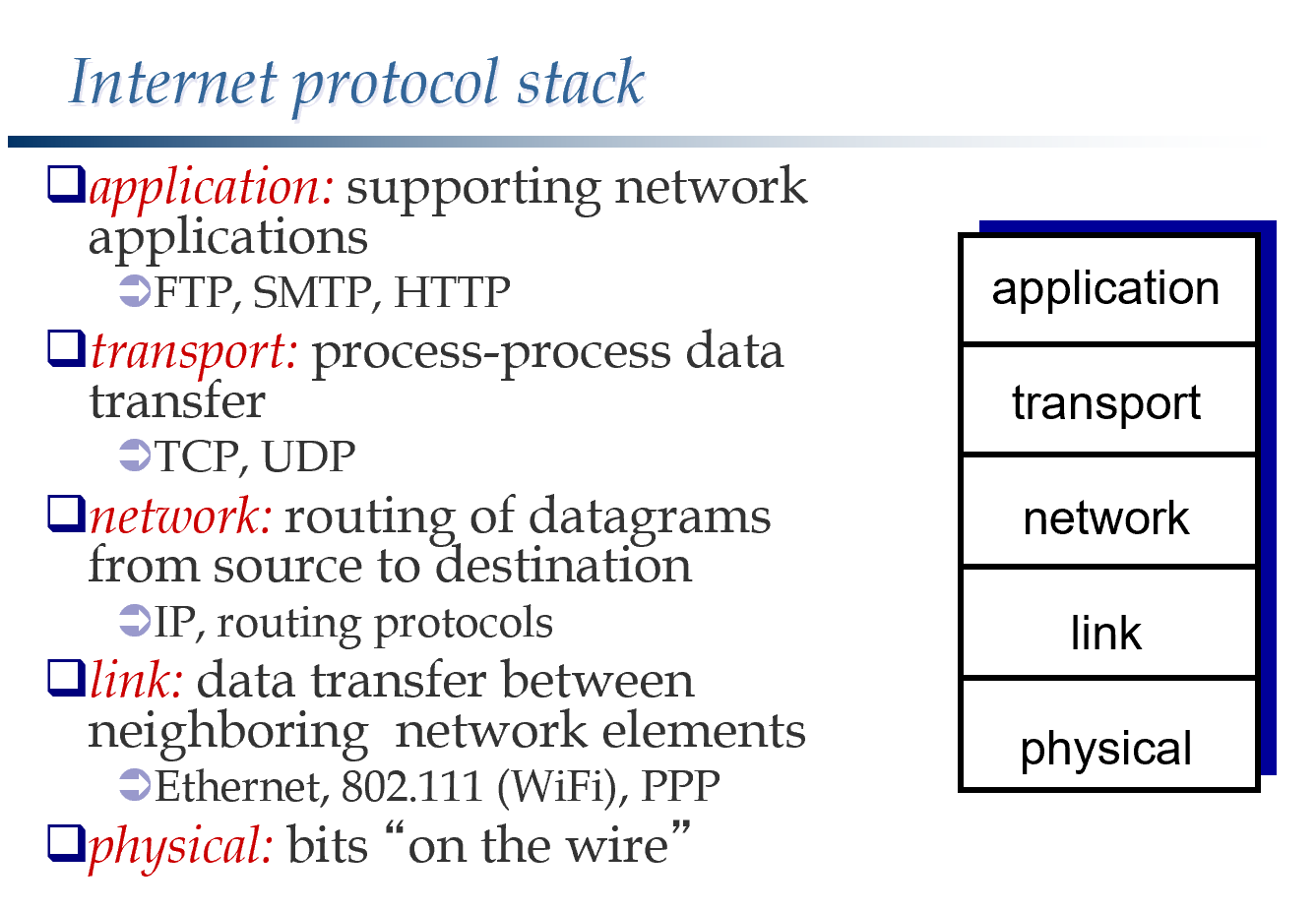
네트워크 간단정리

\*TCP /IP 5계층



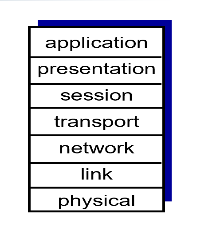
응용 (SMTP, FTP, HTTP)

전송(TCP UDP)

네트워크 (routing 담당)

링크 (이더넷)

피지컬 (물리적 연결선)

\*OSI 7계층

Presentation ( 표현 ): 암호화, 압축전송 이런 것들

Session (세션): 동기화, 데이터 손실회복 등.

Physical : 실제 물리적으로 연결.

\*Link layer 란:

물리적으로 인접한 노드(channel 로 연결)에 데이터 그램을 전송하는 것. (MAC 주소로 확인함)

링형 버스형 이더넷이 존재함.

스위치로 연결된 곳에 뿌림.(허브는 연결된 모든 포트에 데이터 뿌리고), 스위치로 동시접속 제어

스위치랑 라우터 비슷한데 둘 다 포워딩해주는거. (IP 랑 MAC 주소 차이임) , 스위치 포트로 VLAN 가능

신뢰성 있음. Error bit 활용해서 오류 검출, 반이중통신(양방향 가능 동시 x)

에러 :CRC(사이클릭 리던던시 체크) == error check 방법

-다중이용 컨트롤:

채널 분리 (TDMA-타임 디비젼 멀티액세스, FDMA- 프리퀀시)., 턴제

-랜덤액세스 :

알로하 ( 충돌하면 고치고, 슬롯 알로하는 그래서 쪼개서 겹치는 부분만 다시보냄) 운좋을때까지

CSMA 는 지금 채널 쓰고있는지 확인. => 그래도 충돌 가능 => 콜리전 디텍션(주파수)해서 확인

-턴제: token 돌리기 등

ARP 프로토콜

ARP 테이블이란== MAC address 위치를 보관하고있는 것. 찾을 때는 broad cast 해서 나오슈한다.

여기 구글 시나리오나옴 81P

\*network 계층

Router 가 들어온 것들 저장하고 포워딩 함 . HOL(head of line) blocking 이라고 queue 만듬.

* 이게 거의 congestion control 이라고 할 수 있지.

신뢰성 x 상위 tcp 계층 가야됨

Subnet 있는데 함 살펴보기

Sub net 이란 물리적으로 사이에 routing이 없이 도달 가능. 같은 subnet part ip address 를 가짐.

IP4 는 network bit, host bit 으로 구분된다.

CIDR 이라고 classless 해서 주소매기는 법 있음. 원래 ip 는 앞부분에 Aclass 인지 bclass 인지 하는데 얘는 class less 임

호스트가 광역방송으로 DHCP(공유기) 서버 찾고, 대답오면 요청해서 IP 주소 받는거임. Client 주소, DHCP 주소, DNS 서버 주소와 이름 준다. Plug and play

NAT : network address translation

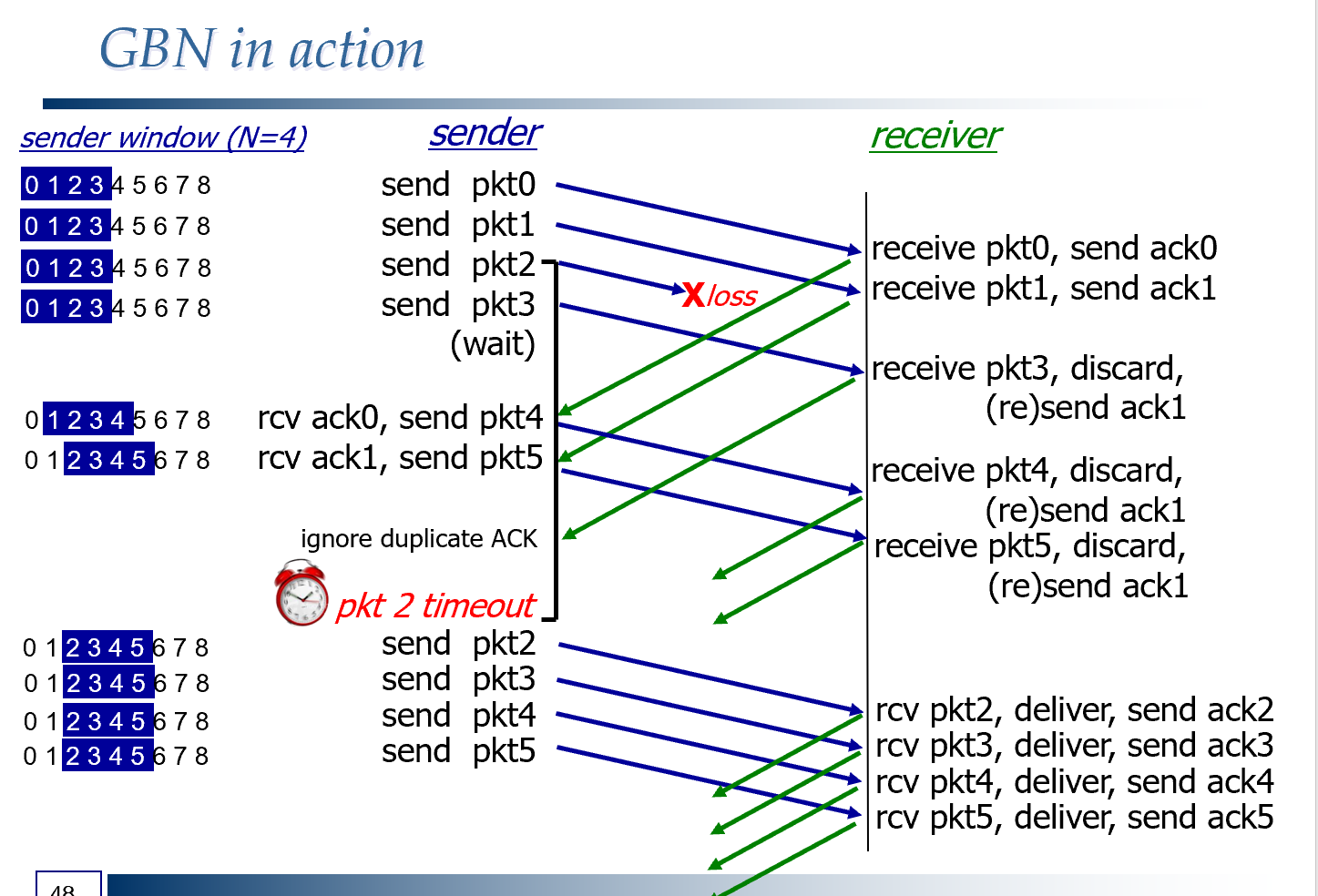
IP v6 사용이유, 더 많은 IP 주소 , 효율적인 Packet 처리 header 부분 많이 변경됨. 효율적인 포워딩.

\*트랜스포트 계층

Port 별로 mux, demux 발생

Pipe lining 으로 ack 받기.

Go back N , selective repeat (selective repeat 은 딜레마 있음)



TCP 특징:

1. 신뢰성이 있고,
2. 순서가 있고,
3. 윈도우 사이즈를 통한 pipelined, congestion control
4. 연결지향적임
5. 전이중 통신 이거와 반대되는게 link layer

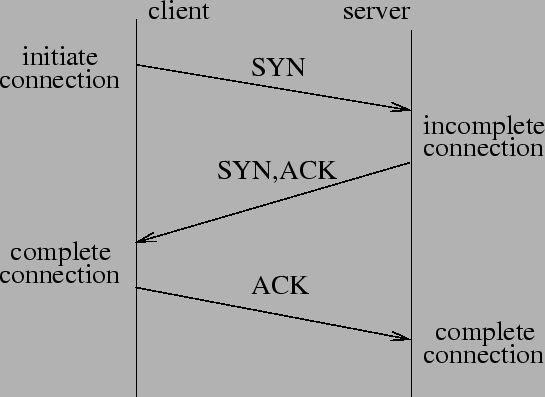
Round trip time: 돌아오는 시간 RTT.

기준으로 time interval 둠. 재 설정은 같은 ack 가 세번 날라올 경우. 왜냐하면 대답 안 한것 중 가장 작은 seq 에 대한 답만 계속할 거임.

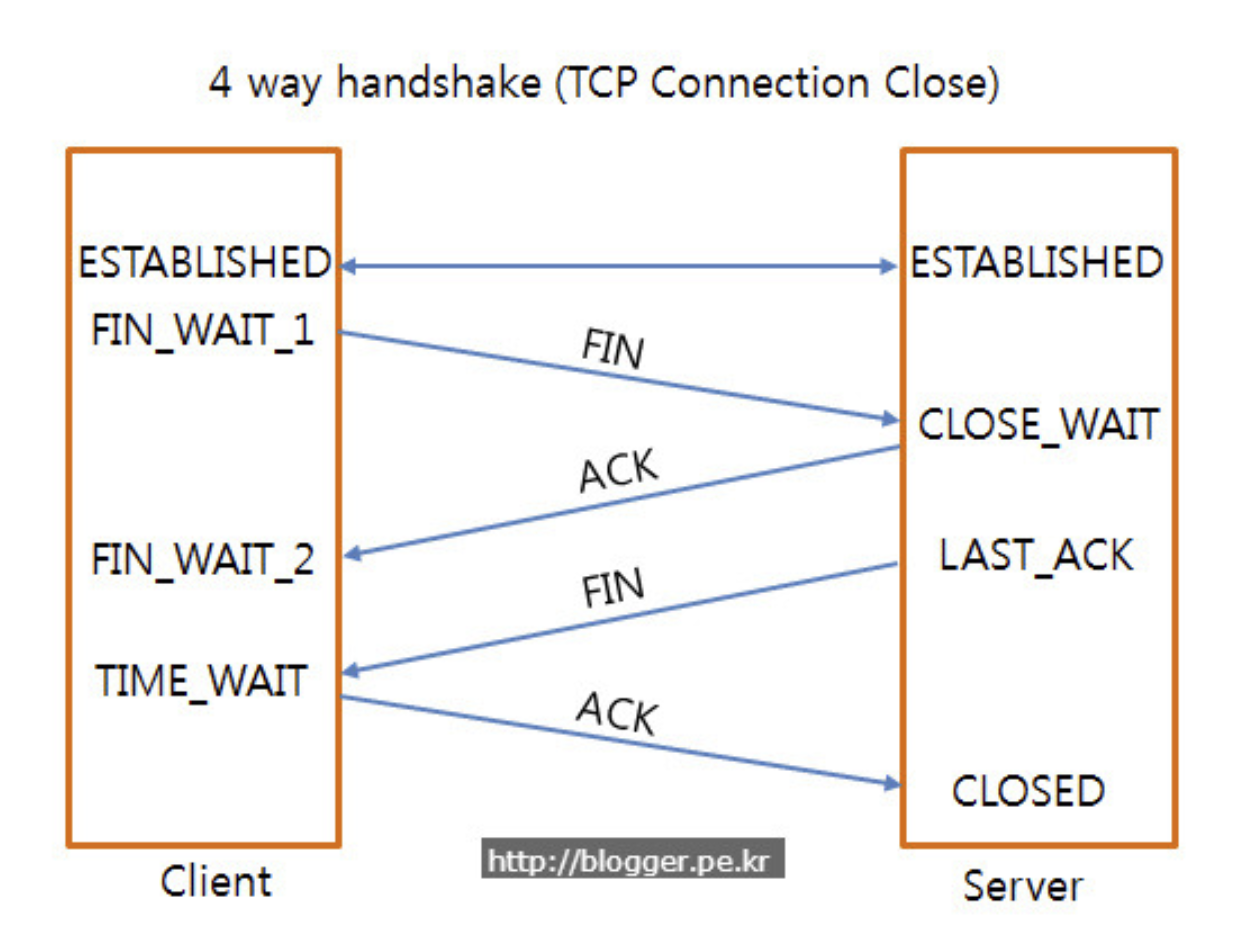
TCP socket receive 버퍼 존재

TCP 의 혼잡성 제어 방법 CWND(congestion window), MSS( maximum segmenation size) 을 점점 올림. 타호가 예시

3way handshake = 클라이언트에게 서버의 존재를 확인하는 것 뿐만 아니라, 서버에서도 클라 존재 확인 필요



4way hand shake =신뢰성 있는 연결인 TCP 특성상 보낸 데이터를 잘 받았는지, 그에 대한 잘 받았다는 응답 역시 잘 받았는지 등을 양측에서 확인하게 된다. 이때 통신하던 중 예기치 않게 연결이 종료된다면 아무래도 TCP의 신뢰성에 큰 영향을 끼칠 수밖에 없다. 그러므로 연결 종료 시에도 양측이 서로 확인하에, 즉 당사자가 연결을 종료하고 싶어하며 종료할 준비가 됬는 지 확인한 후에 각자의 종료 과정을 거치게 되는 것이다.



3way hand shake 공격 방법은 Syn flooding 이 있고

4way 의 문제점에는 종료 FIN 보다 후에 날라오는 데이터들에 있다.

\* routing

Link state 의 다 익스트라 방식 모든 연결 상태를 다 알아야함. OSPF

Distance vector 방식의 bellman ford BGP, (inter AS-네트워크 단위)

SDN NFV 쪽 더보고

\* Application

APP의 아키텍쳐 형태 == p2p, client server

HTTP?

Object 요청 응답 : HTTP

하이퍼 텍스트를 전달하기 위한 프로토콜

웹페이지 HTML 기반 파일에 여러가지 object 를 refence 해서 활용

* HTTP 는 tcp 기반임 왜냐면 준거랑 받은거랑 같아야 하니까. Reliable
* 프로세스 간의 웹서버와 웹브라우저 간에 어떤형식으로 정보를 교환할 것 인가에 대한 약속.
* URL 로 참조. Ex ) JPG 파일
* 요새는 HTTP over quic on UDP => chrome UDP 위에도 동작함
* Non persistent vs Persistent
  + 한번에 하나의 오브젝트 vs 통로 계속 열어놓기 (최근버전)
* 쿠키를 사용한 성능향상
* Version 별로 다양한 기능 추가.

상태코드 아스키, 명령어 추가, 커넥션 유지, keep alive, 파이프라이닝 등의 관리

* GET, RESPONSE
* Proxy server ( caching )
  + Proxy server 가 cache memory 의 역할을 하는 것
  + 만약에 origin server 가 웹페이지를 modification 했다면?
  + 그렇기 때문에 proxy는 origin 한테 수정했는지 물어보고 client 한테 준다
* Proxy, cookie 같은 기능으로 cache 처럼 구현.

FTP, SMTP

DNS

Domain name system. Recur or iter

CDN

\*Security

대칭키 vs 공개키, 개인키

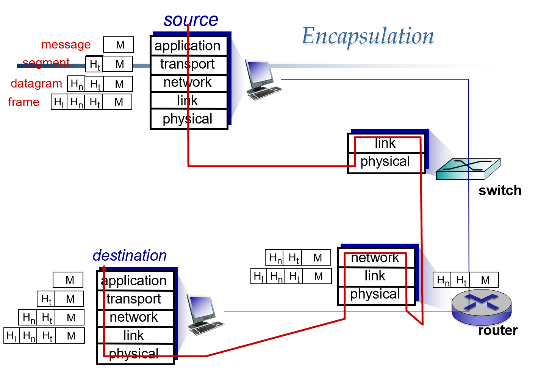
Digital signature

Bob의 private key를 통해서 encryption 한

밥의 메시지와 encryption 한 거와 원래 메시지를 같이 보냄.

=>Bob 의 public key 를 사용해서 decryption

메시지가 같은지 확인

\*네트워크 운영 방식이니까 꼭 한번 보기.

핵심 그림.

Multimedia.

Streaming, VOIP

Streaming 관련

UDP임 , HTTP(TCP)기반 다운을 할 때 도 있음.

Jitter를 고려하여 streaming 시작

Fixed 한 방식과 (일정한 throughput 제공)

Adaptive 한 방식(loss 줄임)

Network delay jitter