물리계층과 데이터 링크 계층의 장비

→ 네트워크 장비

데이터 링크 계층 — NIC, Switch < 전이중통신 / VLAN

물리 계층 — T cable / L.S cable / Herb < 반이중통신 / CSMA/CD

- 물리계층에는 주소 개념이 없다.
  - 호스트 ~ 통신매체 / 통신매체 ⇄ 통신매체 (수신 / 송신)
  - 정보에 대한 조작 및 판단 X

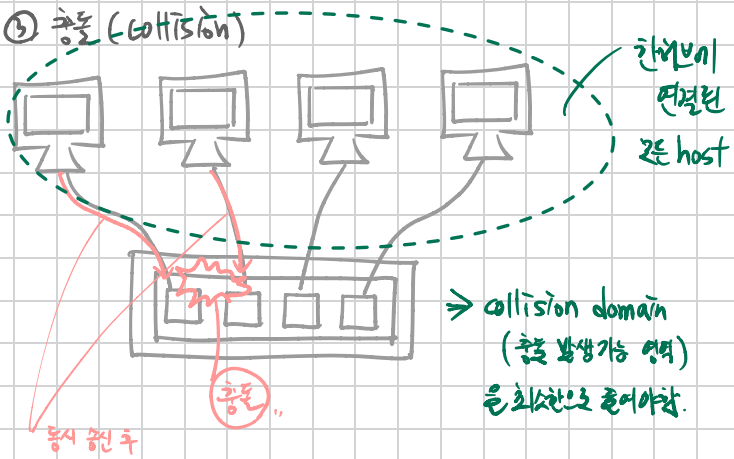
VS

- 데이터링크 계층에는 주소 개념이 있다.
  - MAC 주소.
  - 데이터링크 이상 계층 < 송수신지 특정가능 / 송수신정보 조회가능.

- 허브 (Herb)
  - ① 받은 정보는 모든 포트로 내보냄. (주소개념이 없기 때문)
  - ② 반이중 통신
    - 반이중 (송신 / 수신 일방향.) → 현재 허브에서 하는 통신
    - 전이중 (송수신 양방향) → ex) 전화통화.

  cf) 리피터 (신호증폭.)

- ③ 충돌 (Collision)



⇒ collision domain (충돌 발생가능 영역) 을 최소한으로 줄여야함.

collision 방지 < CSMA/CD / 다른장비 (스위치)

- CSMA/CD ; 반이중 이더넷 네트워크 충돌방지 프로토콜.
  (Carrier Sense Multiple Access With Collision Detection)
  - ① 캐리어 감지
    - 통신매체 사용가능 여부 감지 (현재 전송 중 데이터 확인)
  - ② 다중접근
  - ③ 충돌검출
    - 전송 중단 → 충돌발생 알림신호 (Jam signal)

- 스위치

; 허브와는 달리 특정 MAC주소를 가진 호스트에만 전달 가능.

; 전이중 모든 통신 지원. (CSMA/CD protocol 불필요)

*기능 ① MAC주소 학습

- 특정포트와 해당 포트에 연결된 MAC 주소와의 관계 초기화.
- 찾아는 호스트에만 프레임을 전달. (포트에 할당된 MAC정보 참조.)

1) 플러딩 (초기에는 모든 호스트로 신호송출 → 특정 MAC주소 응답시 응답신호송출)
2) 에이징 (일정시간 특정 포트에서 신호 통신이 되지 않으면, 학습된 MAC 주소 제거.))

② VLAN (Virtual LAN)

- 한 스위치에서 broad cast domain을 분할하여 여러개의 스위치가 있는 것처럼 동작하는 기능.
  - port 할당 (port의 물리적 위치에 따른 VLAN 할당.)
  - MAC 할당. (사전 학습된 MAC 주소에 따른 VLAN 할당.)

[네트워크 계층] ; LAN을 넘어 통신하기

* 데이터 링크 계층의 한계

- 다네트워크 오당경로 (최단거리) 파악 난이도 ↑ … ㉠
- 모든 네트워크에 속한 모든 호스트의 위치를 특정하기 어려움. … ㉡

㉠ 경로문제해결

- routing : 패킷이 이동할 최적의 경로 결정
- router : routing을 수행하는 대표적인 장비

㉡ 주소문제해결

; (MAC주소 + IP주소) ← 기본적으로 IP주소 주로 활용.

* NIC 할당받을때 MAC주소 변동 가능성 (특히 VM에서.)

- IP (Internet protocol)

(IPv4 / IPv6)

* 기능
  - 주소지정 (IP addressing)
  - 단편화 (IP fragmentation.)

cf) RFC
네트워크 / 인터넷 관련 신기술제안, 의견 기록 문서

① 주소지정 (IPv4 기준.)

- IP주소를 바탕으로 송수신 대상 지정.
- 4byte (32bit) / 숫자만 허용. (점으로 옥텟 (octet) 구분.)

② 단편화

- 전송하고자 하는 패킷의 크기를 MTU 이하의 복수의 패킷으로 나누는 것.

· 한번에 전송 가능한 IP 패킷의 최대 크기
· 헤더도 MTU에 포함.
· 일반적인 크기는 1500 byte.

- IPv4 패킷구성.

① 식별자 : 패킷 할당 번호 (to find orginal message before fragmentation)
(identifier)

② 플래그 (flag)

DF (Don't Fragment)
MF (More Fragment)

↓ 고정

③ 단편화 offset (fragment offset)

: 단편화시 데이터의 순서정보 (수신 순서는 단편화 데이터 순서와 상이할 수 있기 때문.)

④ TTL (Time To Live)

: 패킷의 수명 ← 무한히는 패킷이 네트워크 상에 잔존 방지 (트래픽 과부하 방지)

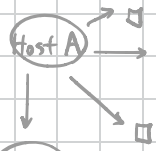
(hop마다 -1) (hop : 

⑤ protocol (상위계층 프로토콜 정보 ex) TCP : 6 / UDP : 17)

⑥ 송신지 / 수신지 IP주소

- IP주소로 MAC 주소 알아내기 (ARP) Host A ↔ Host B

① ARP 요청



; 동일 네트워크 상의 호스트에게 Broad cast Message 발송

② ARP 응답.

UniCast Message (특정 단일 호스트에만 응답하는 메세지.)

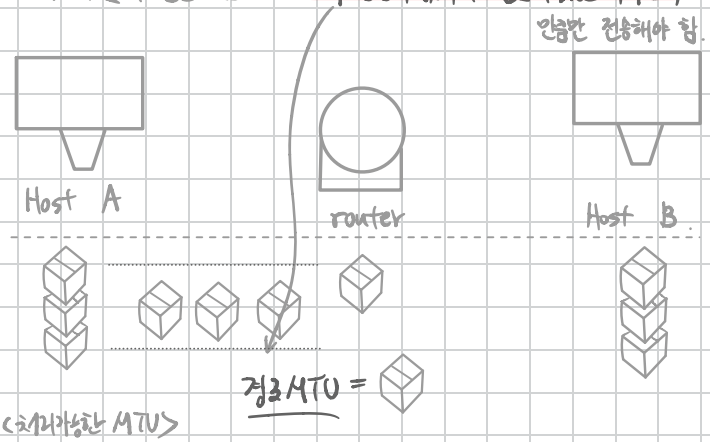
③ ARP 테이블 갱신

ARP Table : ARP 요청 - 응답을 통해 알게된 IP주소와 MAC주소의 연관관계

cf) Terminal (arp -a)

- Voiding Fragment.

: IP 패킷을 주고받는 모든 노드가 IP 단편화 없이 주고 받을 수 있는 최대 크기

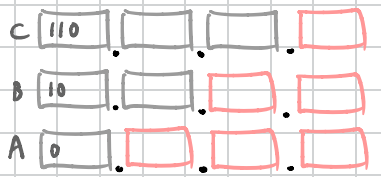


<최소허용한 MTU>

- IP주소 지정.

[네트워크 주소 + 호스트 주소]

클래스 : 네트워크 크기에 따른 IP주소 분류 기준. (A.B.C)



	이론상 할당 N	H
	2^{21}	2^8
	2^{14}	2^{16}
	2^7	2^{24}

④ 호스트 주소가 전부 0인 IP주소 = 네트워크 주소 (네트워크 자체를 의미)

호스트 주소가 전부 1인 IP주소 = 브로드캐스트 주소

한계 : 클래스별 네트워크 크기가 고정되어 있어 낭비되는 IP주소가 많다.

⇓

- Classless addressing

① 서브넷 마스크 (subnet mask)

IP주소상 네트워크 주소 = 1, 호스트 주소 = 0으로 표기한 비트열.

② 서브네팅 (subnetting)

subnet mask로 네트워크 주소 / 호스트 주소로 구분짓는 방법.

↳ IP 주소와 서브넷 마스크를 비트 AND 연산 → 결과 = 네트워크 주소

CIDR 표기법 : "IP주소 / 서브넷 마스크 상의 1의개수" 형식으로 표기.

- IP주소의 분류

① 공인 IP주소 : 전세계에서 고유한 IP 주소
- 네트워크 간의 통신 (인터넷 이용시 사용하는 IP주소)
- ISP나 공인 IP 주소 할당기관을 통해 할당.

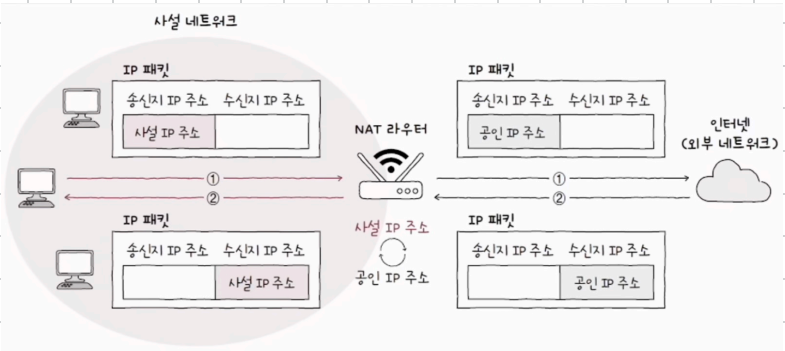
② 사설IP주소 : 사설 네트워크에서 사용하기 위한 IP주소.

범위 | 10.0.0.0/8 | 할당주체 : 라우터(공유기)
192.16.0.0/12
192.168.0.0/16

⇒ 사설네트워크 내에서만 유효한 주소.
일반적으로 네트워크간의 통신은 공인 IP주소를 통해 이루어짐

? 사설 IP주소할당받은 호스트는 외부 네트워크와 어떻게 통신을 하나요?

- NAT (Network Address Translation)
: IP주소 변환기술 사설IP → 공인IP



IP주소부여

① 정적 IP할당
; IP주소 "수동"설정 (IP주소 + subnet주소 + gateway주소 + DNS주소)
(router)

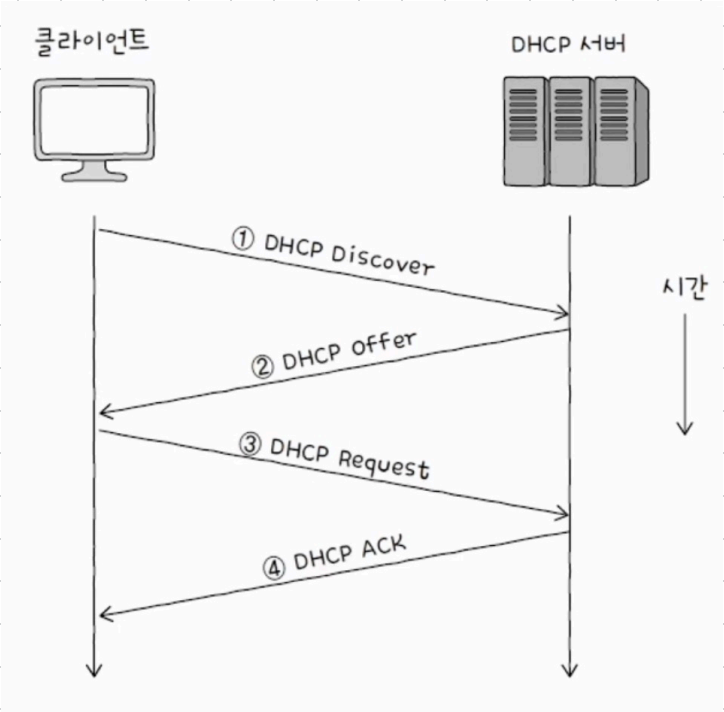
- 기본게이트웨이
: 서로 다른 네트워크를 연결하는 하드웨어적 / 소프트웨어적 수단.
→ 일반적으로 공유기 주소.

② 동적 IP할당 → DHCP

client ~ DHCP server 메세지 송수신을 통한 주소 할당.
↓ 할당대상 호스트
↓ 할당주체 호스트
- 일반적으로 라우터(공유기)가 수행.
- DHCP server는 client에게 할당 가능한 IP 주소 목록을 관리하며, client가 요청시 IP주소를 할당.

- 임대기간. (만료 후 DHCP server에게 반납.)

- 주소 할당 과정.



① DHCP Discover
- client IP 주소 할당요청
- Broad cast : DHCP server 탐색 메세지

② DHCP offer
- client에게 할당가능한 IP주소 정보 제안. (IP주소, subNet Mask 임대기간 등.)

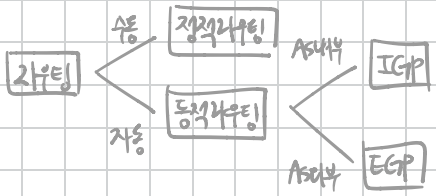
③ DHCP request
- DHCP offer 에 대한 응답
- Broad cast

④ DHCP ACK
- 최종 승인
cf) 임대기간 만료전 임대 갱신 기능.

- 예약 IP 주소 (특수목적주소)

예약 주소	IP 범위	사용 목적
0.0.0.0/8	0.0.0.0 - 0.255.255.255	이 네트워크의 이 호스트
10.0.0.0/8	10.0.0.0 - 10.255.255.255	사설 네트워크
127.0.0.0/8	127.0.0.0 - 127.255.255.255	루프백(loopback) 주소
169.254.0.0/16	169.254.0.0 - 169.254.255.255	링크 로컬(link local) 주소 (호스트가 연결된 링크로 통신 범위가 제한된 주소)
172.16.0.0/12	172.16.0.0 - 172.31.255.255	사설 네트워크
192.0.2.0/24	192.0.2.0 - 192.0.2.255	테스트용
192.168.0.0/16	192.168.0.0 - 192.168.255.255	사설 네트워크
198.18.0.0/15	198.18.0.0 - 198.19.255.255	테스트용
224.0.0.0/4	224.0.0.0 - 239.255.255.255	멀티캐스트(D 클래스)
240.0.0.0/4	240.0.0.0 - 255.255.255.254	미래 사용 용도로 예약(E 클래스)

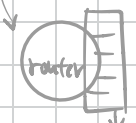
- 라우팅 (routing)



*라우팅 테이블 (routing table)

- 특정 수신지까지 도달하기 위한 정보를 명시한 표.
  ⇒ 수신지까지 도달 경로를 결정하는 요인.

① 수신지 IP주소 + subnet mask
② next hop = 다음 거쳐야 할 IP주소 / 인터페이스
③ Network Interface = 패킷을 내보낼 통로 (NIC명 / 인터페이스 IP주소)
④ metric = 경로비용
(최소화할 수 있는 방향으로 이동.)



Interface 中 하나 선택 후 명시

* Default route : 라우팅 테이블에 경로가 없을 때

* AS (Autonomous System)

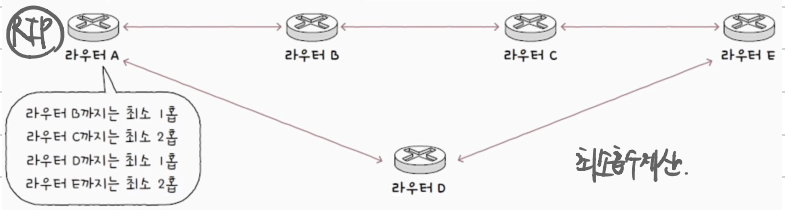
- 한 회사나 단체에서 관리하는 라우터 집합. ← 인터넷에서 고유한 AS번호 할당

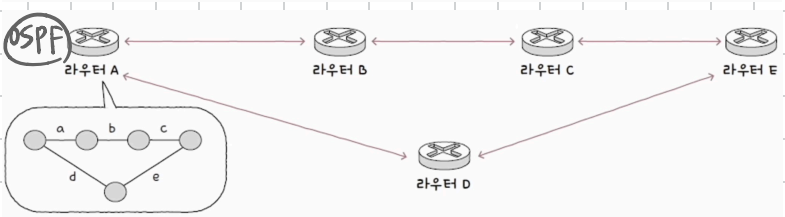
* Routing Protocol

- Router 끼리 자신들의 정보를 공유하여 패킷이 이동할 최적의 경로를 찾기위한 프로토콜

IGP
- RIP 거리벡터활용
- OSPF 링크상태활용

EGP - BGP





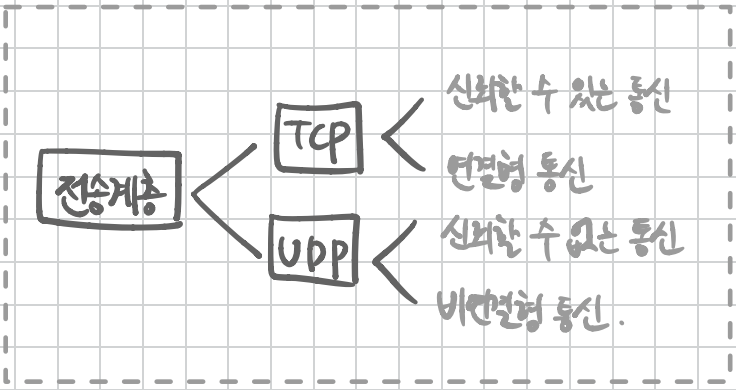
cf) RIP는 라우팅 테이블 갱신을 위해 주기적으로 라우터 간 통신 수행
OSPF는 네트워크 구성 변경시 라우팅 테이블이 갱신됨.

[전송형 계층]

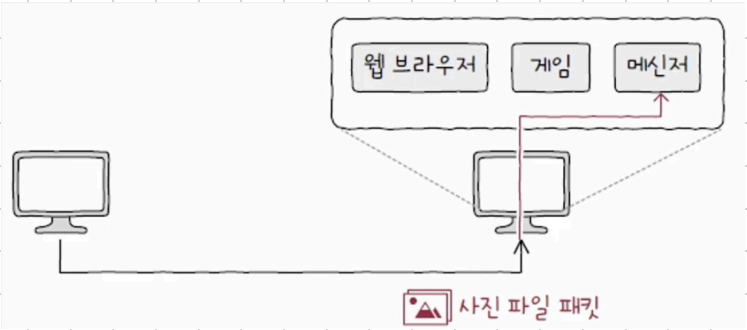
- 신뢰할 수 있는 통신 vs 비연결형 통신 ⇐ 방송최적화 목적.

신뢰할 수 있는 통신
- 패킷이 수신지까지 제대로 전송되었는지 보장O
- (최선형 전달)

비연결형 통신
- 송수신 호스트간에 사전 연결 수립 작업X



* port : 네트워크 상의 어플리케이션 식별 정보



⇒ 패킷 내 수신지/송신지 포트를 통해 송수신지 호스트의 어플리케이션을 식별
⇒ 16비트로 표현 가능. (사용 가능한 포트의수 = 2^16)

포트 종류	포트 번호 범위
잘 알려진 포트	0~1023
등록된 포트	1024~49151
동적 포트	49152~65535

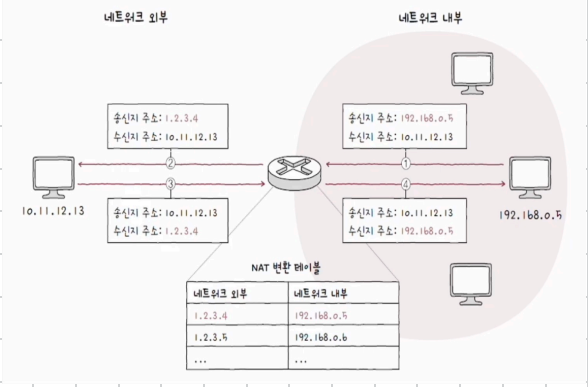
등록된 포트 번호	설명
1194	OpenVPN
1433	Microsoft SQL Server 데이터베이스
3306	MySQL 데이터베이스
6379	Redis
8080	HTTP 대체

잘 알려진 포트 번호	설명
20, 21	FTP
22	SSH
23	TELNET
53	DNS
67, 68	DHCP
80	HTTP
443	HTTPS

`특정 호스트`에서 실행중인 `특정 어플리케이션 프로세스` 식별.
→ IP주소 : 포트 번호

- NAT

NAT 변환 테이블 : 변환의 대상이 되는 IP 주소 쌍.

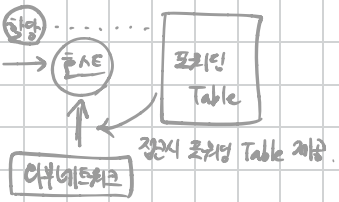


⇒ NAT 단점존재함 ; 공인IP주소 : 사설IP주소 = 1:1 로 대응해야하므로 한정된 공인 IP주소를 모두 사용할 수 없음.

포트기반 NAPT, NAT 활용.

1: 많음 = 공인IP주소 : 사설IP주소 활용 가능.

- 포트 포워딩.



- ICMP (IP 패킷 전송 과정에 대한 피드백 메시지 제공.)

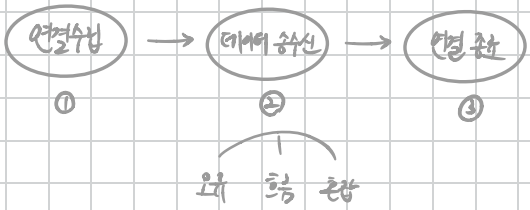
= Type (메시지 유형 번호) + Code (구체적 내용 번호)

타입 이름(타입 번호)	코드 번호	코드 설명
수신지 도달 불가 (3) ; 특정 패킷이 수신지까지 도달할 수 없음을 나타냄	0	네트워크 도달 불가
	1	호스트 도달 불가
	2	프로토콜 도달 불가; 수신지에서 특정 프로토콜을 사용할 수 없음
	3	포트 도달 불가
	4	단편화가 필요하지만 DF가 1로 설정되어 단편화할 수 없음
시간 초과 (11)	0	TTL 만료

* ICMP는 IP 오류보고 / 신뢰성 완전 보장 X

TCP.

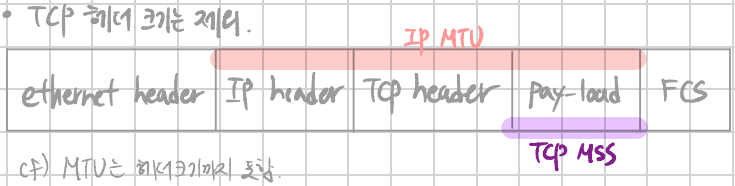
TCP는 통신 (데이터 송수신)하기 전에 연결을 수립하고 통신이 끝나면 연결을 종료.



- MSS (Maximum Segment Size)

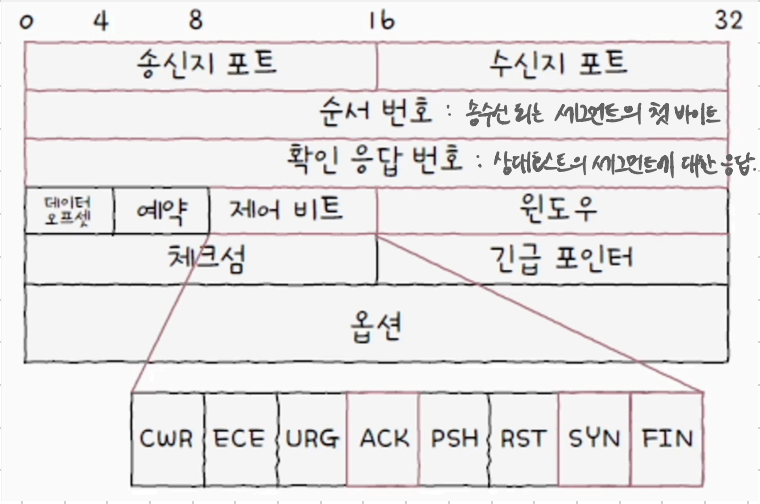
• MSS - TCP로 전송할 수 있는 최대 페이로드 크기.

• TCP 헤더 크기는 제외.

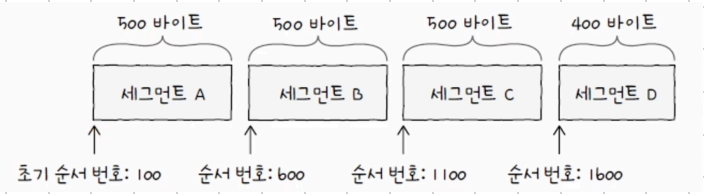


cf) MTU는 헤더크기까지 포함.

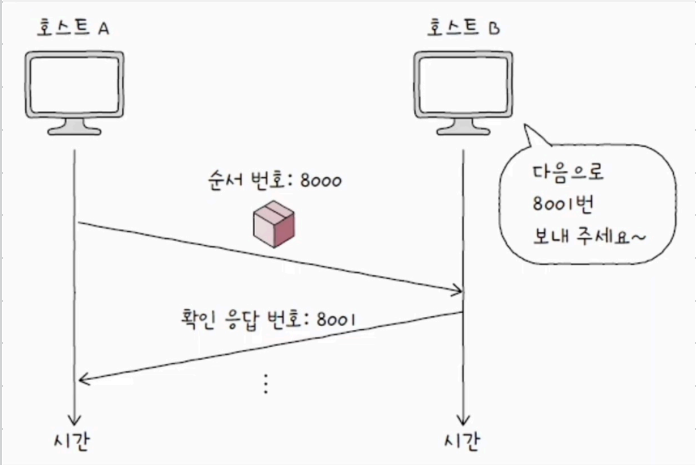
- TCP segment structure



- 순서번호



- 확인응답번호



⇒ 확인응답번호를 보내기 위해서는 세그먼트의 ACK Flag를 '1'로 설정.

- 제어비트

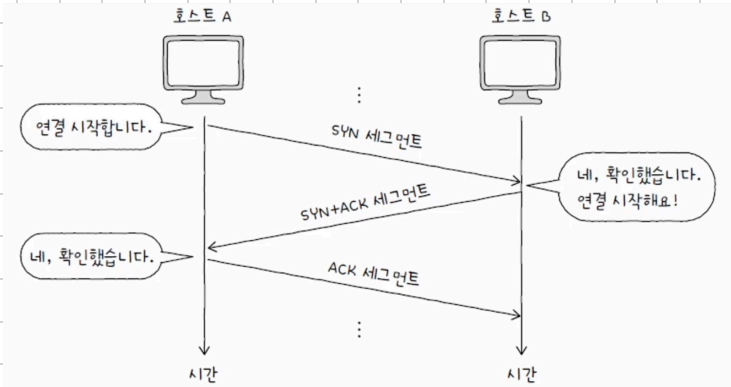
ACK : 세그먼트의 승인을 나타내기 위한 비트

SYN : 연결을 수립하기 위한 비트

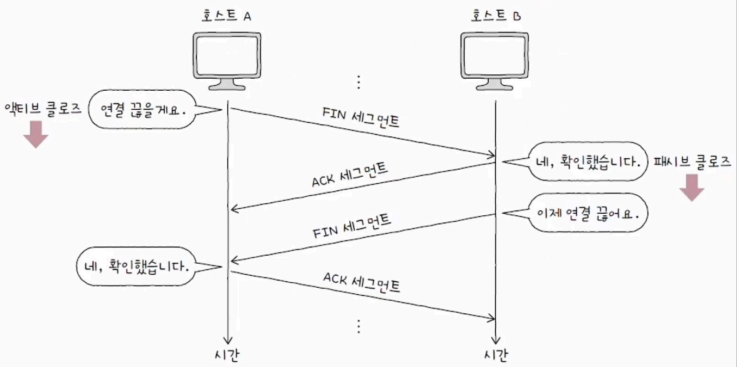
FIN : 연결을 종료하기 위한 비트.

- 윈도우 : 수신윈도우 (한번에 받고자 하는 데이터 크기) 명시.

① 연결수립 (3-way handshake)



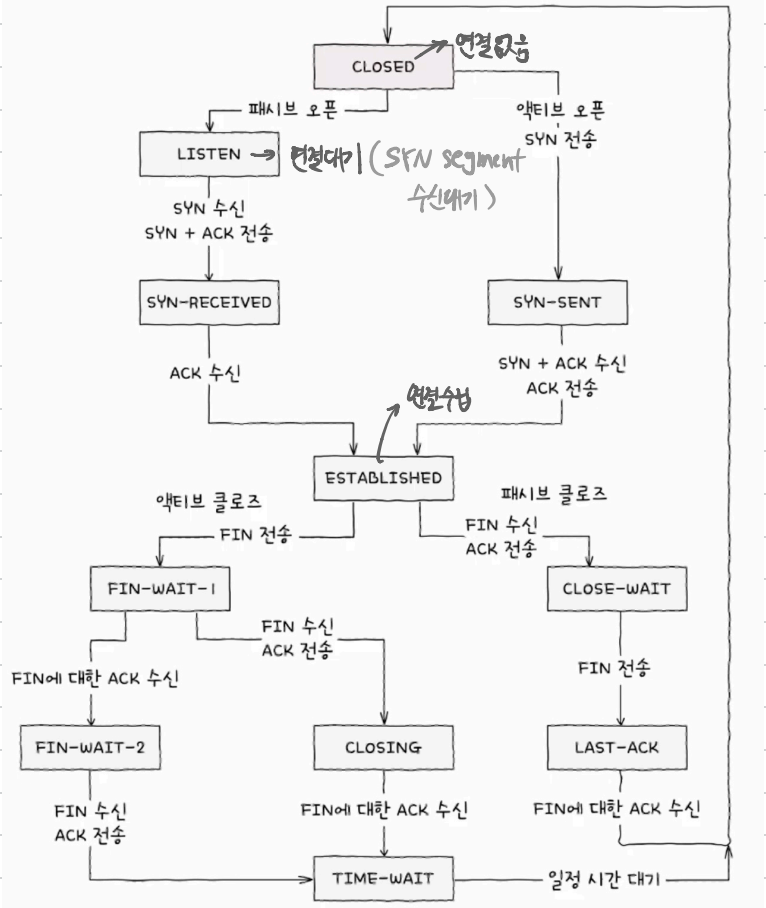
② 연결종료 (4-way handshake)



- TCP state

state : 현재 어떤 통신 과정에 있는지를 나타내는 정보

state를 유지하고 활용하는 TCP = stateful protocol



- UDP stateless protocol. ⇒ 빠른전송 / 비신뢰성 /

(송신/수신지 포트 + 길이 + 체크섬)

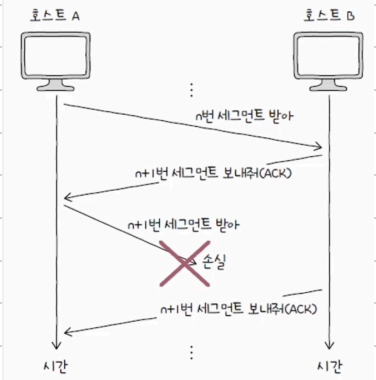
- TCP의 기능

① 흐름제어

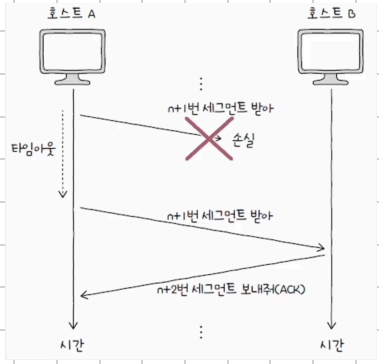
↳ 요구검출 [중복된 Ack segment 수신

타임아웃.

1) 중복된 Ack segment 수신.



2) 타임아웃.



⇒ 재전송 기법 : ARQ (Automatic Repeat reQuest) - 자동 재전송 요구

- Stop-and-wait ARQ
- Go-Back-N ARQ
- Selective repeat ARQ

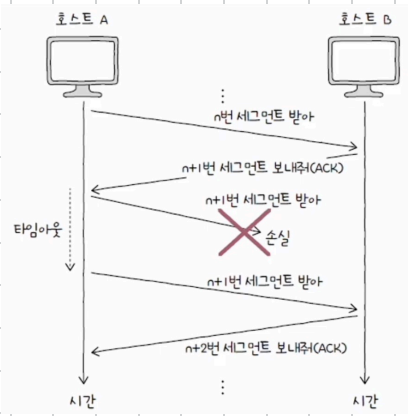
---

1) Stop-and-wait ARQ

제대로 전달됐음을 확인하기 전까지는 새로운 메세지를 보내지 않는 방식

⊕ 높은 신뢰성

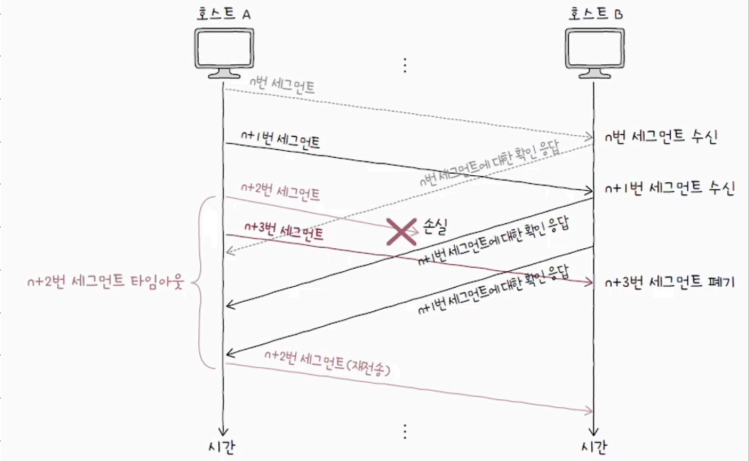
⊖ 네트워크 효율성, 성능저하.



2) Go-Back-N ARQ

[파이프라이닝 기반 ARQ 일종.

여러 세그먼트 전송 중 loss가 발생하면 해당 세그먼트부터 전부 재전송.



⇒ 누적확인응답 (CAck)

* 빠른 재전송 (fast retransmit)

→ 재전송 타이머가 만료되기 전이라도 세 번의 중복된 Ack 세그먼트를 받았다면 곧바로 재전송.

3) Selective Repeat ARQ

- 선택적 재전송 ; 각각의 패킷들에 대한 Ack Segment 발송하는 방식

- 파이프라이닝 기반 Go-Back-N ARQ / Selective repeat ARQ

→ 수신 Host의 수신가능 세그먼트 양 고려해야함.

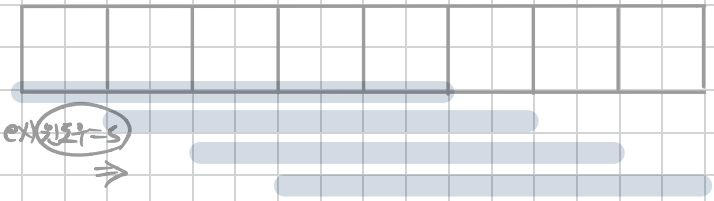
⇓

흐름제어

- 슬라이딩 윈도우 : TCP 흐름제어 기법.

윈도우 = 송신호스트가 파이프라이닝 할 수 있는 최대량.
(확인 응답없이 한 번에 전송 가능한 양)

TCP 헤더에 수신윈도우 값 발송 → 수신호스트



e.g) 윈도우=5 ⇒

② 혼잡 제어

송신 호스트가 혼잡한 정도에 맞춰 유동적으로 전송량을 조절하는 기능.

혼잡 윈도우 : 혼잡 없이 전송할 수 있을 만한 데이터양.
└ 송신호스트에서 직접 계산.

혼잡인식 (↔ 대처) ① ACK 중복수신
② 타임아웃

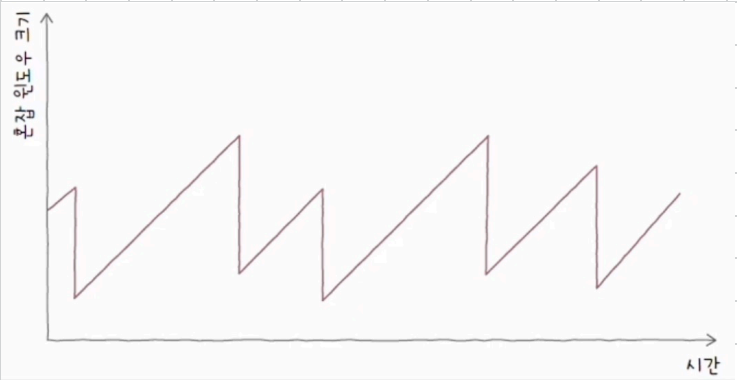
- 혼잡 제어 알고리즘.

1) AIMD (Additive Increase / Multiplicative Decrease)

• 합으로 증가, 곱으로 감소

혼잡없어 x : RTT마다 1씩 선형적으로 증가
혼잡있어 o : 1/2 로 decrease.

* RTT : 데이터 전송 후 답변수신까지 소요시간.



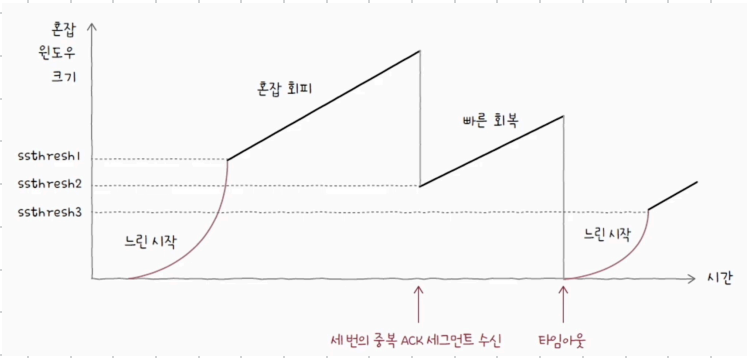
2) 느린시작 알고리즘.

혼잡 윈도우는 1부터 시작해 1회 ACK 수신될 때마다 1씩 증가
⇒ RTT마다 2배씩 지수적으로 증가.

한계지점 ① 느린시작 임계치 - 혼잡타임
② 타임아웃 - 혼잡윈도우 1로 초기화 + 느린시작 임계치 *= 1/2
③ ACK Segment x3 수신 - 빠른회복

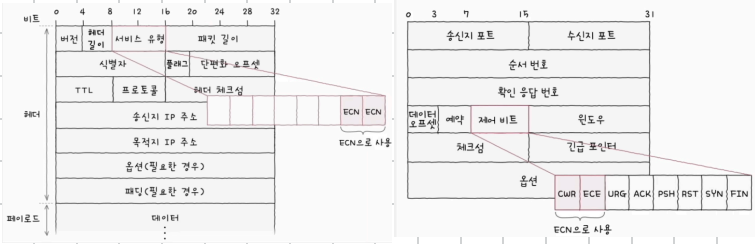
* 혼잡회피 : RTT마다 혼잡 윈도우를 1MSS씩 증가시키는 알고리즘.
알고리즘.

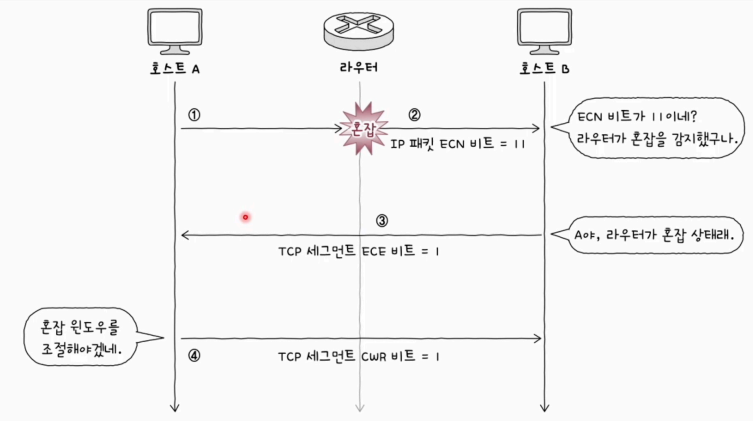
* 빠른회복 : 혼잡발생시 (윈도우 1/2 이후) 느린시작 skip + 바로회피.
알고리즘.



⊕ ECN (+ 중간노드 → 혼잡제어)
주로 라우터.

→ 호스트가 ECN을 지원할경우 TCP/IP 헤더에 ECN 관련 비트추가.





⇒ 소스호스트의 혼잡제어 : 사후처리해결
ECN : 사전방지가능.

8 도메인네임과 네임서버 (DNS 서버)

응용계층

→ IP주소만으로 송수신시 특정이 번거로워 인터 protocol
(상대호스트 특정하기.)

DNS server : 도메인 네임 ~ IP주소 관리. (≒ 공용 전화번호부)

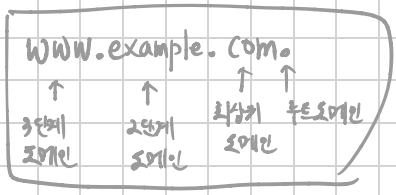
cf) hosts 파일 (개인 도메인네임 ~ IP주소 파일 관리) (≒ 개인 전화번호부)

* 분산 형태. (IP 주소 탐색 과정 – 도메인 네임을 풀이(resolve)한다.)

① 로컬 네임 서버
- Client가 속해있는 네임서버
  - 일반적으로 ISP에서 로컬 네임 서버 주소 할당
  - 공개 DNS 서버 활용 (ex. google 8.8.8.8)

② 루트 네임서버
- 루트 도메인을 관장하는 네임서버
  - 로컬 네임서버가 대응되는 IP 주소를 모를 경우
  - TLD 네임 서버의 IP 주소 반환

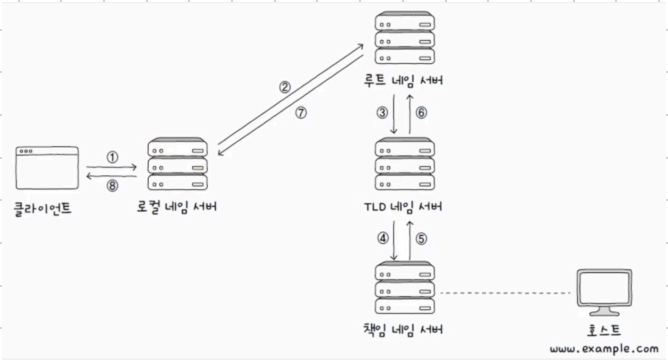
* 전체주소 도메인 네임 (FQDN; Fully Qualified Domain Name)



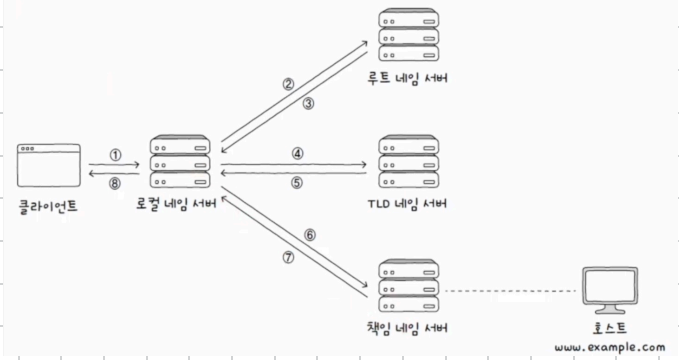
③ TLD 네임 서버
- TLD 관리 네임서버
- DNS 질의에 대해 TLD의 하위 도메인 네임을 관리하는 네임서버주소반환

④ 책임 네임서버
- 특정 도메인 영역(zone)을 관리하는 네임서버

- 재귀적질의



- 반복적질의



⇒ 앞선예시는 8단계를 거쳐야 함. (루트네임서버 과부하)

– DNS캐시 (DNS cache)

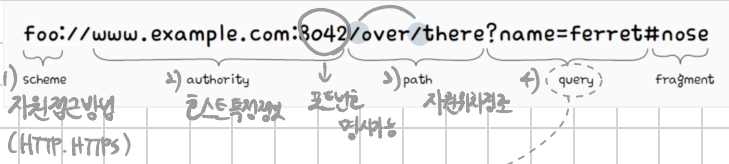
네임서버들이 기존에 응답받은 결과를 임시로 저장했다가 추후 같은 질의에 활용

임시저장된 값은 TTL과 함께 저장
(Time to Live)

- 자원 : 네트워크상의 메시지를 통해 주고 받는 대상.
→ HTTP 요청 메시지의 대상

- URI (Uniform Resource Identifier)
⇒ 자원을 식별할 수 있는 정보

① URL – 위치를 이용해 자원 식별 '/'기준으로 계층



4) Query
장바구니, 정렬순 등 HTTP에 기반한 요청/응답 메시지에서
특정 요청 수행

- query string / query parameter
- ?로 시작하는 <key, value>의 Dictionary 구조.

상품 카테고리: category
브랜드: brand
할인 여부: discounted
정렬 순서: sorted

http://example.com/search?category=books&brand=hanbit&discounted=true&sorted=price_desc

카테고리는 도서, 브랜드는 한빛, 할인은 진행 중인 상품, 정렬은 가격별 내림차순

5) fragment
→ '자원의 한 조각을 가리키기 위한 정보' (HTML 파일과 같은 자원에서 특정 부분을 가리키기 위해 사용.)

② URN
자원에 고유한 이름을 붙이는 방식

⓪ HTTP

→ 사용자와 가장 일반적이게 많이쓰는 프로토콜.

- 특징

① 요청-응답 기반 protocol

HTTP 요청 메세지 ≠ HTTP 응답 메세지

② media-independent protocol

자원과 상호작용하는 interface 만 정의. → not confine feature of resource

cf) Identifying resources = URI (≒ URL)

⇒ Structured in Type/SubType based on '/'

cf) when additional explanation needed.

Structured in "Type/subType; parameter = value"

③ Stateless protcol

- don't memorize the state of client
- regard all of HTTP request

Then, Why?

1) retaining states of all of clients is so burdonsome.

2) prevent to be subordinated to specific server. (that retain states of own client)

3) Scalability + Robustness.

When server downed, substitue other server for the server

④ persistent connection (keep-alive).

⇒ lots of request-answer in "a" TCP connection.

- Structure.

Start Line (line break)
Field Line (line break)* → 0 or more
(line break)
Message text ** → selective

① Start-Line (request : 1) request Line / answer : 2) state Line.)

1) Request Line

Method (blank) request object (blank) HTTP Version (Line)

↓ operation type

↓ URI

2) State-Line

HTTP Version (blank) State Code (blank) Reason phrase (line)

↓

Result of answer (three integer)