ch1.

edge host (client/server)

core packet switch

communication (link)

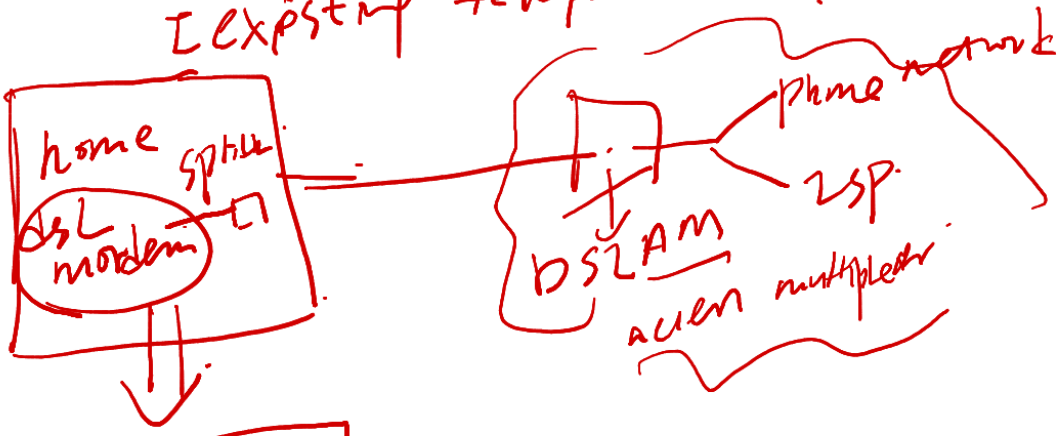
Access network: physical media

1. cable based access

① hybrid-fiber coax

② digital subscribe line (DSL)

[existing telephone line]



router
DSL modem
wifi access
→ all in one

③ wireless
WAN
wide-area cellular access network

2. transmission rate R:

— capacity,

— link bandwidth.

transmission delay = $\frac{L(bit)}{R(bit/sec)}$

physical link

guided media: (solid) → copper, fiber, coax

unguided media (free) → radio

- Twisted pair (TP) — category 5, category 6
- Coaxial cable — bidirectional, multiple frequency
- fiber optic cable — high speed, low error rate
- wireless radio — send to receiver (half-duplex), environment affect

coaxial

wireless LAN (wifi) | wide area (4G cellular) | bluetooth | terrestrial microwave | satellite

forwarding/switching (local) vs routing (global)

demand allocate

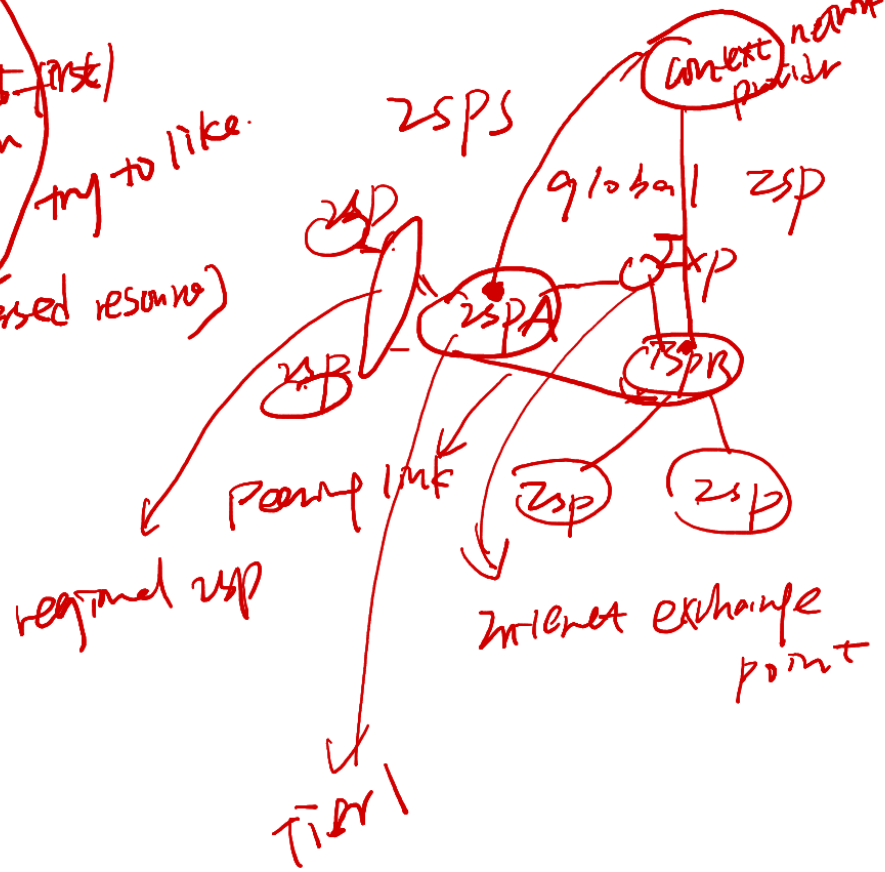
network of networks

packet switching:

- packet transmission delay
- store & forward
- queueing → loss, congestion (packet arrives first, try to like)

circuit switching:

- dedicated resource (reserved resource), no sharing, idle
- telephone
- FDM, TDM

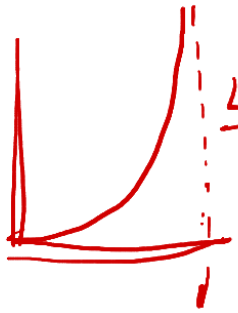


$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$

bit error, out link mis

congestion ↓

$\frac{L}{R}$ · $\frac{d}{s} \rightarrow 2 \times 10^8 m/s$



$\frac{La}{R} \sim 0$ $d \rightarrow 0$

$\frac{La}{R} \rightarrow 1$ $d \rightarrow$ large

> 1 infinite

traffic intensity

throughput = { instantaneous, average

rs, rc, R

security in all layers

- sniffing (from broadcast media, ethernet, wireless)
- ip spoofing (injection, false source address)
- denial of service (bogus traffic, botnet; dos)

- authentication. (SIM card, identity)
- confidentiality (encrypt)
- integrity checks (digital signature. proof, tamper detect)
- firewall (specialized middlebox)

layering

ISO/OSI Layer.

application

presentation → allow application. - compression encryption machine-specific conversion

session → synchronization checkpointing recovery of data exchange.

transport

network

link

physical

ch2.
network app. = end system program.

client-server paradigm
- may dy... ip; each other com... X
- always on host; permanent ip; data center

peer-peer architecture.
- no-always on server
- intermittently connected

process different host : message.
(p2p host = client process & server process)

sockets process identifier ip + port
HTTP 80
mail 25

application protocol :
- type of message.
- message syntax, semantics
- rules
- open protocols [RFCs]
- proprietary protocols (zoom. skype)

App transport service
data integrity (reliable?)   throughput (amount)
timing (delay?)   security (encrypt)

TCP

reliable.
flow control
congestion control
connection-oriented.

[FTP, HTTP, SMTP.
SIP, RTP, DASH. WOW, FPS]

UDP

unreliable.

[SIP. RTP, WOW. FPS]

Vanilla TCP & UDP

no encryption.
clear text.

transport layer security

TLS

security & data integrity
authentication.

Implemented in application layer

(TLS lib)

HTTP (Hypertext transfer)

use TCP

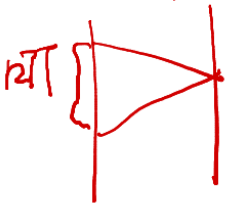
Stateless (server has no client past request info)

non-persistent
TCP connection
one object sent
TCP close.
2RTT/object

parallel

persistent
TCP connection
multiple objects
TCP closed.
(HTTP 1.1)

RTT.



HTTP message: request / response.

ASCII: method -SP ..... cr lf

request:

- post: input in body
- Head: headers only
- Get: send data in URL field (?)
- put: upload to server

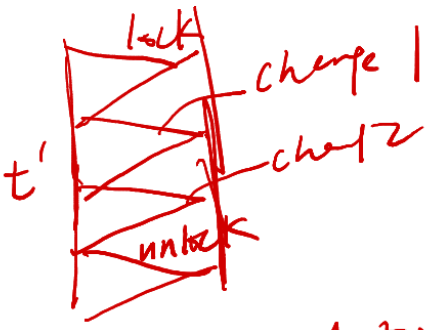
response:

status line.

200 OK 301 Moved permanently 400 bad request

404 not found 505 HTTP version not supported

Stateful protocol: (2 charge)



HTTP = stateless : no need to track request independent

But cookie in message no need recover from t'

CIS

user/server states maintain web sites & browser: cookie (a unique id)

1) Cookie header line (response)
2) cookie header line (next request)
3) cookie file in user's browser
4) cookie file in back-end database at website

function: (cookie)
authorization shopping cards recommendation session state (web e-mail)

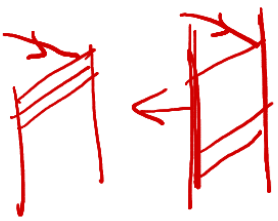
privacy:
- third party
  behavior on given website (first party)
  behavior across multiple website (third party)
  GDPR

How keep state
① protocol endpoints (multiple transaction) → stateful protocol
② in message (cookies in HTTP message)

web caches
configure browser to point
HTTP request → local cache. ← in return from cache.
↘ request server to cache

conditional get (web cached up-to-date?)
C: if-modified-since <date>
S: 304 not modified / 200 ok <data>

HTTP/2
decrease delay in multi-object request
request priority.
push in request
frame of object → mitigate HOL blocking
single TCP → stalls by retransmission
no security over vanilla TCP



HTTP/3
add security
congestion control
move piping over UDP

email

SMTP (simple mail transfer protocol)

User agents : reader, mail client (outlook)

mail servers : mailbox : incoming message for user (□□□□)

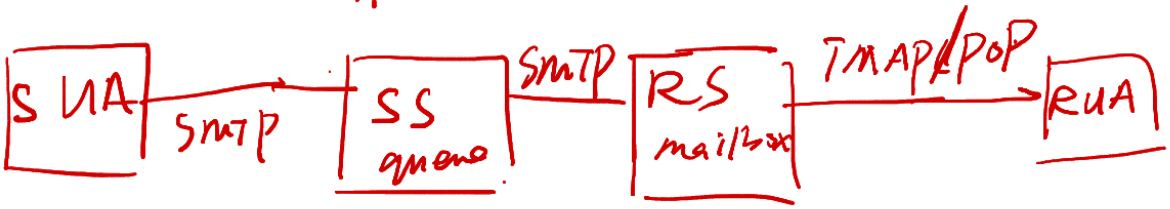
message queue : outgoing (□□□□)

SMTP → { between servers (to receiver server) → queue → mailbox
TCP . 25
(client push)

- SMTP handshake. RFC5321
- tranfer
- closure

header : to
from
subject
(CRLF CRLF)

command ASCII
response. status code . phrase



IMAP → Internet Mail Access Protocol (retrieval from server)
mailbox stuff

HTTP (web based interface)
SMTP / IMAP / POP

DNS (domain name system) → UDP

name ⟺ IP

- distributed database
- application layer (but core internet function)

ROOT DNS   root   13 logical   ICANN

/ | \

.com .org .edu   Top-level domain. (TLD) → [.com .org .cn]

/ \   / \   /\

yahoo.com   pbs.org   Authoritative   organization.

{ DNS registrar author → TLD (RR) NS A

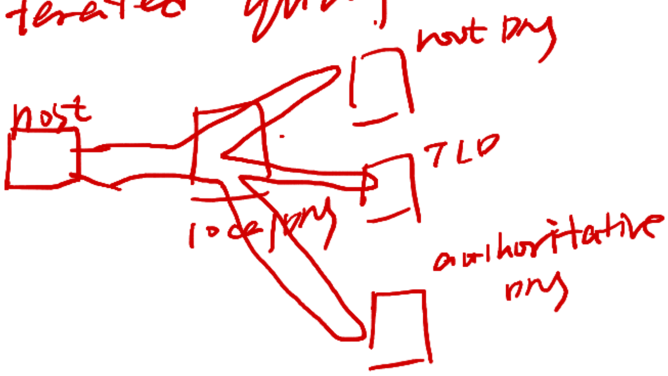
A. MX }

local DNS (by ISP)

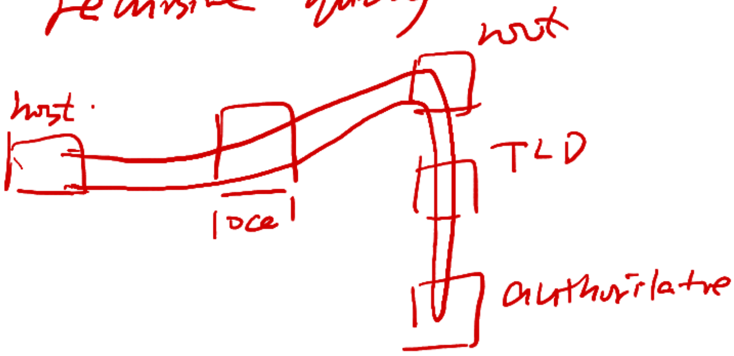
↑ ↓

query

Iterated query



recursive query



cache mapping [use TTL out-of-date]

resource record (RR) [stored]

(name, value, type, ttl)

name	value	type
hostname	ip	A
alias	canonical	CNAME

CNAME: www.ibm.com | servereast.backup2.ibm.com ↑ canonical

TLD domain.	authoritative name	NS
name	SMTP mail server	MX

query/reply :: same format

id. flag (query/reply...)   query ^ name.type   response ^ RR   author?

SEC [DDoS attack

spoofing attack   'DNSSEC' in RFC]

P2P
- no always-on server → (C/S)
- self scalability
- IP change.
  (bittorrent, streaming. VoIP)

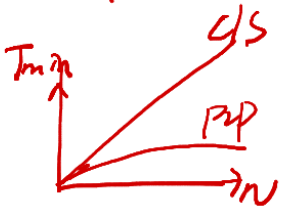
File distribution

C/S

$\geq \max\{\frac{NF}{u_s}, \frac{F}{d_{min}}\}$

d_{min} ↓ download rate

upload download



P2P

at least one copy

$\geq \max\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s+\sum u_i}\}$

upload download

bittorrent =
- tracker: tracks peers in torrent (list)
- torrent: group of peers ← neighbor
- charge peers
- churn: come/go

request chunks

send chunks [tit-for-tat]

better trading partner

[Encoding + DASH + playout buffer]

streaming:

spacial/temporal coding

CBR
VBR (MPEG)

- streaming stored video — network delay
- client side buffer
  playout delay

DASH: dynamic · Adaptive streaming over HTTP (manifest)

encoding rate

CDN: content distribution networks (copies in distributed site)

OTT (over the top)

Socket

UDP

SOCK_DGRAM

StreamSocket = Socket(,)

clientSocket.sendto(port)

close

TCP

clientSocket = Socket(, SOCK_STREAM)

clientSocket.connect(port)

clientSocket.send( )

close

SOCK_DGRAM

serverSocket = Socket(,)

serverSocket.bind(, port)

serverSocket.sendto('', addr)

↓

client IP port

serverSocket = Socket(, SOCK_STREAM)

serverSock.bind(port)

serverSock.listen(1)

{server socket.accept()

connectionSocket.send()

ch3.

logical communication

break — segment — reassemble

transport layer : communication between process demux

network layer : between host postal

Transmission control protocol

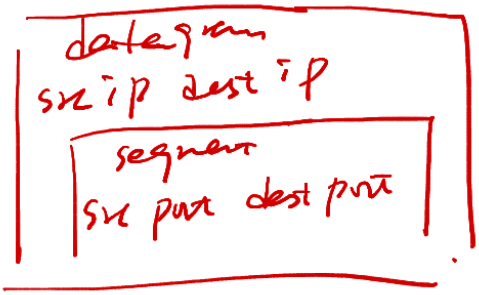
- reliable
- in order
- congestion control
- flow control
- connection set up

multiplex — demultiplex

[segment, datagram]

user datagram protocol

- un-
- unordered
- no-frill



TCP/IP datagram

[src ip src port dest ip dest port] | udp datagram ip/udp

(dest ip port) same

~~src ip port~~ different

↓

socket.

↓

same socket

UDP:

DNS

SNMP

HTTP/3 { reliable, congestion control } at application

principle of reliable data transfer (rdt)

checksum

rdt_send — deliver_data

udt_send ⟷ rdt_rcv

rdt 1.0

rdt 2.0 (acknowledge + retransmission) ack/NAK → (that's why protocols)

stop and wait

[fatal flaw: ack/NAK corrupted]

rdt 2.1 +

retransmission pkt if ack/NAK corrupt

seq number of pkt [pkt 0, pkt 1]

rdt 2.2 NAK free (duplicate ack)

rdt 3.0 +

time out

pipelined

RTT + $\frac{L}{R}$.

RTT + $\frac{nL}{R}$

go-back-n

sender: oldest in flight timer → all n pkt

buffer window: trans but unacked

cumulative ack [ack(n)]

receiver:

highest in-order seq → duplicate Acks,

out of order: buffer or discard

window: N size, consecutive.

Selective repeat

each unacked timer → only the 1

Sender = retrans individual unacked pkt

receiver = individually ack pkt & buffer

window = N consecutive seq

limit seq

dilemma = ack from receiver miss/corrupt

Principle of congestion control

flow control

one sender ⟷ one receiver (two-part)

congestion control

many senders ⟷ one receiver (two-part)

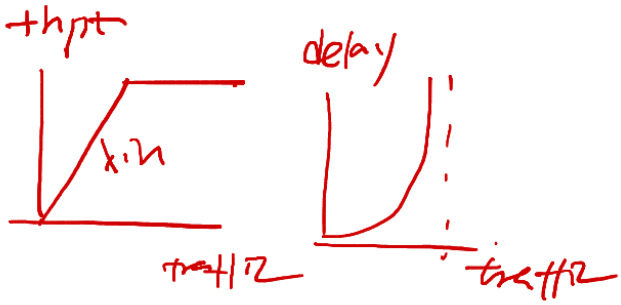
long delays & packet loss

queue

overflow

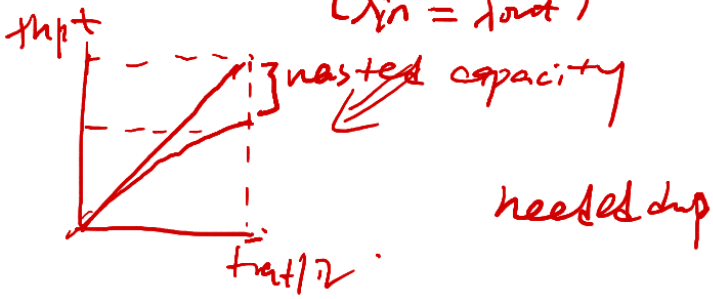
↓ thpt

more work



①

② retransmit $\lambda'_{in} > \lambda_{in}$ ($\lambda_{in} = \lambda_{out}$)



needless dup

③ timeout prematurely

unneeded dup

④ multi-hops

(TCP)

— end-to-end congestion control

inferred from loss/delay

— network-assisted congestion control

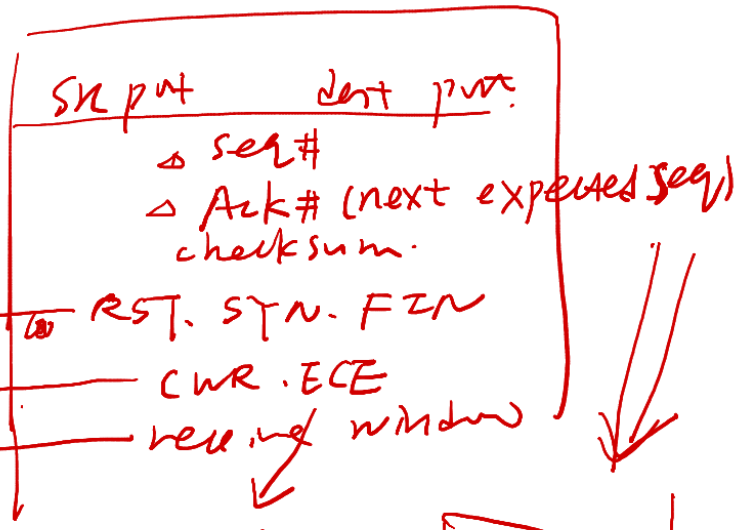
router provide feedback.

indicate level, set sendrate

(TCP ECN, ATM, DECbit)

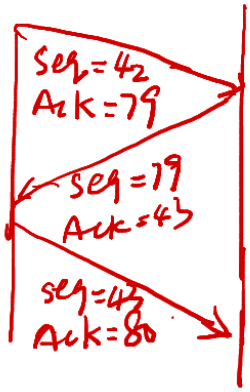
TCP
- point to point
- full duplex
- cumulative Ack
- pipelining
- connection-oriented
- flow control
- in order byte

MSS



congestion

ECN in IP



tcp timeout:
sampleRTT EWMA

estimatedRTT = (1-α) EstimatedRTT + α SampleRTT

α: 0.125

Timeout Interval = estimatedRTT + 4 DevRTT

DevRTT = (1-β) DevRTT + β |sampleRTT - estimatedRTT|

β: 0.25

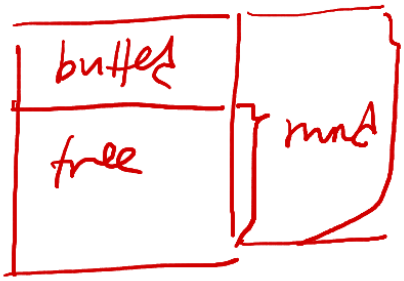
① TCP Ack gen

② TCP retransmission

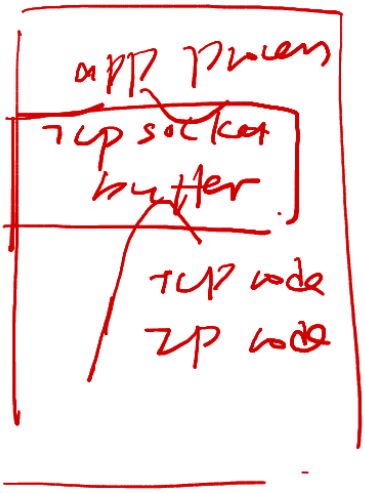
③ TCP fast retrans → 3 Ack

flow control

receive window (rwnd)



receive buffer



receiver advertises free buffer space (rwnd)

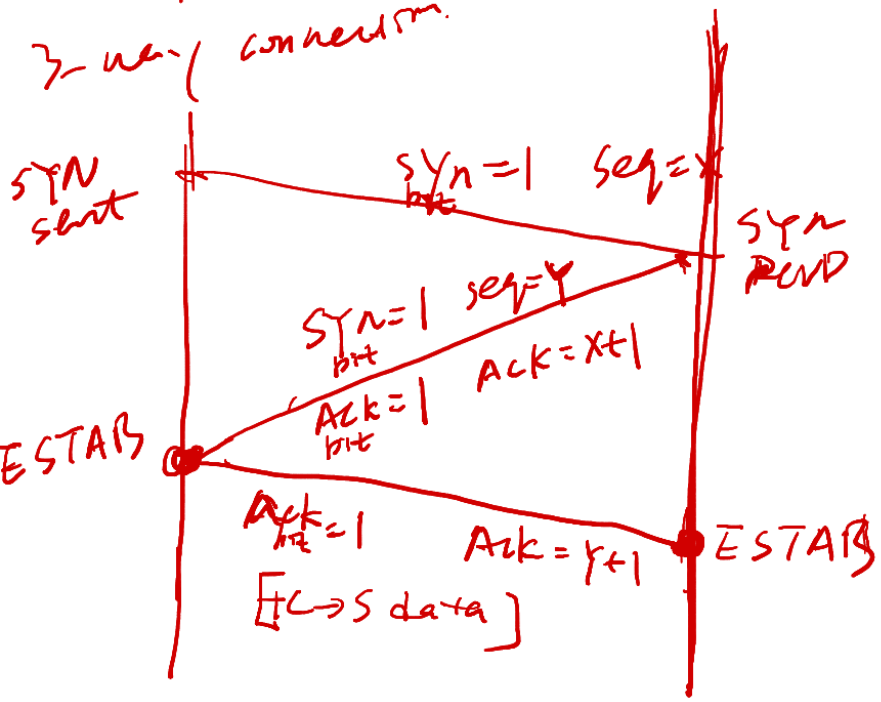
sender limits unACKed inflight data (rcv buffer not overflow)

TCP connection

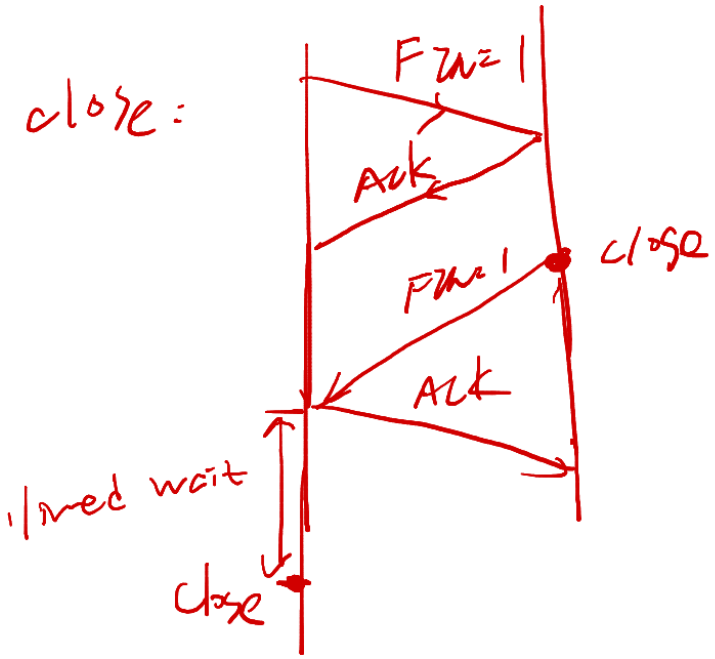
2-way connection problem

duplicate connection

3-way connection



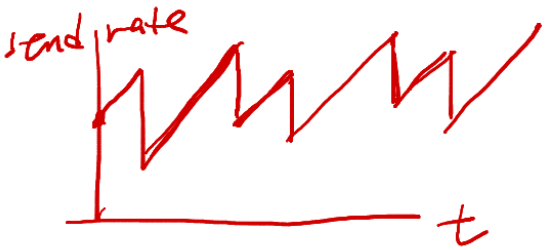
close:



Tcp congestion control

AIMD

sender increase/decrease sending rate until loss/congestion occur



Additive Increase

multiplicative decrease

[probing for bandwidth]

Reno cut in half

Tahoe cut to 1 mss

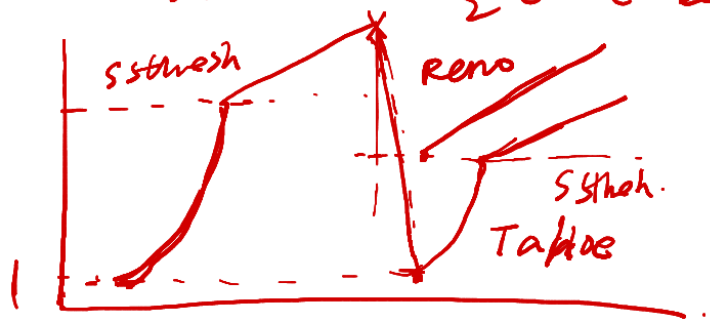
cwnd (congestion window)

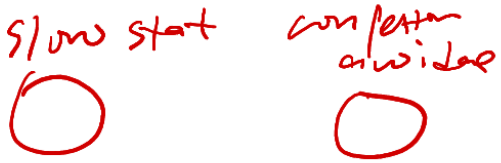
→ dynamically adjusted

TCP rate ≈ $\frac{cwnd}{RTT}$

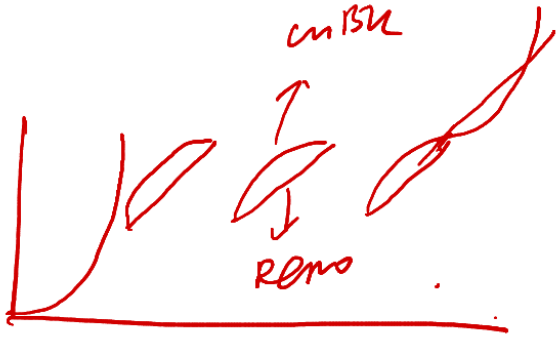
slow start = [cwnd = 1 mss, doubled (1-2-4-...)]

ssthresh = $\frac{1}{2}$ cwnd before loss event.





CUBIC



bottlenecked link (always busy router)

delay based TCP congestion control (BBR) google

pipe = just full enough. but no fuller

RTT_{min} — minum observed RTT

measured throughput $= \frac{\#\text{lost RTT intend byte}}{RTT \text{ measured}}$

maximize thoughput (full)

delay low (not fuller)

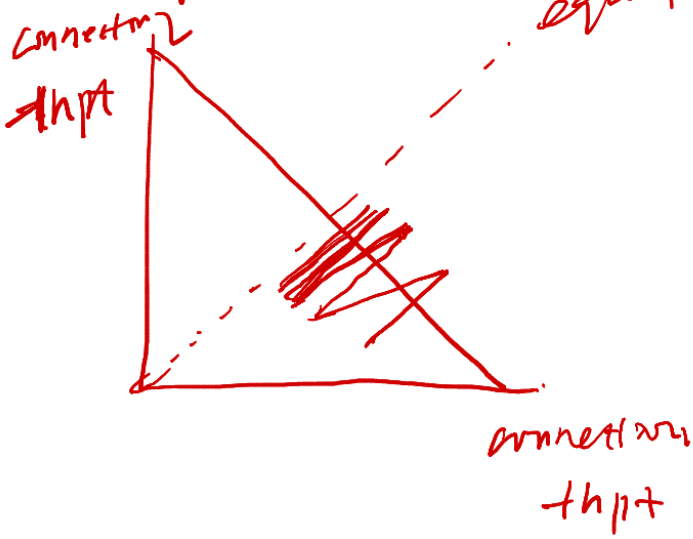
Explict congestion notification (ECN)

ECE on Ack

ip ECN

TCP ECE, CWR

TCP fairness



TCP is fair

HTTP/3 [HTTP/2 (slimmed) / QUIC / UDP / IP]

HTTP/2 / TLS / TCP / IP

no HOL blocking

envlop: different flavors of TCP

UDP + HTTP/3 = QUIC

data & ctrl

Data
local
forwarding

control
wide/global
routing
(routing) routing algorithms
(semi) SDN

service model:

individual — a flow of ...

guaranteed delivery — in order datagram
minimum bandwidth
restriction.

less delay

Internet "best effort"
- no delivery
- no timing
- no in order
- no bandwidth available

(delay)
queuing & loss

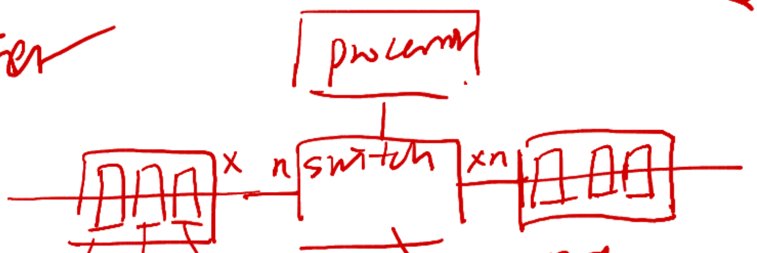
buffering = $\frac{RTT \cdot C}{\sqrt{n}}$ → link capacity

① drop policy
taildrop
priority

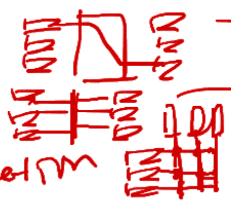
② marking
(ECN RED)
congestion.

③ scheduling discipline
FCFS
priority
round robin
WFQ

router



memory → traditional
bus → contention.
interconnection → n×n parallelism



physical
link (ethernet)
forwarding
queuing → HOL blocking

decentralized switching (match & action)
{ destination-based → longest prefix matching [ternary CAM) TCAM
generalized → SDN

network layer

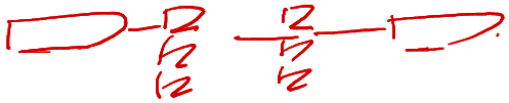
routing = — OSPF, BGP → forwarding table.
— SDN

IP { data format, addressing, packet handling
ICMP { error reporting, router signaling

ver	headlen	service (ECN..)	length (total)
id, flgs, Offset → fragment / assembly			
TTL (max hops)	upperlayer (TCP/UDP)	header checksum	
src IP (32)		dest IP (32)	
⋮		⋮	

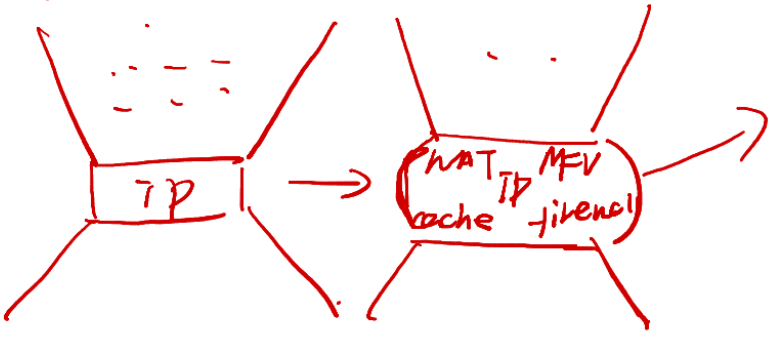
IP address = interface. (by) [connected by wired/wireless]
physical link
host, router → (multiple interface)
IDR2

→ subnet + host
fragment / assembly : MTU (max transfer unit)



subnet : reach without router
- island ..
subnet mask
CIDR = a.b.c.d/x
(classless . InterDomain Routing)

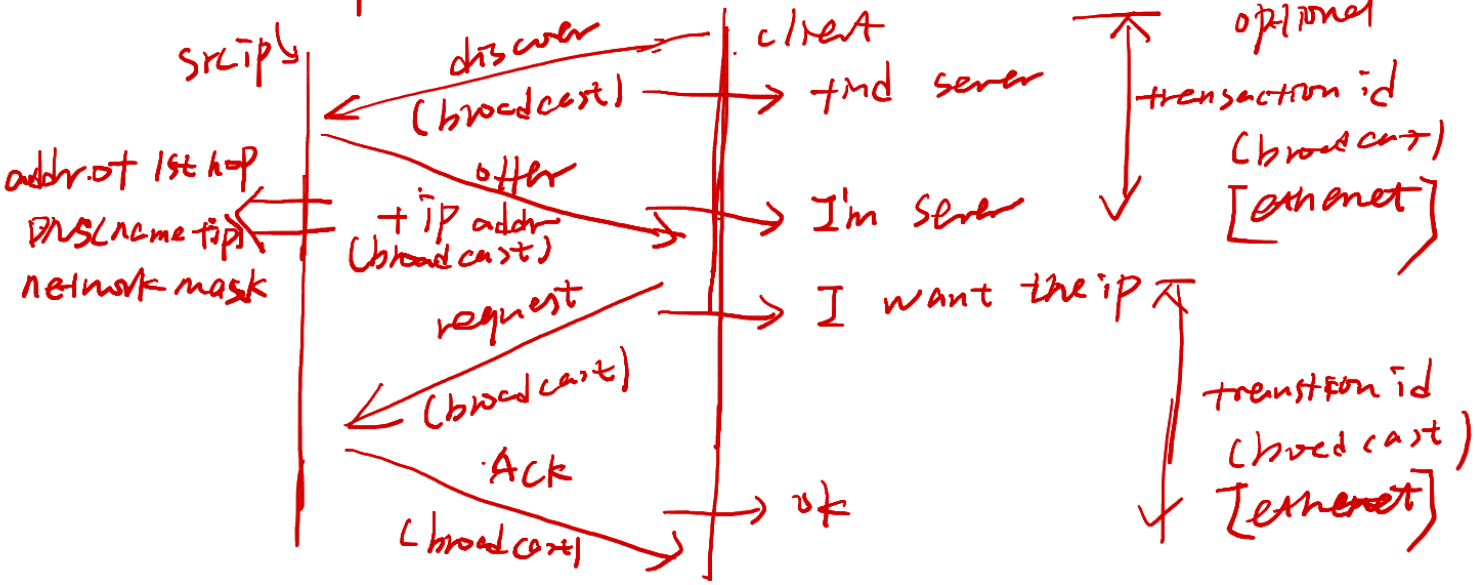
IP hourglass



get ip

host = · hard coded (/etc/rc.config)
② DHCP (UDP)
dynamic host configuration protocol
- plug and play
- reuse address
- discover (host, optional)
  offer (server, optional)
  request (host)
  ack (server)

DHCP server = in router typically

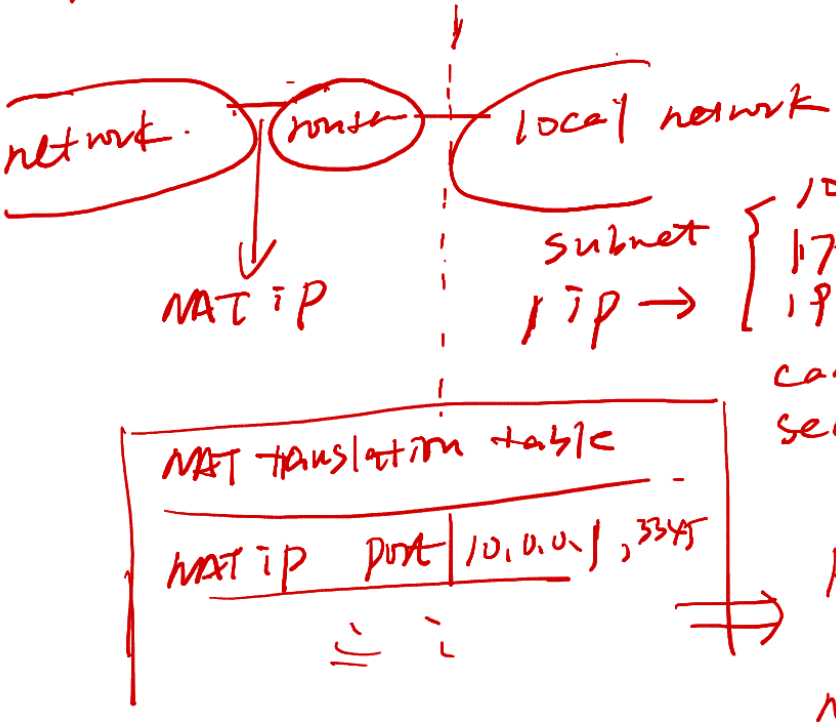


network = allocated portion of provider ISP's address space (mask)

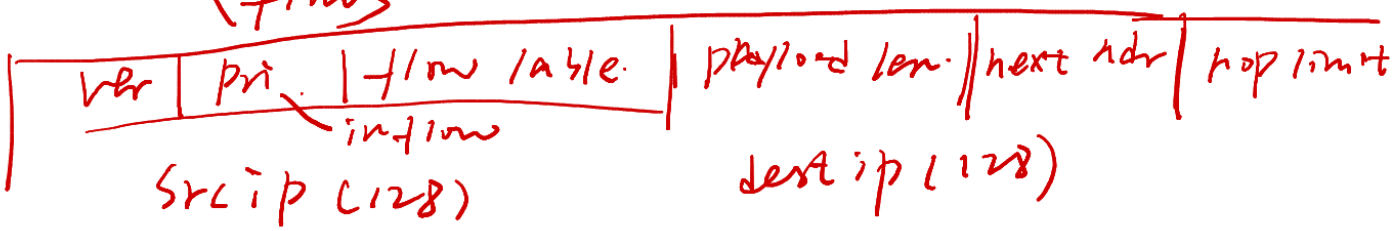
hierarchical addressing

ISP (block of addr): ICANN
(Internet corporation for assigned names and numbers)
through 5 regional registries (RIRs)
DNS root zone

NAT: network address translation [home, institution 4G/5G]

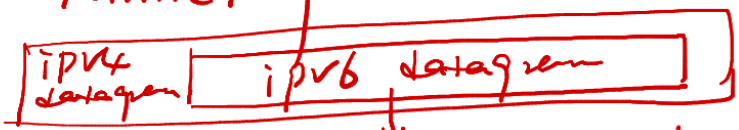


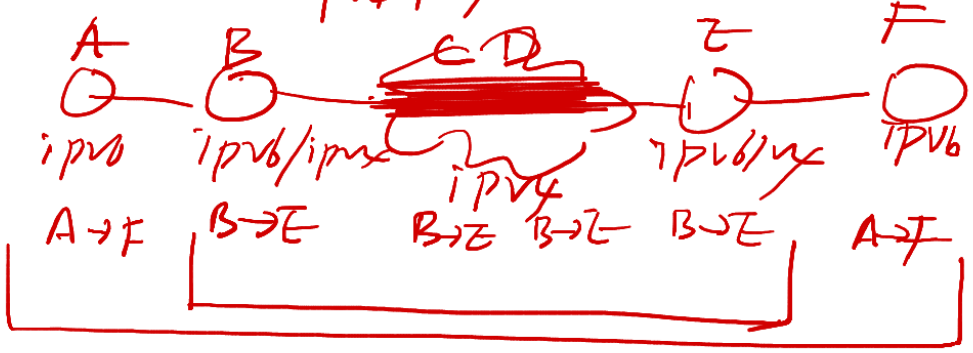
ipv6 :(40 bytes fixed header
flows



NO : checksum
NO : fragmentation/reassembly
NO : option

ipv4 ~ ipv6 : tunneling





generalized forwarding:

flow/forwarding table (match + action)

match, action, priority, counter

↓ (stats)

wildcard

- destination-based (IP, MAC)
- firewall (block)

Router: M = longest prefix IP
A = forward

Switch: M = MAC
A = forward, flood

Firewall: M = IP, TCP/UDP port
A = permit/deny

NAT: M = IP addr, port
A = remote addr, port

Middleboxes = NAT, Application specific (Firewall, load balance, cache, IDS)
with IP

Internet principle:

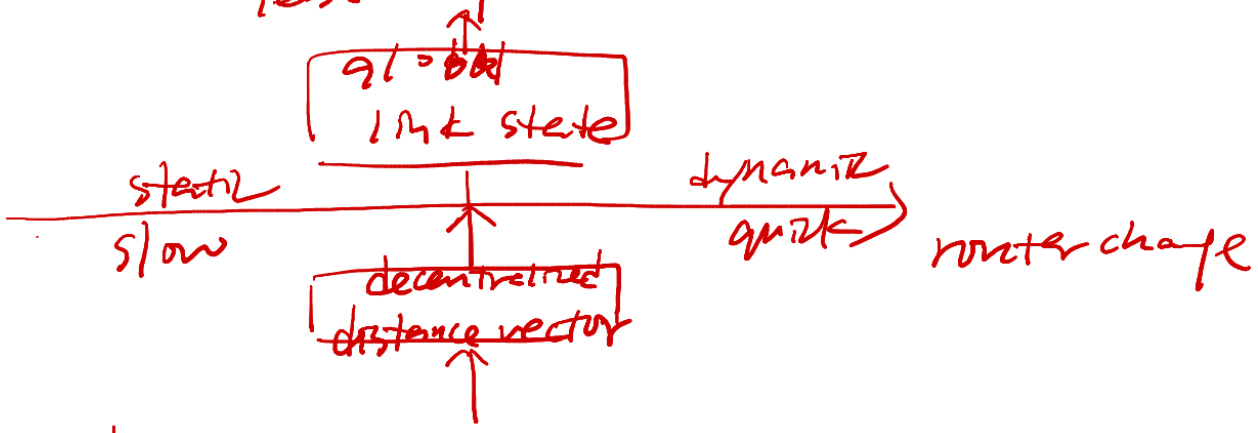
- simple connectivity
- IP
- intelligence at edge

End-end

- end-end
- end_hop by hop-end

routing protocols

"good" = least "cost" → defined by operator
fastest
least congested.



(LS)
dijkstra's
algorithm — one to all
$O(n^2)$. $O(n \log n)$

router broadcast link state info to n routers

$O(n^2)$
↓
router (message)

oscillation possible
bidirect with diff cost
(up stream, down stream)

(DV)
bellman-ford. (dp)

$$D_x(y) = \min_{neighbor} \{ C_{x,neighbor} + D_{neighbor}(y) \}$$

send DV when change → converge

travel fast but travel slow → routing loop

message	convergence	robustness
LS: $O(n^2)$	LS $O(n^2)$	LS: advertise incorrect link cost
DV: neighbor	DV varies	DV: advertise incorrect path cost (black holing) propagate error

Scale. & administrative autonomy

autonomous system (AS) domain.

intra-AS → formerly table ← inter-AS

gateway router

intra-AS routing:

(DV) RIP = routing information protocol

(LDV) EIGRP = Enhanced Interior gateway routing protocol

(LS) OSPF = open shortest path first

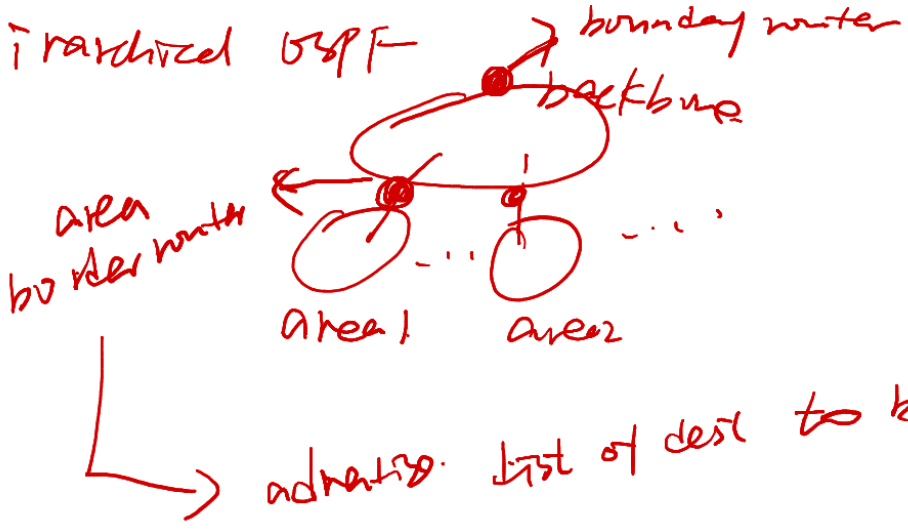
(IS-IS) ISO

OSPF = open shortest path first

(IP) ↳ publicly visible.

Link-state over IP

hierarchical OSPF



↳ advertise: list of dest to backbone

different inter/intra

policy. scale. performance.

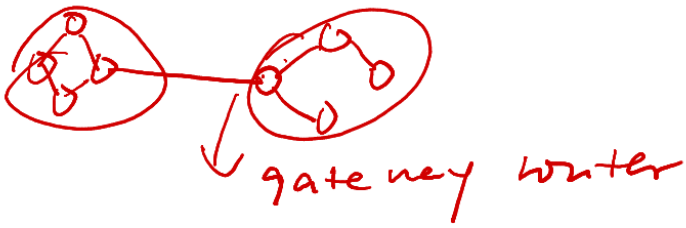
inter-AS routing

BGP (border gateway protocol) TCP/IP

eBGP = subnet reachability info

iBGP = propagate info

info + policy determine → import policy



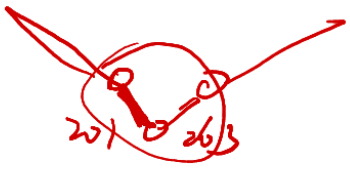
prefix + attributes

< destination, AS-PATH, next-hop >

BGP messages:
- open (TCP)
- update (advertise)
- keepalive (keep, Ack open)
- notification (error, close)

hot potato routing

local gateway least intra cost



Icmp (internet control message protocol)

(ip) host — router

network-level information { error report
echo request/reply (ping)

Icmp message: type, code, first 8 bytes of ip datagram

network management

server = application (human)

device = configurable hardware.

data = device state

network management protocol

CLI = ssh

SNMP / MIB

(udp) simple network management protocol

operator queries / sets device data (MIB)

Management information base

NETCONF / YANG

multi device.

data modeling language

SDN

specific kind of
load balancing
generalizes flow-based forwarding
programmable control application

- openflow
- opendaylight (ODL)
- ONOS

ch6. hosts, routers → nodes

link protocol : may/may not reliable

service framing
reliable
flow control
EDC
half-duplex, full-duplex

in NIC

EDC single bit parity
2-dimensional parity
Checksum (IP)
CRC $R = D \cdot 2^r \bmod G$

multiple access link
point to point link (PPP)
HDLC
broadcast link

Multiple access protocol

distributed algorithm

channel partitioning | random access | taking turns

Channel partitioning

— FDMA

— TDMA

random:

full rate R:

{ detect collision
recover from collision (delay, retry)

— ALOHA

no slotted

— slotted ALOHA

trans next slot

collision, random slots

— CSMA (carrier sense multiple access)

idle — trans

busy — defer

[distance, propagation delay]

— CSMA/CD

aborted when collision

ethernet

idle → ok

busy → abort, send jam signal

binary backoff (exponential)

n collision: random select $[0, 1, 2, \ldots, 2^n - 1]$

efficiency = $\frac{1}{1 + 5\frac{t_{prop}}{t_{trans}}}$

— CSMA/CA

taking turns

— polling

poll

C data controller

overhead (latency, single-point

— token passing

Cable access network

DOCSIS

data over cable service interface specification

FDM : upstream. downstream

TDM : upstream.

MAP [t1, t2]

MAC address (LAN, physical)

48 bit (ROM) / settable (SW)

base 16

ARP : address resolution protocol

Arp table in host / router

<IP, MAC, TTL>

A → B

A broadcast (AMAC, bdcstMAC, AIP, BIP)

B reply MAC (AIP BMAC

<BIP, BMAC TTL>

A — B

A AMAC ; R1 MAC, AIP BIP

R RRMAC., BMAC AIP BIP

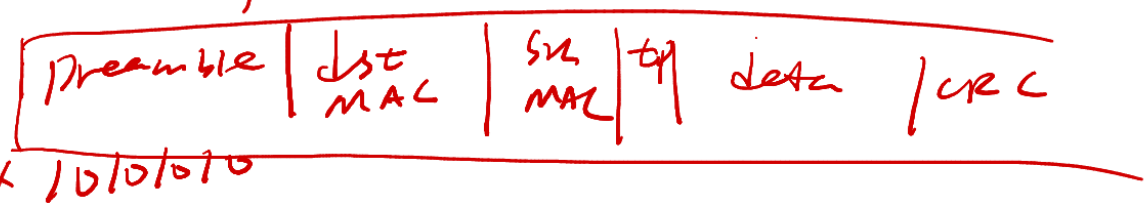
A — R — B

end of switch

ethernet (thick)

bus → switched

collide



7x 10101010

1x 10101011

- connectionless
- unreliable.
- CSMA/CD binary backoff

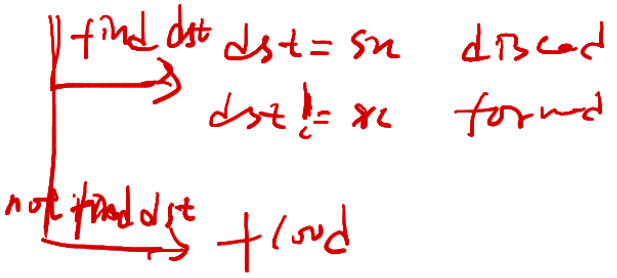
ethernet switch

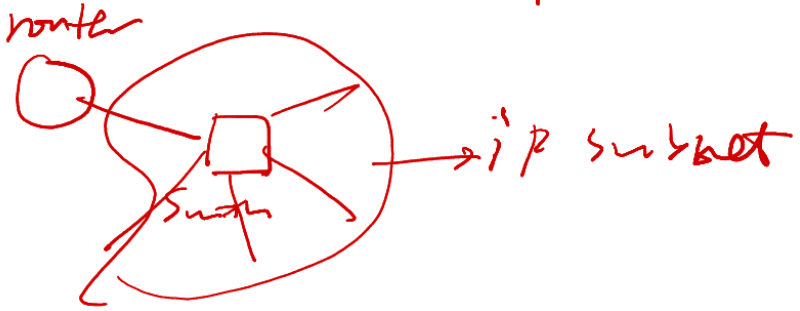
plug and play

↳ self learning

switch table record

- frame reach → (MAC, interface)





VLAN

broadcast: (ARP, DHCP, not learned MAC)

port-based VLAN

trunk port

VLAN ID

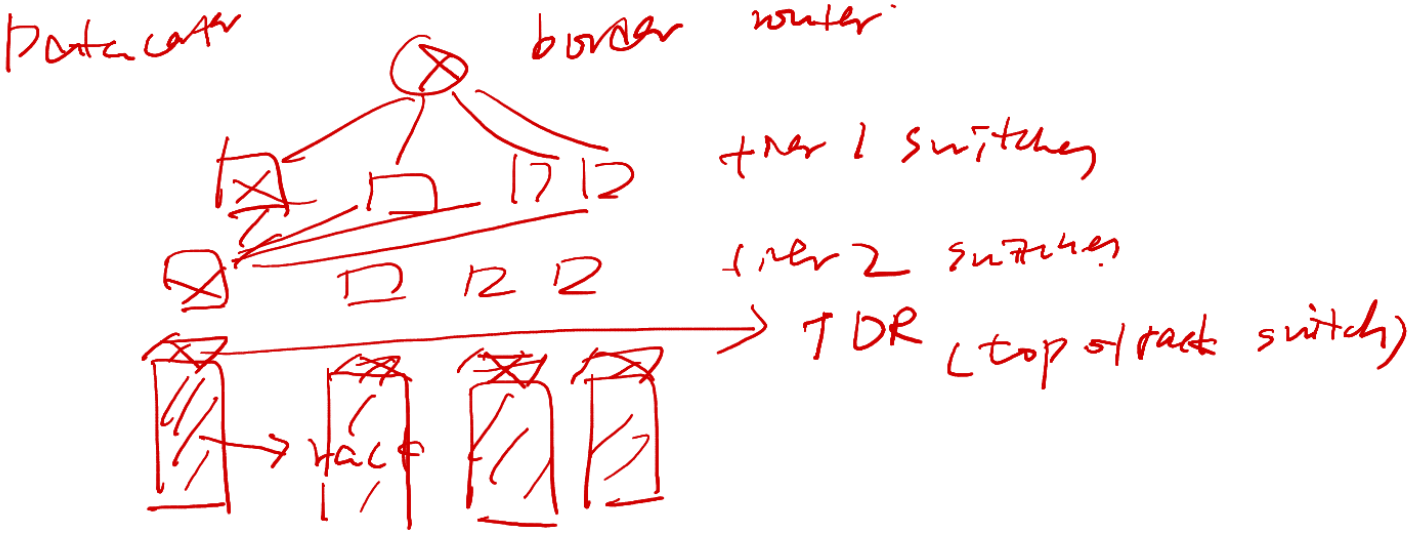
MPLS (multiprotocol label switching)

ip routing: based on dest ip

MPLS routing: { based on src & dest ip
generalized forwarding
fast response

RSVP-TE signaling

Optical-2s control plane
(LLDP)



DHCP

broadcast

client ip name & addr of DNS

first hop ip

MAC of router

ARP

routing (OSPF, BGP)

DNS

TCP

HTTP