11ac	802.11ax
속도	11Mbps	54Mbps	54Mbps	600Mbps	6.7Gbps	9.6Gbps
대역폭	2.4Ghz	5Ghz	2.4Ghz	2.4/5Ghz	5Ghz	2.4/5/6Ghz
최대거리	35m	35m	38m	70m	35m	30m
탄생년도	1999	1999	2003	2009 Wi-Fi 4	2013 Wi-Fi 5	2019 Wi-Fi 6

 ㄴ 802.11n Wi-Fi 4, 802.11ax Wi-Fi 6 등으로 불린다.

* WireShark: 오픈 소스 패킷 분석 프로그램 (다운로드 링크: www.wireshark.org)
 - 리눅스 TCPDUMP와 함께 네트워크 트래픽 분석에 널리 쓰이는 도구

3. 데이터 링크 계층

1) 구성과 역할
- 인접한 네트워크 노드끼리의 데이터 전송, 물리계층에서 발생할 수 있는 오류 감지 및 수정
- 대표적인 통신 규약(protocol)은 이더넷(ethernet), 장비는 스위치가 있다.
- MAC, LLC라는 2개의 부 계층으로 구성된다.
 ㄴ MAC(Media Access Control): 물리적인 부분. 1계층에 연결되며 매체 간 연결방식을 제어한다.
 ㄴ LLC(Logical Link Control): 논리적인 부분. 3계층에 연결되며 Frame을 생성한다.

- MAC 주소는 6바이트(48bit)로 구성되며 16진수로 표현한다.
 ㄴ cmd > ipconpig/all 또는 네트워크 설정에서 확인 가능하다.
 ㄴ OUI(제조사 식별코드, Organization Unique Identifier) 3바이트 + 제조사 내 일련번호 3바이트로 구성된다. 예를 들어 '76-B5-87-58-3A-20'과 같다. (숫자이므로 대소문자 무관)

2) 주요 기능
 (1) Framing: 3계층의 데이터 그램(Datagram)를 캡슐화하여 프레임 단위로 만들고, 헤더와 트레일러를 추가한다.
 - 헤더: 출발지와 목적지 주소, 데이터 내용을 정의
 - 트레일러: 에러 감지 비트
 (2) 회선 제어: 신호간의 충돌이 발생하지 않도록 제어한다.
 - ENQ/ACK 방법: 장비(PC 등) 둘이 1:1로 통신할 때 주로 사용한다. 보내는 쪽은 ENQ 신호로 데이터를 전송할 것임을 알리고, 신호를 수신한 쪽은 ACK 신호로 응답한다.
  ㄴ 전송이 끝나면 EOT 신호를 보내는 것으로 종료한다.
  ㄴ [컴퓨터A -> 컴퓨터B] 방향의 데이터 전송은 ENQ(A) - ACK(B) - 데이터 전송(A) - ACK(B) - EOT(A) 순으로 이루어진다.
 - Polling 방법: 장비가 1:다수로 통신할 때 사용한다. 수신자를 선택해 데이터를 전송한다.
  ㄴ Select 모드: 송신자가 수신자들을 선택하여 한번에 데이터를 전송한다. 데이터를 보낼 장치들에게 위 ENQ와 유사하게 Select 프레임을 보내고, ACK를 받으면 데이터를 보낸다.
  ㄴ Poll 모드: 수신자 각각에 데이터 수신 여부를 확인하고 전송한다.(multi-point) Poll 신호를 보내서 ACK 응답이 온 곳에는 데이터를 보내고, NAK이 오면 보내지 않는다.
 (3) 흐름 제어: 송신자와 수신자의 데이터 처리 속도 차이를 해결하기 위한 제어
 - Feedback 방식의 흐름 제어(Flow Control). 상위 계층에서는 Rate 방식으로 제어한다.
 - Stop&Wait 방식: 수신자에게 Frame을 보내고, ACK(Acknowledgement) 응답이 올 때까지 기다린다. 응답을 받으면 다음 Frame을 보낸다. 구현이 간단하나 비효율적이다.
  ㄴ 송신자는 프레임을 보내고 ACK 응답이 없으면 일정시간(Time-out Interval) 후 다시 프레임을 전송(Retransmit Frame)한다.
  ㄴ 수신자는 같은 데이터(프레임)를 중복해서 받을 경우 이전 프레임을 폐기하고 다시 ACK 신호를 보낸다. 중복 여부는 시퀀스 넘버를 확인한다.
 - Sliding Window 방식: ACK 응답 없이 여러 개의 프레임을 연속으로 전송 가능한 방식.
  ㄴ 송신자(Sender)와 수신자(Receiver)가 둘 다 Window라는 버퍼를 가진다.
  ㄴ 송신자는 Frame을 보낼 때마다 Window 크기를 줄이고, ACK 신호를 받으면 늘린다.
  ㄴ 수신자는 현재 ACK - 이전에 보낸 ACK 신호를 계산하여 Window 크기를 결정한다.
 (4) 오류 제어: 전송 중 오류나 손실 발생 시, 수신측이 에러를 탐지 및 재전송한다.
 - ARQ(Automatic Repeat Request): 프레임 손상 시 재전송이 수행되는 과정
 - Stop&Wait ARQ: 수신자가 받은 프레임이 손상되었을 경우 NAK 신호를 보낸다. 결과적으로 송신자는 NAK 신호를 받거나 Time-Interval이 일어날 때 재전송을 하게 된다.
 - Go Back n ARQ: Frame(0,1,2),(3,4,5)를 보냈는데 3번 프레임에 문제가 있을 경우, ACK 신호와 함께 NAK3 신호를 보낸다. 송신자는 3이 포함된 (3,4,5)를 재전송한다.
  ㄴ 하나만 손상되어도 함께 보냈던 프레임 (3,4,5) 전부를 재전송해야 한다는 단점이 있다.
  ㄴ Selective Repeat ARQ: 위 단점을 보완한 형태로, 손상된 프레임만 선별해서 재전송할

수 있다. 효율적이지만 좀 더 복잡한 알고리즘이 필요하며, 이에 따라 스위치의 CPU나 메모리 요구량이 증가할 수 있다.

3) 이더넷 프레임 구조
- Ethernet v2: MAC 통신과 프로토콜의 형식을 정의
- Preamble + Dest Addr + Source Addr + Type + 데이터 + FCS 로 구성된다.
 ㄴ Preamble(8바이트): 이더넷 프레임의 시작과 동기화. 프레임에 포함되어 있지는 않다.
 ㄴ Dest/Source Addr(각 6바이트): 목적지와 출발지의 MAC 주소
 ㄴ Type(2바이트): 상위 계층(대부분 IPv4)에서 캡슐화된 패킷의 프로토콜 정의.
 ㄴ Data: 상위 계층의 데이터. 46~1500바이트의 크기로, 46바이트 미만이면 뒤에 padding이 붙는다.
 ㄴ FCS(Frame Check Sequence, 4바이트): 에러 체크

4) L2 스위치
- MAC 주소 기반의 통신 장치로, Data Link 계층의 대표적인 장비.
- 허브의 단점을 보완했다.
 ㄴ Half Duplex -> Full Duplex : 특정 시간대에 맞춰 송/수신해야 하는 무전기 방식에서, 자유로운 통신이 가능한 전화기 방식으로 변경
 ㄴ Collision Domain 1개 -> 포트별로 존재 : 패킷 전송 시 연결된 모든 장치에 전달하는 방식에서, 지정된 도메인에만 패킷을 보내도록 변경
- 라우팅 기능이 있는 스위치는 L3 스위치라 부른다.
- 동작 방식: MAC 주소 테이블에서 목적지 주소를 확인하여, 연결된 포트로 프레임 전송
 (1) Learning: 출발지 주소가 MAC 주소 테이블에 없으면 해당 주소 저장
 - 스위치의 포트에 PC가 연결되고 최초로 프레임을 전달할 때, 스위치는 전달된 프레임을 보고 연결된 PC의 MAC 주소를 수집하여 테이블에 저장한다.
 (2-1) Flooding-Broadcasting: 목적지 주소가 MAC 주소 테이블에 없으면 모든 포트에 프레임을 전달
 - Flooding은 전달받은 패킷을 다른 모든 곳으로 전달하는 동작을 의미한다.
 (2-2) Forwarding: 목적지 주소가 MAC 주소 테이블에 있으면 해당 포트로 전달
 (3) Filtering Collision Domain: 출발지와 목적지가 같은 네트워크 영역에 있으면, 다른 네트워크로 전달하지 않는다.
 (4) Aging: MAC 주소 테이블의 각 주소는 일정 시간이 지나면 삭제된다.
 - 테이블 저장 공간을 효율적으로 사용하고, PC가 다른 포트로 옮겨져서 같은 MAC주소가 둘 이상의 포트에 남아 있는 경우의 충돌도 방지할 수 있다.
 - 일반적으로 해당 포트에서 마지막으로 프레임이 들어온 시점에서 300초(Cisco 기준)가 경과하면 삭제한다.

5) ARP (Address Resolution Protocol)
- IP 주소를 통해 MAC 주소를 알려주는 프로토콜
 ㄴ 상위 계층에서 IP 통신을 할 때 전달할 패킷과 목적지의 IP 주소가 2계층으로 넘어오는데, IP 주소에 해당하는 MAC 주소는 ARP를 통해 알아내야 한다.

- 작동 방식: ARP Cache Table 확인 -> 없으면 ARP Request -> ARP Reply
 ㄴ MAC 주소는 ARP Request를 Broadcasting 방식으로 뿌리고, 해당 IP를 자신 수신자로부터 ARP Reply를 받아서 알아낸다. 이후 이 정보를 ARP Cache Table에 저장한다.
 ㄴ 패킷 송신자는 ARP Request를 뿌리기 전에 자신의 ARP Cache Table를 확인한다. 테이블에 목적지의 ARP 주소가 있으면 ARP Request 없이 바로 패킷을 전송한다.
- ARP 헤더 구조: 하드웨어 타입, 프로토콜 타입, 하드웨어 길이, 프로토콜 길이, 명령 코드, 송/수신자의 하드웨어 주소와 프로토콜 주소로 구성된다.
 ㄴ Hardware Type: ARP가 동작하는 네트워크 환경(보통 이더넷)
 ㄴ Protocol Type: 프로토콜 종류(대부분 IPv4, 숫자로는 0x0800)
 ㄴ Hardware/Protocol Length: MAC 주소 6바이트, IP주소 4바이트
 ㄴ 명령 코드(Operation): ARP Request면 1, Reply면 2가 된다.
 ㄴ Sender/Target Hardware Address: 송/수신자의 MAC 주소
 ㄴ Sender/Target Protocol Address: 송/수신자의 IP 주소

6) STP (Spanning Tree Protocol)
- 루핑(Looping): 2개 이상의 스위치들끼리 프레임을 서로 계속 전달하여 생기는 문제
 ㄴ 회선 및 스위치의 이중화 또는 증축으로 인해, 같은 네트워크 대역 내에서 스위치에 연결된 경로가 2개 이상인 경우 발생한다. 2계층에서 가장 빈번히 발생하는 이슈.
 ㄴ 스위치들이 서로의 프레임을 끝없이 전송한다. '브로드캐스트 스톰' 이라고도 표현한다.
- STP: 자동으로 루핑을 막아주는 알고리즘(스패닝 트리 알고리즘)에 사용되는 프로토콜
- Bridge ID와 Path Cost의 두 가지 우선순위를 가진다. 낮을수록 우선순위가 높다.
 ㄴ Bridge ID: 스위치의 우선순위. 0~65535로 설정된다. 디폴트값은 중간값인 32768.
 ㄴ Path Cost: 링크의 속도(대역폭). 1Gbps는 4, 100Mbps는 19, 10Mbps는 100.
 ㄴ Path Cost는 초기에 '1000/대역폭' 으로 계산되었으나, 1Gbps 대역폭 등장 이후 우선순위가 소수점 이하로 떨어져서 IEEE에서 대역폭별 숫자를 정의했다.

6-1) STP 구성요소
- STP는 Root Bridge, Root Port, Designated Port의 세 가지 요소로 구성된다.
 ㄴ Root Port, Designated Port로 지정되지 않은 포트는 모두 막는다.
 (1) Root Bridge: 네트워크에서 1개의 스위치를 지정
 - 각 스위치는 고유 BID(Bridge ID)를 가진다.
  ㄴ 우선순위 2바이트 + MAC주소 6바이트로 구성된다.
  ㄴ 우선순위는 기본값이 32768이며 명령어로 지정할 수 있다.
 - 스위치끼리 BPDU를 교환하여 가장 낮은 BID를 가진 스위치가 Root Bridge로 지정된다.
 (2) Root Port: Root Bridge가 아닌 각 스위치마다 1개의 포트를 지정
 - Path Cost를 고려하여 루트 브릿지와 가장 빠르게 연결 가능한 포트를 지정한다.
 (3) Designated Port: 세그먼트마다 하나씩 지정
 - 각 세그먼트별로 루트 브릿지와 가장 빠르게 연결되는 포트를 지정
  ㄴ 세그먼트: 임의의 두 스위치 사이의 연결선.
 - 우선순위: 루트 브릿지 ID > Path Cost > 브릿지 ID > 포트 ID
  ㄴ 루트 브릿지와 연결되는 세그먼트는 무조건 루트쪽 포트가 Designated Port가 된다.

- 셋 모두에 속하지 않는 포트는 Non-Designated 포트라고 한다.

6-2) BPDU(Bridge Protocol Data Unit)
- STP 구조에서 스위치 사이에 주고받는 제어 프레임
 ㄴ BPDU들은 약 2초당 한 번씩 발생한다.
 (1) Configuration BPDU: 구성 관련 BPDU
 - Root BID: 루트 브릿지로 지정될 스위치 정보
 - Path Cost: 루트 브릿지까지의 경로 비용
 - Bridge ID, Port ID: 나머지 스위치와 포트의 우선순위
 (2) TCN(Topology Change Notification) BPDU: 네트워크 내 구성 변경 통보
 - 구성이 변경될 때마다 업데이트를 하기 위한 프레임

6-3) 상태 변화
- STP는 5가지 상태를 가진다.
 (1) Disabled
 - 포트가 Shut Down인 상태
 - 데이터 전송 불가, MAC 학습 불가, BPDU 송수신 불가
 (2) Blocking
 - 부팅 또는 Disabled 상태를 Up 했을 때 처음 거치는 단계
 - BPDU 송수신만 가능
 (3) Listening
 - Blocking 포트가 루트 또는 Designated 포트로 선정되는 단계
 - BPDU 송수신만 가능
 - 15초간 지속된 후 Learning 상태로 넘어간다.
 (4) Learning
 - MAC 주소 학습 시작, BPDU 송수신 계속함
 - 15초간 지속된 후 Forwarding 상태로 넘어간다.
 (5) Forwarding
 - 데이터 전송 시작, BPDU 송수신 계속함

6-4) RSTP & MST
- RSTP(Rapid STP): STP를 적용하면 포워딩까지 30~50초가 소요된다(3.6-3 참고). 이 시간을 Listening & Learning 단계를 삭제하여 1~2초로 줄인 형태의 STP.
 ㄴ 대부분의 네트워크 환경은 RSTP로 구성되어 있다.
- MST(Multiple Spanning Tree): 여러 개의 STP를 묶어서 효율적으로 관리하는 구조
 ㄴ 네트워크 그룹이 많아지면 BPDU 프레임이 많아져 스위치에 부하가 발생하기 때문

7) VLAN (Virtual LAN)
- (물리적 구성이 아닌) 논리적인 가상의 LAN을 구성하는 기술
- 한 스위치 내에서도 가상으로 브로드캐스트 도메인을 나눌 수 있다.
 ㄴ 관리 및 보안: 불필요한 데이터 전송을 막고, 물리적인 이동 없이 LAN 그룹 변경 가능

ㄴ 비용 절감: 큰 스위치 하나를 사서 가상으로 나누어 쓸 수 있어 효율적

7-1) 종류
- Port, MAC, IP 기반으로 구성할 수 있다.
(1) Port 기반 VLAN
- VLAN의 각각의 그룹에 물리적 포트들을 지정한다.
- VLAN 변경은 물리적인 포트 또는 스위치의 VLAN 설정을 변경함으로써 이루어진다.
(2) MAC 주소 기반 VLAN
- VLAN에 각 호스트(PC 또는 네트워크 장비)의 MAC 주소를 정의한다.
- 포트가 바뀌어도 MAC 주소만 같다면 VLAN 변경이 불필요하다.
- 신규 호스트 연결 시에는 해당 호스트의 MAC 주소를 확인하여 그룹에 포함시키는 초기 설정이 필요하다.
(3) IP주소 기반 VLAN
- IP주소 서브넷 기반으로 VLAN을 나눈다.
ㄴ IP(Internet Protocol): 3계층에서 사용하는 프로토콜
ㄴ 서브넷: IP주소들의 네트워크 영역을 일정 크기로 구분한 것

7-2) 트렁크(Trunk) 프로토콜
- 이더넷 프레임에 식별용 VLAN ID를 삽입하여 데이터를 구분하는 것
ㄴ 2개 이상의 스위치에서 하나의 물리적 연결만으로 VLAN 그룹을 공유하기 위해 사용
ㄴ 태그를 단다고 표현하며, VLAN ID 정보를 'VLAN Tagging' 이라고도 한다.
ㄴ IEEE 802.1q 에 정의되어 있다.
- 802.1q tagged format: 이더넷 프레임에 삽입되는 4바이트의 태그
ㄴ TPID(16bit)와 TCI(PCP 3bit, DEI 1bit, VID 12bit, 총 16bit)로 구성된다.
ㄴ TPID(Tag Protocol IDentifier): 태그된 프레임인지 아닌지 구분하는 식별자
ㄴ TCI(Tag Control Information): 태그 제어 정보. PCP, DEI, VID로 구성된다.
ㄴ PCP(Priority Code Point): 프레임의 우선순위
ㄴ DEI(Drop Eligible Indicator): (트래픽 혼잡 시) 제거 대상 프레임 구분
ㄴ VID(VLAN IDentifier): VLAN이 어느 프레임에 속하는지 결정

7-3) VLAN 구성 방법
- VLAN 구성 전에 그룹 구분, 종류, 트렁크에 관해 정한다.
(1) VLAN 그룹을 어떻게 구분할지 정한다.
(2) VLAN 구성방법을 정한다.
- Port, MAC, IP 기반으로 구성할 수 있다. 위의 3.7-1 참고.
(3) 어떤 트렁크 포트를 사용할지 정한다.
- 다양한 태깅 프레임이 오가므로 큰 대역폭이 필요하다.
- 프레임에 어떤 태그를 허용할지 정한다. 정의되지 않은 태그는 통신 불가.
- 이후 VLAN을 설정한다.
ㄴ 제조사별로 다르니 실제로 설정할 때는 홈페이지 매뉴얼 참고. 여기서는 CISCO 기준으로 설명한다.

(1) VLAN 그룹을 설정한다.
(2) 액세스 모드를 설정한다.
- 사용할 포트에 1개의 VLAN ID를 설정한다.
(3) 트렁크 모드를 설정한다.
- 사용할 포트에 여러 개의 VLAN ID를 설정한다.
(4) 다이나믹 모드를 설정한다.
- 다이나믹 모드: 연결된 포트들의 상태에 따라 액세스 또는 트렁크로 변경되는 모드

4. IP (Internet Protocol)

1) 역할
- 네트워크 계층의 주요 통신 프로토콜
 ㄴ 네트워크 계층은 패킷 포워딩과 네트워크간 패킷의 패킷의 전송 경로 지정(라우팅)을 수행한다.
- 라우팅을 구현하고 본질적인 인터넷을 구축하는 계기가 되었다.
 ㄴ 1974년 IEEE 논문에서 IPv4가 발표된 후 기존의 RFC760,761를 사용하던 전송 제어 프로그램이 RFC791(IP),793(TCP) 으로 대체되었다. TCP/IP 모델의 기원.
 ㄴ TCP/IP: 서로 다른 시스템의 컴퓨터를 연결하고 데이터를 전송하는 통신 프로토콜 집합
- IP주소: 네트워크에 연결된 기기를 식별하는 고유 주소
 ㄴ ipconfig(윈도우) / ifconfig(리눅스) 명령어로 확인 가능
 ㄴ IPv4 기준 32비트로 구성되며 약 42억 여 개의 할당이 가능하다.
 ㄴ 현재는 IPv4 사용중이나, IP 주소 개수가 고갈되어감에 따라 IPv6가 릴리즈되었다.

2) 구조
- 네트워크 계층의 IP 패킷은 헤더와 페이로드로 구성되어 있다.
 ㄴ 헤더(IP Header): 출발지와 목적지의 IP 주소
 ㄴ 페이로드(Payload): 전송되는 데이터
- IP Fragmentation: IP패킷을 작은 패킷으로 나누어 전송하고 목적지에서 재조합하는 방식
- MTU(Maximum Transmission Unit): IP패킷을 전송할 수 있는 최대 크기. 네트워크 경로나 라우터가 처리할 수 있는 최대 크기. 이를 초과하면 Fragmentation이 된다.

2-1) IP 헤더의 구성요소
- IPv4 기준, 옵션 미 지정 시 최소 20바이트(4바이트 단위로 최소 5개)로 구성된다.
- 0~4바이트: 버전(4bit), 헤더 길이(4bit), TOS(8bit), 전체 패킷 길이(16bit)
 ㄴ 버전: IP 버전. 여기서는 IPv4.
 ㄴ 헤더 길이: 이 IP헤더의 길이 (최소 20 ~ 최대 60바이트)
 ㄴ TOS(Type of Service): 서비스 품질
  ㄴ 전체 패킷 길이: IP패킷 전체(헤더+페이로드)의 길이. 최대 65535.
- 5~8바이트: Identifier(16bit), Flags(4bit), Offset(12bit)으로 구성.
 ㄴ IP Fragment 필드로 단편화와 재조합, 큰 패킷을 작은 패킷으로 나눠 보낼 때 사용한다.

- 9~12바이트: TTL(8bit), Protocol ID(8bit), Header Checksum(16bit)
 ㄴ TTL(Time To Live): IP패킷의 수명. 해당 시간 경과 시 폐기한다.
  ㄴ 보통 64(리눅스 계열) 또는 128(윈도우)홉. 홉(Hop)에 관해서는 라우팅에서 후술.
 ㄴ Protocol ID: 데이터에 포함된 상위 계층의 프로토콜 정보
  ㄴ 번호가 지정되어 있다. 예를 들어 TCP는 6번, UDP는 17번.
 ㄴ Header Checksum: 오류 검출에 사용된다.
- 13~20바이트: 출발지(Source), 목적지(Destination)의 IP 주소. 각 4바이트.
- 그 밖에 IP헤더 옵션과 Padding이 붙을 수 있는데, 디버깅 외엔 거의 사용되지 않음

3) IP 주소
- 네트워크 부분과 호스트 부분으로 나뉜다.
 ㄴ 네트워크: ip주소의 앞 3바이트. 라우터에 해당하며 브로드캐스트를 수행한다.
 ㄴ 호스트: ip주소의 마지막 8비트. 개별 단말기(PC)에 해당한다.
 ㄴ 예를 들어 192.168.1.16에서 192.168.1은 네트워크, 16은 호스트
- 같은 라우터 내의 호스트끼리는 브로드캐스트 스위칭으로 통신이 가능하지만, 라우터가 다를 경우 라우팅이 필요하다.
 ㄴ 다른 시에 거주한다면 시외버스를 타야 하는 것에 비유할 수 있다.
- IP주소 클래스: IP주소를 네트워크 크기에 따라 A~E의 5개 단계로 구분한 것.
 ㄴ A클래스: 0~127(127.255.255.255까지) 사용 가능 (호스트는 0.0.0~255.255.255, 2^{24}개)
 ㄴ B클래스: 128 ~ 191 사용 가능 (호스트는 0.0~255.255, 2^{16}개)
 ㄴ C클래스: 192 ~ 223 사용 가능 (호스트는 0~255, 256개)
  ㄴ 가장 많이 사용되는 클래스
 ㄴ D클래스: 멀티캐스트용으로만 사용한다. 224 ~239 사용 가능.
 ㄴ E클래스: 연구용으로만 사용한다. 240~255(255.255.255.254) 사용 가능.
- 네트워크 별 첫 번째 숫자와 마지막 숫자는 호스트에게 할당하지 않는다.
 ㄴ 첫 번째 숫자는 네트워크 영역을 알리는 용도, 마지막 숫자는 해당 주소로 오는 패킷을 브로드캐스팅 하는 브로드캐스트 주소이다.

4) 서브넷
- 부분망. 할당된 네트워크를 효율적으로 사용하기 위해 쪼개어 사용하는 서브 네트워크.
- 서브넷 마스크: 네트워크를 여러 개의 서브넷으로 구분하는 개념
 ㄴ 서브넷 마스크 계산법: 비트가 1인 부분은 네트워크, 0인 부분은 호스트가 된다.
 ㄴ 'AND' 비트 연산을 통해 서브넷 네트워크 주소만 추출할 수 있다.
- 서브넷은 2^n개 단위로 구성한다. 서브넷 마스크로 비트 연산하기 용이한 형태이기 때문.
 ㄴ 예를 들어 254개의 네트워크를 할당받았으나 라우터1개 - 호스트3개만 있을 경우, 각 호스트(3개) + 게이트웨이 주소 + 네트워크 영역 알림 + 브로드캐스트 주소 = 총 6개의 IP가 필요하다. 이 경우 2의 3승인 8개의 IP로 서브넷을 구성한다.
- 디폴트 게이트웨이: 다른 네트워크로 패킷 전송 시 거쳐야 하는 거점
 ㄴ 여러 경로를 가진 라우터는 보통 1개의 경로를 디폴트 게이트웨이로 지정한다.
- Prefix 표기법: 네트워크 영역에서 '1'인 비트의 개수를 '/개수' 형태로 표현한다. 서브넷 마스크 표기를 간단히 표현하기 위해 사용한다.

ㄴ 예를 들어 ip가 1.2.3.4고, 서브넷 마스크가 255.255.255.240(=> 비트 1의 개수 28개)이라면 '1.2.3.4/28'로 표기할 수 있다. 반대로 '/28' 표기를 보고 마지막 8비트가 11110000임을 알 수 있다.
ㄴ 단, LAN 설계 시 C클래스 단위로 나누는 것이 가장 일반적이긴 하지만, C클래스가 아닌 경우에는 서브넷 마스크가 255.255.255.x 형태가 아닐 수 있으니 유의.

* 서브넷 마스크 계산기: www.subnet-calculator.com

5) ICMP (Internet Control Message Protocol)
- 인터넷 제어 메시지 프로토콜. IP통신의 에러 상황을 출발지에 전달, 메시지를 제어한다.
ㄴ IP통신에서 에러 발생 시 처리가 불가한 점을 보완하기 위해 1981년 소개됨(RFC 792)
- ICMP는 IPv4 패킷으로 캡슐화된다.
ㄴ IP 헤더의 Protocol ID에 1번으로 표기된다.
- ICMP의 내용(payload)에는 Type(1byte), Code(1byte), Checksum(2bytes)가 포함된다.
ㄴ 추가적으로 타입과 코드에 따라 가변길이가 추가된다.
ㄴ Type: ICMP 메시지 종류
ㄴ Code: 메시지 타입별 세부 코드 정보
ㄴ Checksum: ICMP 헤더 손상 여부를 확인하는 값
- 터미널에선 ping 또는 traceroute 명령어를 통해 사용된다.
ㄴ ex) ping 8.8.8.8

5-1) ICMP Type 종류
- 0~254까지 정의되어 있다. (링크 참고)
- 주로 쓰이는 타입은 0,8,9,10(정보용), 3,5,11,12(오류 보고용) 이다.
- Type 8,0(Echo Request, Reply): 네트워크 문제 진단 시 사용한다.
ㄴ ICMP Echo Request를 보내면 목적이는 Echo Reply로 응답한다.
ㄴ 목적지 도달 여부, 소요시간(RTT, Round-Trip delay Time), TTL을 확인할 수 있다.
- Type 9,10(라우터 광고, 라우터 정보 요청): 자신이 라우터임을 알리거나, 라우터가 누구인지에 대한 정보를 요청할 때 사용한다.
- Type 3(Destination Unreachable): IP패킷 라우팅에 실패했을 때 발생한다.
ㄴ Code가 0인 경우: 네트워크 도달 실패
ㄴ Code 1: 호스트 도달 실패
ㄴ Code 2: 프로토콜에 문제가 있음
ㄴ Code 3: 포트에 문제가 있음
ㄴ Code 4: fragmentation에 문제가 있음
ㄴ Code 5: 소스에 문제가 있음
- Type 5(Redirect): 로컬 네트워크에 2개 이상의 경로가 있을 때 더 좋은 경로를 알려준다.
ㄴ 네트워크가 최적의 경로로 구성되어 있지 않다는 뜻으로, 라우터 설정을 다시 해줘야 함
- Type 11(Time Exceeded): 시간 초과. TTL 값이 0이 될 때 출발지에 응답한다.
ㄴ Code 0: TTL이 만료되었을 때
ㄴ Code 1: 여러 개로 쪼갠 패킷(Fragment)에서 일부가 TTL만료로 누락되었을 때

- Type 12(Parameter Problem): IP옵션을 잘못 사용하여 패킷 폐기 요청

6) DHCP (Dynamic Host Control Protocol)
- 동적 호스트 구성 프로토콜. 네트워크 장치에 IP 주소를 자동으로 할당해준다.
 ㄴ 요청에 의한 IP 할당과 회수로 인해 효율성 극대화
 ㄴ 잘못된 IP 설정으로 인한 장애 예방
 ㄴ IP 변경이 잦은 호스트 관리 용이
- DHCP 할당 정보는 윈도우 기준 cmd에 ipconfig/all을 입력하여 확인할 수 있다.
 ㄴ ipconfig/renew, ipconfig/release로 ip를 갱신 또는 해제할 수 있다.
- DHCP 메시지 포맷
 (1) OpCode, Hardware Type, Hardware Address Length, Hop Count (각 1바이트)
 - OpCode(Operation Code): 1번이면 서버에 IP 요청(Request), 2번이면 서버가 클라이언트에 응답(Reply)
 - Hardware Type: 대부분의 경우 이더넷(Ethernet). 1번으로 표현된다.
 - Hardware Address Length: 대부분 MAC 주소이므로 6이다.
 - Hop Count: 0에서 시작하여 1홉(Hop)마다 카운트
 (2) Seconds, Flags (각 2바이트)
 - Seconds: IP할당 후 경과한 시간 (초)
 - Flags: 서버 응답 구분 값. 0이면 unicast, 1이면 broadcast.
 (3) Transaction ID (4바이트)
 - 통신 시 Key값으로 사용된다. 요청과 응답이 맞는지 매칭할 때 사용된다.
 (4) 클라이언트 IP 주소 (4바이트)
 - 최초엔 0.0.0.0이며, 할당될 IP가 들어갈 자리이며 추가로 옵션을 받는다.
 - 옵션에는 DHCP 메시지 타입을 알려주는 값(1~8)이 들어간다
  ㄴ 1(DISCOVER): 클라이언트가 서버를 찾기 위한 브로드캐스팅 메시지
  ㄴ 2(OFFER): 서버가 클라이언트에게 할당할 IP주소 제시
  ㄴ 3(REQUEST): 클라이언트가 원하는 구성을 서버에 요청
  ㄴ 4(DECLINE): 이미 사용중인 IP가 요청되었다면 서버에서 거부
  ㄴ 5(ACK): 요청 수락
  ㄴ 6(NAK): 요청 거부
  ㄴ 7(RELEASE): 클라이언트가 서버에 IP 해제 요청
  ㄴ 8(INFORM): 클라이언트가 추가 설정 정보 요청
 (5) 서버 IP 주소 (4바이트)
 (6) 게이트웨이 IP 주소 (4바이트)
 (7) Client Hardware Address (16바이트)
 (8) Server Host Name (64바이트)
 (9) Boot File Name (128바이트)

6-1) DHCP 동작 예시
- IP 할당 과정
 DHCPDISCOVER(1): PC가 DHCP 서버 발견

DHCPOFFER(2): DHCP 서버가 PC에게 IP 제안
DHCPREQUEST(3): PC는 제안 받은 IP를 할당해달라고 요청
DHCPACK(5): 서버가 요청 수락

- IP 갱신 과정 (IP 갱신 타입이 도래하면 갱신을 요청해야 함)
DHCPREQUEST(3): PC는 기존 IP를 계속 쓰겠다고 재할당을 요청
DHCPACK(5): DHCP 서버가 확인 후 요청 수락
- IP 해제 과정 (사용중인 PC 전원이 off 되는 경우)
DHCPRELEASE(7): PC는 더 이상 IP 할당이 필요없음을 알림

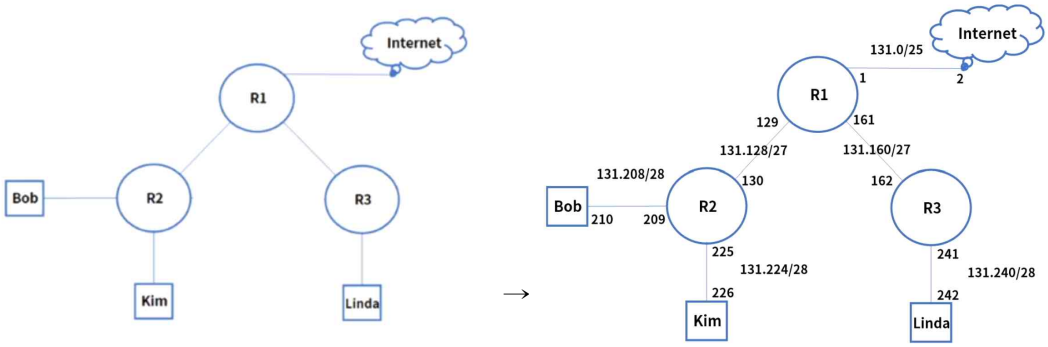
5. 라우팅 (Routing)

1) 라우팅 프로토콜
- 라우팅(Routing): 목적지 IP주소를 확인하고 하나 혹은 여러 개의 네트워크 간 패킷 경로를 선택하여 전송하는 과정. 네트워크 계층(L3)의 주요 역할.
- 전세계 네트워크 호스트는 IP라우팅을 통해 연결된다. 패킷은 Hop, TTL로 관리된다.
 ㄴ Hop: 소스와 목적지 간의 경로. 패킷이 경로 하나를 지날 때마다 1hop으로 계산한다.
 ㄴ TTL(Time to Live): 패킷의 남은 hop카운트. 0이 되면 폐기한다.
- 터미널에 traceroute 또는 tracert 명령어를 통해 라우팅 경로를 확인할 수 있다.
 ㄴ ex) tracert -d www.naver.com
- 크게 정적 라우팅, 동적 라우팅으로 구분된다.
 ㄴ 정적(static) 라우팅: 경로 정보를 라우터에 미리 저장하여 패킷 전송
 ㄴ 동적(dynamic) 라우팅: 경로 정보가 네트워크 상황에 따라 더 빠른 경로로 유동적으로 변경되며 패킷 전송

2) 라우터
- 라우터(Router): 네트워크 간 패킷을 전송해주는 장비. 기호로는 ⊗ 형태로 표현된다.
 ㄴ 라우티드(Routed): 라우터가 라우팅하는 대상. 즉, IP를 뜻한다.
- 인터페이스: 라우터에 있는 접속 가능한 포트. 통신용과 관리용으로 구분된다.
 ㄴ 통신용: UTP, 광, 무선으로 구성. WAN(라우터 간 연결)은 시리얼 포트도 존재한다.
 ㄴ 관리용: 보통 콘솔 포트라고 부른다. 원격에서 접속이 불가할 때(접속 장애 등) 장비에 직접 연결하기 위해 사용한다.
ㄴ 일반적으로 RS232 인터페이스를 사용한다.

3) 정적 라우팅 (Static Routing)
- 수동으로 경로를 라우터에 입력하여 패킷을 처리하는 방식
 ㄴ 라우팅 테이블에 목적지 IP 주소와 인터페이스 정보를 설정하여 경로를 결정한다.
 ㄴ 가장 기본적인 라우팅 방식
- 라우팅 테이블을 구성하는 과정은 아래와 같다.
 (1) 서브넷 마스크를 이용하여 네트워크 대역을 할당하고, 할당한 네트워크 대역 내에서 IP 주소를 할당한다. (처음, 마지막 숫자는 할당하지 않는다. 4.2-2 참고)

- 예시) xxx.xxx.131.0~255 네트워크를 이용해 서브넷을 구성한 모습



(2) 각 PC별로 IP주소, 서브넷 마스크, 게이트웨이를 지정한다.

- 보통 기본(default) 게이트웨이만 인접한 라우터로 설정한다.
  ㄴ 위 예시에서 Bob의 세팅값은 아래와 같다.
  IP 주소: xxx.xxx.131.210
  서브넷 마스크: 255.255.255.240
  기본 게이트웨이: xxx.xxx.131.209

(3) 경로 별 네트워크 대역을 확인해서 connected, static, default 라우팅을 설정한다.

- 커넥티드(connected): 라우터의 포트에 직접 연결된 네트워크 대역. 설정할 필요 없음.
- 정적(static): 직접 연결되지 않은 네트워크 대역으로 이동하는 경로를 지정
  ㄴ 예시) 위 예시에서 R1의 정적 라우팅 설정
  131.208/28 via 131.130 (131.208/28로 이동하려면 131.130으로 보낸다)
  131.224/28 via 131.130
  131.240/28 via 131.162
- 디폴트(default): 테이블에 없는 네트워크 대역을 전송할 경로
  ㄴ 정적 라우팅 설정에서 0.0.0.0/0 에 대한 경로를 지정하면 디폴트가 된다.
  ㄴ 호스트가 패킷을 보낼 경우 외부(인터넷)로 내보낼 경로로 사용된다.
  ㄴ 예시) 위 예시에서 R1, R2 라우터의 디폴트 라우팅 설정
  R1: 0.0.0.0/0 via 131.2 (인터넷으로 보낸다)
  R2: 0.0.0.0/0 via 131.129 (R1으로 보낸다)

4) 라우팅 알고리즘

- 목적지까지 최적의 경로를 계산하고 라우팅 테이블에 업데이트하는 알고리즘
  ㄴ 동적으로 라우팅 테이블을 관리하는 데 사용한다.
- 대표적으로 Distance Vector와 Link State 알고리즘으로 구분된다.

(1) Distance Vector 알고리즘

- 각 라우터가 연결되는 지점들에서 분산 업데이트. 각 라우터들에 의해 최소 비용 경로가 계산하여 인접 노드와 교환한다.
  ㄴ 주기적이며 비동기 방식. 소규모 네트워크에 적합하다.
- 목적지까지 거리(= Hop Count)는 '라우터 사이의 거리 + 인터페이스 방향'이 된다.
- 장점: 비교적 구성이 간단하고, 인접 라우팅 테이블만 관리하므로 메모리가 절약된다.
- 단점: 주기적 라우팅 테이블 업데이트로 무의미한 트래픽이 발생할 수 있고, 네트워크 전

체의 라우팅 테이블 업데이트 시간이 느리다.
ㄴ 라우터A -> 라우터B -> 라우터C -> ... 순서대로 주기를 기다려가며 업데이트되기 때문
- 최단 경로 알고리즘: Bellman-Ford 알고리즘
ㄴ 연결된 각 링크의 비용이 변경됨에 따라 주기적으로 최단 거리를 업데이트하여 인접 노드에 전달한다.
ㄴ '목적지 A까지 최소 거리 = (나와 연결된 라우터까지 거리 + 해당 라우터에서 A까지 최소 거리) 중 가장 작은 값' -> A와 연결된 라우터에 도달할 때까지 반복
(2) Link State 알고리즘
- 중앙 집중형 업데이트, 네트워크 전체 정보를 통해 최소 비용 경로 계산
ㄴ 이벤트 기반의 라우팅 테이블 관리. 대규모 네트워크에 적합하다.
- 회선의 대역폭을 고려하여 가중치를 부여한다.
- 네트워크 토폴로지 경로를 모든 라우터들에게 전달한다.
ㄴ 라우팅 정보가 변경되는 이벤트 건에 대해서만 전파하여 트래픽을 감소시킨다.
ㄴ 전체 네트워크상의 라우터들이 가진 라우팅 테이블 정보가 동일하게 유지된다.
- 최단 경로 알고리즘: 다익스트라(Dijkstra) 알고리즘
ㄴ 출발지와 목적지 사이의 최단 경로를 구하고, 이 정보를 기반으로 각 라우터들은 최상의 경로를 계산한다.
ㄴ 자신을 제외한 모든 라우터까지의 거리를 무한으로 설정해두고, 링크된 각 라우터까지의 거리를 계산한다. 계산 값이 현재 값(초기값=무한)보다 작으면 계산 값으로 업데이트한다.

5) 동적 라우팅 구분
- 동적 라우팅은 '자율 시스템(AS, Autonomous System)'에 따라 구분된다.
ㄴ AS: 하나의 회사 또는 단체 내에서, 같은 라우팅 정책으로 관리되는 라우터들의 그룹
- EGP(Exterior Gateway Protocol): AS와 다른 AS 간의 라우팅 프로토콜.
ㄴ 대표적으로 BGP가 있다.
- IGP(Interior-): 같은 AS 내에서 동작하는 라우팅 프로토콜.
ㄴ 대표적으로 RIP, OPSF가 있다.

5-1) BGP (Border Gateway Protocol)
- 대표적인 EGP 라우팅 프로토콜로, ISP to ISP 연결에 사용한다. 현재 버전은 BGP4.
- eBGP와 iBGP로 구성된다.
ㄴ eBGP: AS 간의 연결 및 라우팅 정보를 교환
ㄴ iBGP: 동일 AS 내에서 BGP 라우팅 정보 교환
- Distance Vector 알고리즘에서 루핑 방지를 위해 개선한 경로 벡터 라우팅 프로토콜 사용
- 빠른 속도보다 안정성을 중요시하며, 조직간 정책에 의거하여 최적의 경로를 결정한다.
- BGP 설정은 라우터ID, Neighbor, Network 설정을 통해 가능하다.
ㄴ Router ID 설정: 각 라우터의 식별용 고유 IP 설정
ㄴ Neighbor 설정: 인접 라우터의 AS 번호 설정. 안정성을 중요시하므로 자동 탐지가 불가하며, 수동으로 직접 연결된(connected) 인터페이스로만 Next Hop을 설정해야 한다.
ㄴ Network 설정: 전파할 네트워크 대역
- BGP는 4가지 메시지로 라우팅 정보를 교환한다.

(1) OPEN: 인접 라우터와 연결된 후 보내는 메시지
- BGP버전, AS번호, Hold Time, Operation Parameter
(2) UPDATE: 라우팅 테이블 경로에 대한 속성 값
- Unreachable Route(끊어진 부분이 있는지), Path Attribute(Path의 속성이 변경되었는지), Network Layer Rechability(네트워크 계층 복구 기능)
(3) NOTIFICATION: 에러가 감지되면 에러 코드를 보내고 BGP 연결 종료
(4) KEEPALIVE: 주기적으로 인접 라우터와의 연결을 확인. 이 메시지에 응답이 없으면 해당 라우터에 문제가 있다고 판단한다.
- BGP FSM(Finite State Machine): BGP는 인접(peer) 라우터와의 동작을 결정하기 위해 6 자기 유한 상태 머신(FSM)을 사용한다.
ㄴ 1. Idle: 모든 자원을 초기화하고 인접 라우터 연결 준비 상태 (-> 2번 상태로 이동 가능)
ㄴ 2. Connect: 연결 완료를 기다리는 상태 (-> 4,3)
ㄴ 연결 성공 시 4번, 연결 실패 시 3번으로 이동
ㄴ 3. Active: 연결 실패 이후 다시 연결을 시도하는 상태 (-> 4,1,2)
ㄴ 4. Open Sent: OPEN 메시지를 보내는 상태 (-> 5,1,3)
ㄴ 오류 발생 시 NOTIFICATION, 정상이면 KEEPALIVE 메시지를 보낸다.
ㄴ 5. Open Confirm: OPEN 메시지를 확인한 상태 (-> 6,1)
ㄴ KEEPALIVE를 받은 경우 6번, NOTIFICATION을 받은 경우 1번으로 이동
ㄴ 6. Established: KEEPALIVE 메시지의 송수신이 가능함이 확인된 상태 (-> 1)

5-2) RIP (Routing Information Protocol)
- Distance Vector 기반의 IGP용 라우팅 프로토콜.
ㄴ RIPv1: Classful 라우팅. 라우팅 업데이트 시 서브넷마스크를 사용하지 않고, 브로드캐스팅 방식을 이용한다. 초기에 많은 IP가 활용될 것을 모르고 사용했다.
ㄴ RIPv2: Classless 라우팅. 라우팅 업데이트 시 서브넷마스크 정보를 전달한다. 멀티캐스팅 방식을 사용하며, Triggered Update(변경된 부분만 업데이트) 설정이 가능하다.
- 속도가 아닌 거리(hop) 기반으로 경로를 선택한다.
ㄴ 최대 hop count는 15로, 이를 초과하면 해당 라우팅 패킷은 폐기된다.
- 주기적으로 전체 라우팅 테이블을 업데이트한다. (보통 30초)
- 구성이 간단하고 메모리 사용량이 적다. 소규모 네트워크에서 주로 사용한다.

5-2-1) RIP 메시지 포맷
(1) Command(1byte), Version(1byte), Reserved(2byte)
- Command: 명령 구분. 1이면 Request, 2명 Response.
- Version: RIP 버전. 1 또는 2.
(2) Family(2byte), Route Tag(2byte)
- Family: 프로토콜 정보. IP 사용 시 2가 된다.
(3) IP Address: 목적지 주소
(4) Subnet Mask
(5) Next Hop
(6) Distance: 목적지까지의 hop count

5-2-2) RIP의 동작

(1) 요청 메시지 발송 방식

- 라우터가 초기화(전원 on 포함) 또는 특정 라우팅 테이블의 기간 만료 시 메시지 발송
- 특정 네트워크 주소 또는 전체 라우팅 정보를 요청

(2) 응답 메시지 발송 방식

- 요청 메시지 수신 시 또는 주기적(30초)으로 자신의 라우팅 정보를 전파
- 일정 시간(180초) 특정 경로에 대한 응답이 없으면 hop count 16으로 설정 -> 폐기

(3) RIP 메시지 수신 시

- 신규 메시지일 경우 -> 라우팅 테이블에 추가한다.
- Next Hop 정보가 변경된 경우(더 좋은 경로 발견 시) -> 해당 값 변경
- Next Hop이 같은 경우 Hop Count 비교 -> 현재 값보다 작으면 변경, 아니면 무시

5-3) OSPF (Open Shortest Path First)

- Link State 라우팅 알고리즘을 사용하는 IGP용 라우팅 프로토콜
  ㄴ 규모 있는 단체/회사는 대부분 OSPF를 사용하고 있다.
- RIPv1의 단점을 보완
  ㄴ hop count 제한 없음 -> 대규모 네트워크에서 활용 가능
  ㄴ VLSM(Variable-Length Subnet Mask) 사용 -> 서브넷 마스크를 통한 효율적 IP관리
  ㄴ 변경된 정보만 전파 -> 적은 양의 라우팅 트래픽 유발
  ㄴ 라우터 hop 대신 링크의 속도(대역폭)까지 확인하여 경로 설정
  ㄴ Link State 알고리즘 -> 전체 토폴로지가 한 번에 업데이트되므로 Convergence Time이 빠르다.
    ㄴ Convergence Time: 토폴로지 변화가 전체 네트워크에 반영되기까지의 시간
- CPU 부하가 크고 메모리 소모가 많다는 단점이 있다.
- 계층적 구조로, 여러 개의 Area를 나누어 각 영역은 독립적으로 라우팅을 수행한다.

5-3-1) OSPF에서 사용되는 요소들

- LSDB(Link State Database): 각 OSPF Area 내의 전체 망 정보, 링크 상태와 경로 정보
  ㄴ 이를 바탕으로 경로를 설정한다.
- LSA(Link State Advertisement): 라우팅 기초 정보가 담긴 패킷. 각 라우터들이 링크 상태, 인접 관계, 요약 정보(네트워크/링크의 경로 비용 포함)를 교환할 때 사용된다.
- DR(Designated Router): 중복되는 LSA 교환을 방지하기 위해 Area당 하나씩 선출된다.
  ㄴ BDR(Backup DR): DR과 같은 역할로 선출되며 동일한 정보를 공유받는다. DR이 일정 시간동안 제대로 응답하지 않으면 BDR이 DR로 선출된다.
  ㄴ 선출 기준은 OSPF Priority(사용자가 설정 가능) > Router ID 순.
  ㄴ DR/BDR은 다른 라우터들과 LSA정도를 교환하여 인접 네이버 관계를 형성한다.
  ㄴ DR/BDR 외의 라우터들은 LSA 정보 교환 없이 Hello만 교환하고 네이버(neighbor) 관계를 형성한다. 관계 형성 시 DR에게 보고한다.

5-3-2) OSPF 메시지 종류
- 메시지는 인접 라우터 발견 및 관계 유지, 멀티캐스트에 사용된다.
 ㄴ Protocol ID는 89를 사용
 (1) Hello
 - 인접 라우터 및 로컬 링크 상태를 검색, 관계를 설정하고 주요 매개변수를 전달한다.
 - 일정 간격으로 Hello 메시지를 보내서 상태를 확인(Keepalive)한다.
 (2) DBD (DataBase Description)
 - OSPF 정보 구축을 위해 LSDB 내용을 전달한다.
 (3) LSR (Link State Request)
 - 상대 라우터에게 링크 상태 정보를 요청한다.
 (4) LSU (Link State Update)
 - 네트워크 변화 발생 시 인접한 라우터에게 상태를 전달한다.
 (5) LSAck (Link State Acknowledgement)
 - 패킷을 정상적으로 받았다는 수신 확인 메시지. 신뢰성 확보를 위해 사용.

5-3-3) OSPF 테이블 종류
 (1) OSPF 네이버(neighbor) 테이블
 - 네이버를 성립한 인접 라우터들의 정보를 라우터 ID를 통해 관리한다.
 (2) OSPF DB 테이블
 - 네이버에게 수신한 라우팅 업데이트 정보를 관리한다.
 - LSA 메시지를 이용해 LSDB를 동기화하고, 이를 기반으로 최적 경로를 선출한다.
 (3) 라우팅 테이블
 - 최적 경로를 등록한다.
 - Area 내의 라우팅 정보, 다른 Area의 업데이트 정보, 외부 AS의 업데이트 정보가 명시되어 있다. 이러한 정보를 취합하여 패킷의 최적 경로를 선택하게 된다.

5-3-4) 네이버 테이블의 상태 변화
 (1) Down: 전원 off, 작동하지 않는 상태
 (2) Init: 인접 라우터에게 Hello 메시지 전송
 (3) 2way: 서로에게 Hello 메시지를 보내서 정보(라우터 ID)를 교환
 - 서로를 Neighbor로 확인하고, Neighbor List에 업데이트한다.
 - DR(Designate Router)에게 Hello 메시지로 이 내용을 보고한다.
 (4) ExStart(실행): DBD 메시지를 통해 마스터/슬레이브를 선출한다.
 - 마스터는 Request를, 슬레이브는 Response 전송하게 된다.
 (5) Exchange(교환): DBD 메시지를 교환하여 링크 상태 정보를 교환한다.
 (6) Loading(전송): LSR, LSU, LSAck 메시지를 통해 라우터 테이블을 로딩한다.
 (7) Full: 로딩이 정상적으로 끝난 상태. 인접한 라우터들의 정보를 유지한다.

5-3-5) 링크 종류
 (1) Point to Point: 라우터와 라우터가 1:1로 직접 연결
 (2) Transient: 여러 개의 라우터가 하나의 Area 내에서 버스를 통해 연결

(3) Stub: 하나의 Area에 1개의 라우터만 연결
(4) Virtual: 가상으로 연결 (물리적 연결이 어려운 상황에서 사용)

6. 전송(Transport) 계층

1) 역할
- OSI 4계층으로 End to End의 연결 지향(Connected-oriented) 서비스.
 ㄴ IP 통신은 Host to Host
- TCP, UDP를 가장 많이 사용한다.
- 소켓을 통해 프로세스별로 통신이 가능하다.
 ㄴ 5 tuple(Source IP, Source Port, Dest IP, Dest Port, Protocol) 이라는 다섯 가지 정보를 통해 연결을 수행한다.
- Port: 특정 프로세스를 구분하는 단위. 0~65535번으로 구분된다.
 ㄴ 0~1023: well-known port. 표준 서비스들이 정의되어 있다.
  ㄴ 예를 들어 웹 연결 포트는 80 이다.
 ㄴ 1024~49151: registered port. 특정 프로토콜이나 응용 프로그램들이 사용한다.
 ㄴ 49152~65535: dynamic port
 ㄴ 윈도우/리눅스의 콘솔 창에 netstat -an 입력하여 자신의 프로세스별 포트와 연결된 외부 주소를 확인할 수 있다.

* IP통신 ↔ TCP/UDP 구분 ↔ Port ↔ Process

2) TCP (Transmission Control Protocol)
- 전송 제어 프로토콜.
- 연결 지향(Connected-oriented), 인터넷을 구성하는 핵심 프로토콜.
- 1:1 통신을 하며, 신뢰성을 기반으로 데이터를 에러 없이 전송한다.
 ㄴ 패킷의 상태 정보를 확인하며, 에러 발생 시 재전송을 요청하고 에러를 복구한다.
 ㄴ Segment는 TCP Header + Data로 구성 (Packet은 IP Header + Segment)
- Protocol ID는 6이며 1981년 릴리즈되었다. (RFC 793)

2-1) TCP 헤더 포맷
- 4바이트 5줄로 최소 20바이트 (+Option/Padding)
 (1) Source Port(2byte), Dest Port(2byte)
 - 출발지와 목적지의 포트
 (2) Sequence Number: 데이터의 순서 번호
 - 중복 패킷을 방지하며, 여러 개로 나뉜 패킷의 재조립에 필요하다.
  ㄴ 패킷을 보낸 순서와 도착하는 순서가 반드시 일치하지 않음
 (3) Acknowledgement Number: 승인 번호
 - 수신 측에서 수신을 확인하고, 다음 송신 데이터를 요청할 때 사용한다.
 - 어떤 수신에 대한 응답인지 구분하기 위해, 받은 신호의 Sequence Number + 1 값을 넣

어 보낸다.
(4) HLEN(1byte), Reserved(1byte), Window Size(2byte)
- HLEN(Header Length): 헤더 길이. 최소20 ~ 최대60 바이트.
- Reserved: TCP 제어 플래그. TCP 회선을 제어하고 관리하는 중요한 부분.
  ㄴ URG, ACK, PSH, RST, SYN, FIN으로 구성되며 활성화된 비트는 1로 표현한다.
  ㄴ URG: 긴급. 우선 순위를 높여 먼저 송신
  ㄴ ACK: 확인. 수신자가 송신자에게 패킷을 정상적으로 받았다고 알림
  ㄴ PSH: 버퍼링 없이 바로 송신하라는 플래그
  ㄴ RST: 비정상 상황에서 연결 끊기
  ㄴ SYN: 연결을 맺기 위해 처음 보내는 패킷
  ㄴ FIN: 정상 종료. 송신측이 수신자에게 연결 종료 요청 (데이터를 다 보냈을 때 등)
- Window Size: 데이터를 처리할 수 있는 버퍼 크기. 수신 버퍼의 여유 용량을 통보한다.
(5) Checksum(2byte), Urgent Pointer(2byte)
- Checksum: 데이터 무결성 확인용 값. (데이터 위/변조 방지)
- Urgent Pointer: 긴급 데이터를 알림
(6) Option, Padding: 옵션. MSS(Maximum Segment Size), 타임스탬프 등.

2-2) TCP 통신 과정
- 연결 방식: 3-way handshake
ㄴ TCP는 연결 지향 프로토콜로, 두 호스트가 통신하기 전에 연결을 위한 관계를 수립한다.
(1) 클라이언트가 서버에게 SYN 신호 전송
- 신호를 주는 곳은 SYN_SENT 상태, 서버는 신호를 받으면 SYN_RCVD 상태가 된다.
(2) 서버가 클라이언트에게 SYN + ACK 신호로 응답
- 이 때 승인 번호(Acknowledgement Number)에 (1)에서 받은 Sequence Number + 1 값을 넣어서 보낸다. 어떤 신호에 대한 응답인지 구분하기 위함.
- 응답을 받은 클라이언트는 ESTABLISHED 상태가 된다.
(3) 클라이언트가 서버에 ACK 신호 전송
- 승인 번호(Acknowledgement Number)는 (2)에서 받은 Sequence Number + 1
- 연결 종료: 4-way handshake
(1) 클라이언트가 서버에 FIN 신호 전송
- 클라이언트는 FIN_WAIT_1 상태
(2) 서버가 클라이언트에 ACK로 응답
- 서버도 CLOSE_WAIT 상태가 된다.
- 응답을 받은 클라이언트는 FIN_WAIT_2 상태 (서버가 종료시켜주길 기다리는 상태)
(3) 종료 절차를 끝낸 후 서버가 LAST_ACK 상태가 되며 FIN 신호 전송
(4) 신호를 받은 클라이언트는 TIME_WAIT 상태가 되며 ACK 신호 전송
- ACK 신호를 받은 서버는 CLOSED 상태가 되고 완전히 연결이 종료된다.

2-3) TCP 타이머
(1) Retransmission
- 송신 측이 패킷을 전송할 때마다 카운트

- 특정 시간(RTO, Retransmission Time Out) 내에 ACK 응답이 오지 않으면 재전송
ㄴ RTO는 RTT(Round Trip Time)에 따라 가변적으로 변한다. (네트워크 상태 등에 대응)
ㄴ RTO는 샘플링된 RTT값인 SRTT(Smoothed RTT), RTT의 변동 계수인 RTTVAR(RTT Variation), alpha(1/8), beta(1/4), R(측정된 RTT값), G(clock granularity)으로 계산한다.
ㄴ RTTVAR는 (1 - beta) * RTTVAR + beta * |SRTT-R| 로 계속 업데이트된다.
ㄴ SRTT는 (1 - alpha) * SRTT + alpha * R 로 계속 업데이트된다.
ㄴ RTO = SRTT + max(G, 4*RTTVAR)
ㄴ 이에 관해서는 https://tools.ietf.org/rfc/rfc6298.txt 참고
(2) Persistence
- 윈도우(버퍼의 여유 공간) 사이즈 용량 부족 시 주기적 재전송 요청을 위한 타이머
- 수신측이 용량이 부족할 때 알리고, 용량에 여유가 생기면 다시 송신측에 요청한다.
- 수신측이 용량 부족(윈도우 사이즈 = 0)을 보낼 경우 송신측에서는 Persistence 타이머가 가동된다.
ㄴ 용량에 여유가 생겼으니 다시 보내달라는 ACK 신호가 중간에 유실되어 송신측이 무한히 대기하는 데드락 상황 방지
ㄴ Persistence 타이머가 종료되면 Probe(ACK 재전송 요청)를 보내고 타이머 재가동
ㄴ 다시 타이머가 종료되면 시간을 2배로 늘리고(최대 60초) Probe 재전송
(3) Time waited
- TCP 연결 종료 후 특정 시간동안 연결을 유지하기 위한 타이머
ㄴ 지연되는 패킷을 기다리는 시간
ㄴ 일반적으로 MSL(Maximum Segment Lifetime, 120초)의 2배 정도 기다린다.
- 다른 연결이 맺어진 상태에서 이전 연결의 패킷이 뒤늦게 오는 문제 방지
(4) Keepalive
- TCP 연결 유지 타이머
- TCP 연결을 맺은 후 수신측이 2시간동안 패킷을 못 받으면 수신측에서 75초 단위로 Probe를 전송한다.
- Probe 9개를 보낼 때까지 응답이 없으면 연결 종료, 응답이 있으면 타이머 재설정

2-4) 흐름 제어 (Flow Control)
- 송/수신 측의 데이터 처리 속도 차이 해결 (느린 쪽에 맞추어야 함)
- Sliding Window 기법: 버퍼(TCP Data + 여유 공간인 Window로 구성)에서 TCP Data가 처리되면, 그만큼 Window Size를 늘리는 방식.

2-5) 혼잡 제어 (Congestion Control)
- 수신측으로 유입되는 트래픽의 양이 정해진 대역폭을 초과하지 않게 제어
(1) AIMD (Additive Increase/Multiplicative Decrease)
- 패킷 전송 시 문제없으면 Window Size를 1씩 증가시킨다.
- 타임아웃 또는 loss시 패킷 속도를 1/2로 감소시킨다.
- 초기에 높은 대역폭을 사용할 수 없고, 혼잡 상태를 미리 감지할 수 없다는 단점이 있다.
(2) Slow Start
- 패킷 전송 시 문제없으면 Window Size를 2배씩 증가시킨다.

- 혼잡 상태 발생 시 Window Size를 1로 변경하고 발생 시점을 기록한다.
ㄴ 이후에는 기록된 혼잡 상태에서의 Window Size 절반까지만 Window Size를 2배씩 증가시키고, 이를 초과하면 1씩 증가시킨다.
(3) TCP Tahoe
- Fast Retransmit 알고리즘 사용
- 수신측에서 먼저 와야 하는 패킷이 오지 않고 다음 패킷이 올 경우에도 ACK를 보낸다.
- 송신측은 타임아웃 시간을 기다리지 않고 중복된 순번의 패킷을 3개 받으면 재전송한다.
- 개선된 버전으로 Fast Recovery 알고리즘을 추가로 사용하는 TCP Reno, TCP New Reno, TCP SACK, TCP Vegas 등이 있다.

3) UDP (User Datagram Protocol)
- 신뢰성이 낮으나 데이터 전송이 빠르다.
- 1:다수 통신 가능. 송신측은 데이터를 보내고 확인하지 않으며 재전송도 불가능하다.
- 비연결형(Connectionless), 실시간 데이터 전송(스트리밍 서비스 등)에 적합
ㄴ 예를 들어, 스포츠 경기 생중계 등은 전송 문제가 발생해도 끊어진 부분을 다시 보내주는 것보다, 현재 데이터를 실시간으로 전송하는 것이 더욱 중요하다.
- Protocol ID는 17이며 1980년 릴리즈되었다. (RFC 768)

3-1) UDP 헤더 포맷
- 8바이트로 구성
(1) Source Port(2byte), Dest Port(2byte)
- 출발지와 목적지 포트
(2) Length(2byte), Checksum(2byte)
- Length: 전체 데이터 길이 (헤더 + 데이터)
- Checksum: 데이터 무결성 확인용 값

3-2) TCP와 UDP 비교

	TCP	UDP
Protocol ID	6	17
헤더 길이	최소 20바이트	8바이트
순서 확인	가능	불가능
신뢰성	높음	낮음
통신 방식	1:1	1:n
연결성	Connection-oriented	Connectionless
제어	흐름/혼잡 제어 가능	관련 기능 없음
속도	느림	빠름

4) NAT (Network Address Translation)
- 네트워크 주소 변환. 사설 IP를 라우팅 가능한 공인 IP로 변환한다.
ㄴ 3계층 이상의 장비 또는 방화벽에서 처리 가능

- 보안: 내부 IP 주소를 외부에 공개하지 않음
- 유연성: 공인 IP 대역에 영향을 주지 않고 내부 네트워크 구성 변경 가능
- 비용: 공인 IP 할당에는 주기적으로 비용이 발생하므로, 내부에선 사설 IP를 쓰고 외부와 통신할때만 공인 IP를 사용함으로써 공인 IP 할당 비용 감소

4-1) 공인 IP와 사설 IP
- 공인 IP: 공인기관(ICANN)에서 인정하는 IP주소. 인터넷을 통한 외부망에서 식별되고 통신 가능한 IP
 ㄴ http://ipconfig.kr/에서 자신의 PC가 외부와 통신 시 사용하는 공인IP 확인 가능
- 사설 IP: 내부망에서 사용 및 식별 가능한 IP. IPv4 개수의 한계로 등장했다.
 ㄴ 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 범위 내에서 사용하도록 정해져 있다.
 ㄴ 콘솔 창에서 ipconfig 또는 ifconfig 명령어로 확인 가능

4-2) NAT 종류
 (1) Static NAT
 - 사설 IP 1개를 공인 IP 1개로 매핑하는 1:1 NAT
 - 주로 외부 공개형 서버에 구성한다.
 (2) Dynamic NAT
 - 내부 IP 주소와 외부 IP 주소가 일정 범위 내에서 매핑되는 n:n NAT
 - 내부 PC들은 외부로 통신 시 공인 IP 대역(NAT Pool) 내에서 하나를 할당받게 된다.
 (3) PAT (Port Address Translation)
 - 여러 개의 내부 사설 IP들이 1개의 공인 IP로 변환되는 1:n NAT
 - 공개형 서버가 아닌 내부 -> 외부로 접속이 필요한 PC들이 사용한다.
  ㄴ IP가 중복되므로 Port로 세션을 구분한다.
 (4) 포트포워딩 (Port Forwarding)
 - 공인 IP 1개로 여러 대의 사설 IP를 포트로 구분하여 연결
 - PAT와 유사한 방식이나 공인 IP 1개로 여러 대의 공개형 서비스 구축 시 사용한다.
  ㄴ 예를 들어 포트 번호 80이면 내부의 서버들 중 웹서버로 연결하는 식

4-3) Hairpin NAT
- NAT 장비에서 출발지 IP를 NAT 장비에 매핑된 IP로 변경한다.
- PC가 공인 IP로 목적지 서버에 접속했는데 알고 보니 목적지가 내부(동일 네트워크 내)에 존재하는 경우 발생하는 이슈에 대한 해결책
 ㄴ 서버는 동일 대역 IP로 확인하여 (NAT를 거치지 않고) 바로 PC에게 응답 패킷을 보낸다. 이 경우 PC는 기존 커넥션이 아닌 신규 패킷으로 판단하여 통신이 불가해진다.
- Hairpin NAT를 사용하면 서버에게 전달되는 출발지 주소는 PC가 매핑된 NAT의 IP이므로, 응답 패킷은 NAT로 전송된 후 PC에 전달된다.

5) 텔넷(TELNET)
- 원격 호스트 컴퓨터에 접속하기 위해 사용되는 프로토콜
 ㄴ 장비 관리 또는 서버 접속 시 사용된다. (CLI / Shell 환경)

ㄴ 윈도우 cmd나 리눅스 터미널에서 접속 가능. 무료 오픈소스 Putty를 많이 사용한다.
- 클라이언트의 경우 포트 테스트 용도로 사용한다. ('telnet 주소 포트번호' 형태)
ㄴ ex) 콘솔 창에 telnet www.naver.com 80 입력 시 정상 연결 가능 확인됨
ㄴ ex) 이상한 포트 번호로 telnet www.naver.com 30000 입력 시 연결되지 않음
- 데이터 변환 가상장치인 NVT(Network Virtual Terminal) 지원
- 프로세스와 터미널이 1:1 동기화 관계
ㄴ 클라이언트에서 보는 화면과 서버에서 보는 화면이 동일하다
- 협상 가능한 옵션이 있다. Negotiation Commands를 통해 제어할 수 있다.
(1) WILL / WON'T : 옵션 활성화를 원한다 / 원하지 않는다
(2) DO / DON'T : 옵션 활성화를 요청한다 / 요청하지 않는다
- Sender가 WILL을 보내면 Receiver는 DO/DON'T로 응답해야 한다.
- Sender가 DO를 보내면 Receiver는 WILL/WON'T로 응답해야 한다.

6) SSH (Secure Shell)
- 원격지에 있는 컴퓨터를 명령어를 통해 제어하며, 강력한 인증 방식과 암호화를 제공한다.
ㄴ TELNET이 보안에 취약한 점을 보완하여 대체하기 위해 1995년 개발, TCP 22번 사용.
ㄴ SSH 버전으로는 SSHv1, SSHv2가 있다
- 특징
(1) 인증(Authentication)
- 사용자가 서버 접속 시 패스워드 또는 공개 키 기반의 인증방식 지원
ㄴ 공개 키 방식: 공개키로 데이터를 암호화하고 개인키로 복호화하는 방식. 후술할 대칭키가 유출 위험성이 큰 점을 보완한다. 개인 키는 일종의 비밀번호로 유출되면 안 된다.
ㄴ 사용자 인증은 개인키를 통해 인증 데이터를 암호화하고 공개키를 전달하는 방식으로 이루어진다. 서버 측은 암호화된 데이터 + 공개키를 통해 신원을 확인한다. 전자서명에서 많이 사용된다.
(2) 암호화 (Encryption)
- 대칭키 방식 사용 (AES, Blowfish, 3DES 등의 암호화 알고리즘)
ㄴ 대칭키: 동일한 키로 암호화/복호화를 둘 다 할 수 있는 방식
ㄴ 상대의 공개키와 자신의 개인키를 사용하여 비밀 키를 생성하여 데이터를 암호화한다.
(3) 무결성 (Integrity)
- 데이터 위변조 방지
ㄴ MAC(Message Authentication Code)을 사용하여 서로 맞춰보는 방식
(4) 압축 (Compression), 다중화 통신 등
- 통신 과정
(1) TCP 연결을 완료한다. (서버-클라이언트 모두 Established 상태)
(2) SSH 버전을 확인한다.
(3) 암호화 키 알고리즘을 협상하고, 키를 교환한다.
(4) 키가 교환되면 해당 키를 통해 암호화된 데이터를 교환한다.

7. 애플리케이션(Application) 계층

1) 역할
- TCP/IP 모델 최상위 계층
- 사용자에게 가장 가까운 소프트웨어로, 사용자와 응용 프로그램 간 인터페이스를 제공한다.
 ㄴ HTTP, DNS, SSH, SMTP, BGP, DHCP 등이 속한다.

2) DNS (Domain Name Service)
- 도메인 주소를 IP 주소로 변환해주는 서비스
 ㄴ L2~L3 계층에서 MAC과 IP주소를 변환해주는 ARP와 유사한 역할.
- 계층적 구조: Root DNS(전세계적으로 13개) -> Top Level(kr, com, org 등) -> Second Level (naver, google 등) -> Sub Level (www, mail, search 등)의 계층으로 나뉜다.
 ㄴ ex) search.naver.com = com + naver + search
- 쿼리 과정: Recursive Query. Local DNS 서버가 재귀적으로 여러 서버에게 질의하여 응답을 받는다.
 ㄴ ex) www.naver.com의 IP 요청 시
 -> Local DNS 서버가 모르는 내용 -> Root DNS에게 문의
 -> .com에 속한다고 응답 -> com DNS에게 문의
 -> naver.com에 속한다고 응답 -> naver.com DNS에게 문의
 -> 해당 서버에는 www.naver.com에 대한 정보가 있으므로 매핑된 IP주소 반환
 -> Local DNS 서버가 받아서 PC에 전달
 ㄴ 여러 군데에 반복적으로 질의한다는 점에서 'Iterative Query'라고도 부른다.
- Resource Record: DNS 레코드. DNS 서버가 가지고 있는 IP 매핑 정보 테이블
 ㄴ Name, Value, Type, TTL(Time To Live)로 구성된다.
 ㄴ Type은 A(호스트), NS(네임서버), CNAME(별칭), MX(메일서버)의 4가지가 있다.
- DNS 메시지: 쿼리(Query)와 응답(Response)의 2가지가 있다.
 ㄴ Query: Header + Question
 ㄴ Response: Header + Question + Answer + Authority + Additional
 ㄴ 헤더는 Identifier(쿼리/응답 구분), Flag(DNS 쿼리 속성), Question 개수, Answer 개수, Authority 개수, Additional Records 개수의 6가지로 구성된다. (각 2바이트)
- Hosts.txt: 호스트 이름과 IP주소가 매핑되어 저장된 파일
 ㄴ Local DNS에 질의하기 전 우선적으로 이 파일에 매핑 정보가 있는지 확인한다.
 ㄴ 윈도우 기준 C:\windoes\system32\drivers\etc\hosts 경로에 있다.
- DNS 캐시 테이블: 기존에 질의하여 응답받은 DNS 정보를 일정시간(TTL) 저장해두는 것.
 ㄴ 웹 사이트 www.daum.net 로 접속한다고 가정할 경우 동작
 -> Hosts.txt 에 매핑 정보가 있는지 확인한다.
 -> DNS 캐시 테이블에 매핑 정보가 있는가 확인한다.
 -> 없을 경우 Local DNS에 질의한다.

3) HTTP (HyperText Transfer Protocol)
- WWW상에서 정보를 공유하는 프로토콜
 ㄴ 포트 번호 80, 버전은 1.0(1996), 1.1(1999), 2(2015)가 있으나 1.1을 가장 많이 사용

ㄴ WWW(World Wide Web): 전 세계에 연결된 인터넷망. 1989년 팀 버너스 리의 WWW 프로젝트로 제안되었다.
- HTML(HyperText Markup Language): 웹 페이지 언어
- URL(Uniform Resource Locator): 웹 페이지를 찾기 위한 주소
ㄴ ex) http://www.naver.com:80/index.html (naver.com만 입력해도 기본값이 들어감)
- 간소화한 웹 통신 형태: [웹 브라우저 -> HTTP Request -> 웹 서버 -> Select, Update -> DB 서버 -> Response -> 웹 서버 -> HTTP Response -> 웹 브라우저]
ㄴ 웹 브라우저: Chrome, IE, Opera, Firefox, Safari 등
ㄴ 웹 서버: Apache(리눅스 계열), Microsoft IIS 등
ㄴ DB 서버: MySQL 등

3-1) HTTP Request
- 클라이언트가 서버에 특정 함수를 사용하여 요청한다.
ㄴ 크게 Head와 Body로 구성된다.
- Start Line: 첫 번째 줄. HTTP Method + Request Target + HTTP Version로 구성.
(1) Start Line
- HTTP Method: 요청의 목적
ㄴ GET: 리소스 요청
ㄴ POST: 내용 전송
ㄴ PUT: 내용 갱신
ㄴ HEAD: 리소스 내용이 아닌 리소스에 대한 정보만 요청
ㄴ DELETE: 리소스 제거
- Request Target: 리소스 경로. 웹 서비스도 디렉토리 구조로 되어 있다.
(2) Head
- Accept: 클라이언트가 허용 가능한 파일 형식 (텍스트, 이미지 등)
- Uset-Agent: 클라이언트의 OS, 브라우저 정보
- Host: 서버의 도메인 이름

3-2) HTTP Response
- 클라이언트 요청에 따른 서버의 응답
ㄴ 마찬가지로 Head + Body로 구성된다.
- Start Line은 HTTP Version + Status + Status Message + Date + Content-location + etag + Last-modified(사이트의 최종 수정일) + Content-Length 등으로 구성된다.
ㄴ etag: 캐시 정보 업데이트
- Status Code: 요청에 대한 처리 결과
ㄴ 200번대: 성공(Success)
ㄴ 300번대: 리다이렉션(Redirection)
ㄴ 400번대: 클라이언트 에러
ㄴ 500번대: 서버 에러

3-3) HTTP 쿠키 (Cookie)

- 클라이언트 웹 브라우저 로컬에 저장되는, 키와 값이 들어있는 파일
 ㄴ 크롬에서 F12 - Application탭 - Cookies에서 확인 가능
- 세션(Session): 일정 시간 내 같은 웹브라우저의 요청이 들어오면 하나의 상태로 유지
 ㄴ 클라이언트에 세션ID를 발급하여 쿠키로 전달하고, 일정 시간 내에 동일한 세션ID로 접속하면 해당 ID에 매핑된 정보를 확인하여 관련 서비스를 제공한다.
- HTTP 속성 - Stateless: 통신이 끝나면 상태 정보를 유지하지 않는다.
 ㄴ 요청에 대한 응답을 보낸 후 접속을 끊어서, 커넥션 리소스 비용을 줄인다.
 ㄴ 새로운 페이지를 접속할 때마다 신원을 알 수 없음

3-4) SSL/TLS (Secure Socket Layer/Transport Layer Security)
- TCP/IP 네트워크 통신간 보안을 제공하는 암호화 프로토콜. HTTPS 구현에 사용된다.
 ㄴ HTTPS(HTTP Secure): 쿠키는 로컬에 저장되므로 유출 또는 조작에 취약하며, 세션은 세션 하이재킹을 통한 탈취 위험이 있어서 생긴 암호화 통신 프로토콜
- 기능
 ㄴ 인증: RSA, DSS를 사용한다.
 ㄴ 무결성: 해시 메시지 인증 코드(HMAC, Hash Message Authentication Code)로 MD5, SHA-2 알고리즘 사용
 ㄴ 기밀성: 데이터 암호화 알고리즘으로 3DES, RC4 사용
- TLS계층: 상위 3개, 하위 Record 프로토콜로 구분되며, 상위 계층에서 정책 협상 후 Record 프로토콜에서 Application 데이터를 분할, 압축, 암호화한다.
 (1) 상위 프로토콜
 - HandShake: 키 교환 방식, 암호화 방식, HMAC 방식, 압축 방식 등을 협상
 - Change Cipher Spec: 암호화 알고리즘 등 어떤 정책을 적용하게 되었는지 클라이언트와 서버에 알림
 - Alert: 협상 과정에서 상대가 제시한 암호화 방식을 내가 지원하지 못하는 경우 알림
 (2) 하위 프로토콜
 - 어떤 정책을 사용할지 결정한 이후 사용된다.
 - Record: 데이터 교환, 메시지 전송. Record 단위로 Plain Text를 암호화해서 보낸다.
- SSL/TLS 동작 과정
 (1) Client Hello
 - 클라이언트가 지원 가능한 암호화 알고리즘(cipher suite) 전달
 (2) Server Hello
 - 클라이언트가 지원하는 chiper suite를 확인하고 서버도 지원하는 chiper suite 중 적합한 것을 정해서 전달
 (3) 서버 인증서 전달
 - 서버 인증서(Certificate), DH키(ServerKetExchange), 인증서 요청(CertificateRequest) 등을 전달한다.
 (4) 클라이언트 인증서 전달
 - 인증서(Certificate), DH키(ClientKeyExchange), 인증서 확인(CertificateVerify) 등을 전달한다.
 (5) Change Cipher Spec

- 클라이언트와 서버가 공통된 암호화 알고리즘이 정해졌음을 서로에게 알린다.
(6) 통신 시작
- 데이터(Application Data)를 주고받기 시작한다.
- 기존의 Plain Text를 TLSCiphertxt로 암호화하여 주고받는다.

4) 메일 서비스
- Email(Electronic mail): 전자 메일. id@domain 형태로 사용한다.
ㄴ 웹메일, ERP 기업용 메일, 아웃룩(outlook) 등이 있다.
ㄴ 1973년 등장(RFC561) -> 1982년 현재 쓰이는 방식인 SMTP(RFC821) 릴리즈
- 사용 프로토콜
ㄴ SMTP(Simple Mail Transfer Protocol): 메일 발신 프로토콜
ㄴ POP3(Post Office Protocol Version3): 서버에서 메일을 가져오거나 삭제하는 프로토콜
ㄴ IMAP4(Internet message Access Protocol4): 중앙서버에서 메일을 관리하여, 접속하여 확인할 수 있는 프로토콜

4-1) SMTP (Simple Mail Transfer Protocol)
- 전자 메일 전송을 위한 표준 프로토콜. 기본적인 클라이언트-서버 통신에 사용된다.
ㄴ TCP 25 - RFC 821, 2821에 명시되어 있음
- SMTP 명령어를 통해 명령을 전송하고 수행하는 방식으로 통신한다.
ㄴ HELLO: 인사, 세션 초기화
ㄴ MAIL: 메일 전송 시작, 송신자 이름 전송
ㄴ RCPT: 수신자 이름 전송
ㄴ DATA: 데이터 전송 시작
ㄴ QUIT: 세션 종료
- 서버는 SMTP 응답을 통해 상태를 알리거나 명령 수행 결과를 전달한다.
ㄴ 220: 세션 준비
ㄴ 221: 세션 종료
ㄴ 250: 요청한 명령 정상 수행됨
ㄴ 421: 서비스 불가
ㄴ 450: 다른 프로세스에 의해 접근이 불가함
ㄴ 500: 명령이 잘못됨
ㄴ 551: 사용자 요청이 잘못됨
- 통신 예시
(1) 클라이언트가 서버에 TCP25로 접속, 서버는 220으로 서비스가 준비됨을 알림
(2) 클라이언트는 MAIL + myAddr@naver.com 으로 내가 메일을 보낼 것이라고 알림
(3) RCPT + othersAddr@naver.com 으로 상대 주소를 알림
- 만약 없는 주소라면 거절(550 reject) 응답을 보낼 수도 있다.
(4) 클라이언트는 메일 내용을 보내며 DATA 신호 전송
- 서버는 354 start mail input으로 응답한다.
(5) 메일 전송이 끝나면 클라이언트는 QUIT 전송, 서버도 221 Bye로 응답.

4-2) POP3 (Post Office Protocol Version3)
- 수신서버의 메일 박스에서 메일을 가져오고 삭제하는 프로토콜. 회사에서 쓰는 outlook 등의 메일 클라이언트 프로그램에서 사용한다.
 ㄴ TCP 110 - RFC 1939, 2449에 명시
- POP3 명령어
 ㄴ USER: 사용자 ID
 ㄴ PASS: 사용자 패스워드
 ㄴ STAT: 서버 상태
 ㄴ LIST: 메시지 리스트와 크기 확인
 ㄴ DELE: 메시지 삭제
 ㄴ QUIT: 연결 종료
- POP3 응답은 두 가지뿐이다.
 ㄴ +OK: 정상
 ㄴ -ERR: 에러

4-3) IMAP4 (Internet message Access Protocol4)
- 메일서버로 접속하여 메일을 읽거나 삭제하는 프로토콜. 네이버, 다음 등의 포털 메일에서 사용한다.
 ㄴ TCP 143 - RFC 3501에 명시
- 장점: 원하는 메일 메시지만 전송, 다중 접속 가능, 보관함 연동 가능, 자유로운 삭제 가능
- 단점: 메일을 당겨오는 것이 아니므로 모든 사용자 메일을 서버에 보관하고 있어야 한다. 즉 메일 서버의 자원 사용률이 높아진다.
- IMAP4 명령어
 ㄴ LOGIN: 사용자 접속
 ㄴ SELECT INBOX: 메일 박스 선택
 ㄴ FETCH: 리스트 보기
 ㄴ UID FETCH: 메시지 가져오기
 ㄴ STATUS: 메일 박스의 상태 확인
- 메일 서버가 다를 경우에는 메일 서버 간 전송한다.
 ㄴ MTA(Mail Transfer Agent): 메일 전송
 ㄴ MUA(Mail User Agents): 메일 송수신 프로그램
 ㄴ MDA(Mail Delivery Agent): MUA가 수신한 메일을 수신자 우편함에 기록
 ㄴ MRA(Mail Retrieval Agent): 서버의 우편함에서 사용자에게 메일을 가져옴