

Computer Networking

A TOP-DOWN APPROACH

SEVENTH EDITION

KUROSE • ROSS



Computer Networking

—A Top-Down Approach

Chapter 1

Introduction

1.1 What is the Internet?

What's the Internet?

"nuts and bolts" view

- Connected computing devices
- Communication links
- Packet switches

Internet:

interconnected ISPs

protocols

standards: RFC (by IETF)

What's a protocol?

protocols define {
format
order
actions

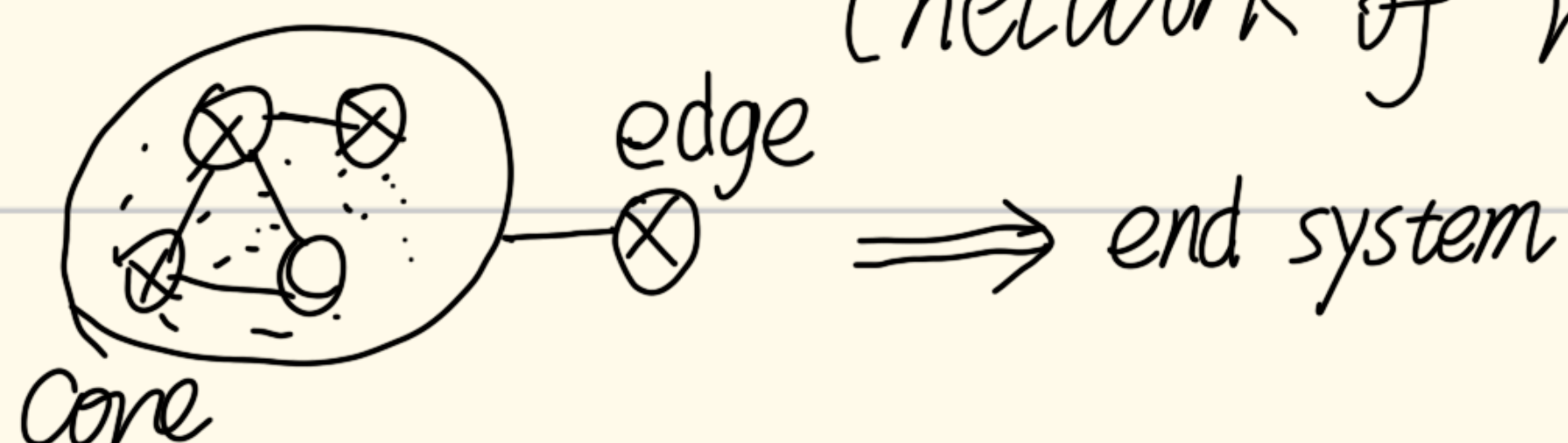
1.2 Network Edge

Network structure

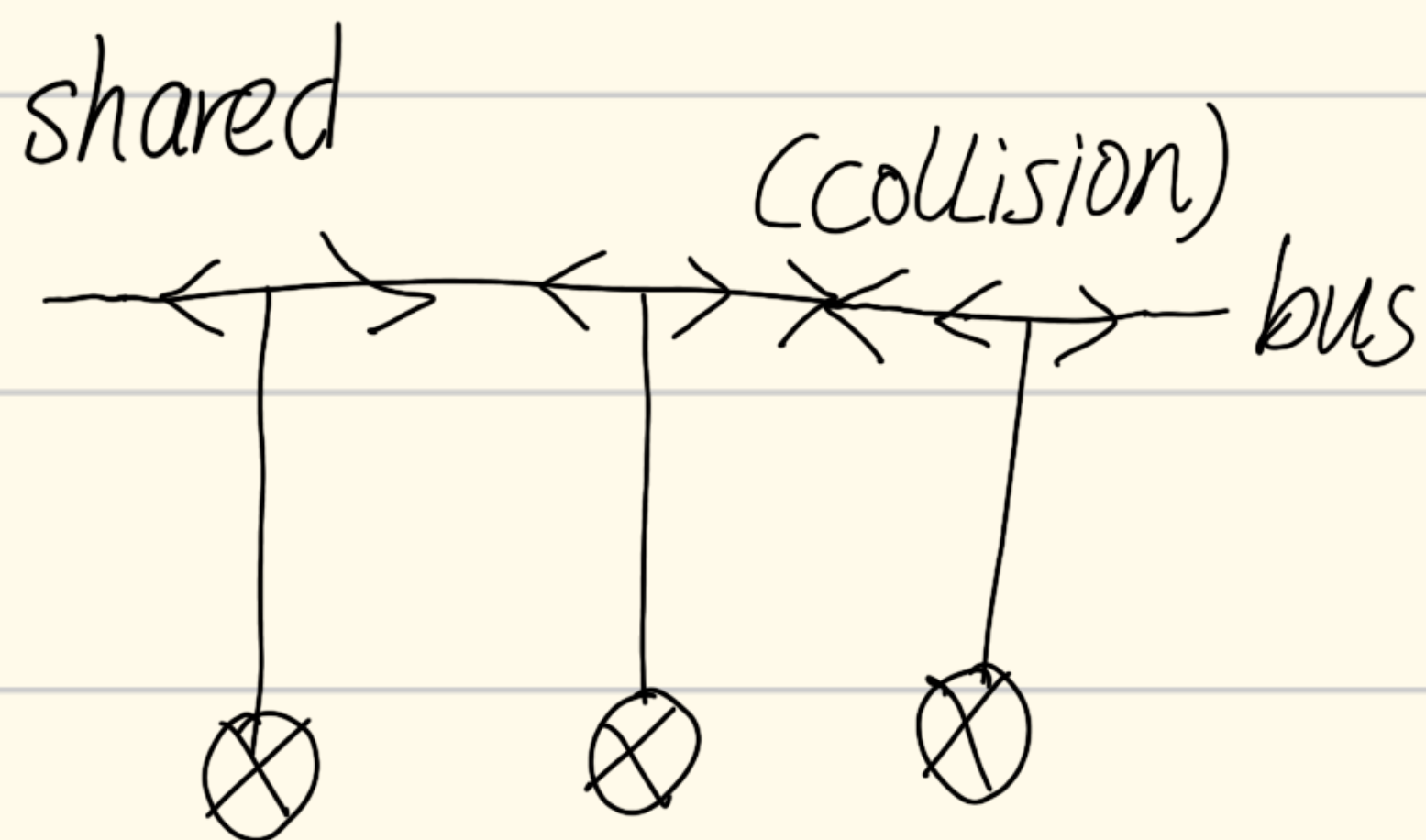
network edge: hosts (clients and servers)

access network, physical media

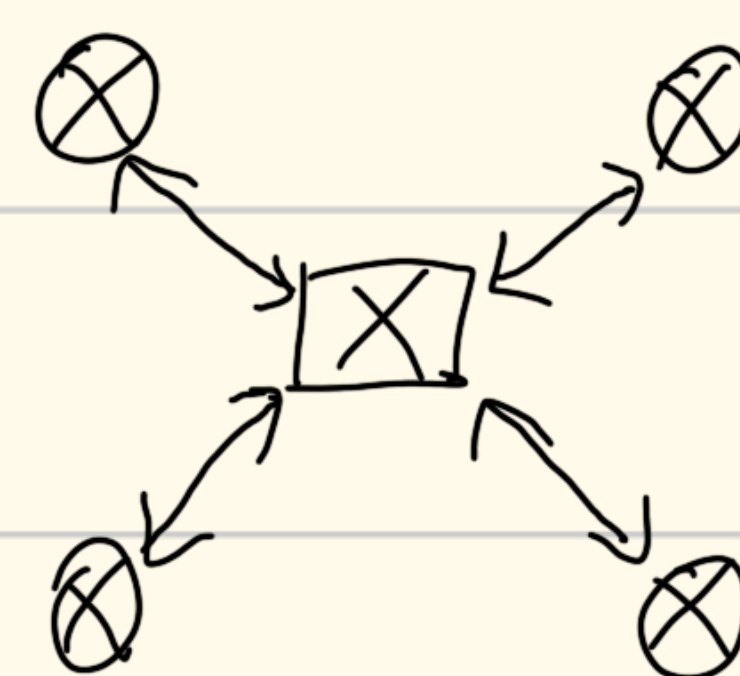
network core: interconnected routers
(network of networks)



Shared or dedicated



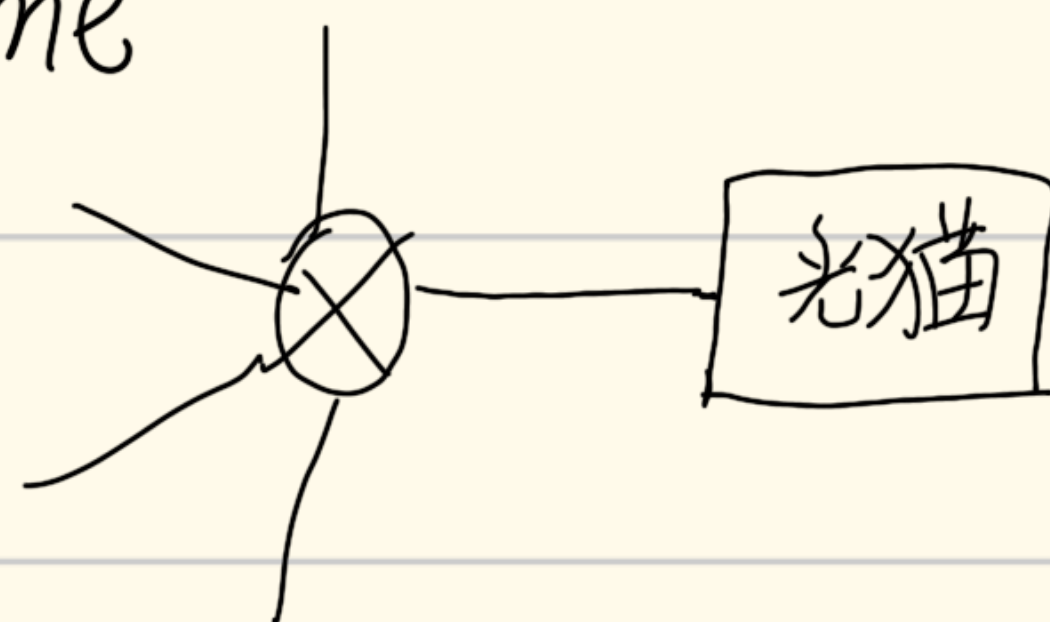
dedicated



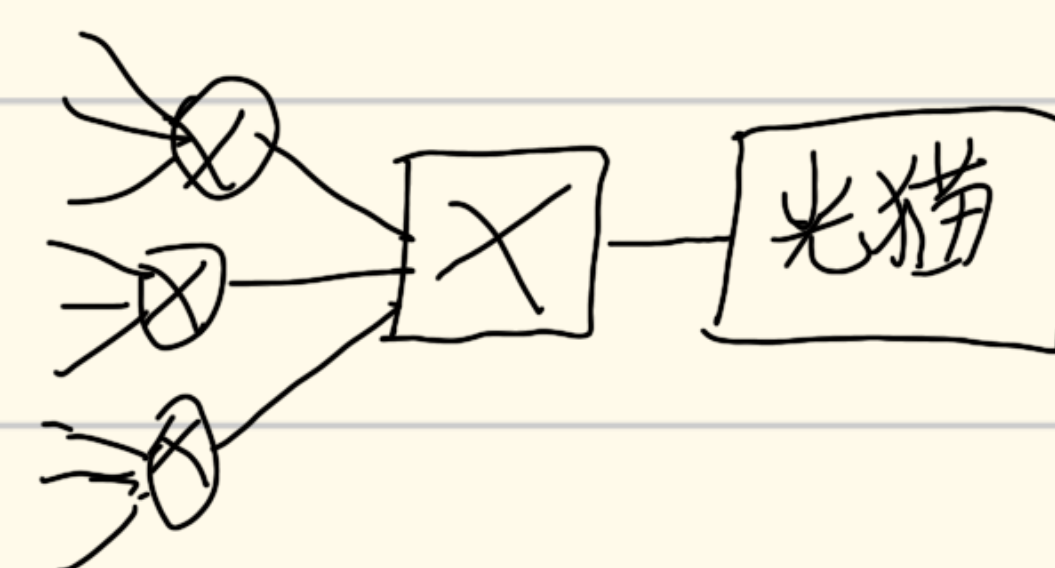
Access network

home or enterprise

home



enterprise



wireless access network

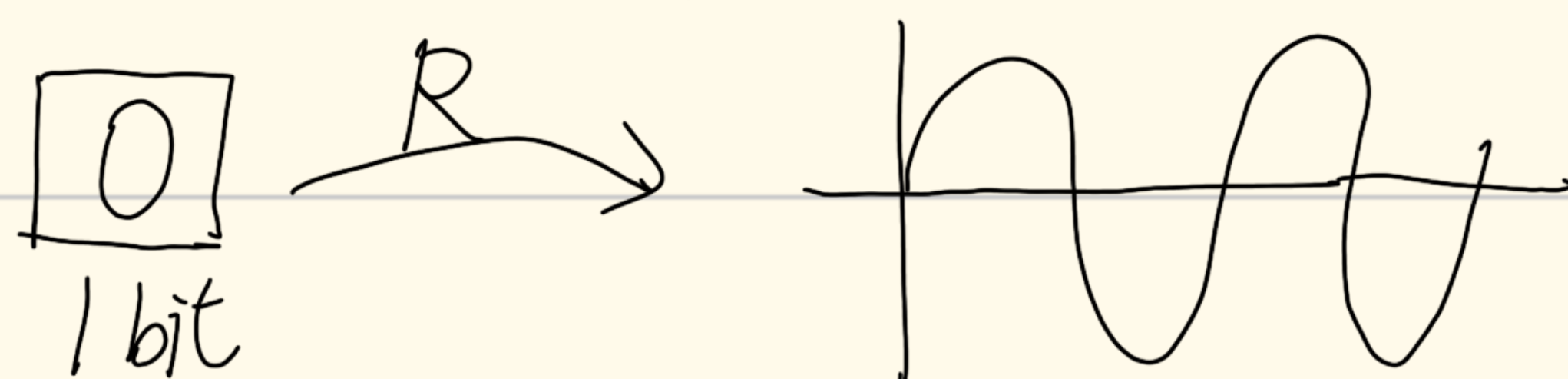
WLAN: wireless local area network

蜂窝通讯: wide-area wireless access

Hosts

transmission rate R

(aka link capacity, bandwidth)



Physical media

twisted pair (TP) $\left\{ \begin{array}{l} \text{Category 5} \\ \text{Category 6} \end{array} \right.$

1.3 Network Core



Packet-switching:

store and forward

store:

entire packets arrive $\rightarrow$ transmit on next link

forward:

$\frac{L}{R}$ (s) to transmit L -bit packet into link at R bps
queueing delay and loss

if arrival rate exceeds R :

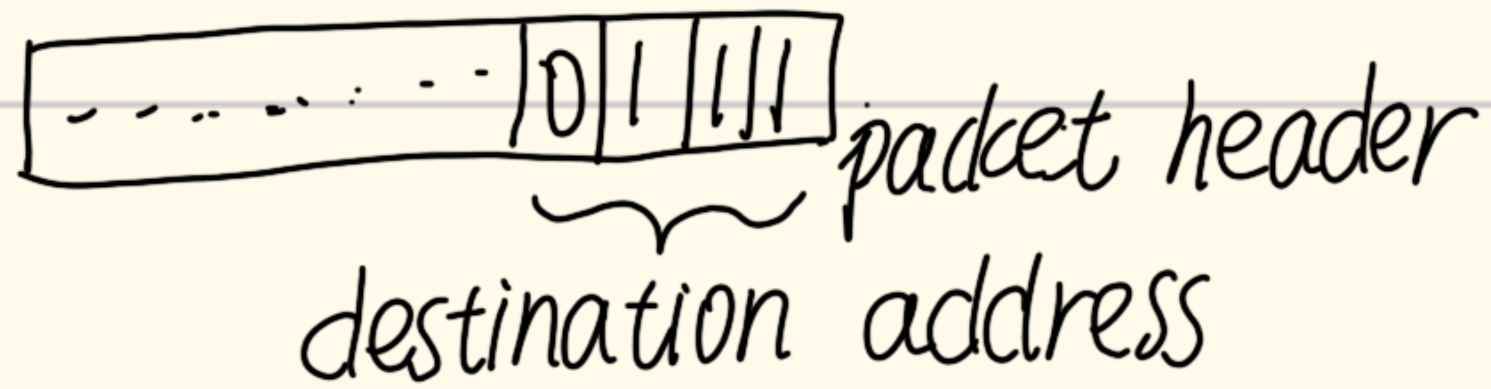
queue, waiting to be transmitted

be dropped if buffer fills up (FIFO 去head; 去tail; 加权算法)

2 key functions as network-core

routing

forwarding



Circuit-Switching:

dedicated resources

no sharing: circuit segment idle if not used by call
end-end resources allocated to + reserved for
"call" between src & dest

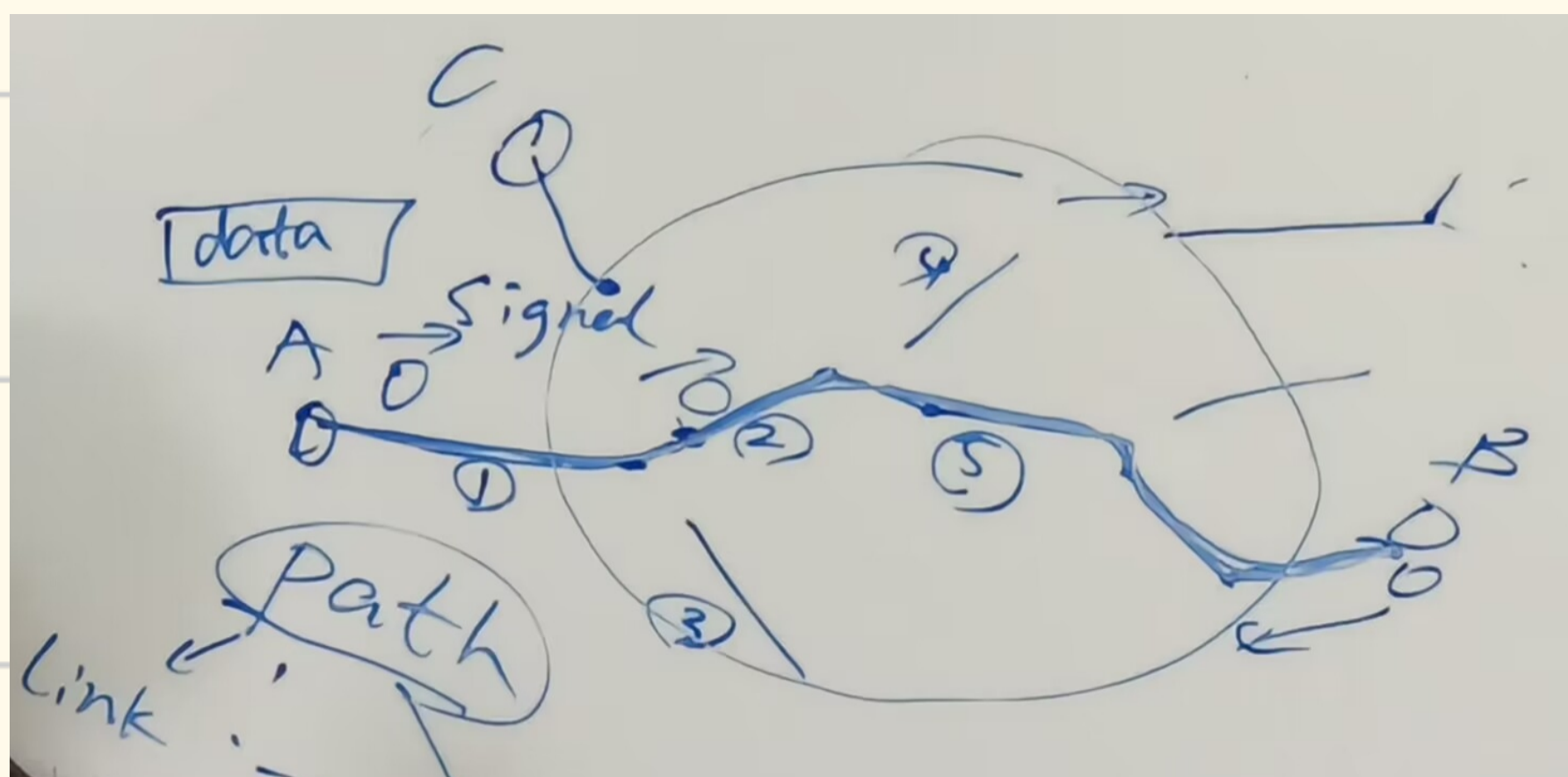
Difference:

circuit-switch:

dedicated

单点失效, 全部重连

call setup

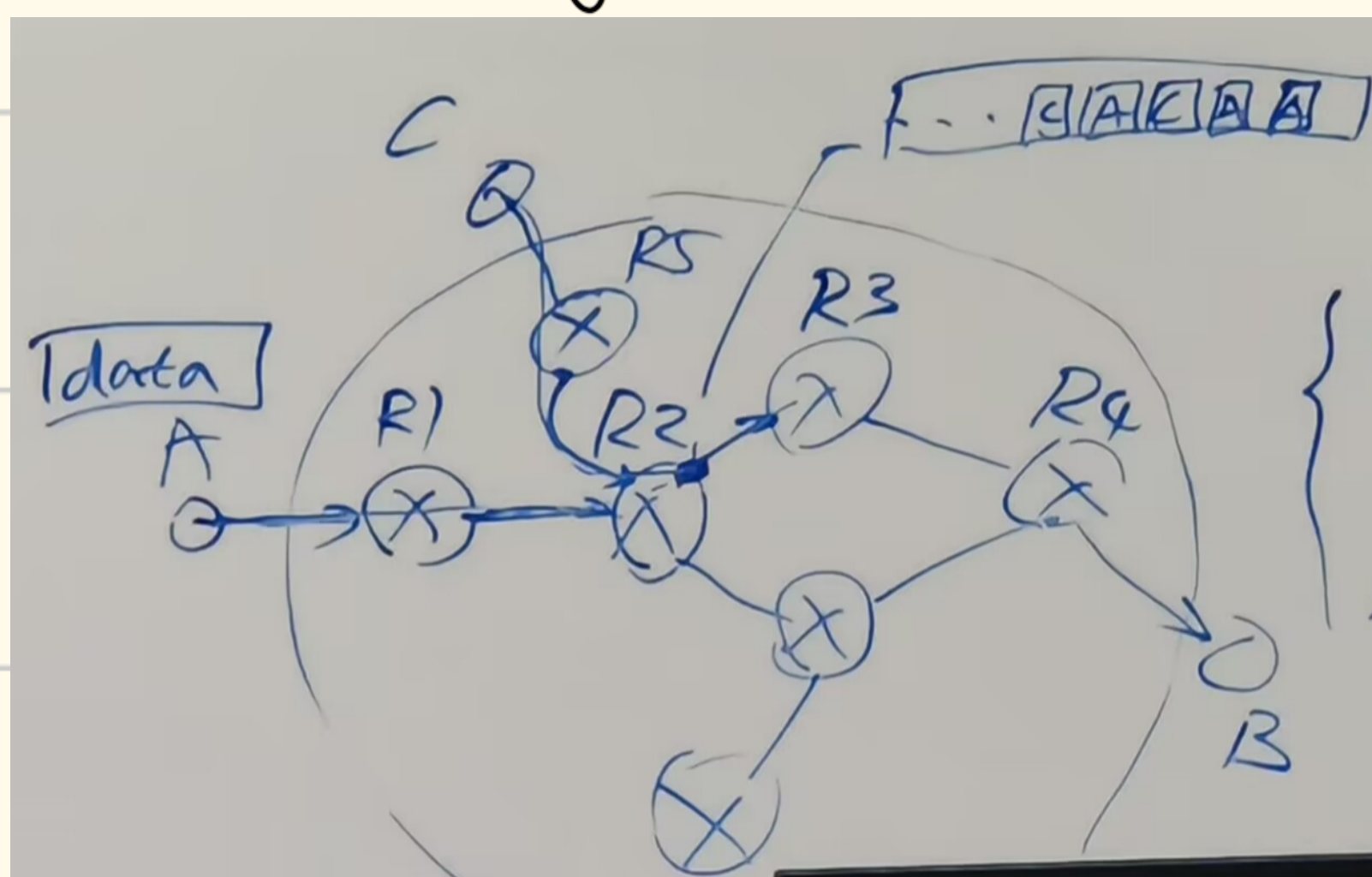


packet-switch:

delay and loss

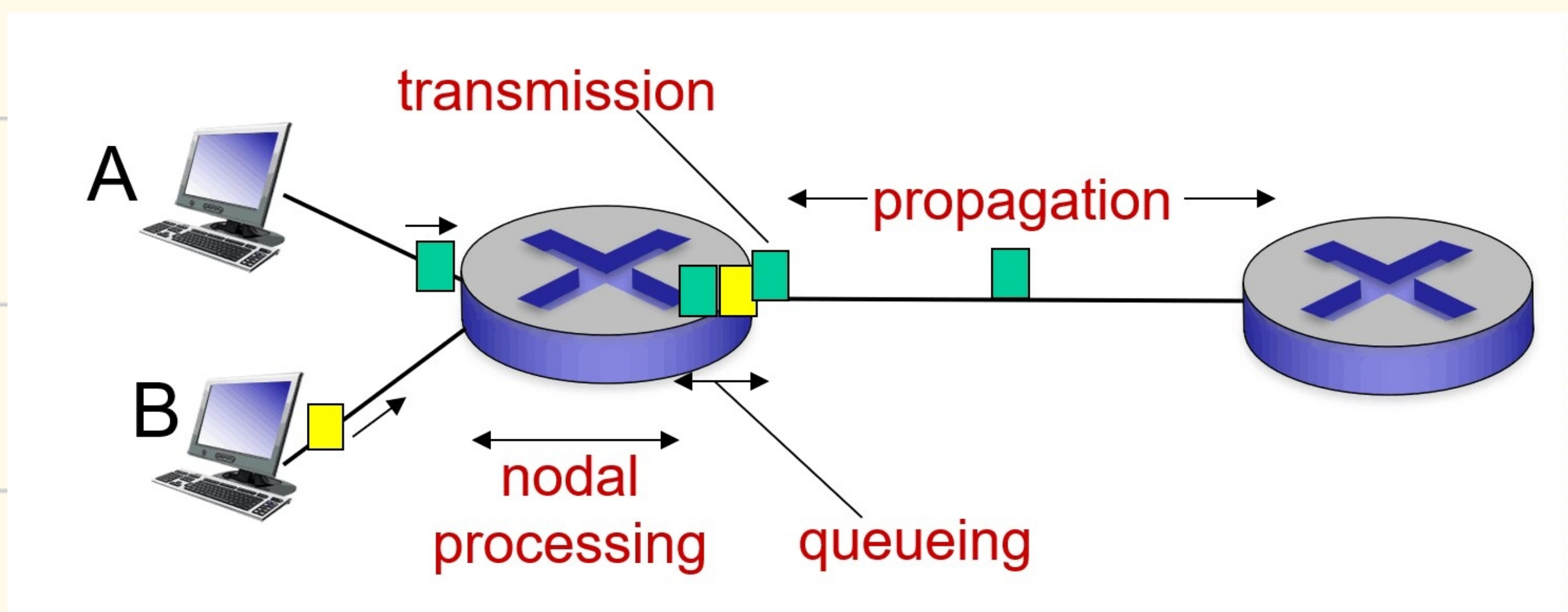
congestion control

arrive out of order



1.4 Delay, Loss, Throughput ☆

Four sources of packet delay



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

① nodal processing

check bit errors (CRC + checksum)

determine output link

② queueing delay

depends on congestion

③ transmission delay

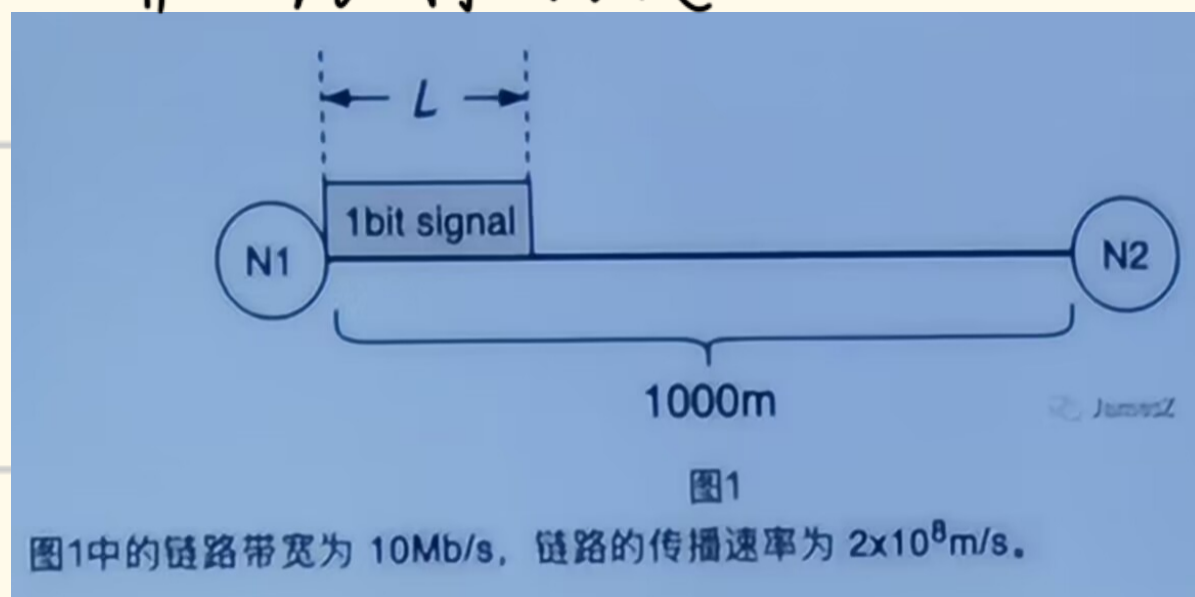
$$d_{\text{trans}} = \frac{L(\text{bit})}{R(\text{bps})} = \frac{\text{packet length}}{\text{bandwidth}}$$

④ propagation delay

$$d_{\text{prop}} = \frac{d(\text{m})}{S(\text{m/sec})} = \frac{\text{physical link length}}{\text{propagation speed}}$$

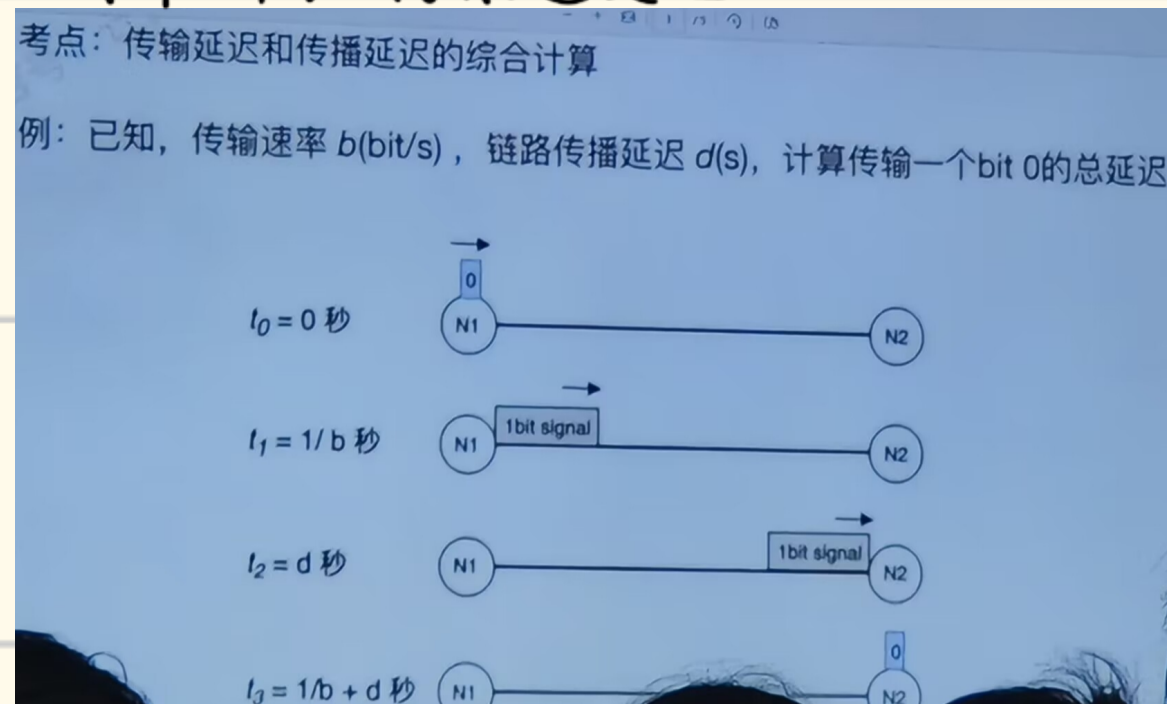
Examples:

计算一个比特的长度

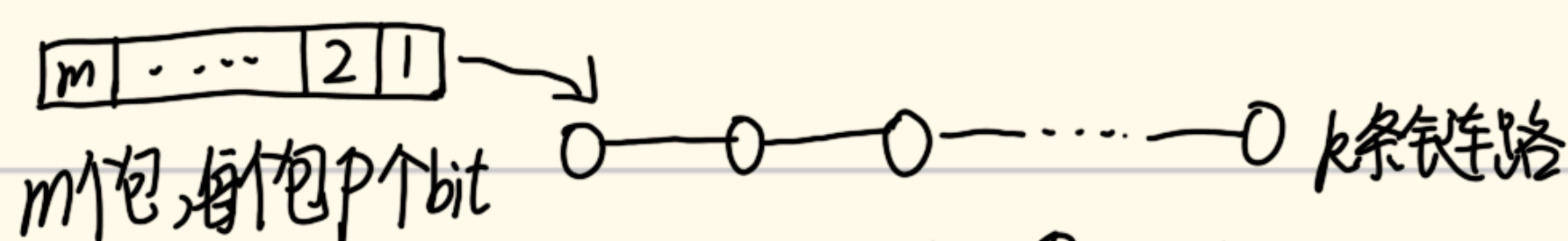


$$L = \frac{1(\text{bit})}{R(\text{bps})} \cdot S(\text{m/sec})$$

计算一个比特的总延迟



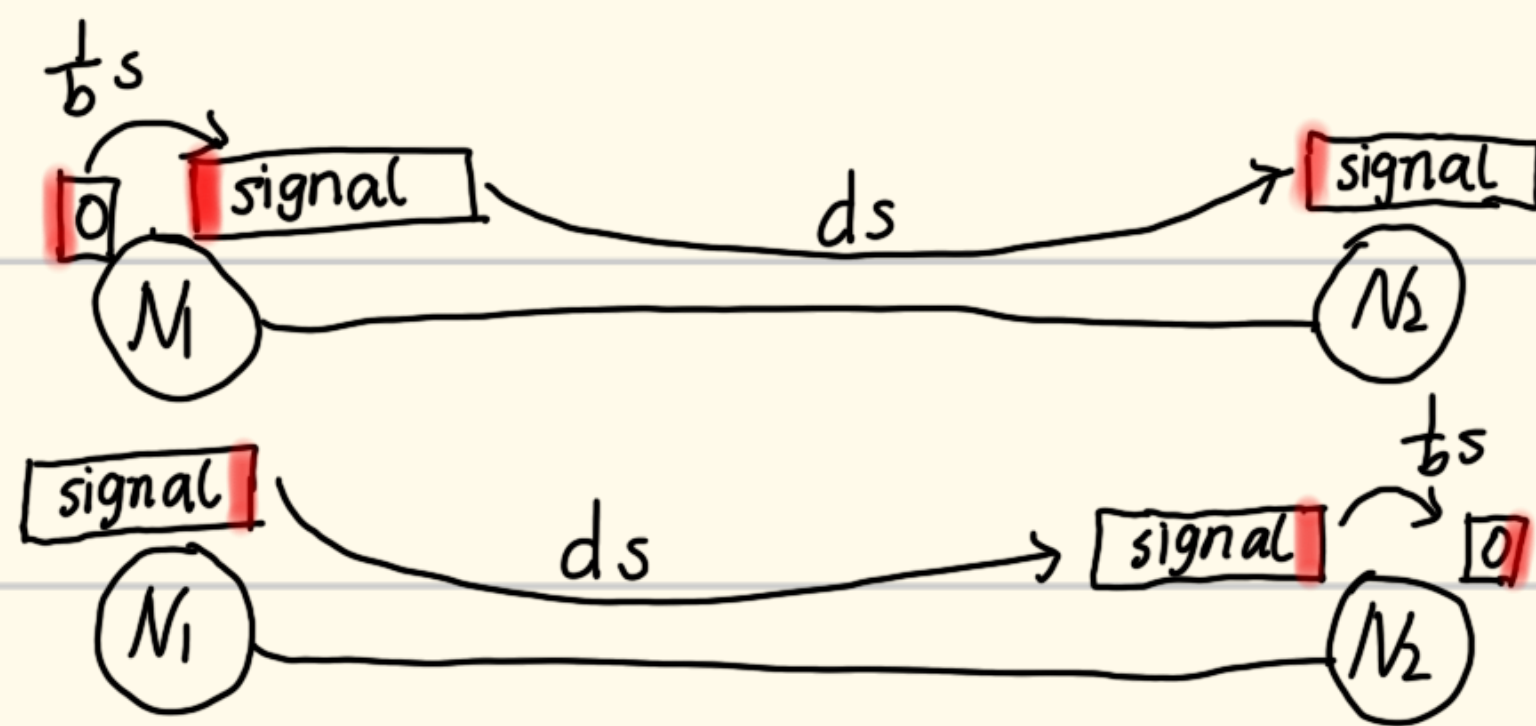
注意中转节点处等待数据包
全部接收后才能转发的时间
bit-error-checking



$$1 \text{ 号包到达: } \left(d + \frac{p}{b}\right) \times k$$

等待 p 个 bit 全部到达

$$\text{等待 } m \text{ 个包全部到达: } \frac{p}{b} \times (m-1)$$



注 {
 网络速度 $G \xrightarrow{10^3} M \xrightarrow{10^3} K$
 文件速度 $G \xrightarrow{2^{10}} M \xrightarrow{2^{10}} K$

带宽时延积

考点：带宽时延积
 带宽时延积 bandwidth-delay product
 James.Z JamesZ 5天前

图1
 图1中的链路带宽为 10Mb/s，链路的传播速率为 2×10^8 m/s。

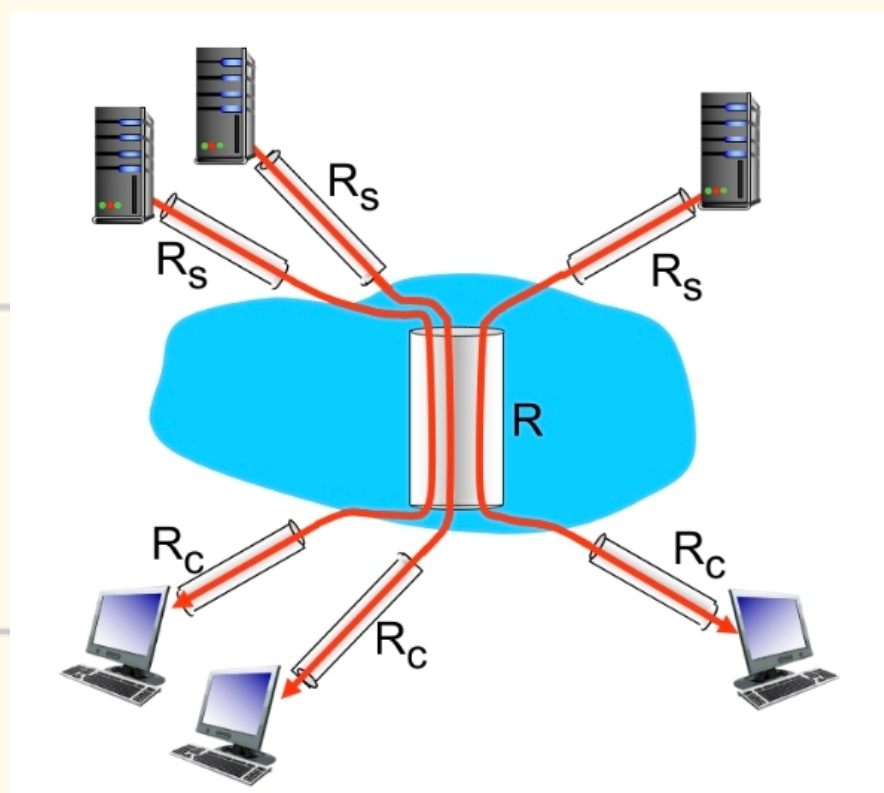
Question :
 上述场景中，N1节点向N2节点连续发送一个大小为1G bits的文件。
 请问，
 1) 从N1开始发送到N2收到完整文件一共需要多少时间？
 2) 在文件传输过程中的任意时刻，出现在链路上bit的最大数目是多少？

$$\text{product} = d_{\text{prop}} \times R$$

转换 | bit 时间 = $\frac{1}{R}$ s
 在 propagation 内转换比特数 $\frac{d_{\text{prop}}}{\frac{1}{R}} \text{ bit}$

Throughput

Rate at which bits transferred between sender & receiver
 (单位时间通过一个系统的比特数)



$$\text{throughput} = \min\{\text{bandwidths in system}\}$$

Real Delay

"traceroute"

将数据包中防止循环转发的TTL字段由32强制置1
 3 probes $\rightarrow$ average

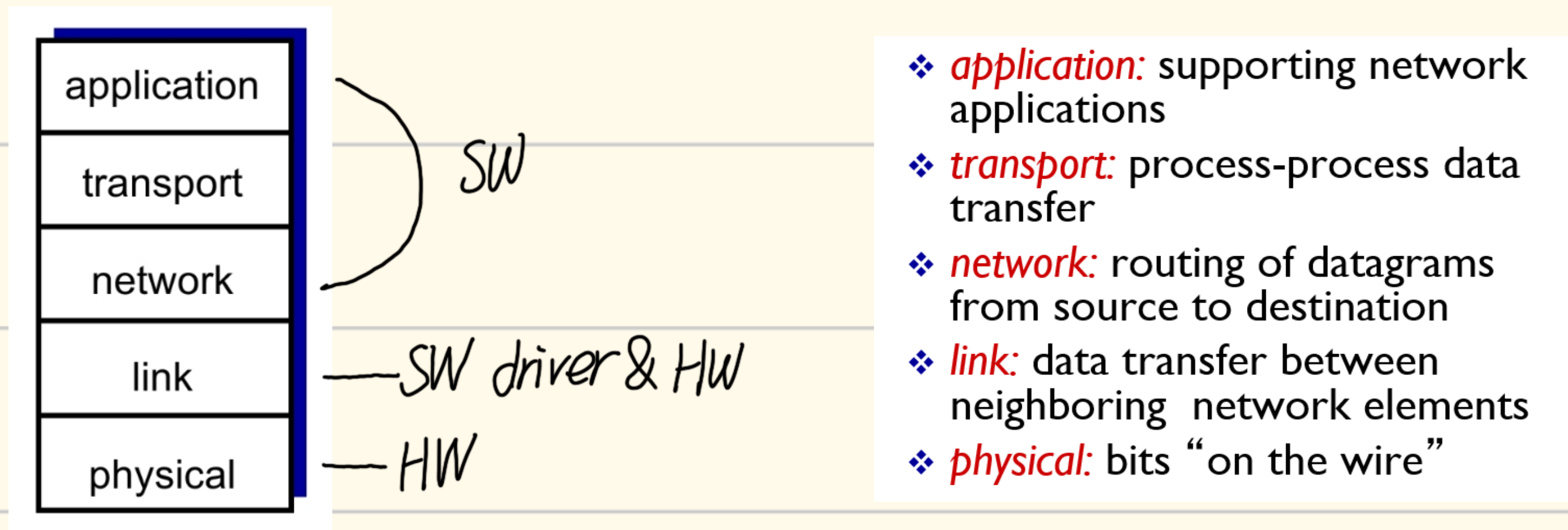
1.5 Protocol Layers, Service Models

Why layering?

allows identification, relationship of complex system's pieces
erase maintenance, updating of system

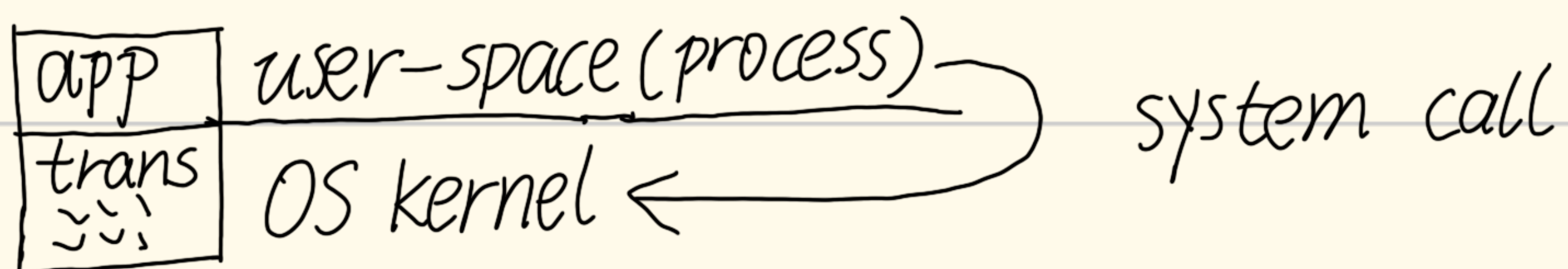
— considered harmful:
data redundancy
repeated function

Internet Protocol Stack



application layer:

business



transport layer:

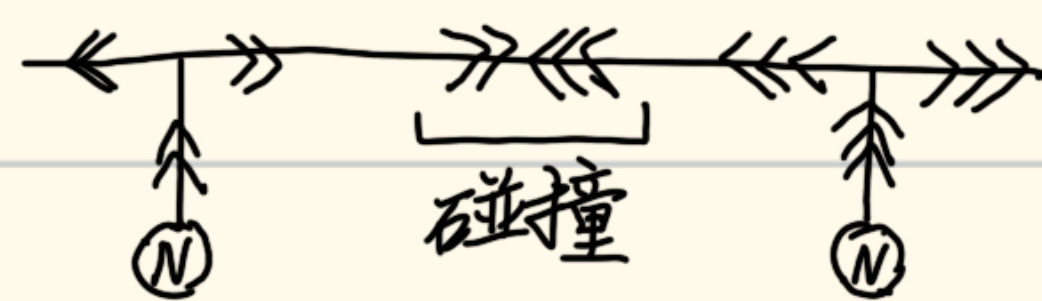
解决丢包问题 (queueing, TTL=0, bit error checking)
解决乱序到达的 sequence number gap

network layer:

routing (Dijkstra Algorithm)

link layer:

data transfer between neighboring network elements
解决共享链路的监听、碰撞检测与重发 (二进制退避算法)



补充: 二进制指数退避算法

Step 1:

将到站往返时间 $2t$ (冲突窗口、争用期) 作为基本退避时间

Step 2:

$k = \min\{\text{此时冲突次数}, 10\}$

Step 3:

从序列 $\{0, 1, 2, \dots, 2k-1\}$ 中任选一数 R , 等待时延为 $2t \cdot R$

当冲突次数超过 16 次后, 发送失败

解决链路速度不匹配问题

(N)
1Gbps

(N)
10Mbps

不会丢包, 只会降速
(路由器间速度不匹配会丢包)

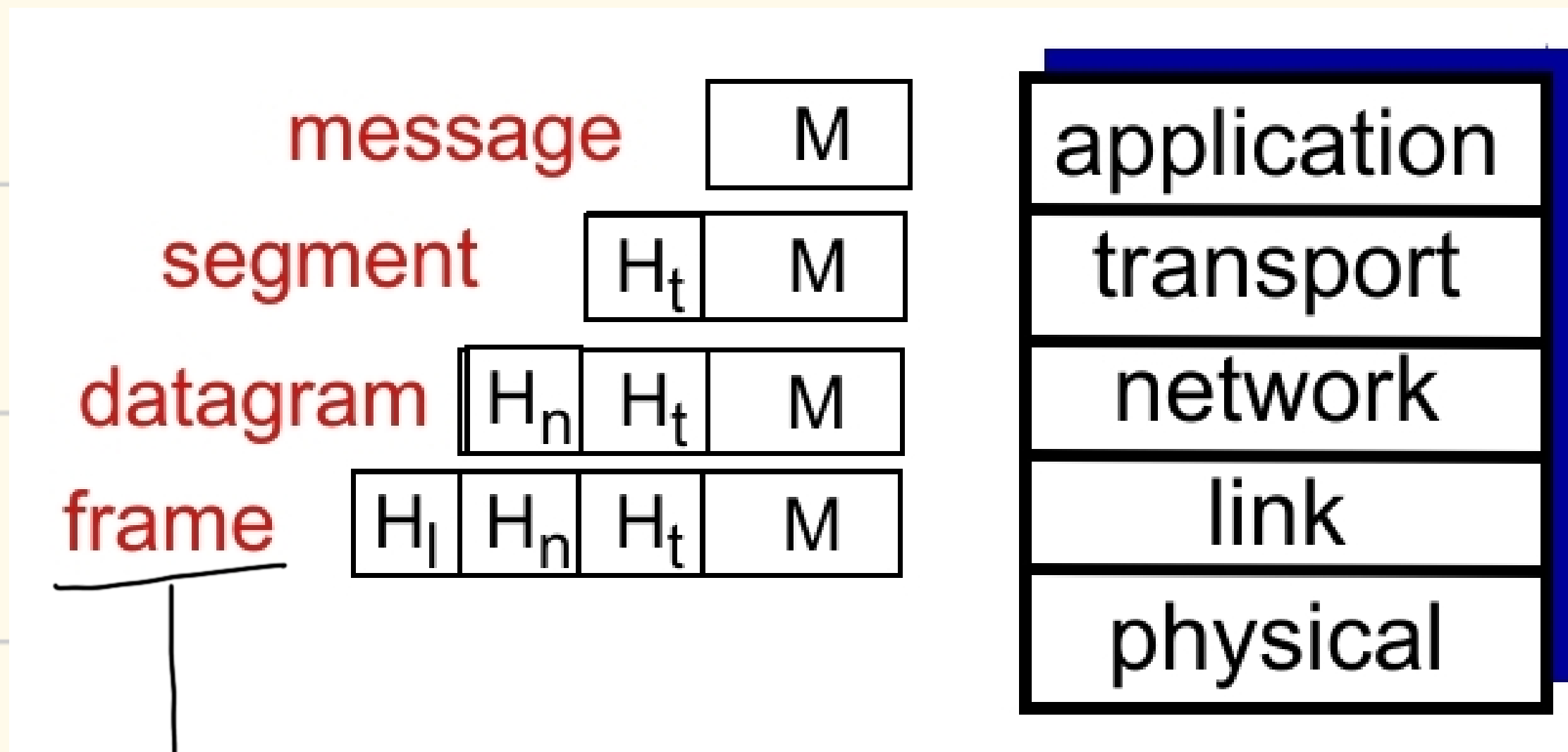
physical:

解决信号的调制、解调 (NIC)

0/1 $\longleftrightarrow$

Encapsulation

for a datagram:



在链路中传输的数据包均为 frame

1.6 Network Under Attack: Security (X)

Securing TCP

TCP & UDP = no encryption, traverse in cleartext

Securing TCP:

secure socket layer (SSL)
transport layer secure (TLS)

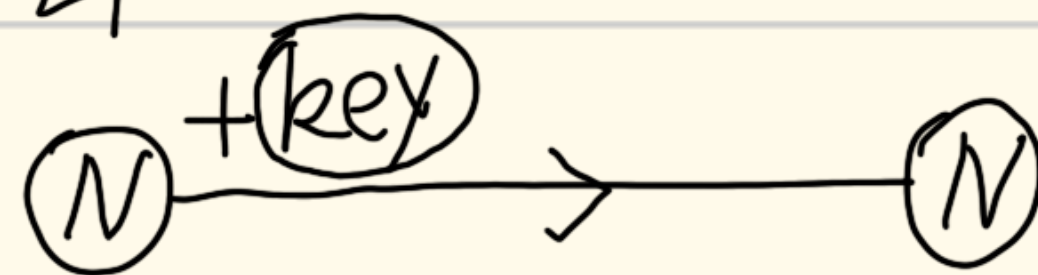
Digest (hash) Algorithm:

md5, sha

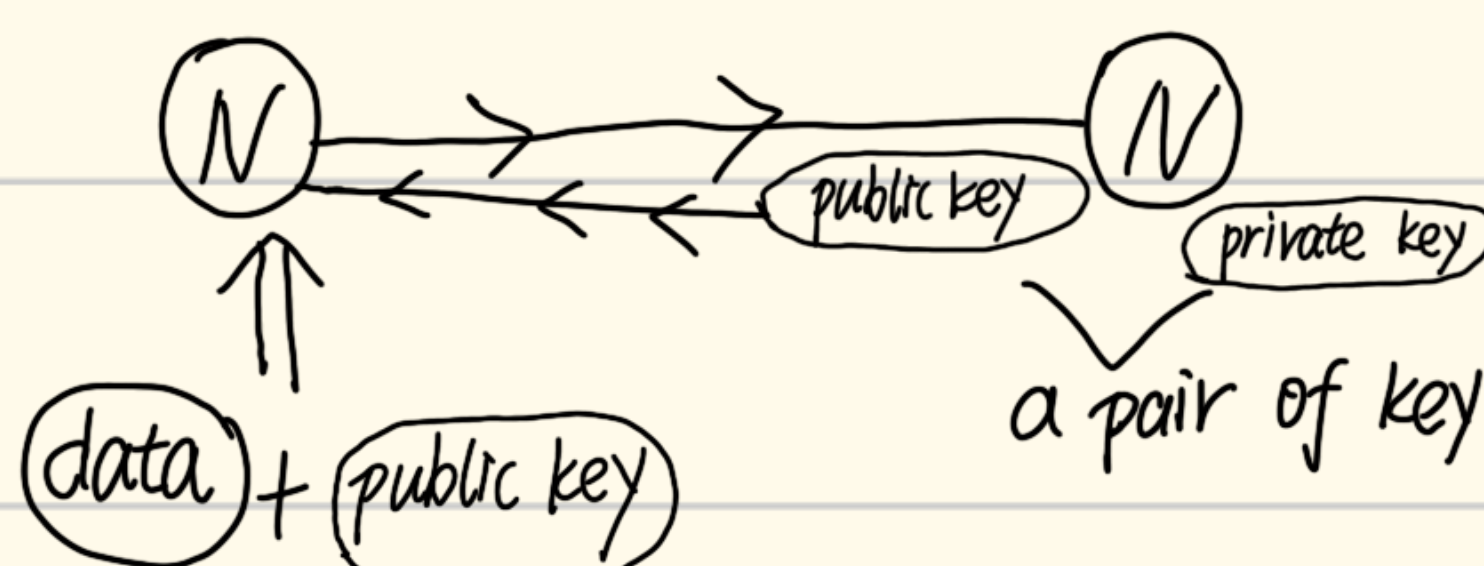
(features = 防篡改、不可逆、速度极快)

应用 = 用户密码不存储, 只存加 salt 密码的 hash value

对称加密



非对称加密



Chapter 2

Application Layer

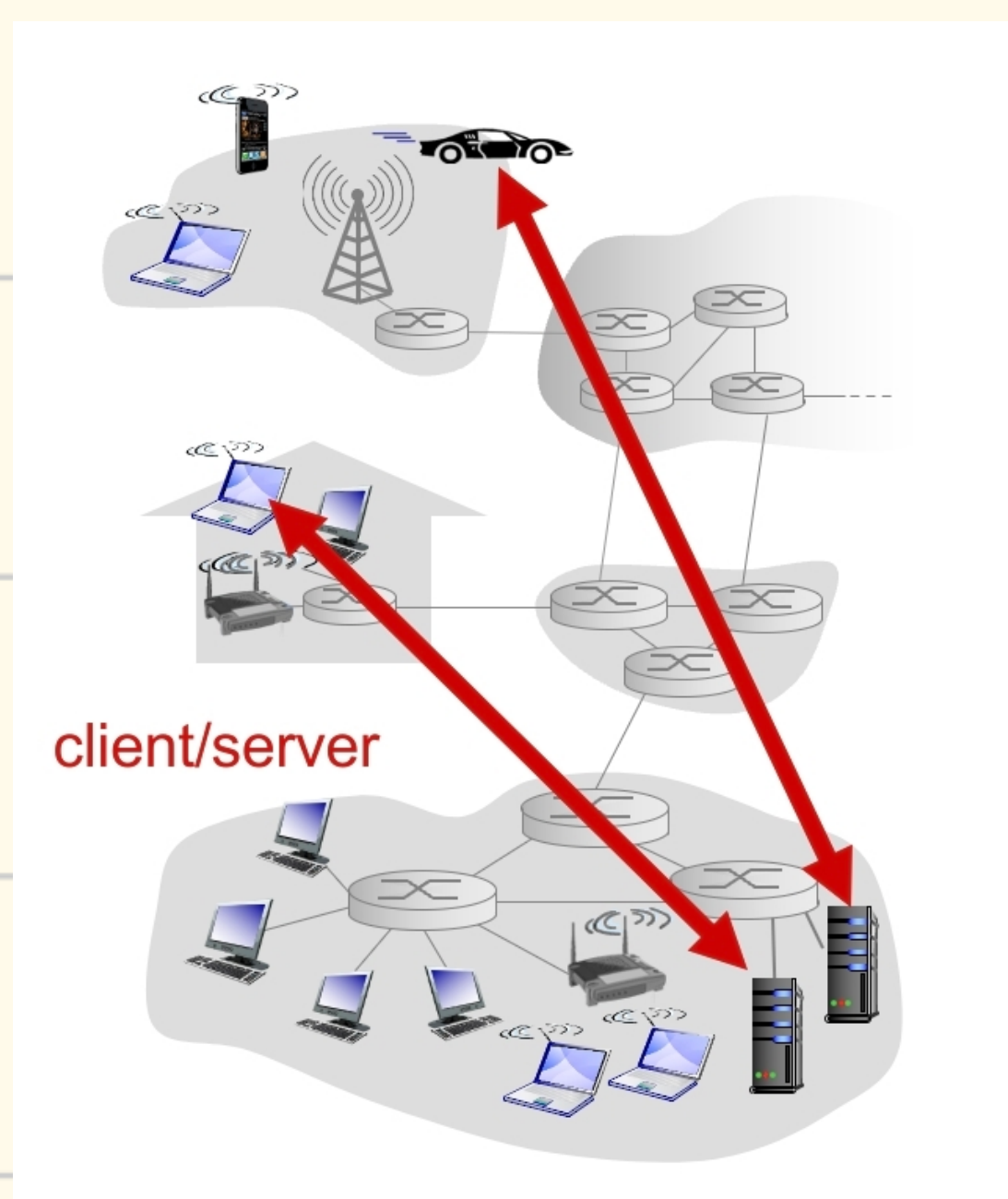
2-1 Principle Of Network Applications

Structure of application:

Client - Server

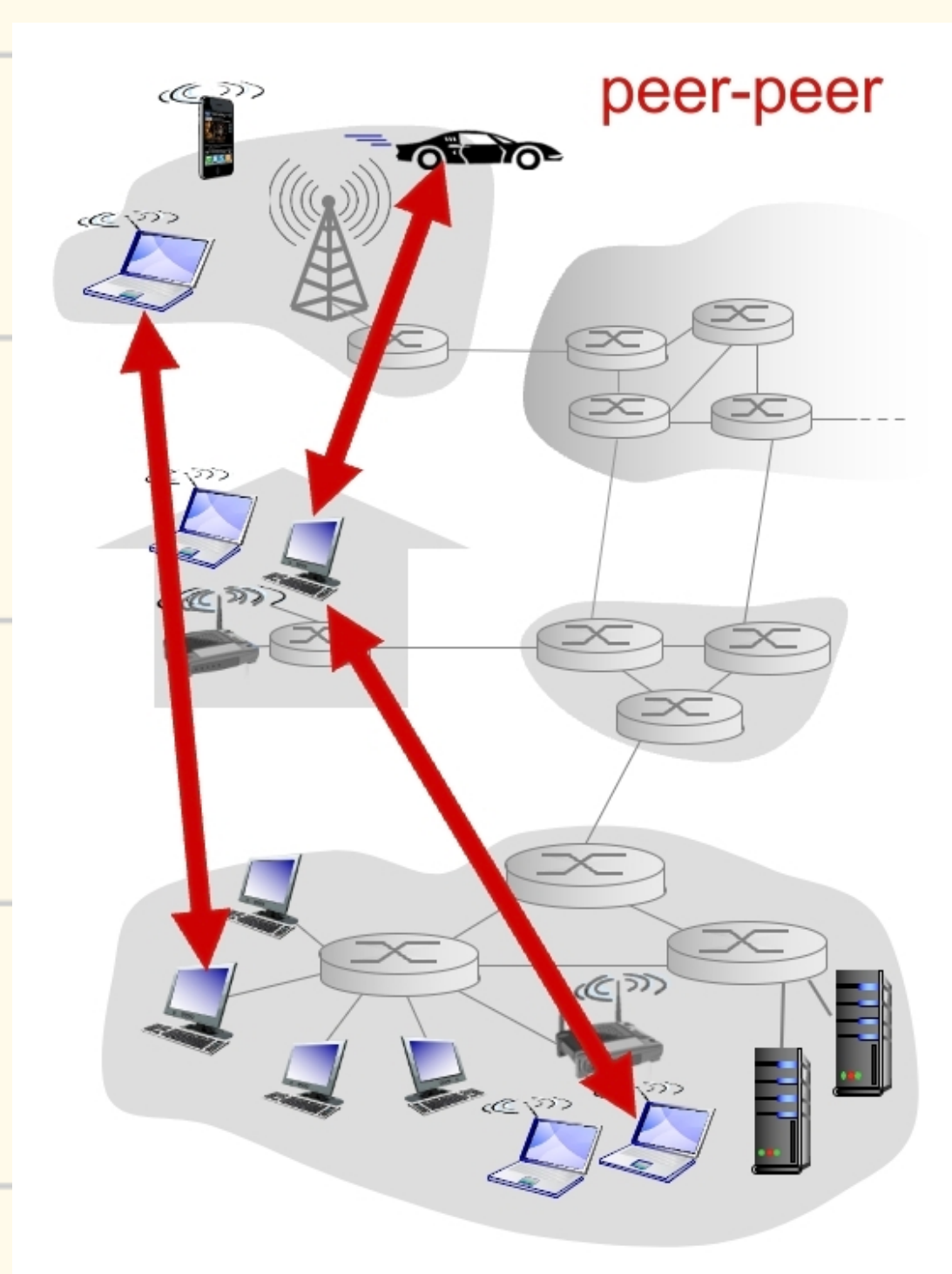
Peer-to-Peer (P2P)

Client-server Architecture



client/server 是就 process 而言的
client: 发起连接请求
server: 接收连接请求
(如 XunLei 既是 client, 又是 server)

P2P Architecture



Process Communicating

inter-process communication (IPC): pipe, shm, FS

process in different hosts: exchanging messages

Internet transport protocols services

TCP service:
reliable transport
flow control
congestion control

UDP service:
unreliable data transfer

2.2 Web and HTTP ☆

Web objects

HTML files

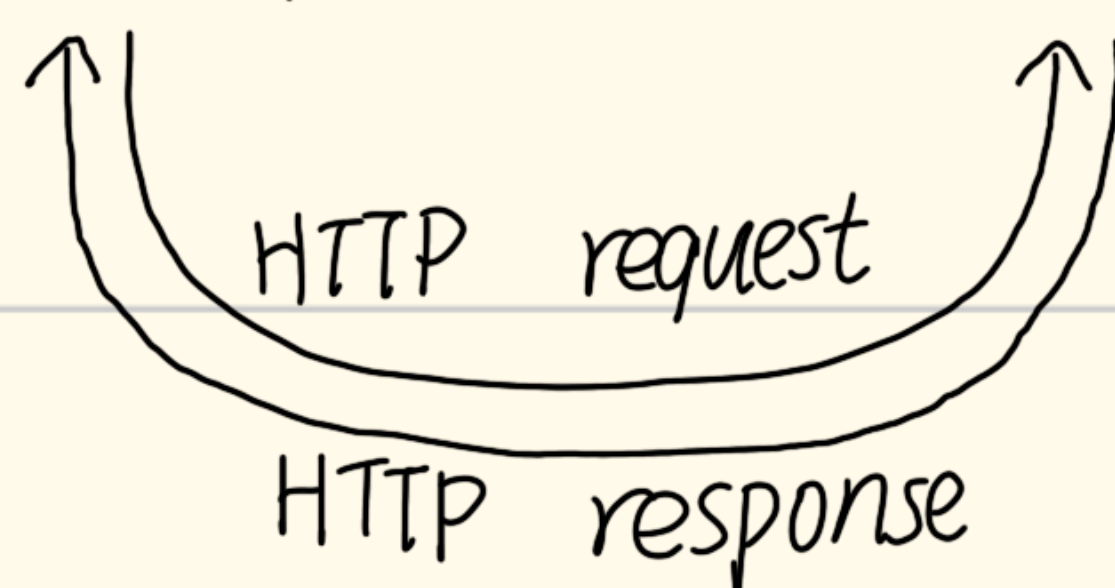
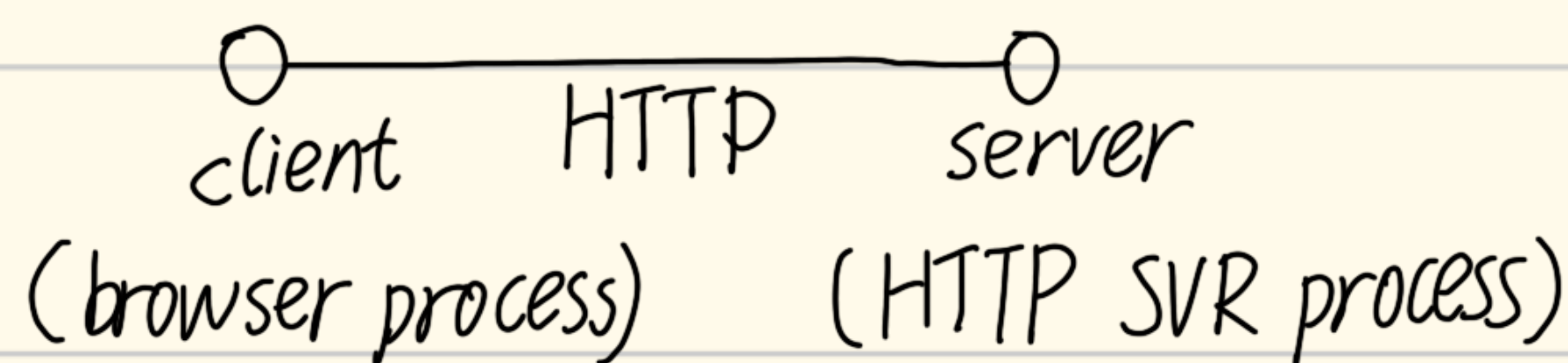
image files

CSS files

JS files

HTTP overview

hypertext transfer protocol

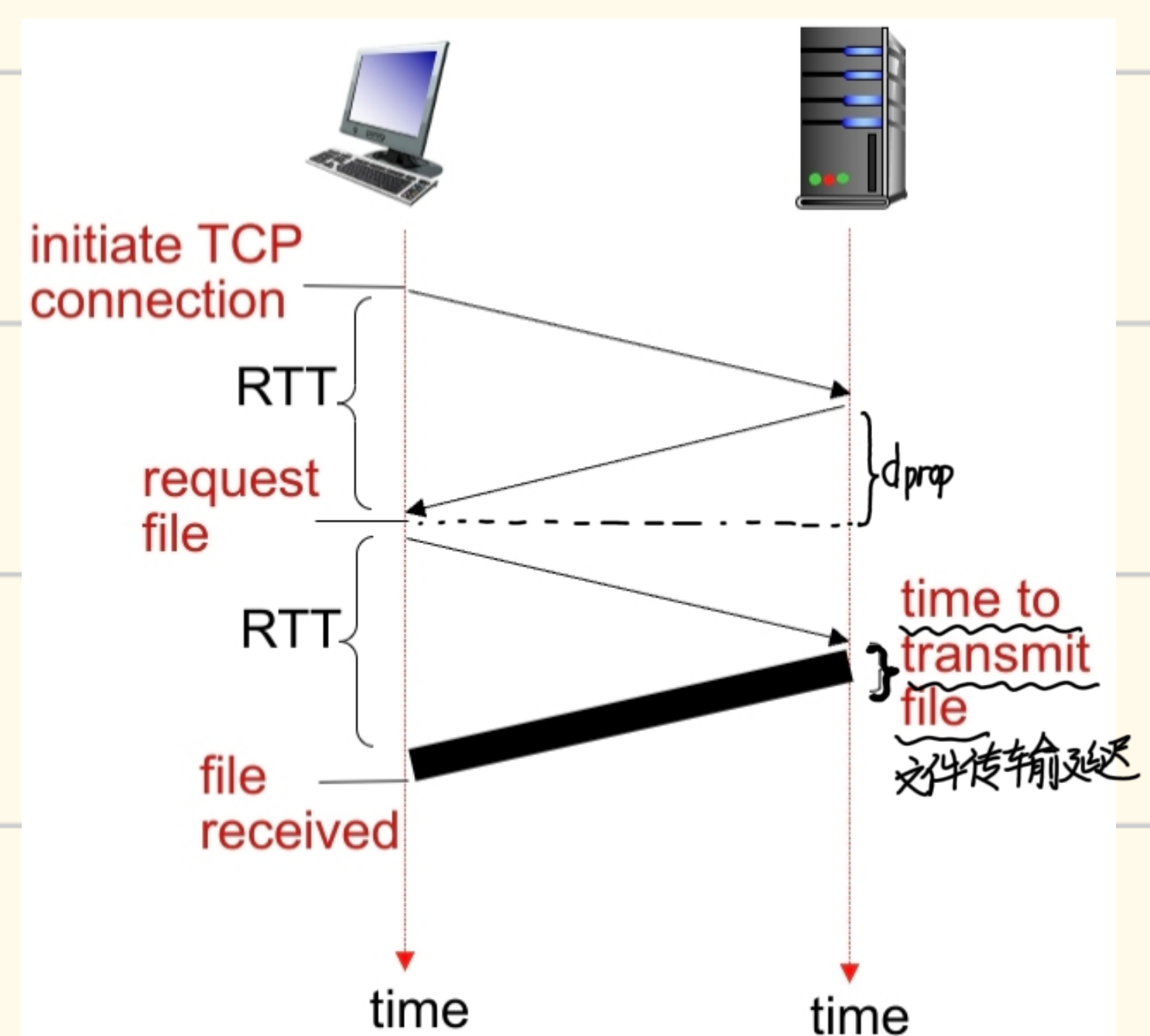
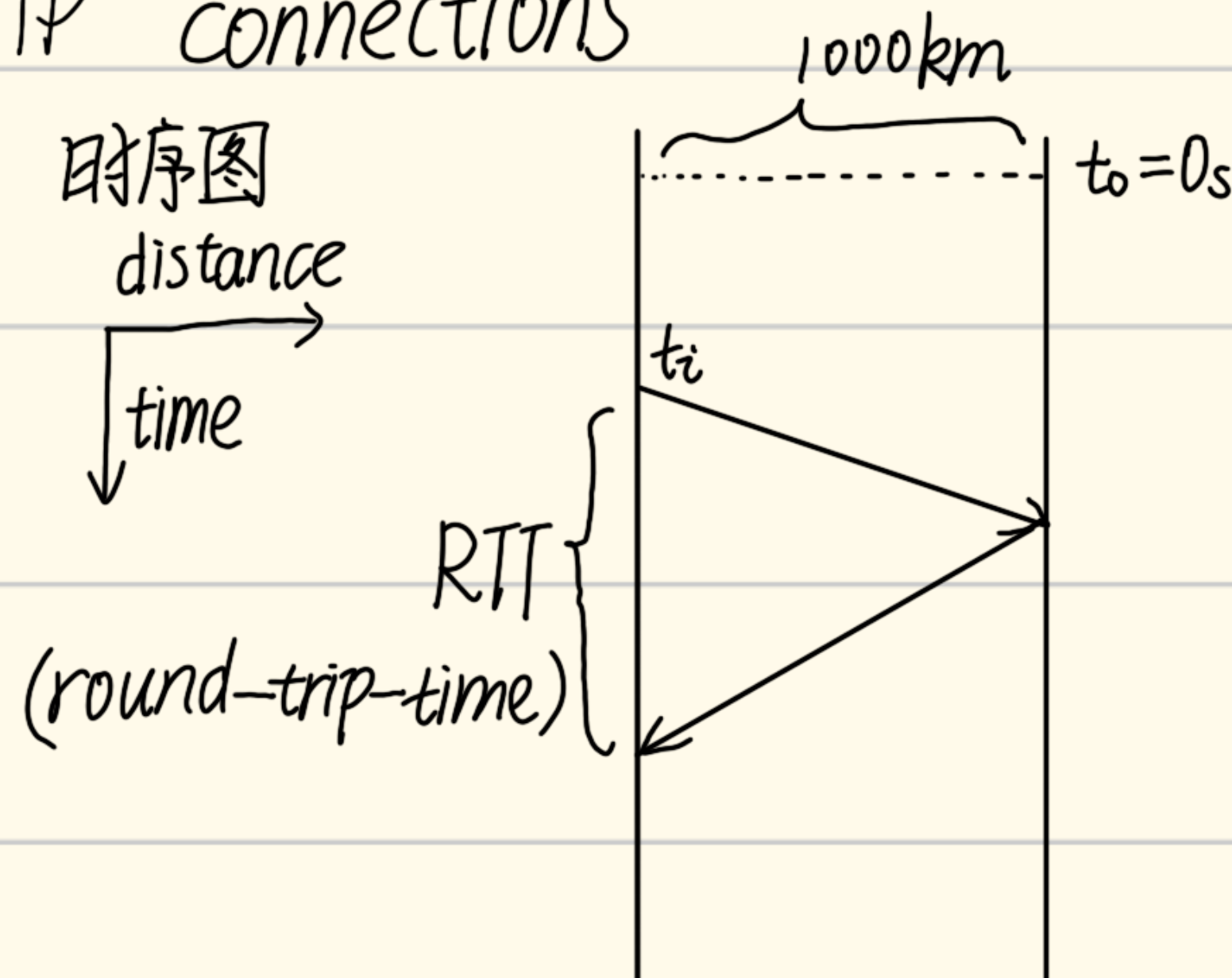


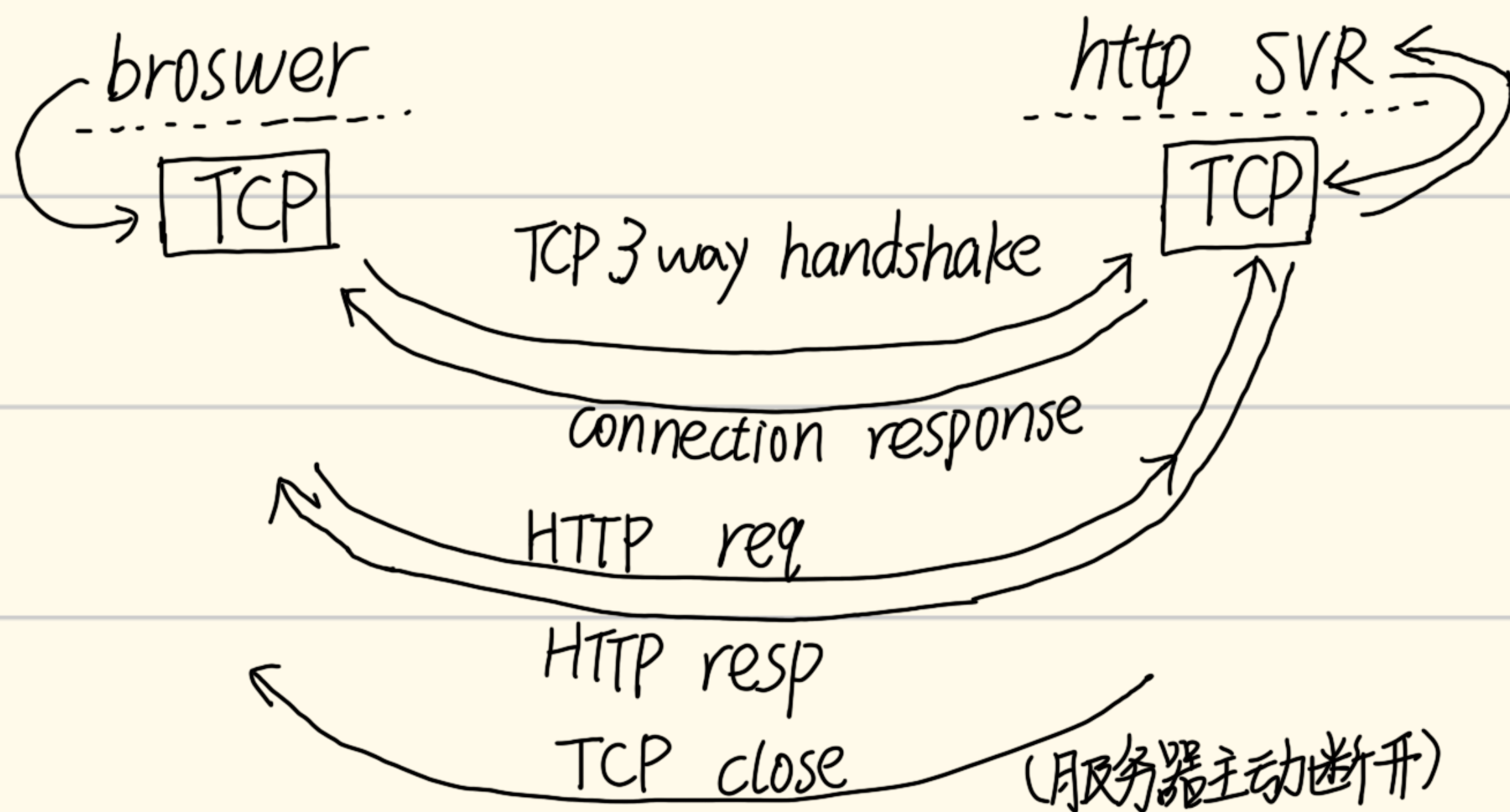
uses TCP

HTTP	message
------	---------

 segment by TCP

HTTP connections





non-persistent HTTP_{1.0}

persistent HTTP_{1.1} 优势 < 一次连接, 多次传输
pipelined 请求模式, 而非 stop & wait

传送 k 个 object 时, non-persistent 需 2k 个 RTT 时间
persistent 需 k+1 个 RTT 时间

pipelined:

优点: 更接近带宽时延积, 利用率更高

缺点: http 1.1 head of line blocking (必须有序返回)

Example:

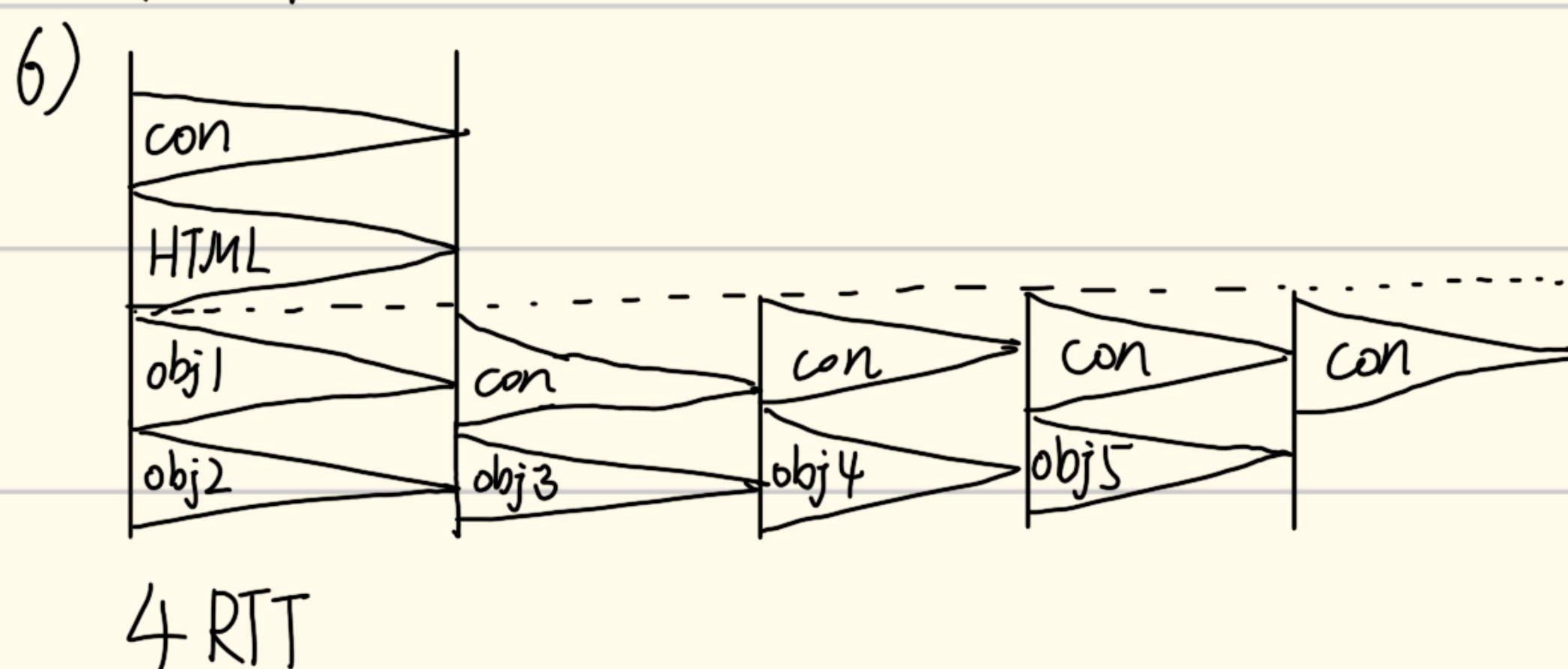
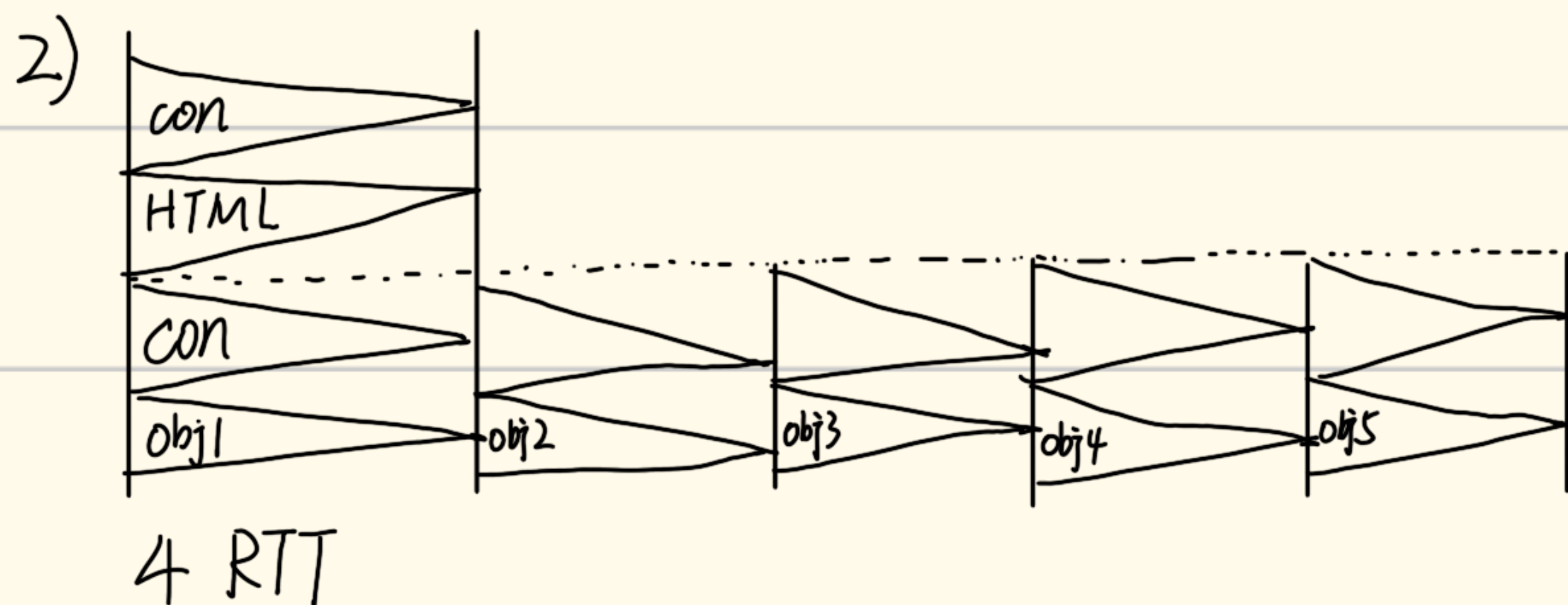
Suppose a browser sends an HTTP request to get a web page from the web server. The page consists of 1 HTML object and 5 reference objects. All these objects are stored on the server. Let RTT denote the round-trip time between the client and the server.

a. Calculate how many RTT it takes to get the above web page in the following scenarios. Ignore transmission delay.

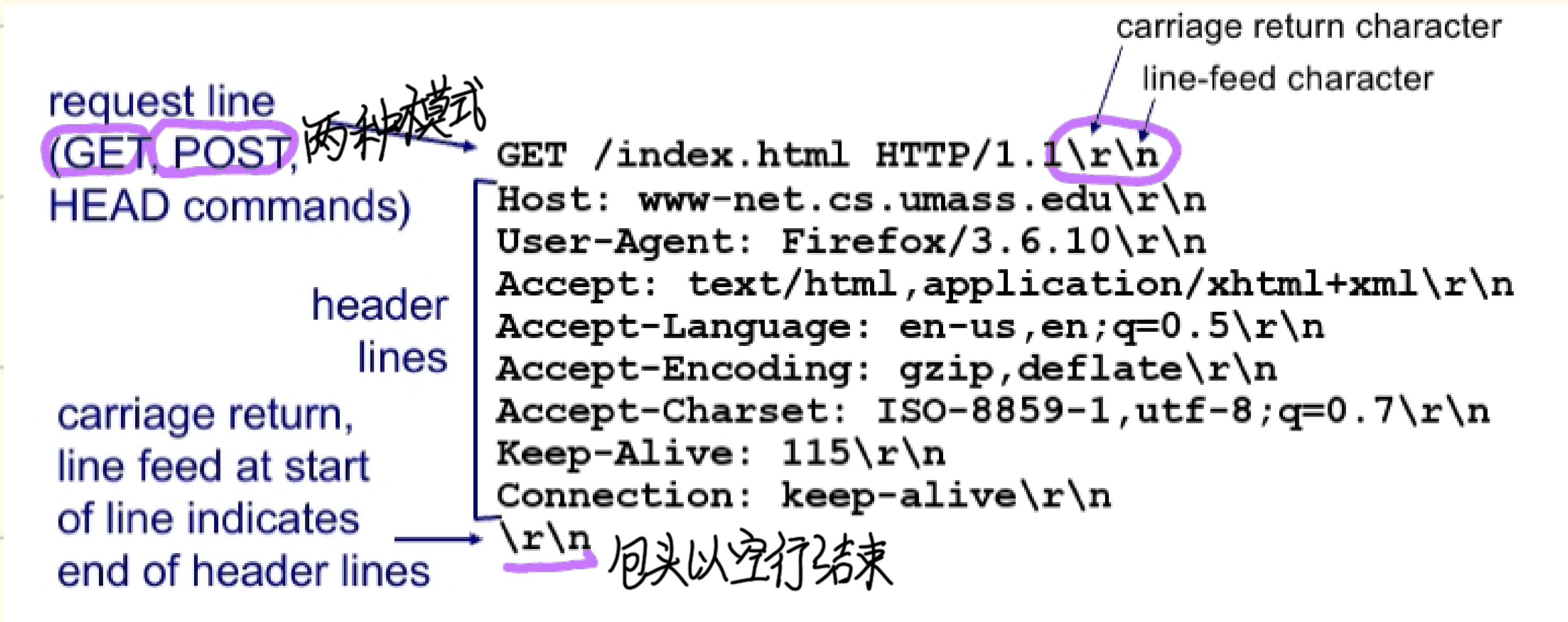
- 1) HTTP 1.0, Default connection model, Without parallel connections
- 2) HTTP 1.0, Default connection model, 5 parallel connections
- 3) HTTP 1.0, Default connection model, 3 parallel connections
- 4) HTTP 1.1, Default connection model, Without parallel connections, Pipelining
- 5) HTTP 1.1, Default connection model, Without parallel connections, Stop & Wait
- 6) HTTP 1.1, Default connection model, 5 parallel connections, Stop & wait
- 7) HTTP 1.1, Default connection model, 3 parallel connections, Stop & wait

b. Discuss why pipelined protocols outperform stop-and-wait protocols in throughput.

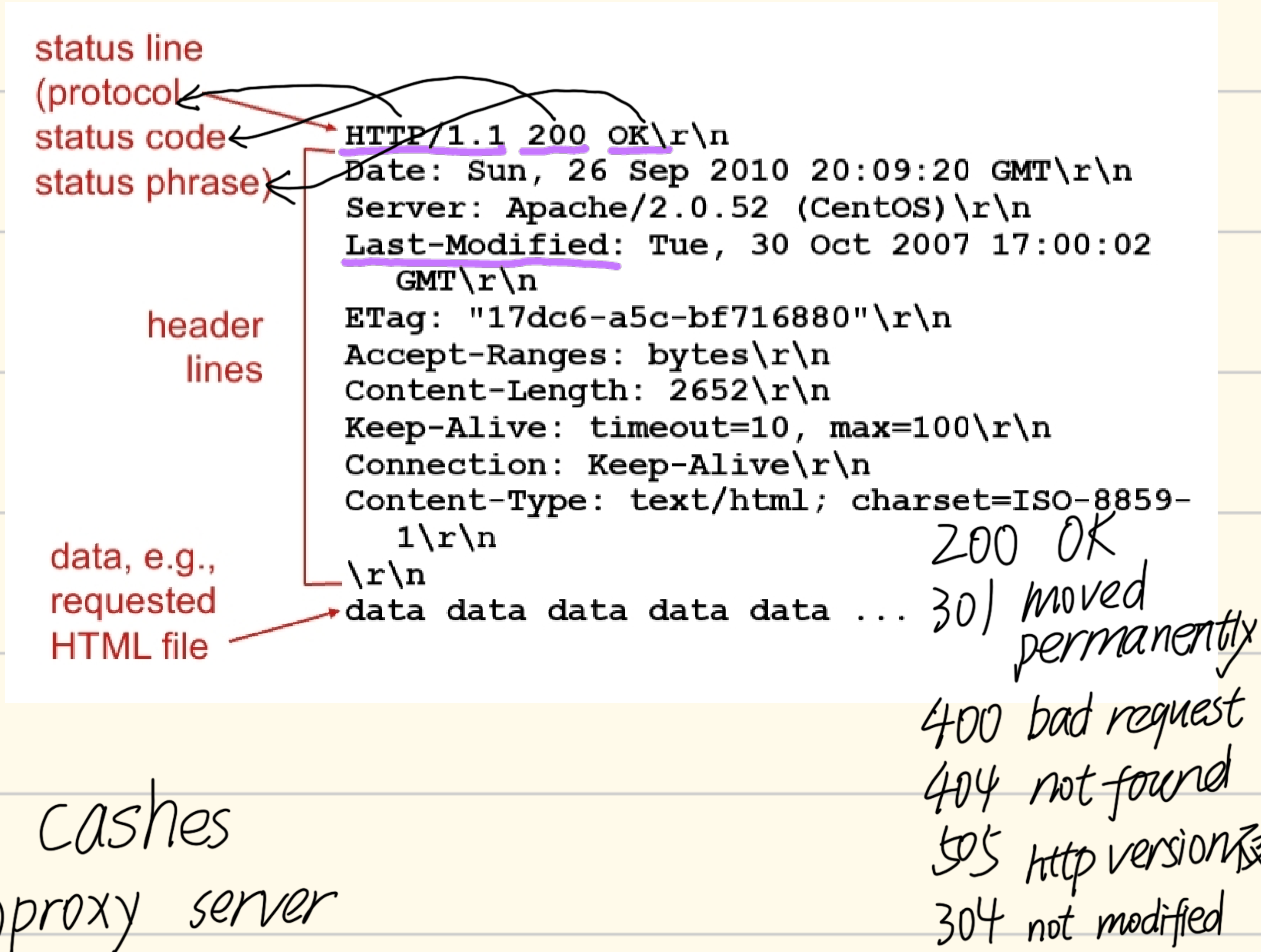
a. 1) 12 RTT



http request message

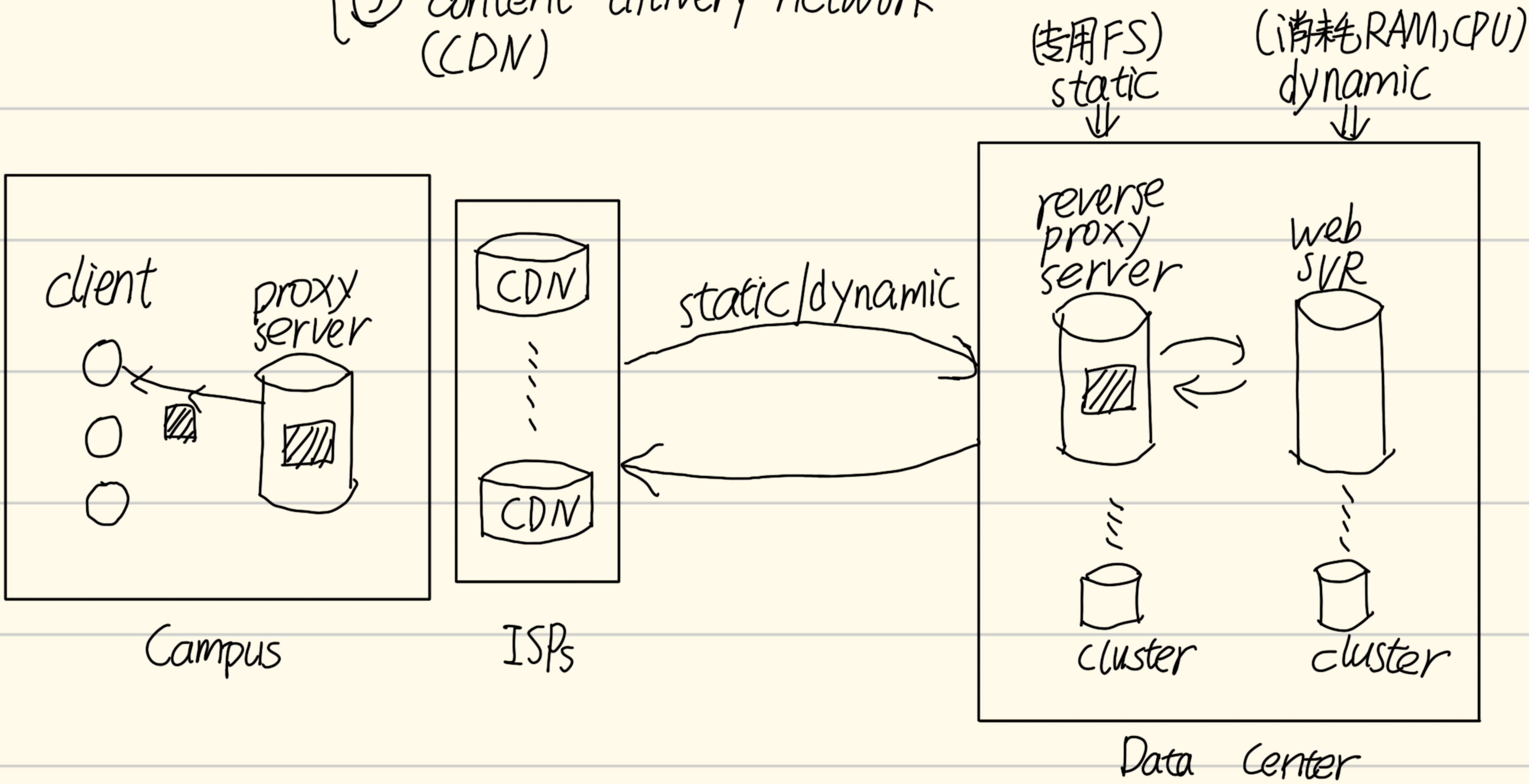


http response message

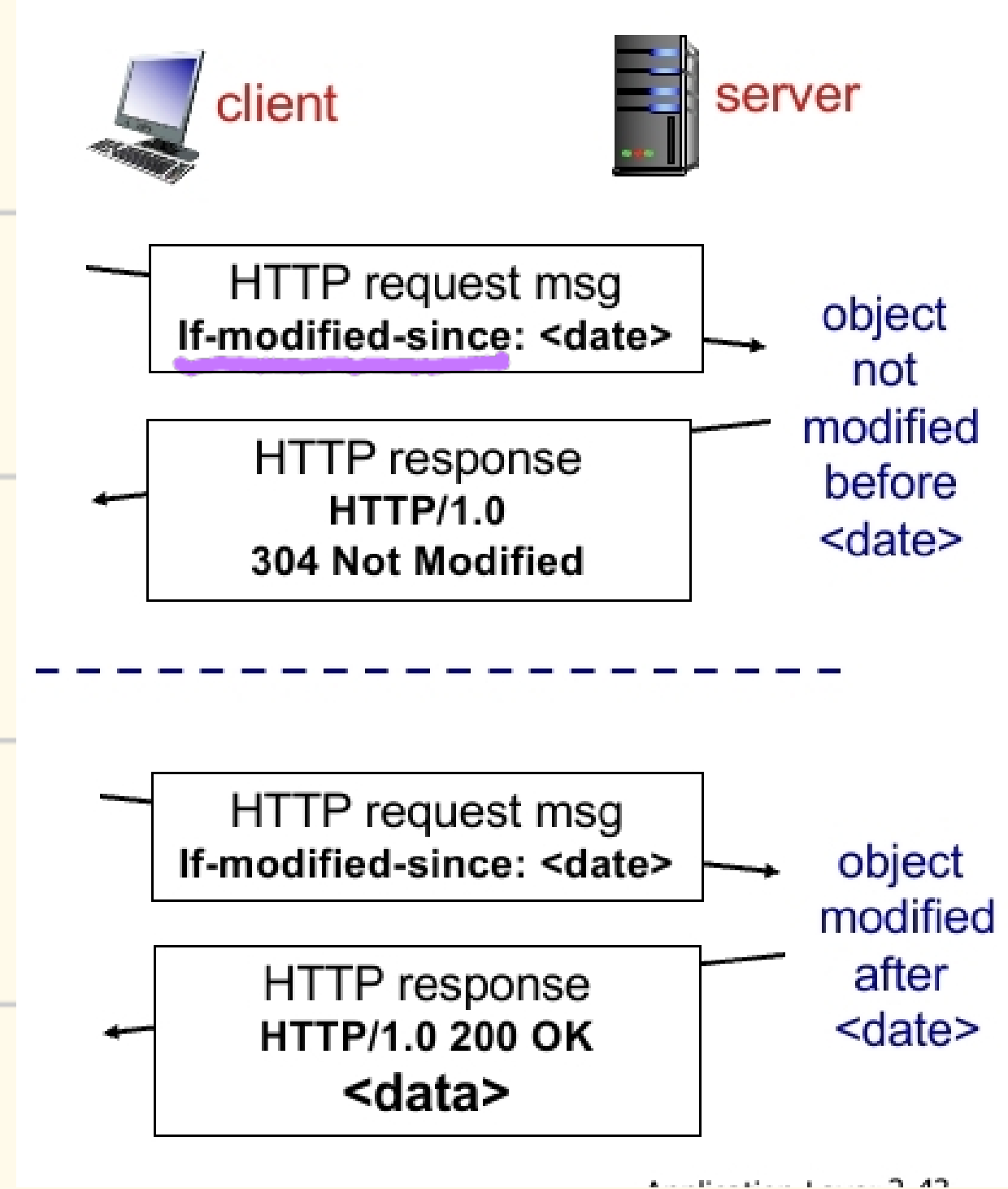


Web caches

- HW {
- ① proxy server
 - ② reverse proxy server
 - ③ content delivery network (CDN)



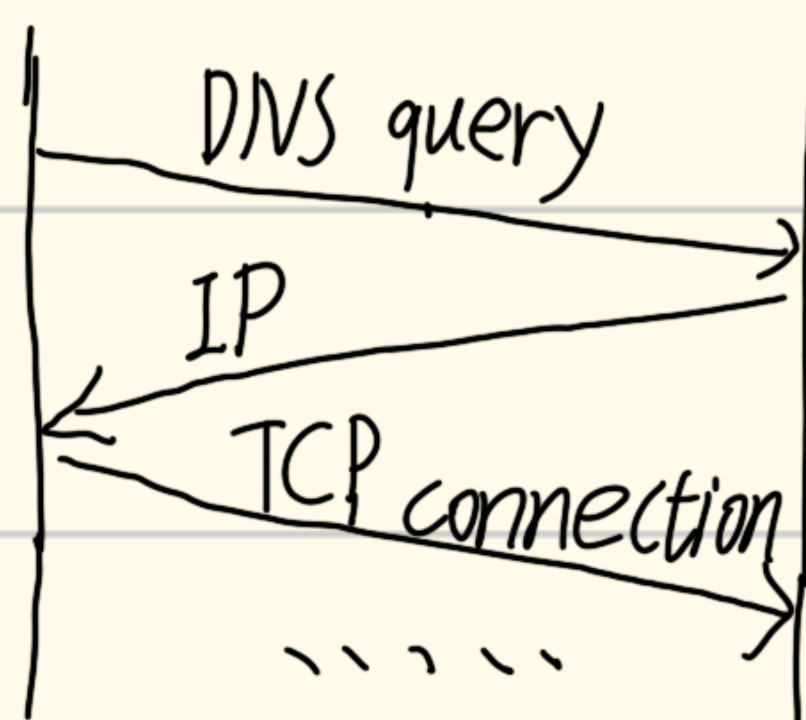
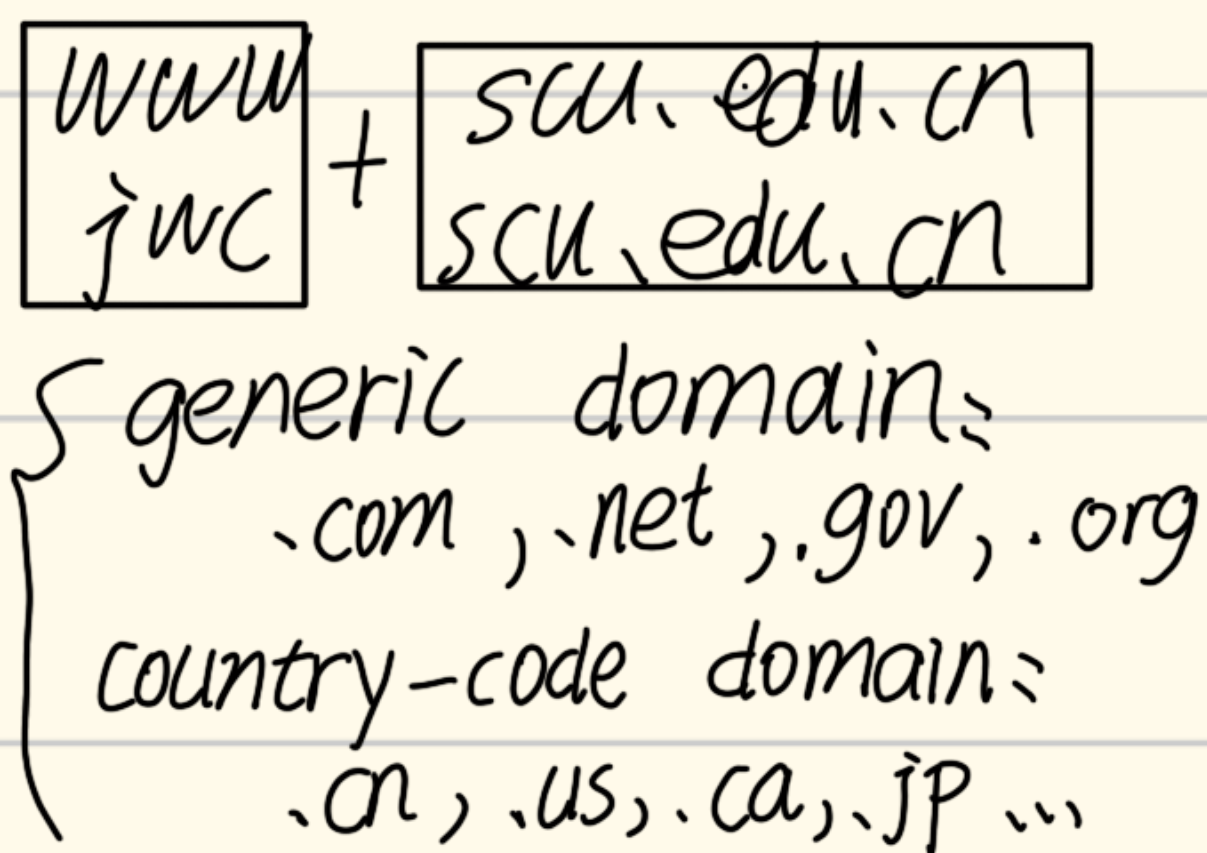
SWF ④ conditional GET



2.5 DNS ★

DNS = domain name system

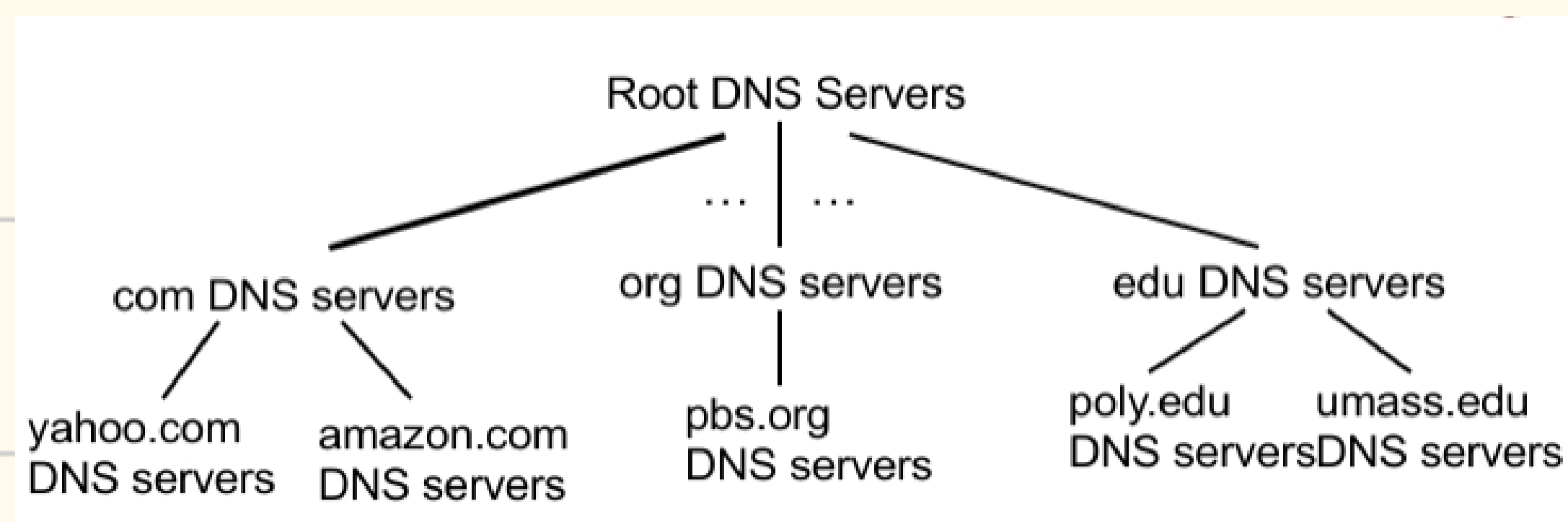
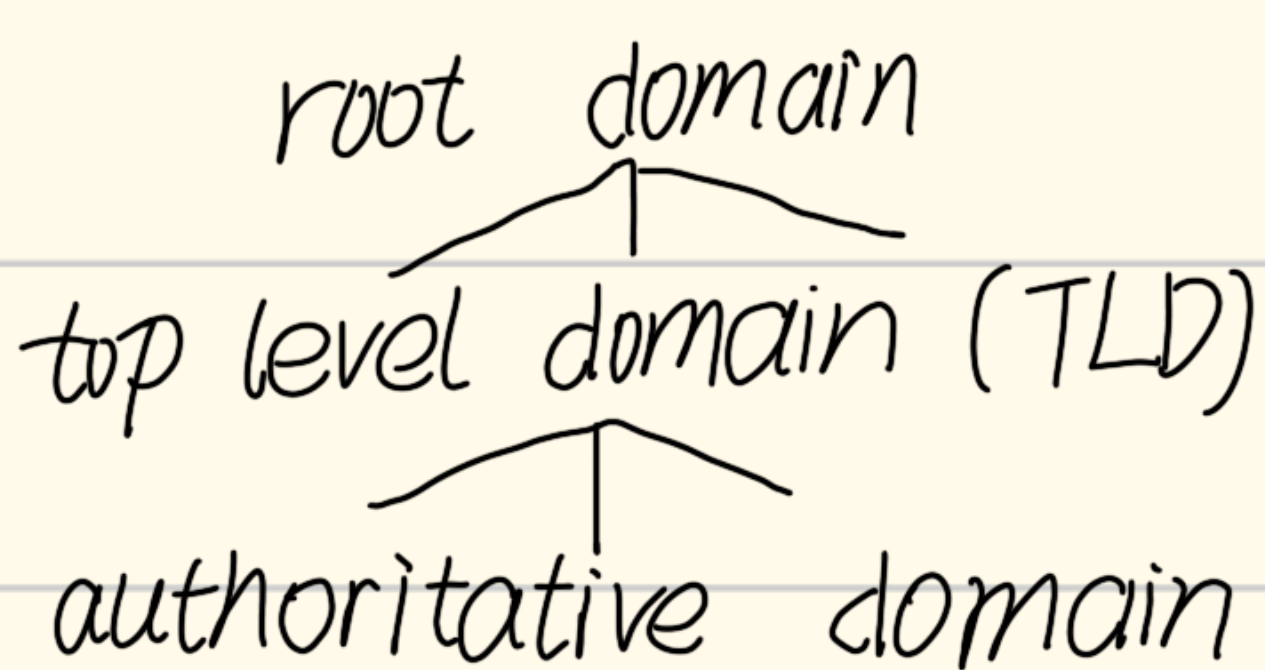
域名: hostname + domainname



why not centralize DNS?

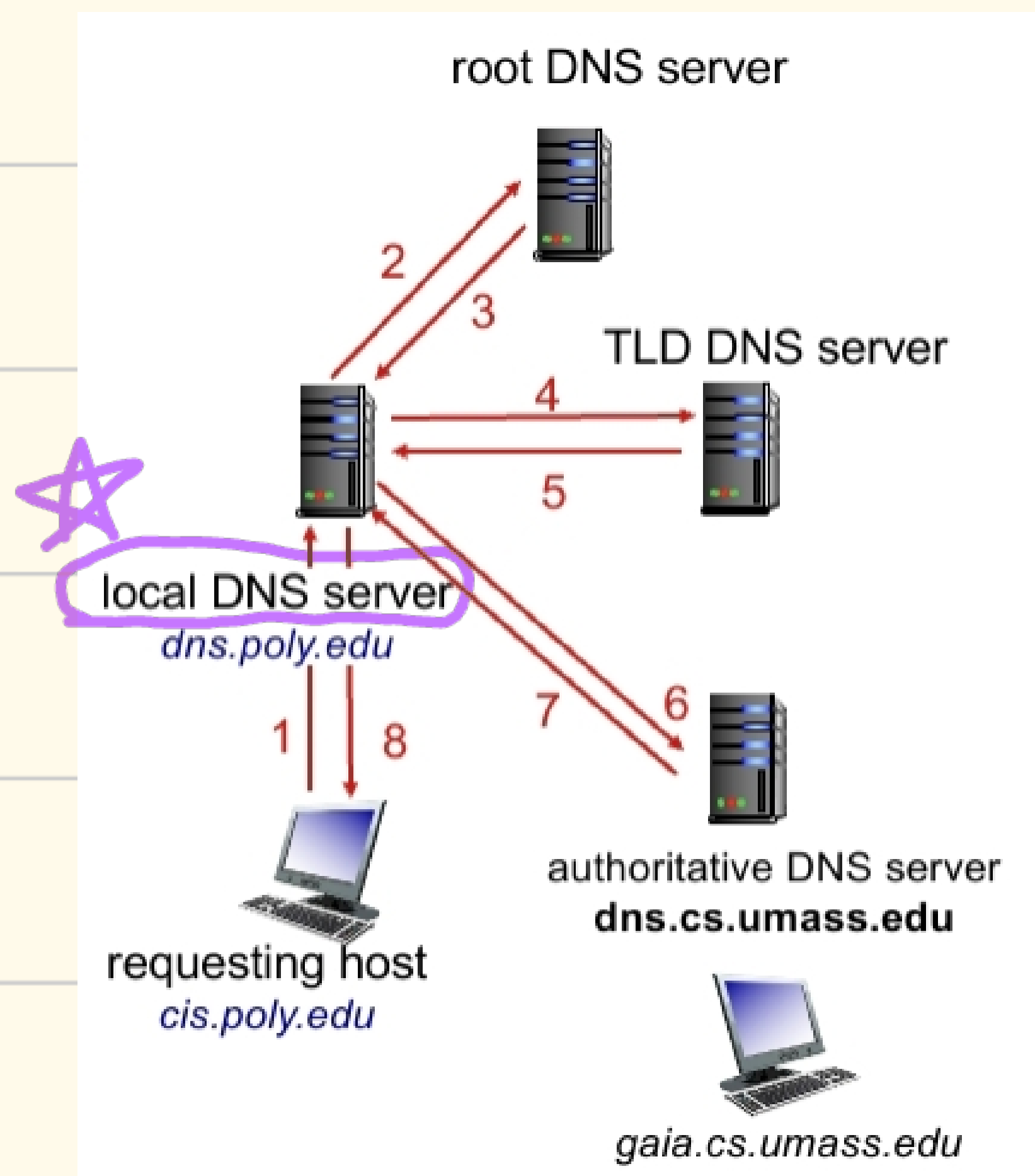
- single point of failure
- traffic volume
- distant centralized database
- maintenance

DNS = a distributed, hierarchical

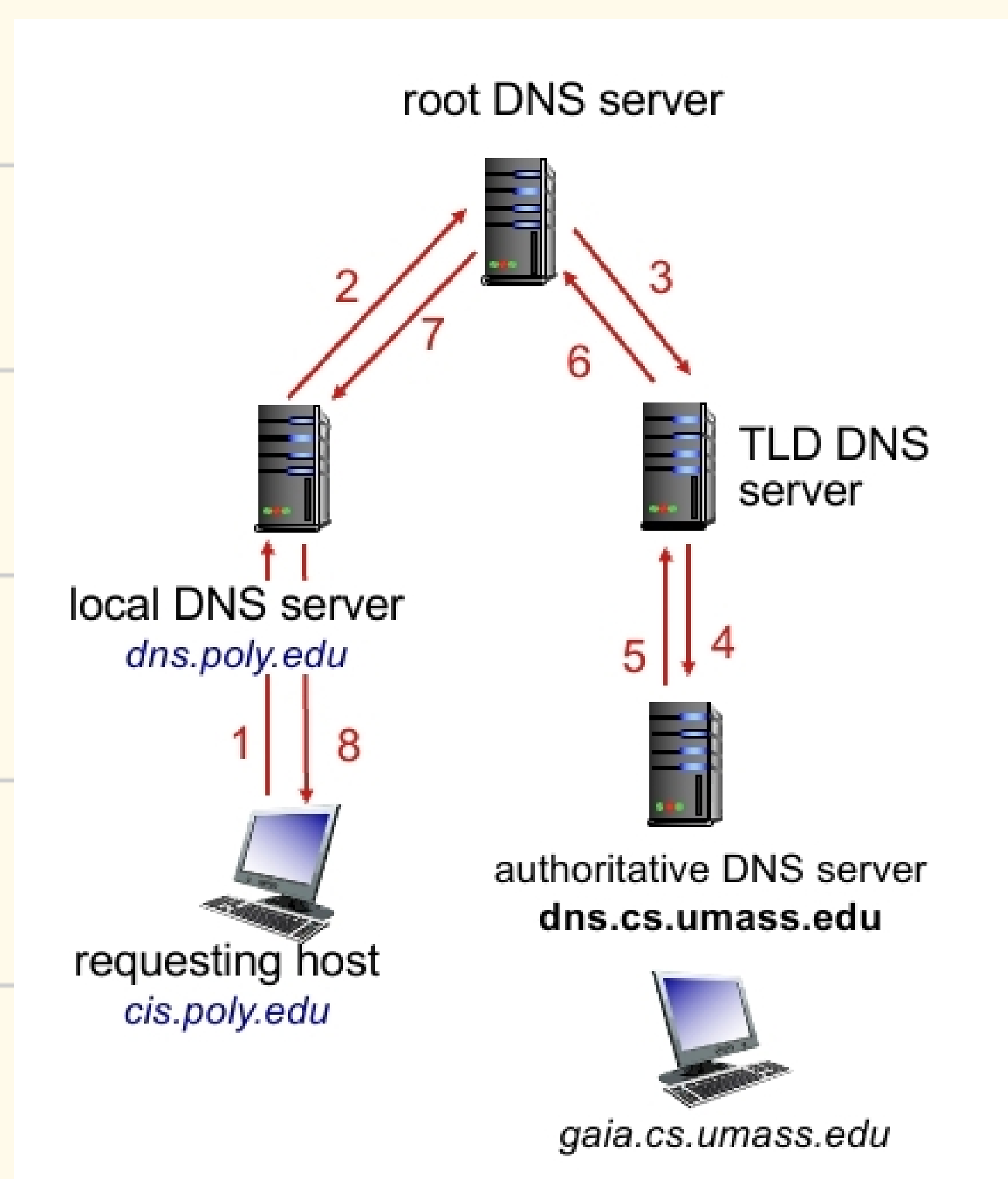


"dig + trace www."

example =



iterated query (v)



recursive query (x)

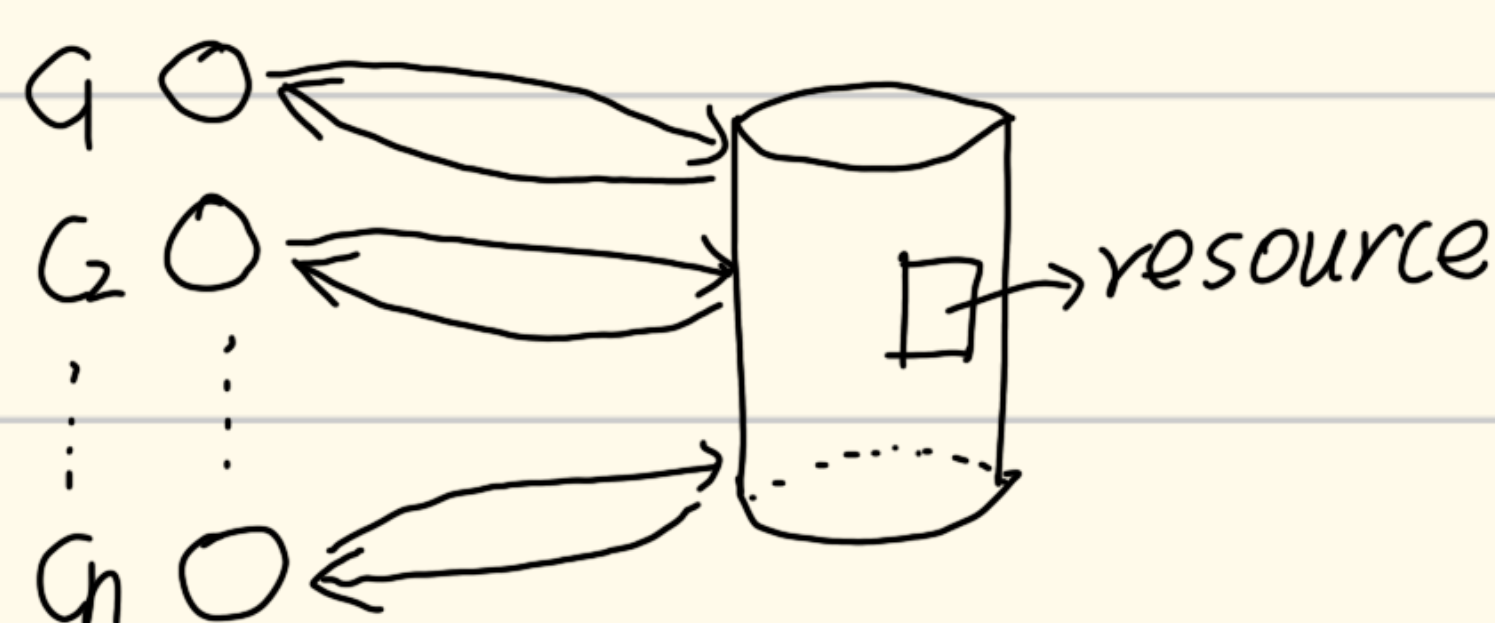
DNS protocol $\left\{ \begin{array}{l} \text{query messages} \\ \text{reply messages} \end{array} \right.$ (same format)
 (UDP)
 ↓
 distributed database storing resource records (RR)

RR format: (name, value, type, ttl)

type {
A: name $\rightarrow$ hostname
 value $\rightarrow$ IP address
 eg. DNS Load Balance
 同一个域名, 多个A记录映射至不同IP
CNAME: name $\rightarrow$ alias name
 value $\rightarrow$ canonical name
 NS (domain, hostname)
 MX (mailserver)

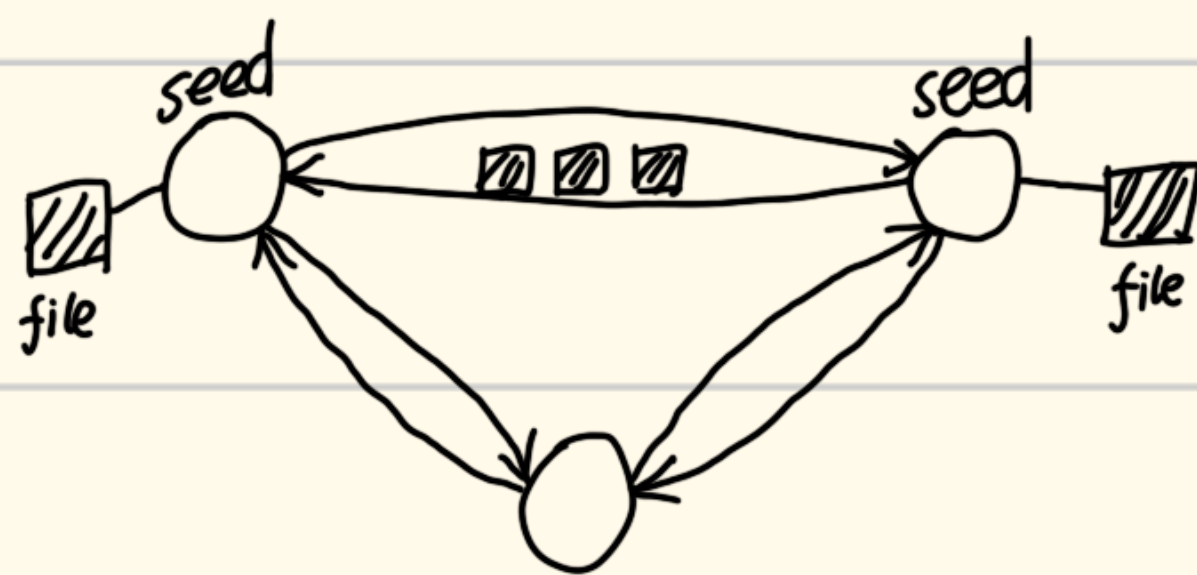
2.6 P2P Application (x)

Client-Sever Architecture (cls)



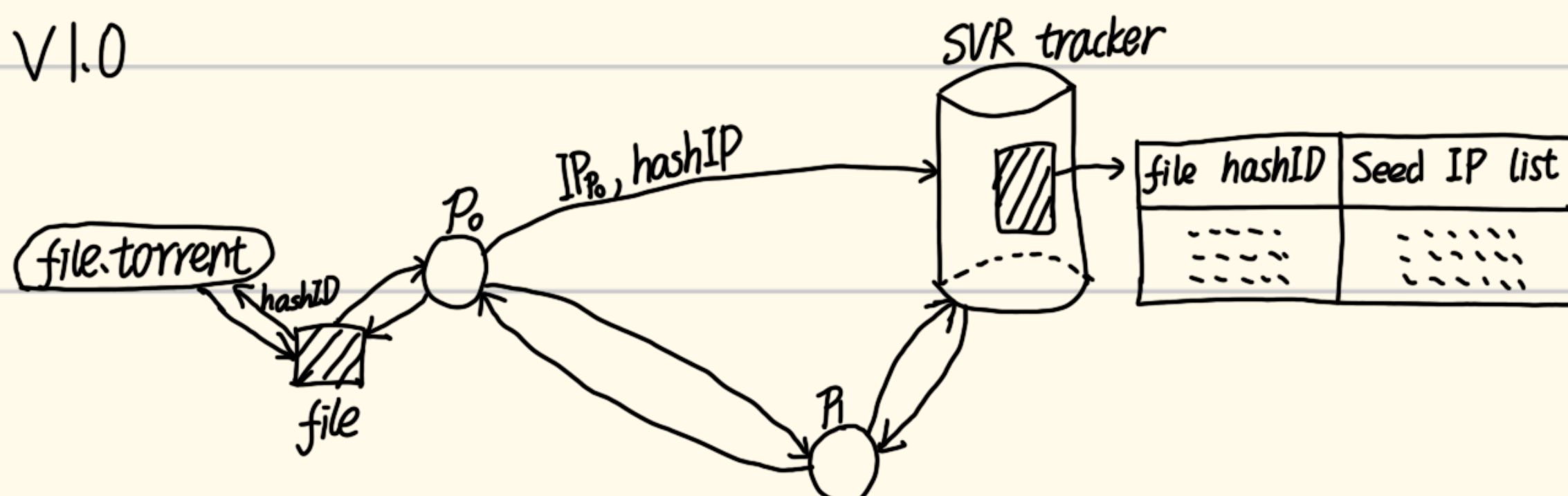
bottleneck:
 Server CPU/RAM
 Bandwidth
 single point of failure

Peer-to-Peer Architecture (P2P)

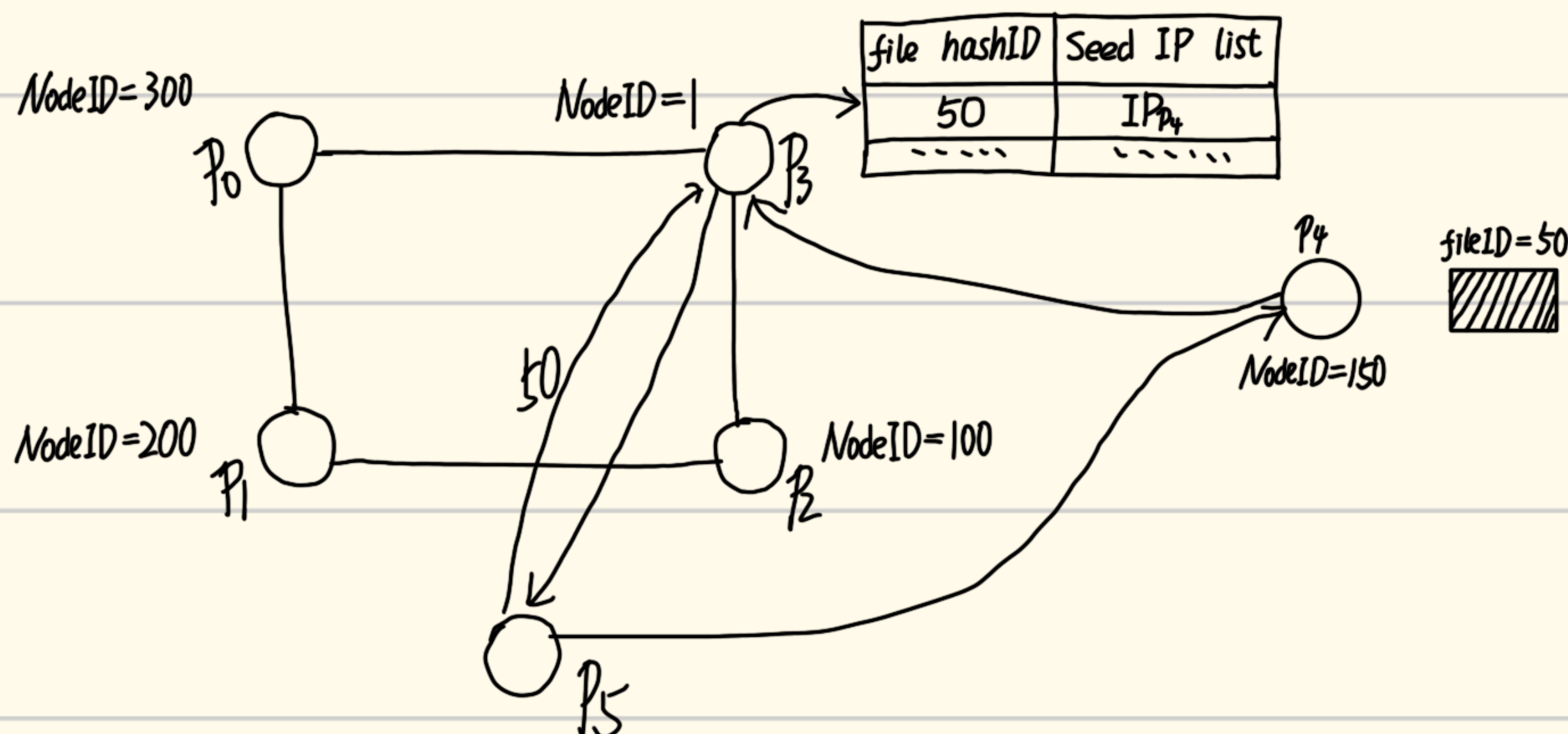


缺点：
架构复杂

V1.0



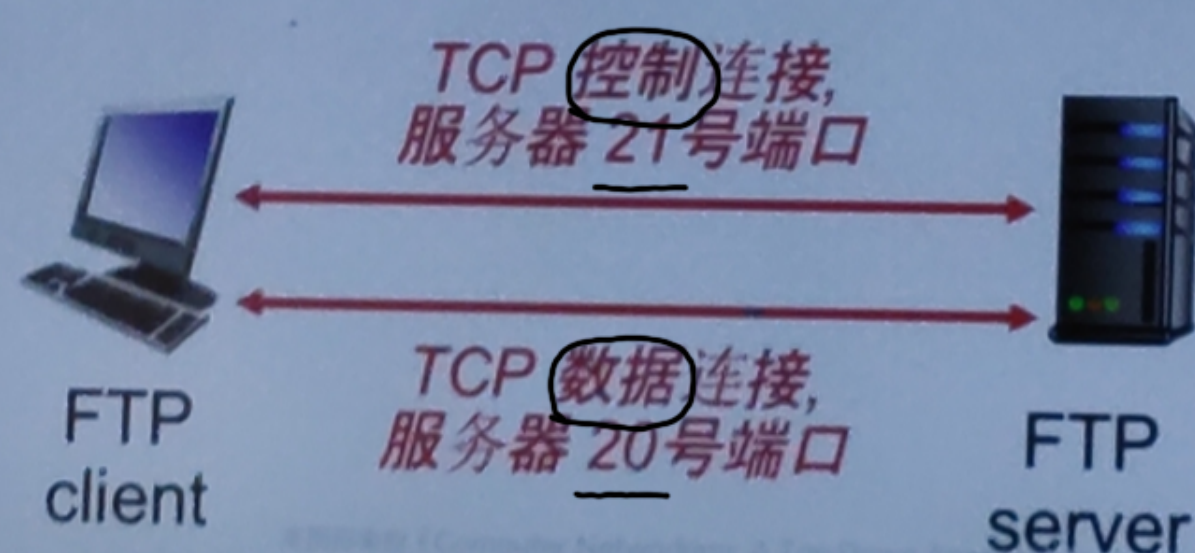
V2.0



附: electronic mail
FTP

FTP文件传输协议的两个连接

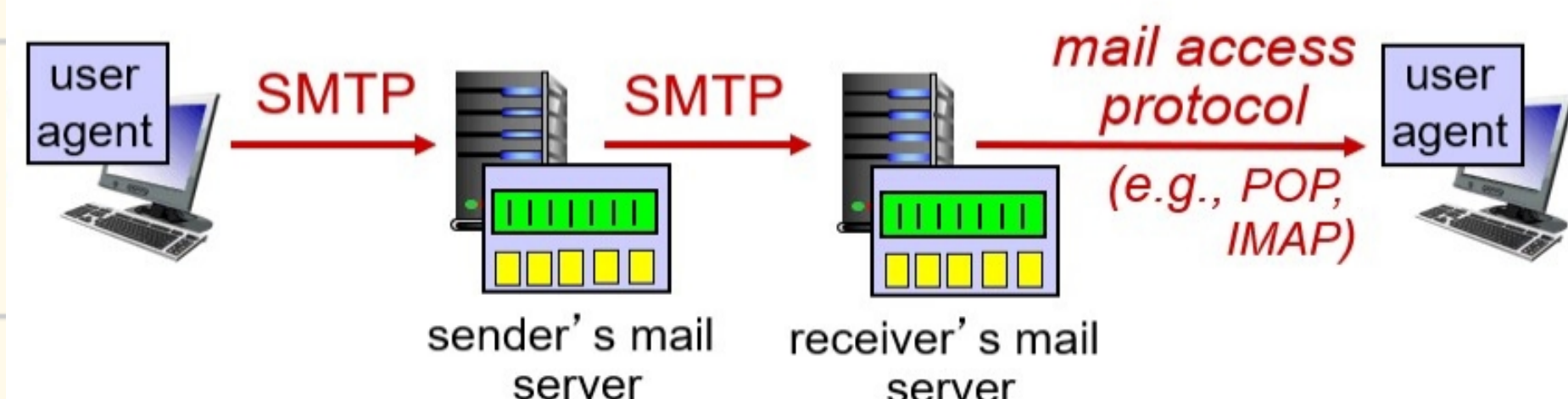
- 各种控制指令（包括下载、上传、中止传输、删除文件、获取文件夹中的文件列表等等指令）在控制连接的通道中传输。
- 文件内容在数据连接的通道中传输。在数据连接之外的另一个连接通道中传输控制命令，又称为带外传输。



FTP = file transfer protocol

邮件发送SMTP和接收POP3

- 发邮件用SMTP协议, push to 25号端口
- 收邮件用POP3协议, pull from 110号端口



发 { SMTP = simple mail transfer protocol
收 { POP3 = post office protocol
IMAP = internet mail access protocol

Chapter 3

Transport Layer

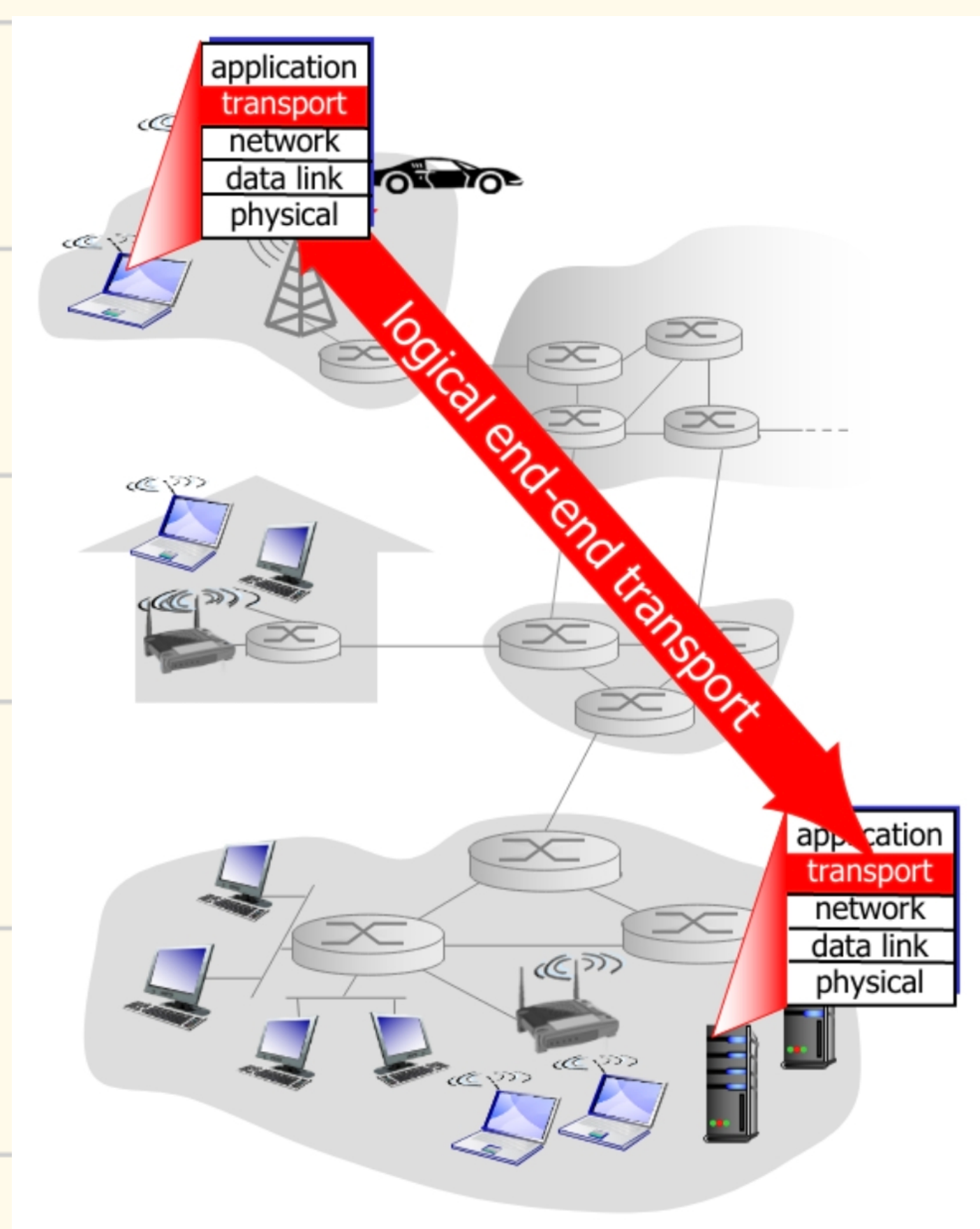
3.1 transport-layer services

- provide logical communication
- send side: messages $\rightarrow$ segments
rcv side: segments $\rightarrow$ messages
- network layer: logic communications between hosts
transport layer: logic communications between process

- UDP, TCP

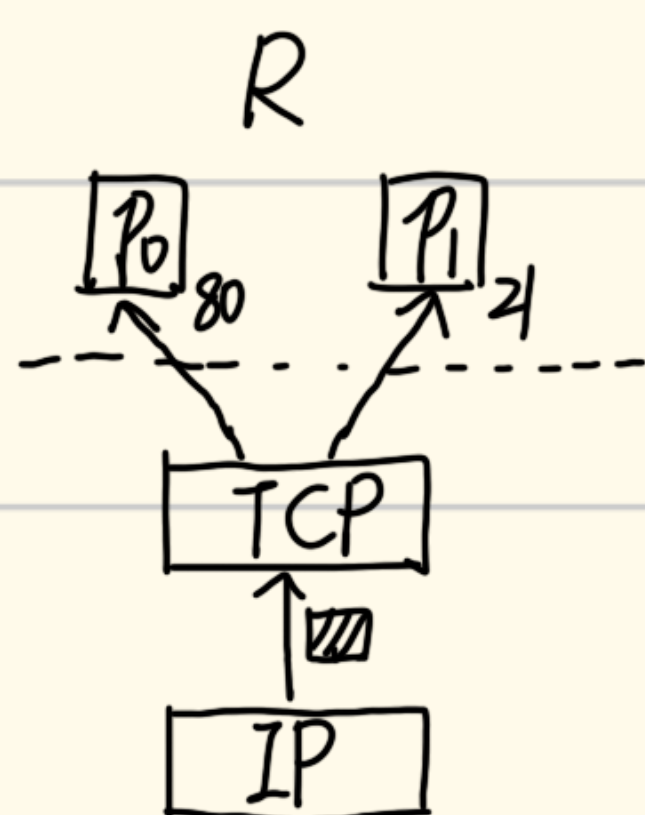
- reliable, in-order delivery (TCP)
 - congestion control
 - flow control
 - connection setup
- unreliable, unordered delivery: UDP
 - no-frills extension of "best-effort" IP
- services not available:
 - delay guarantees
 - bandwidth guarantees

(UDP会check-sum,
能够解决bit error)

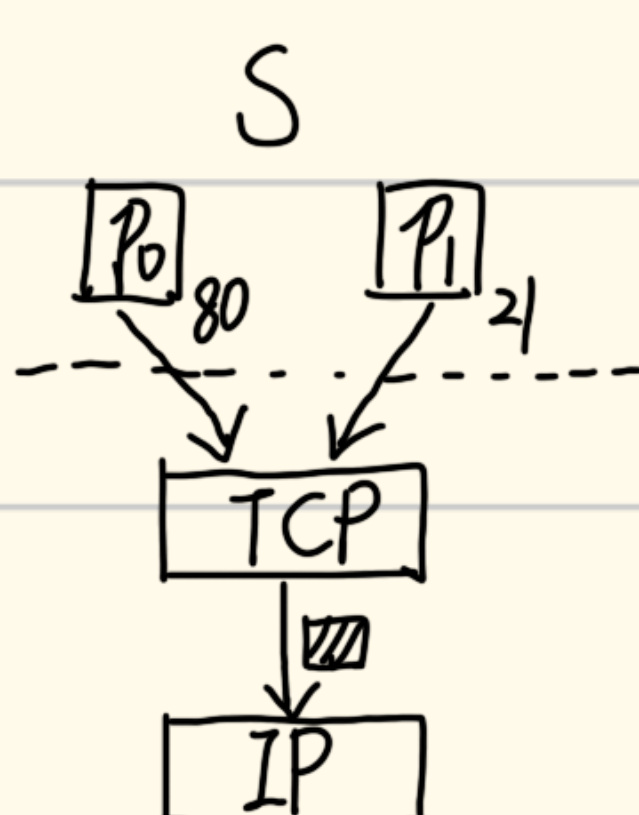


3.2 Multiplexing & Demultiplexing

- Demultiplexing



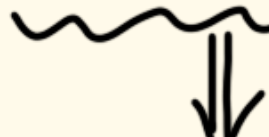
- Multiplexing



- Process

pid (必须有)

port (联网才有): (0, 65536] integer ID



(0, 1023] well-known port (只分配给服务器进程)

- Demux

for UDP,

只看 dest port

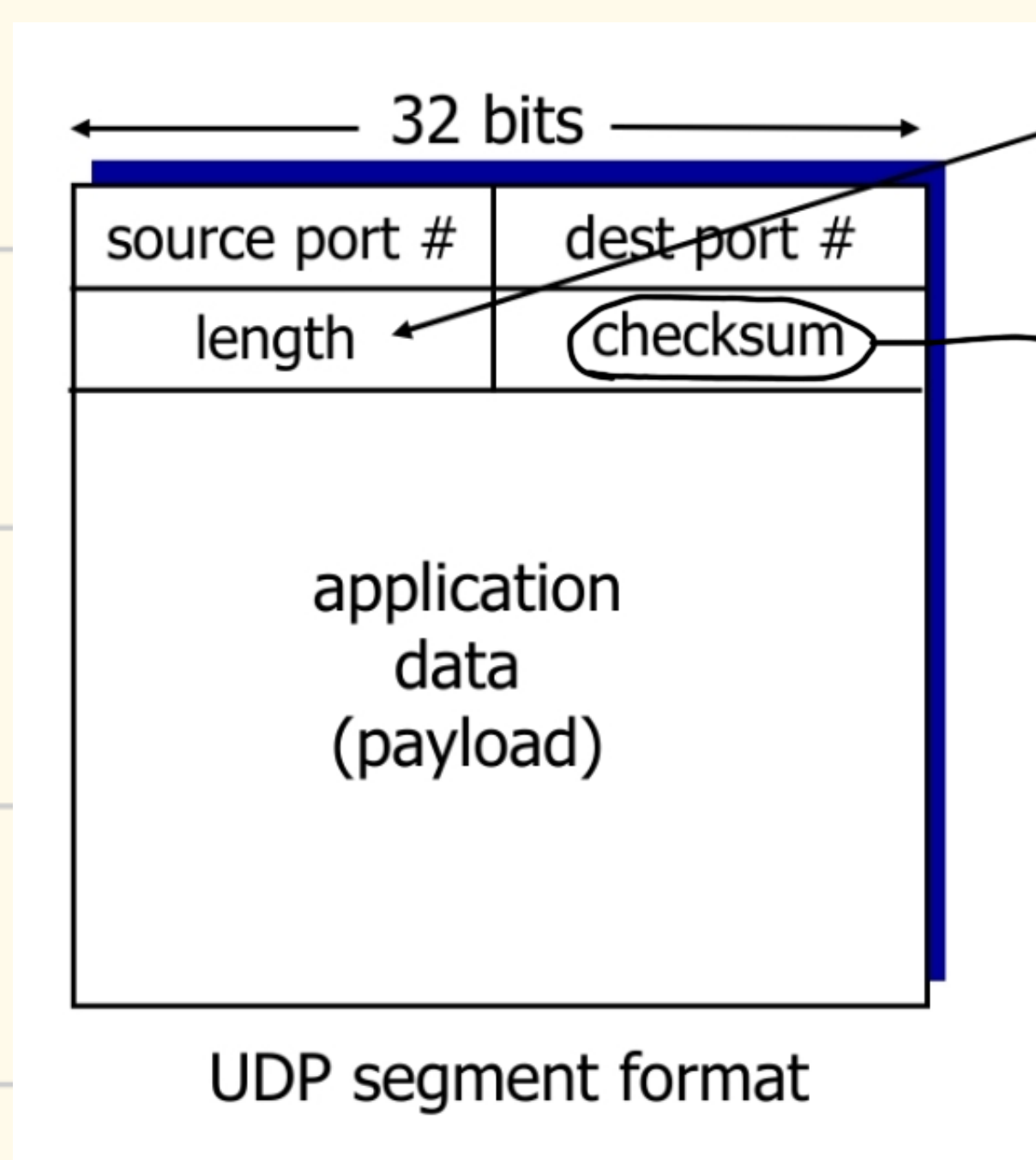
for TCP,
需匹配四个信息

TCP socket identified
by 4-tuple:

- source IP address
- source port number
- dest IP address
- dest port number

3.3 connectionless transport: UDP

- UDP: User Datagram Protocol
"no frills" "bare bones" "best effort service"
- UDP segments maybe:
lost
delivered out-of-order to apps
- Connectionless:
no handshaking
each UDP segment handled independently
(UDP: message-segment -- 对应 "xyz" "abc" → "xyz" "abc"
TCP: message-segment 多对多 (m-n) 对应 "abc" "xyz" ↔ "ab" "xyz" [由于 flow control 等])
- UDP use
use in DNS, SNMP
reliable transfer over UDP: 谷歌 QUIC 库
- UDP segment header



UDP segment 长度, including header

每16 bits一组

example: add two 16-bit integers

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
wraparound	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
sum	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

注: UDP 不存在 MSS, 只有 MTU=1500

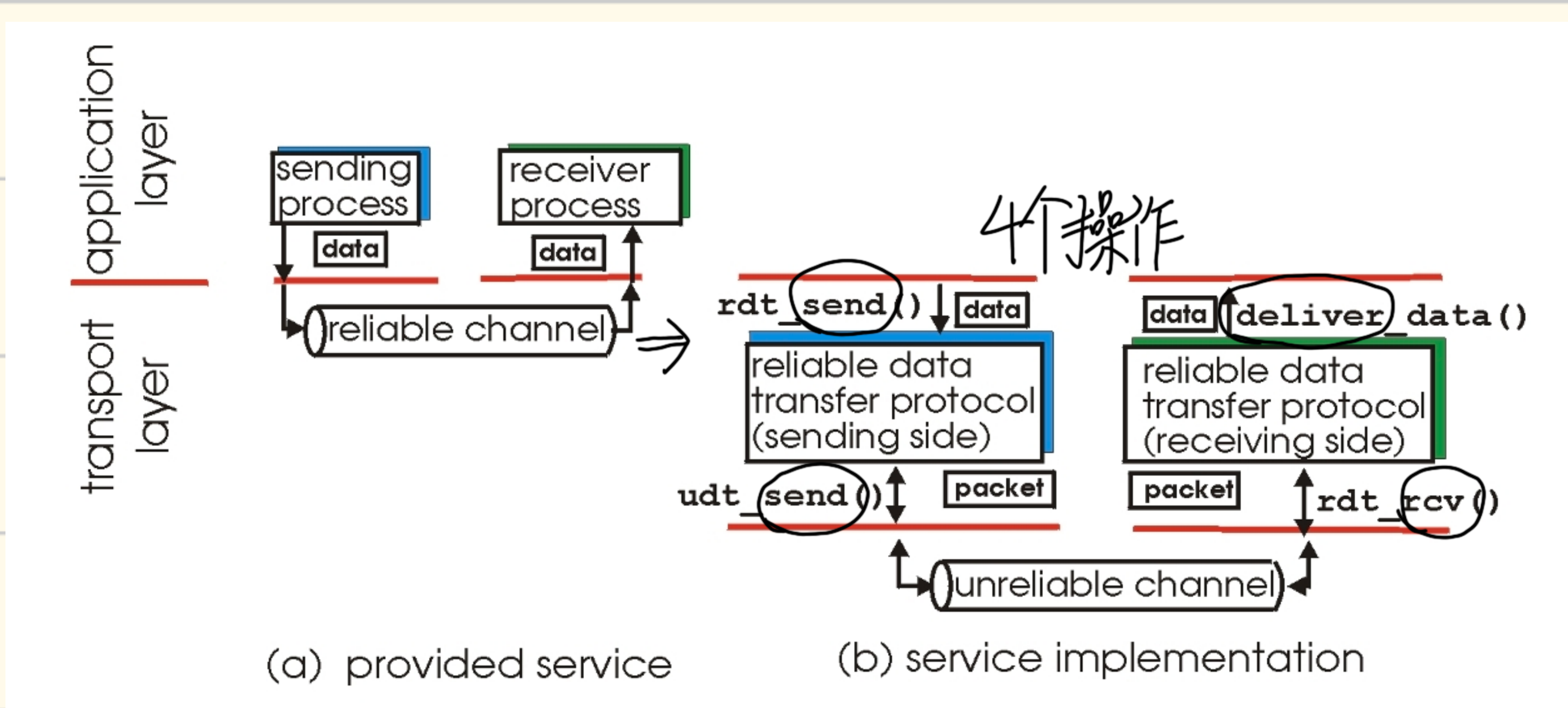
最大 UDP segment data 段长度 = 1500 - 20 - 8 = 1472



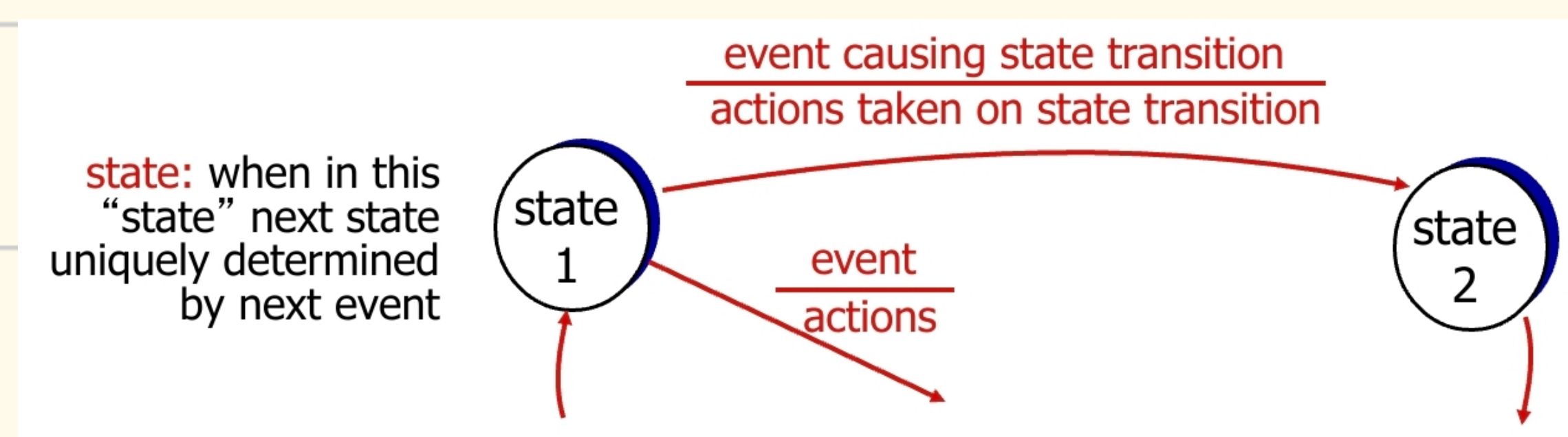
3.4 Principle of reliable data transfer

- Principles

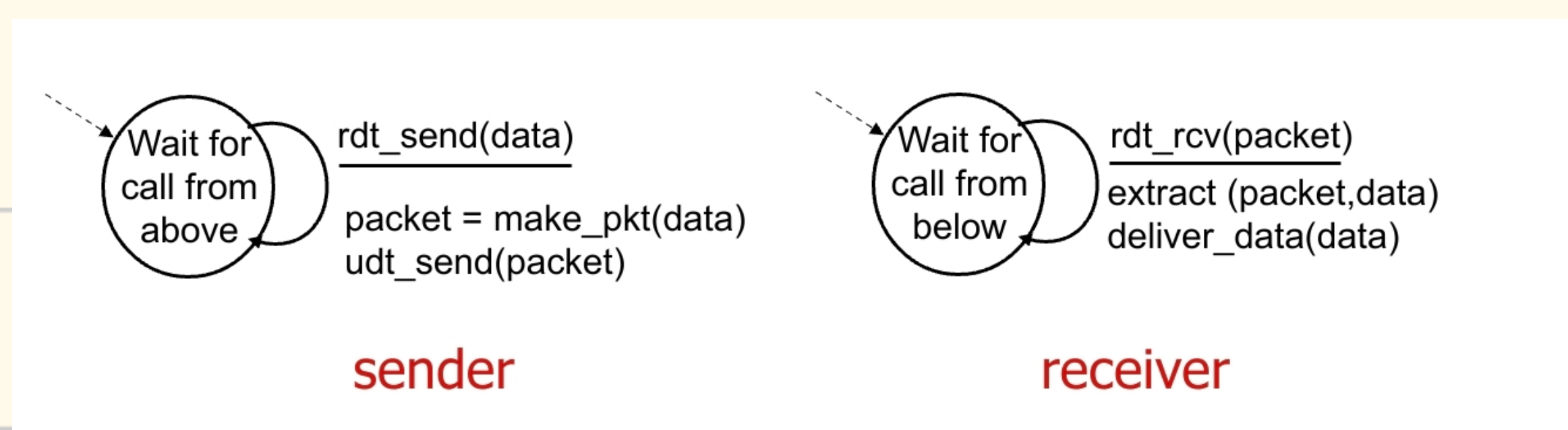
提供可靠传输的协议层: app trans link



finite state machines (FSM)



- rdt 1.0 = a reliable channel

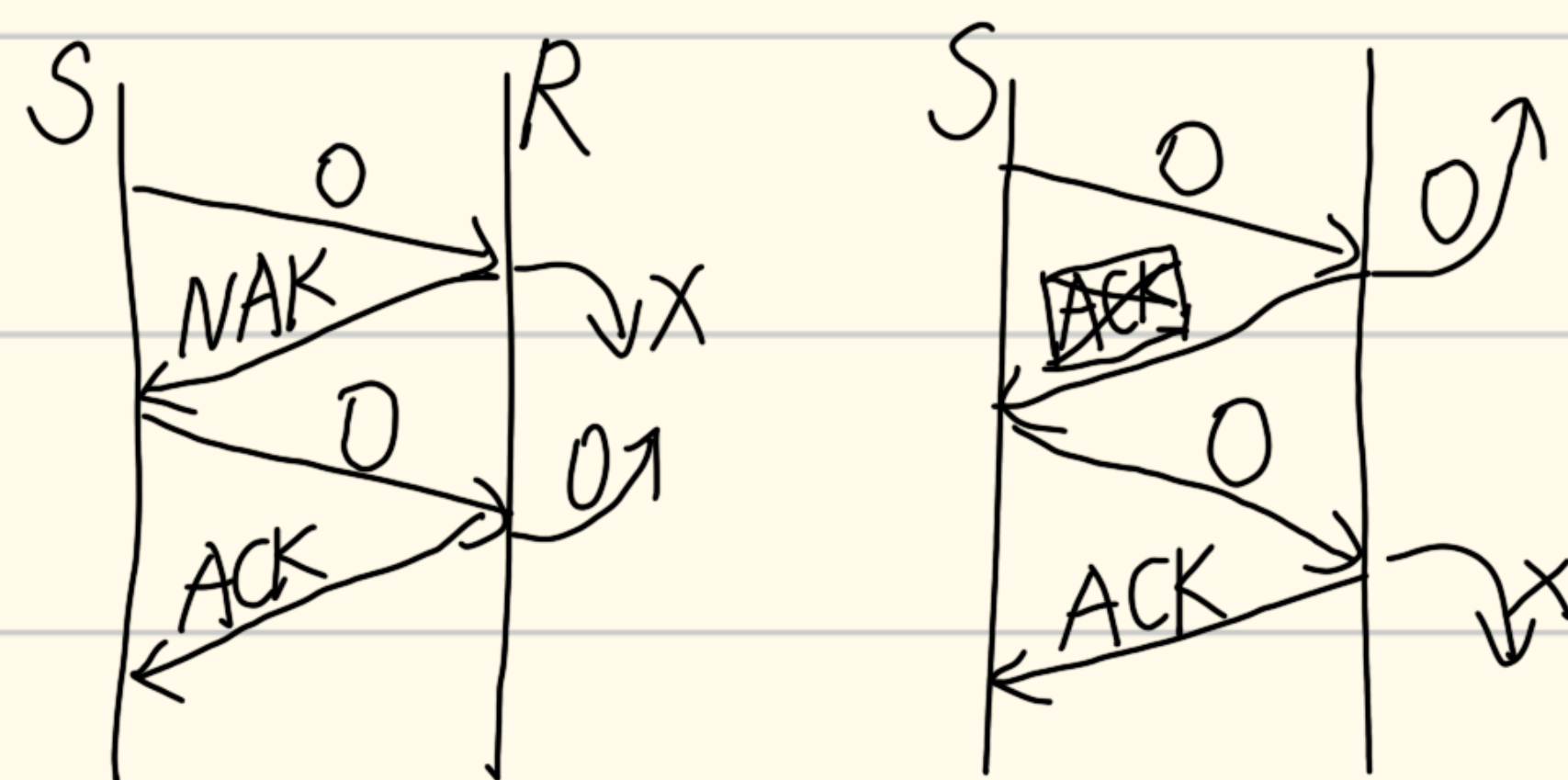


- rdt 2.0 = channel with bit errors
 - error detection = checksum
 - feedback = acknowledgements (ACK)
 - negative acknowledgements (NAK)
 - a fatal flaw = ACK/NAK 出错 → possible duplicate

- rdt 2.1 = handles garbled ACK/NAK

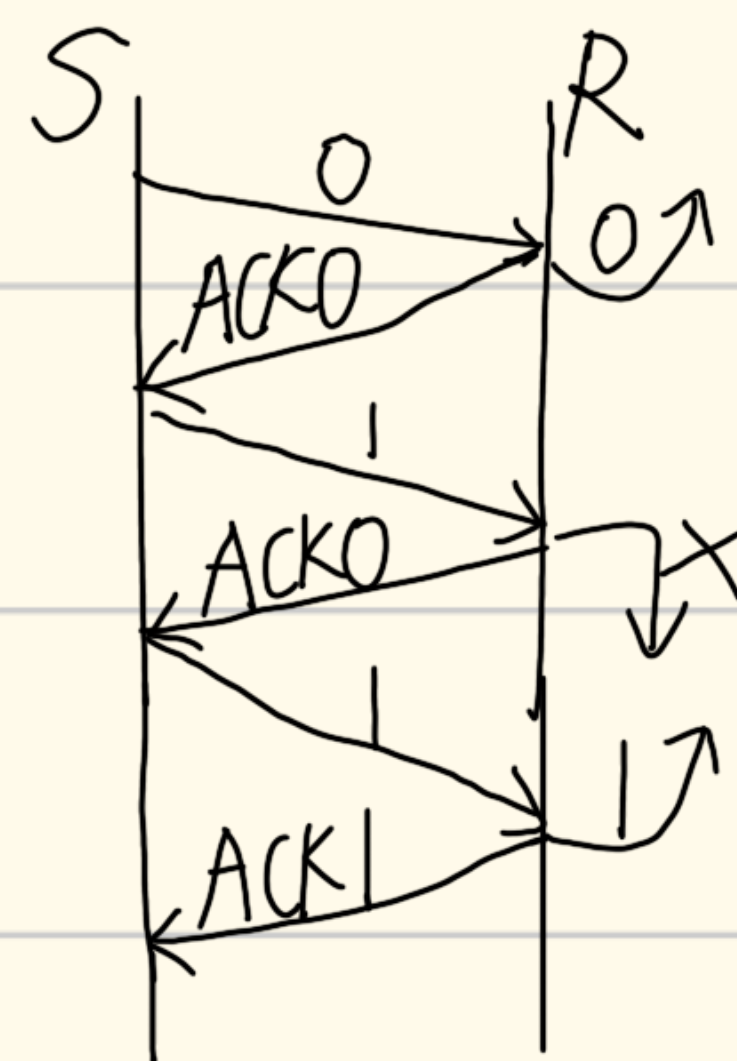
- retransmits
- add sequence number to pkt
- discards duplicate pkt

stop and wait mode



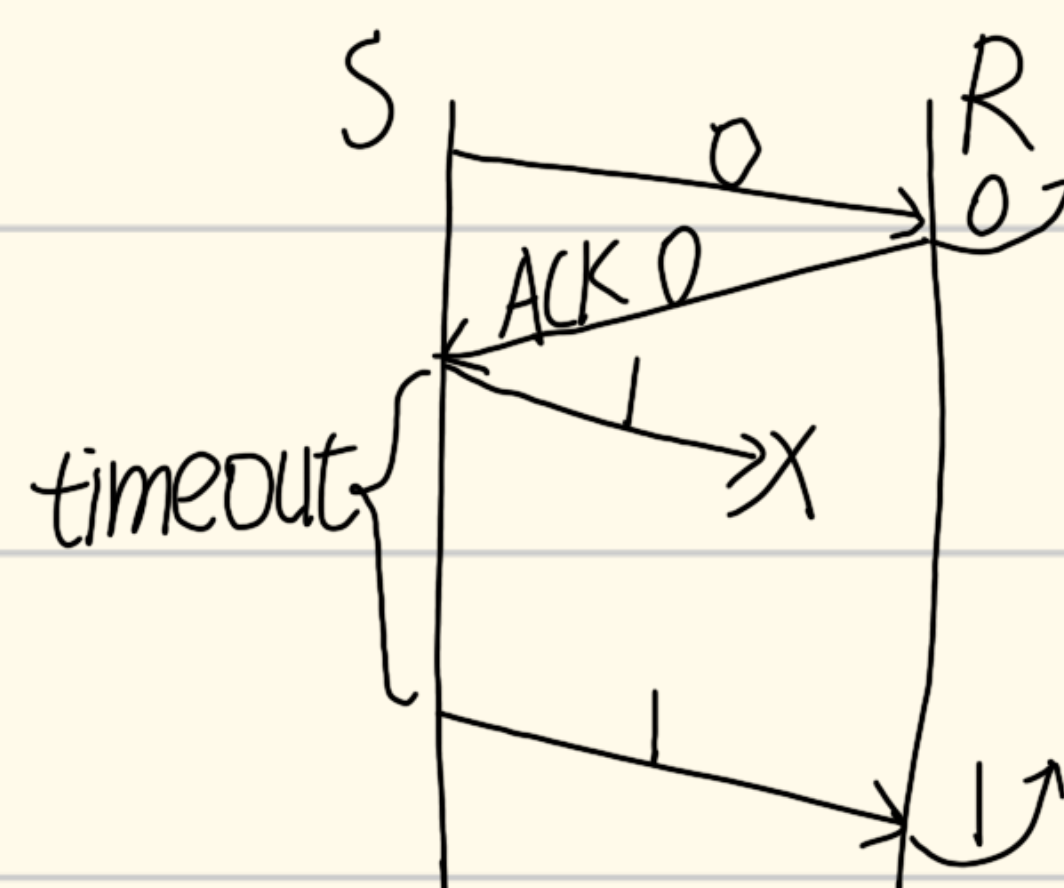
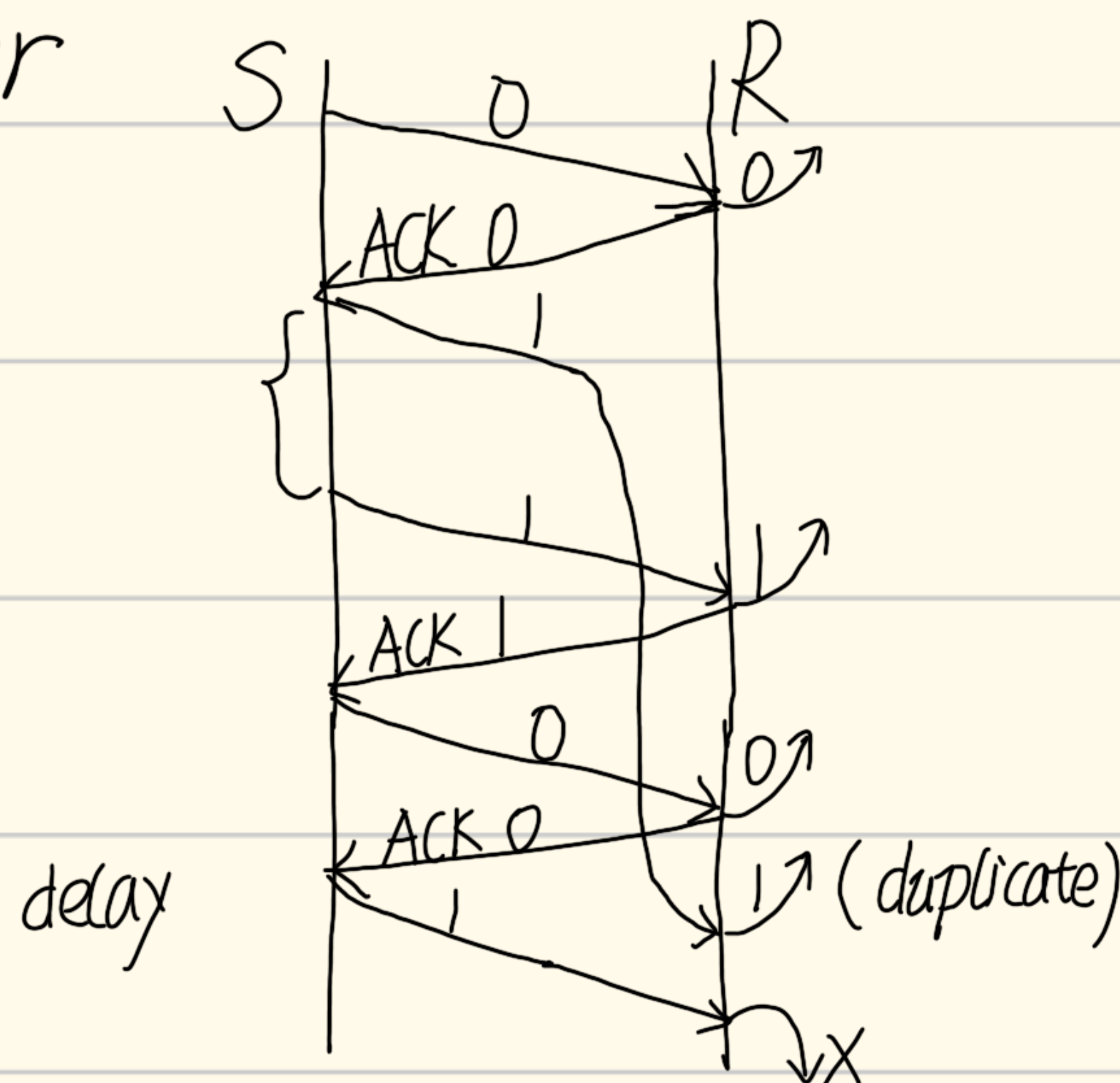
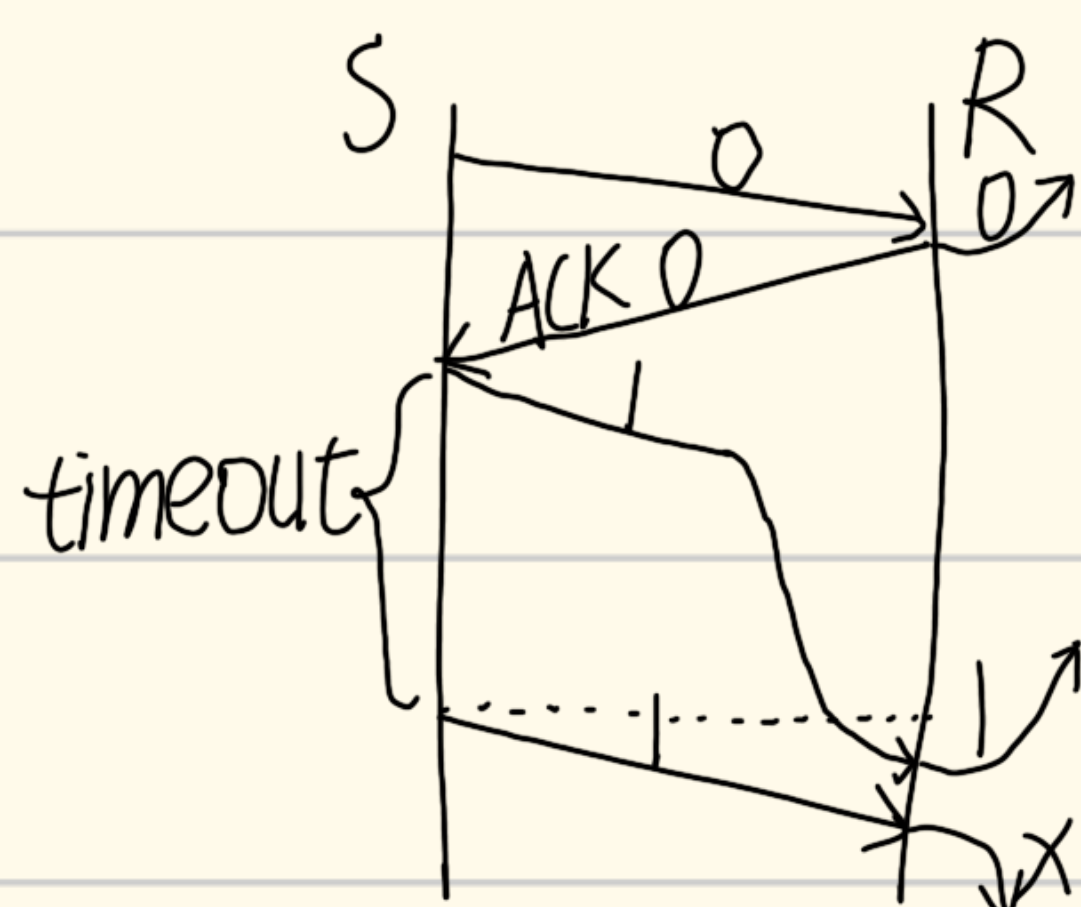
- rdt 2.2: a NAK-free protocol

ACK pck includes seq#
retransmit if duplicate ACK



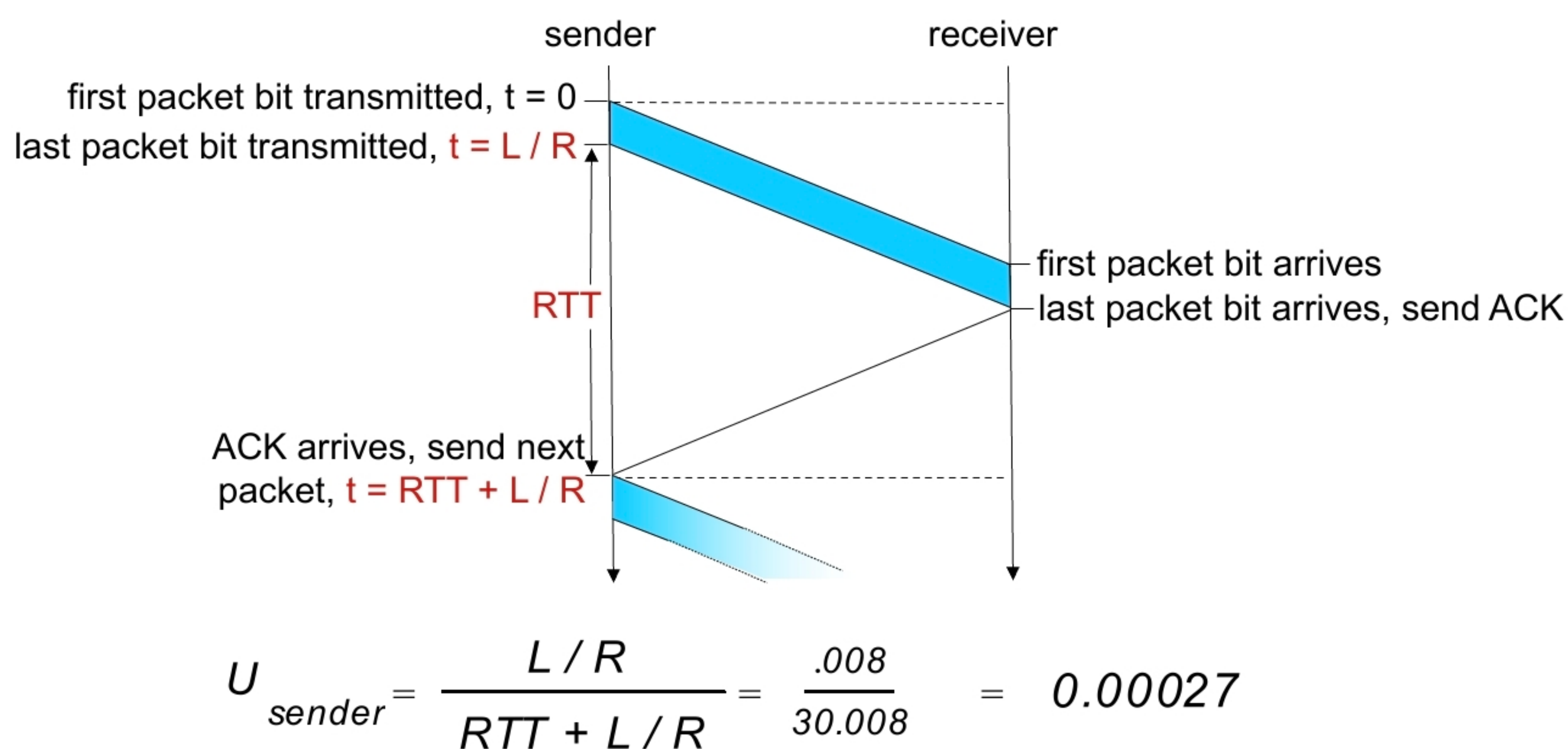
- rdt 3.0: channels with loss

— count down timer



loss

低效的 stop-and-wait
(尤其是 $d_{prop} \gg d_{trans}$)



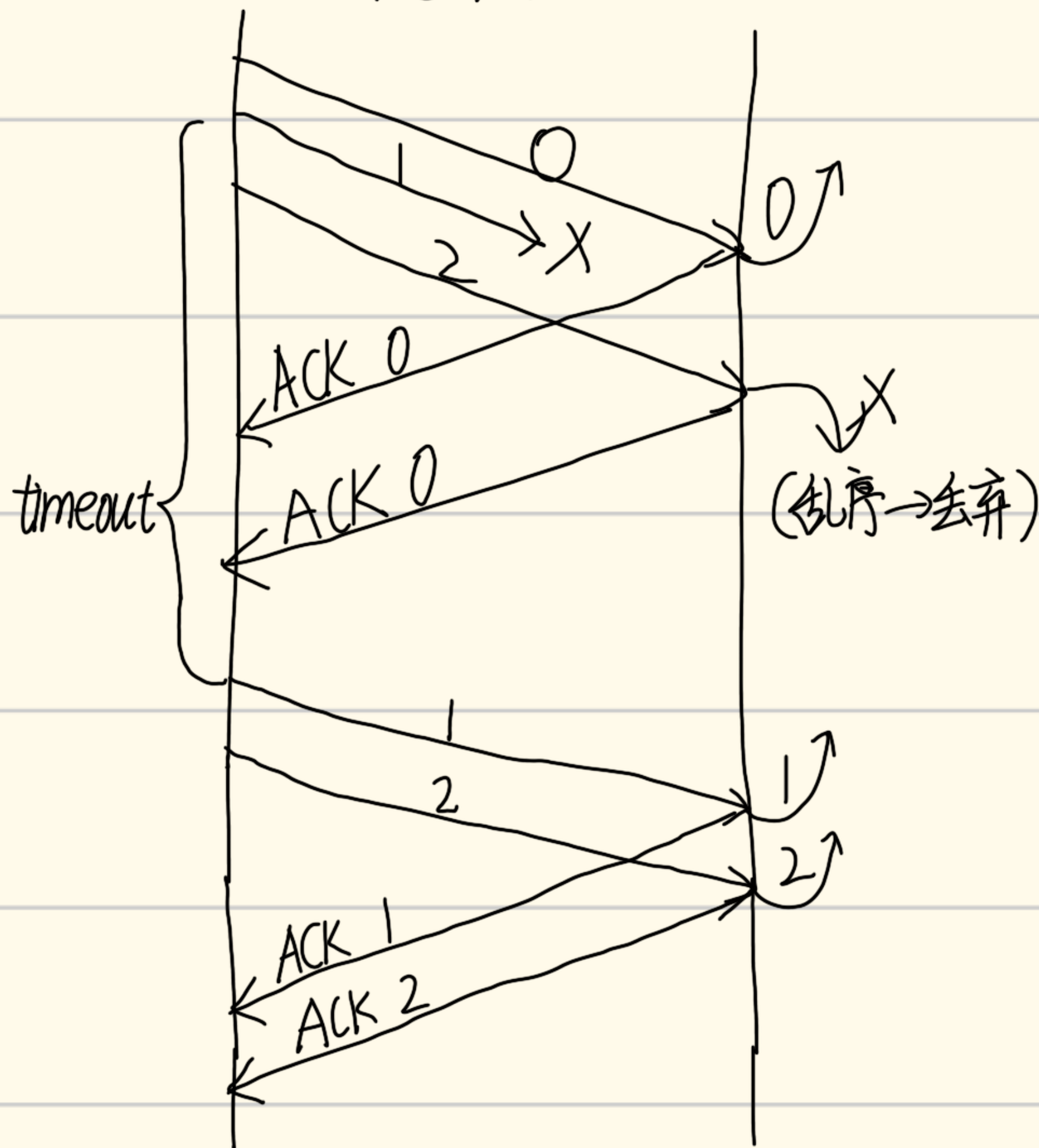
• pipeline

pipelining: sender allows multiple, "in-flight", yet-to-be-acknowledged pkts

eg. Go-Back-N

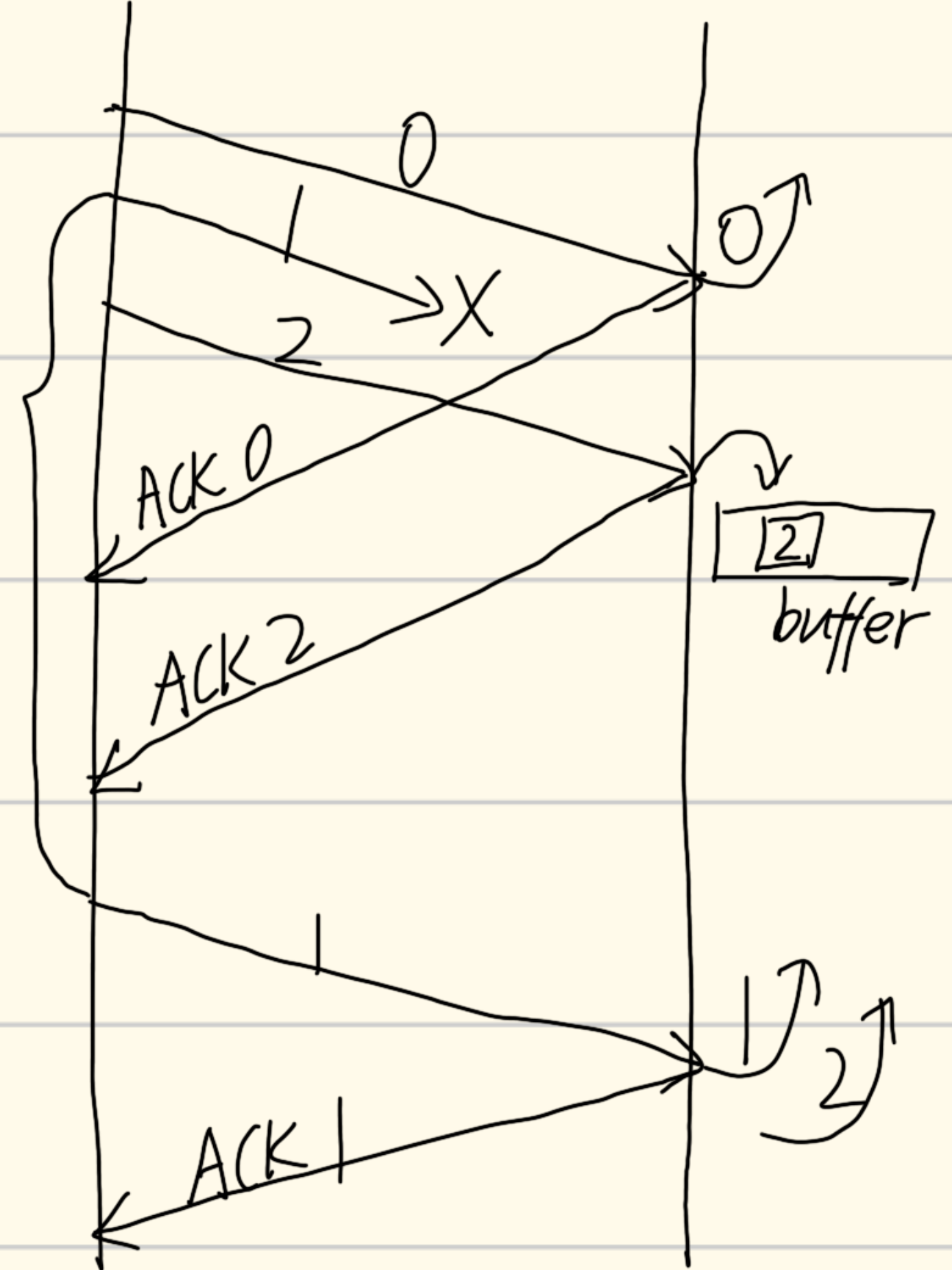
Selective-Repeat

GBN

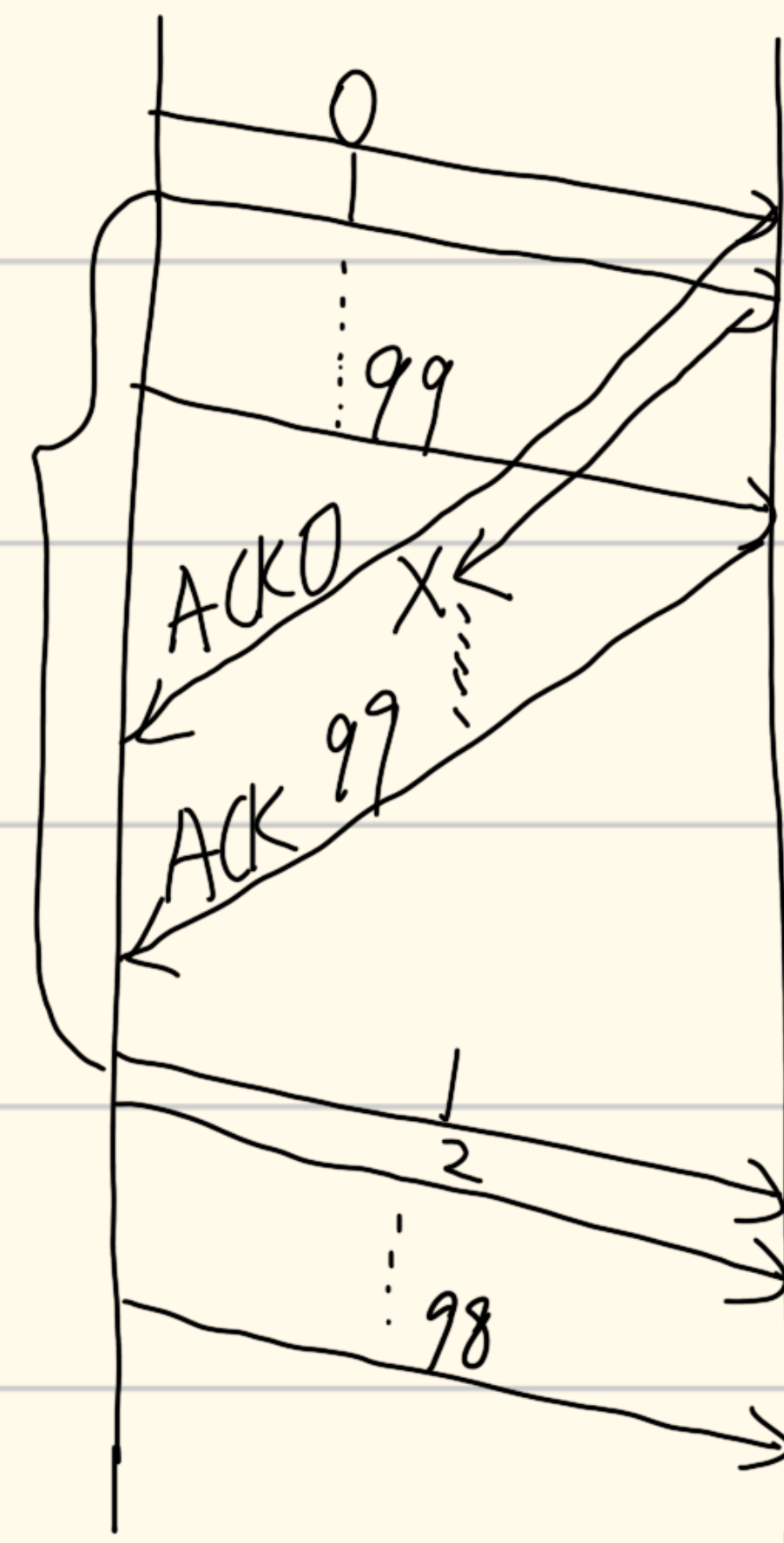
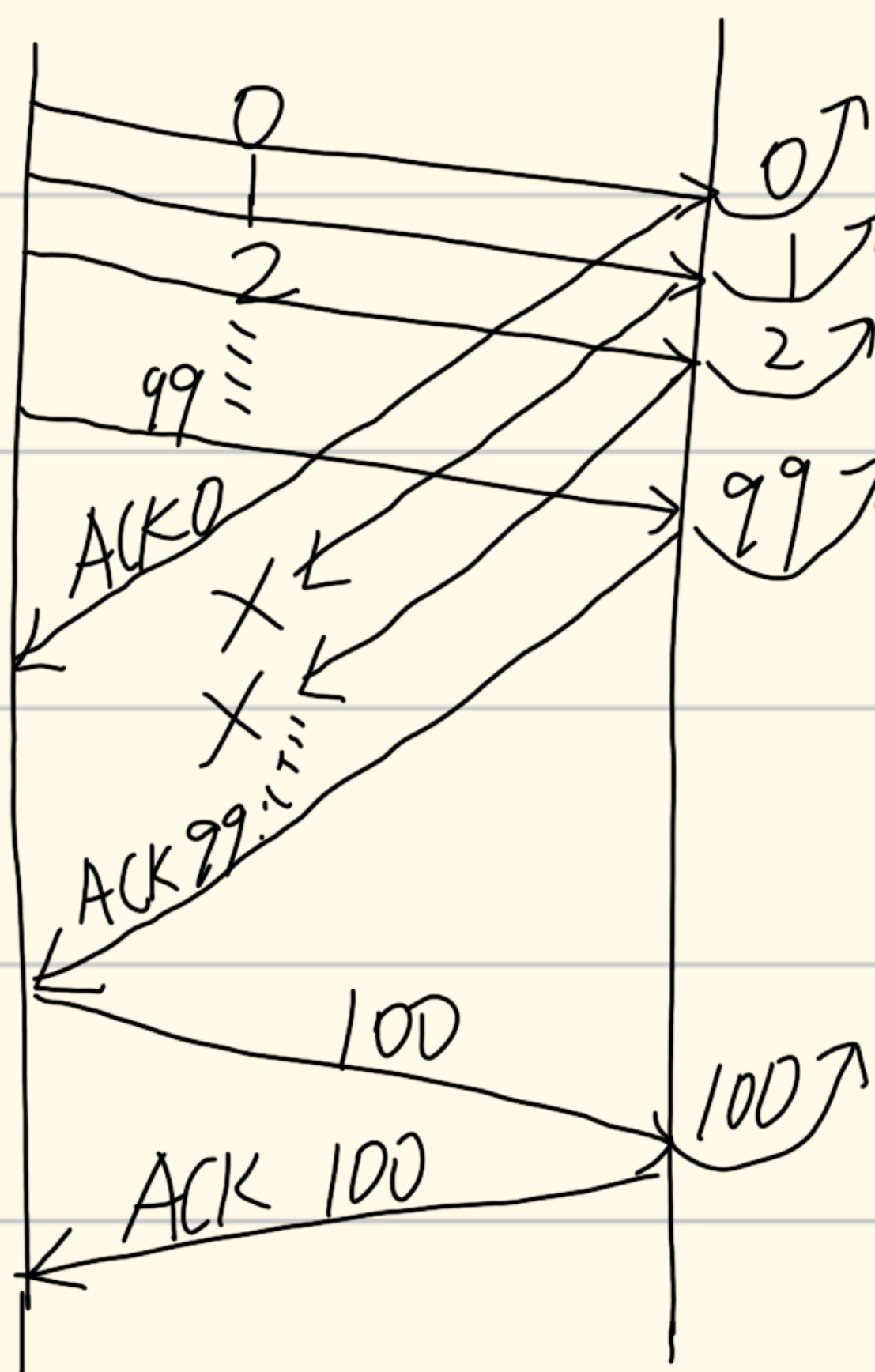


cumulative ack
only one timer (重传 timer)

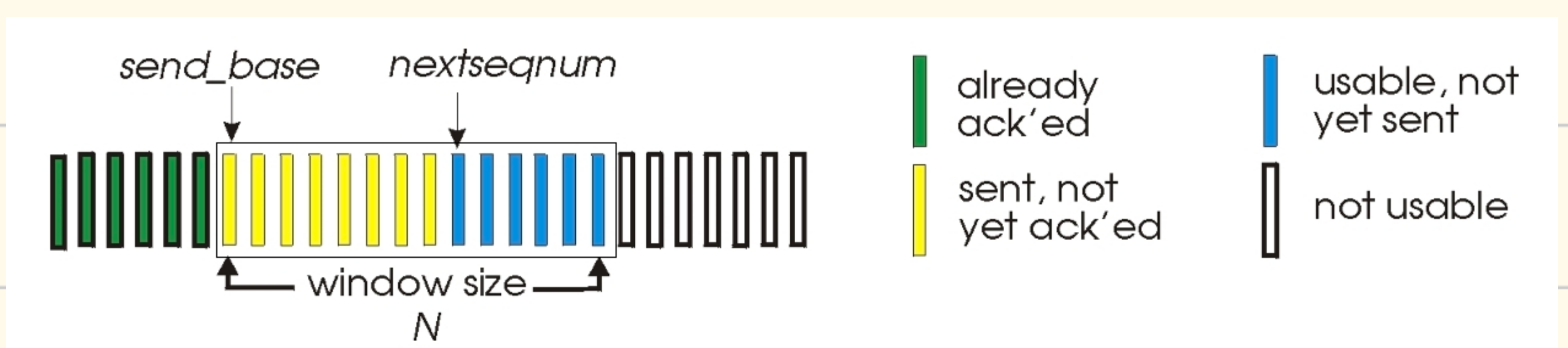
SR



individual ack
timer for each packet

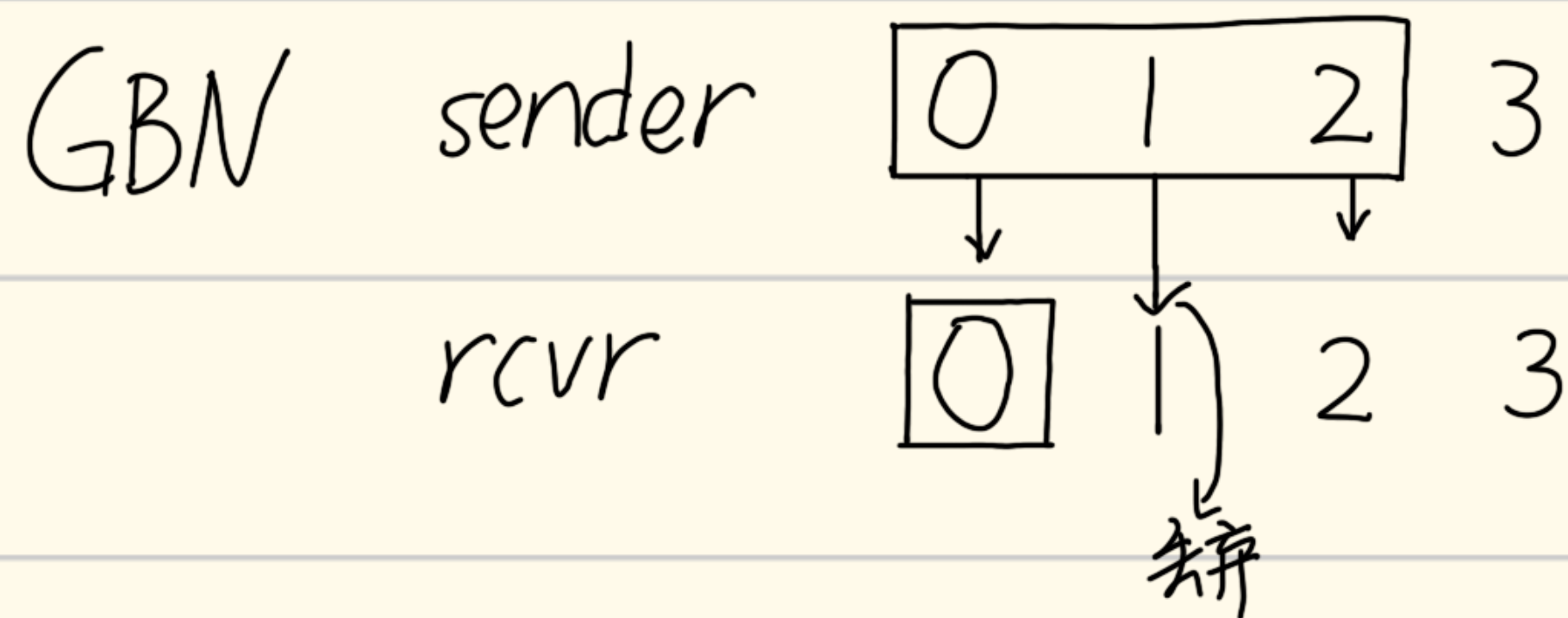


• Window

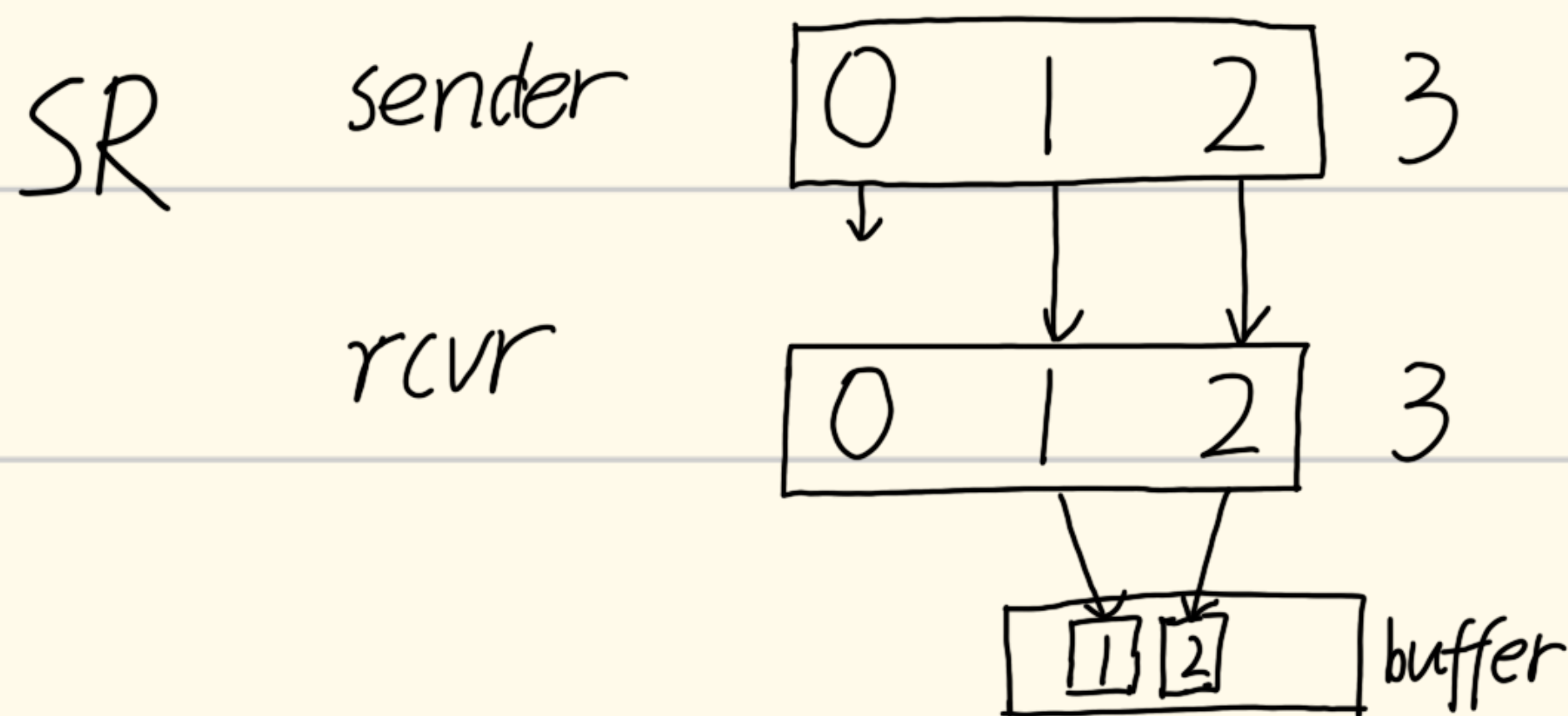


$\left\{ \begin{array}{l} \text{size: 处理 fast-sender \& slow-rcvr (如线程死锁)} \\ \text{由 rcvr 的 buffer 大小决定} \\ \text{GBN、SR 为固定 size, TCP 为动态 size} \\ \text{position: 左边界} \end{array} \right.$

注: GBN 接收端窗口大小永远为 1

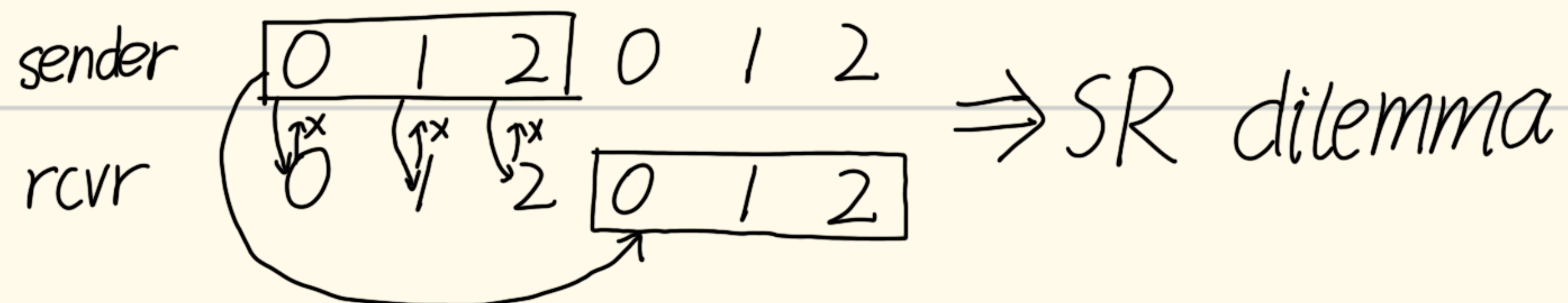
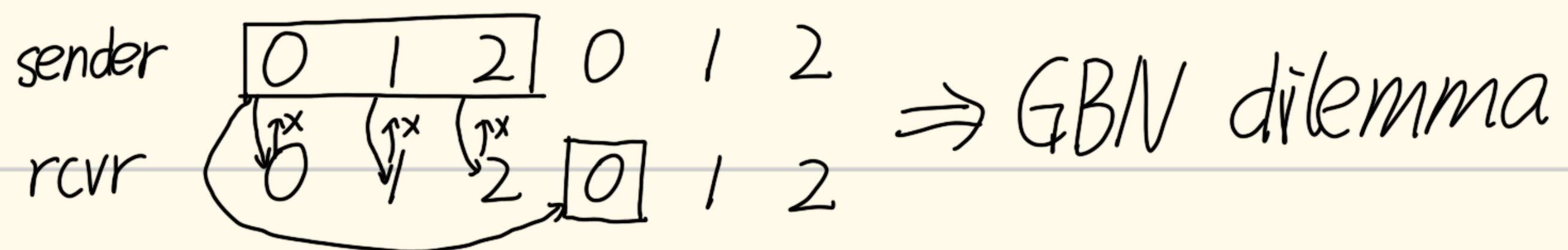


GBN: 收到窗口任一 ack, sender 即可右滑



SR: 收到窗口最左边 ack, sender 才可右滑

当 Seq# size $\leq$ Window size 时,



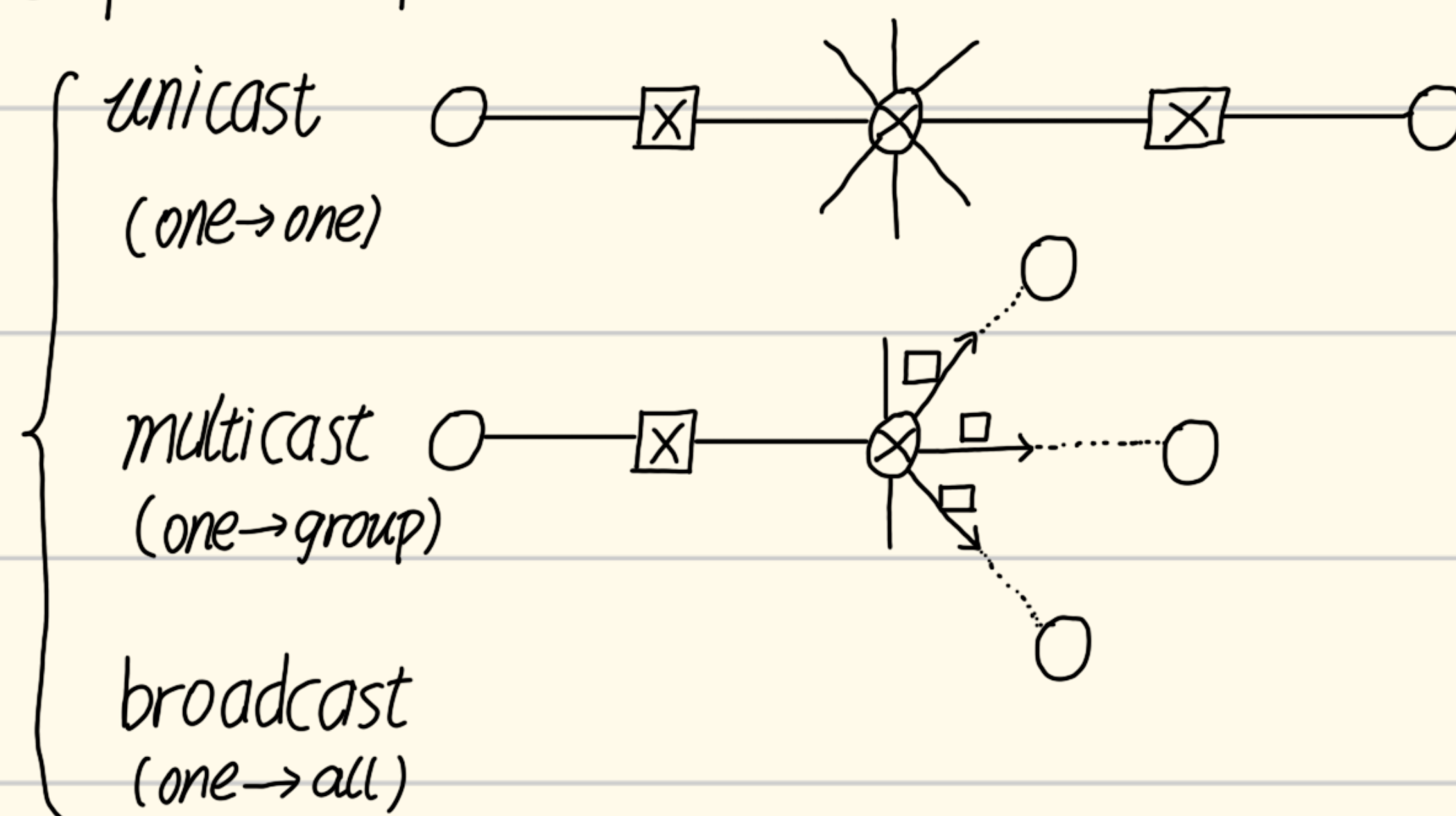
3.5 Connection-oriented

Transport: TCP

TCP = transmission control protocol

TCP: Overview

① point to point

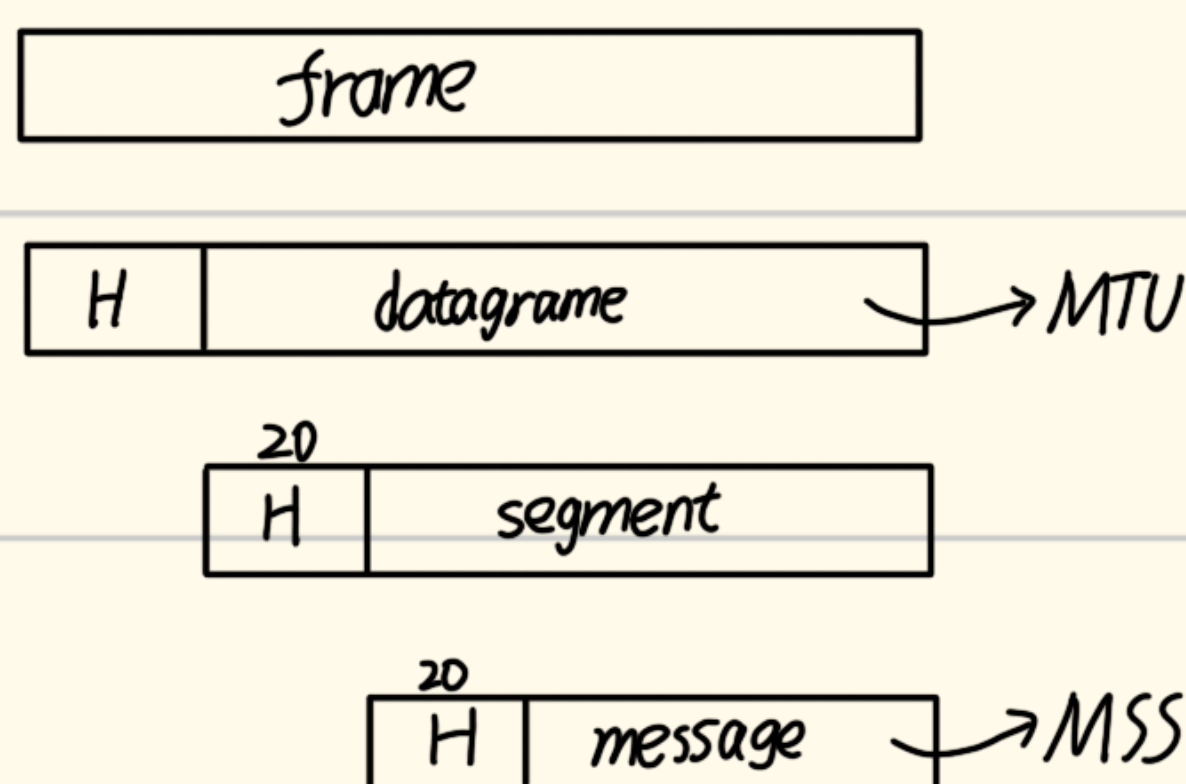


② no message boundaries

③ pipelined: TCP congestion & flow control → set window size

④ full duplex data: MSS [maximum segment size = 1460]
(只算seg包体)

MTU [maximum transfer unit = 1500]
(只算frame包体)

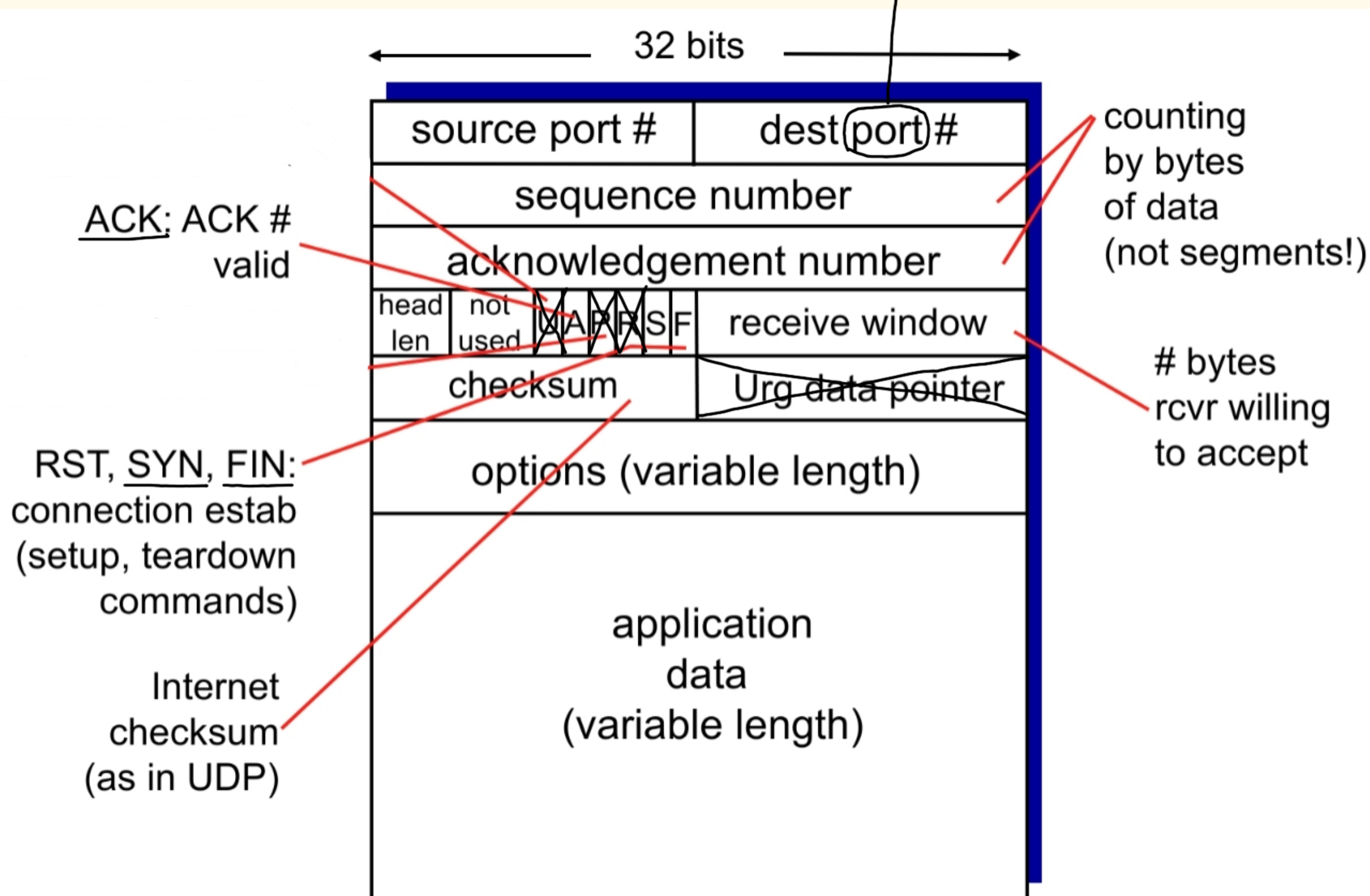


⑤ connection-oriented: handshaking

⑥ flow controlled: sender NOT overwhelm rcvr

TCP Segment

80 http(web) 21 ftp 25 smtp 110 pop3 53 dns 443 https

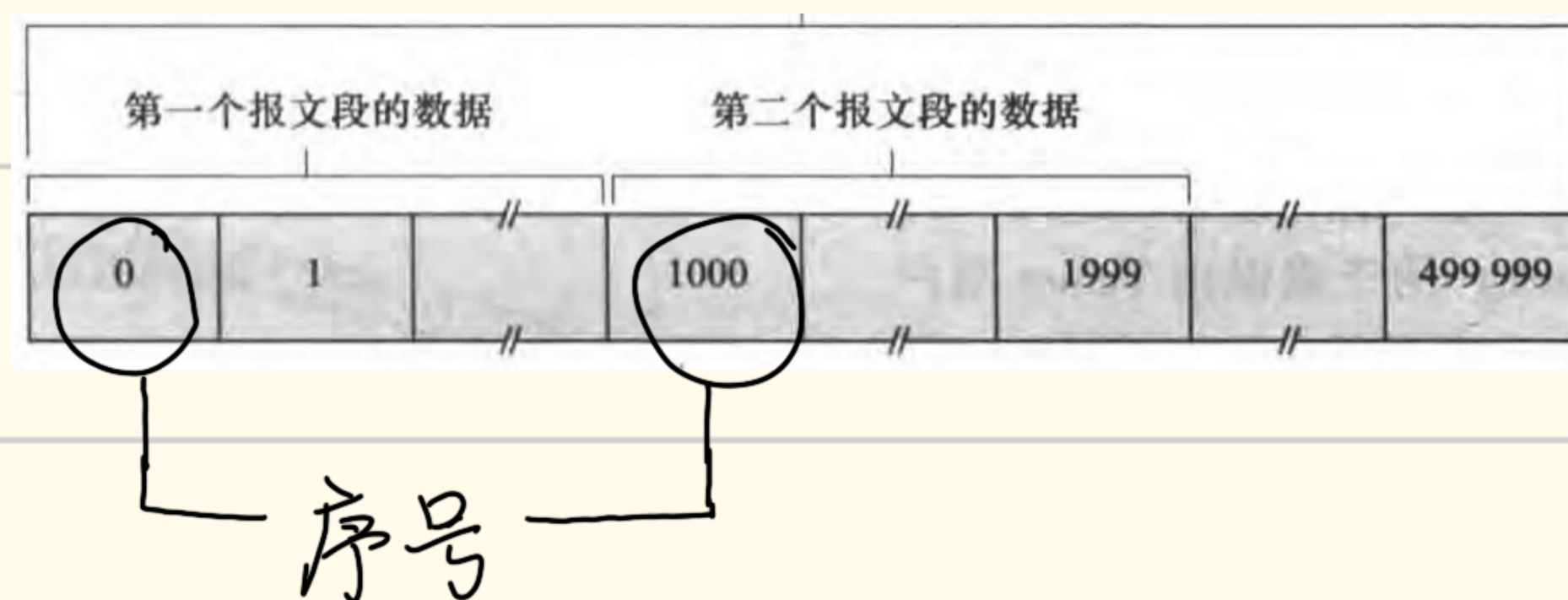


TCP seq. numbers, ACKs

sequence number for a segment

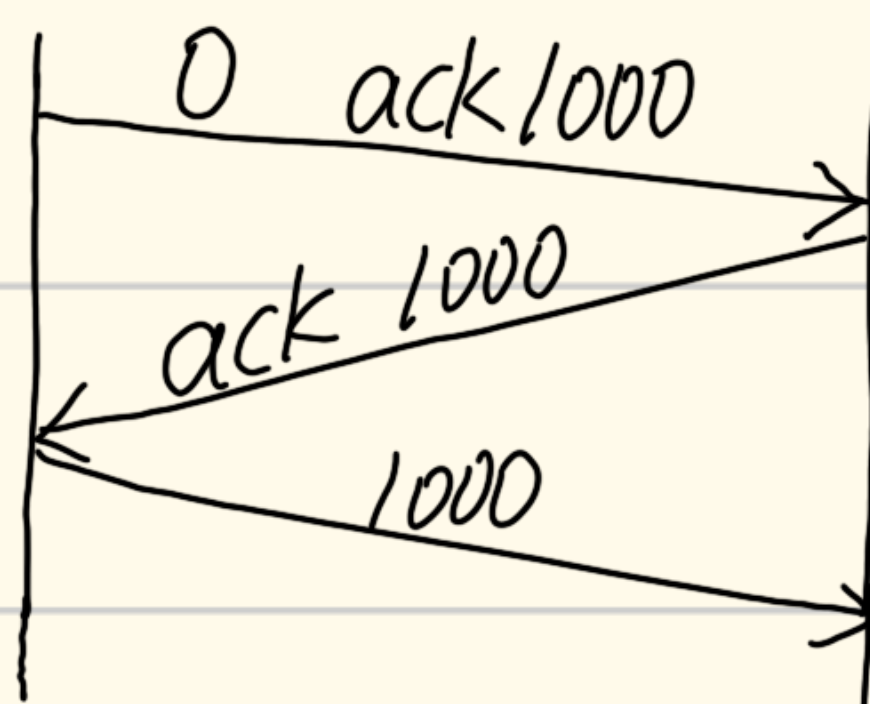
eg. 数据流 500,000 bytes 对 message 字节编号

MSS = 1000 字节

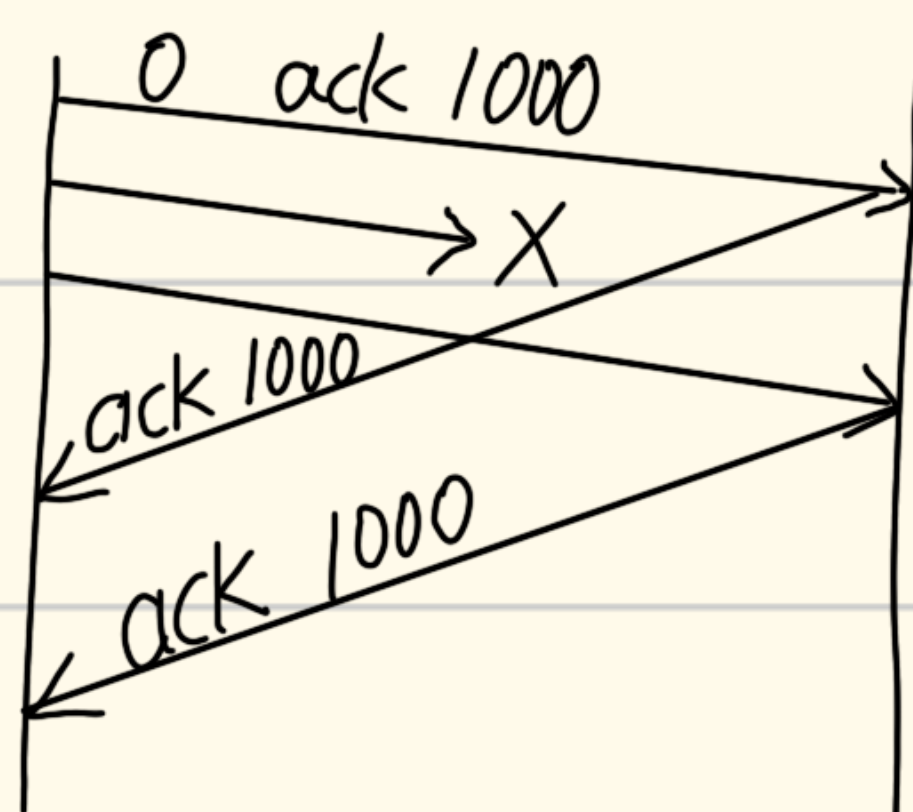


acknowledgement

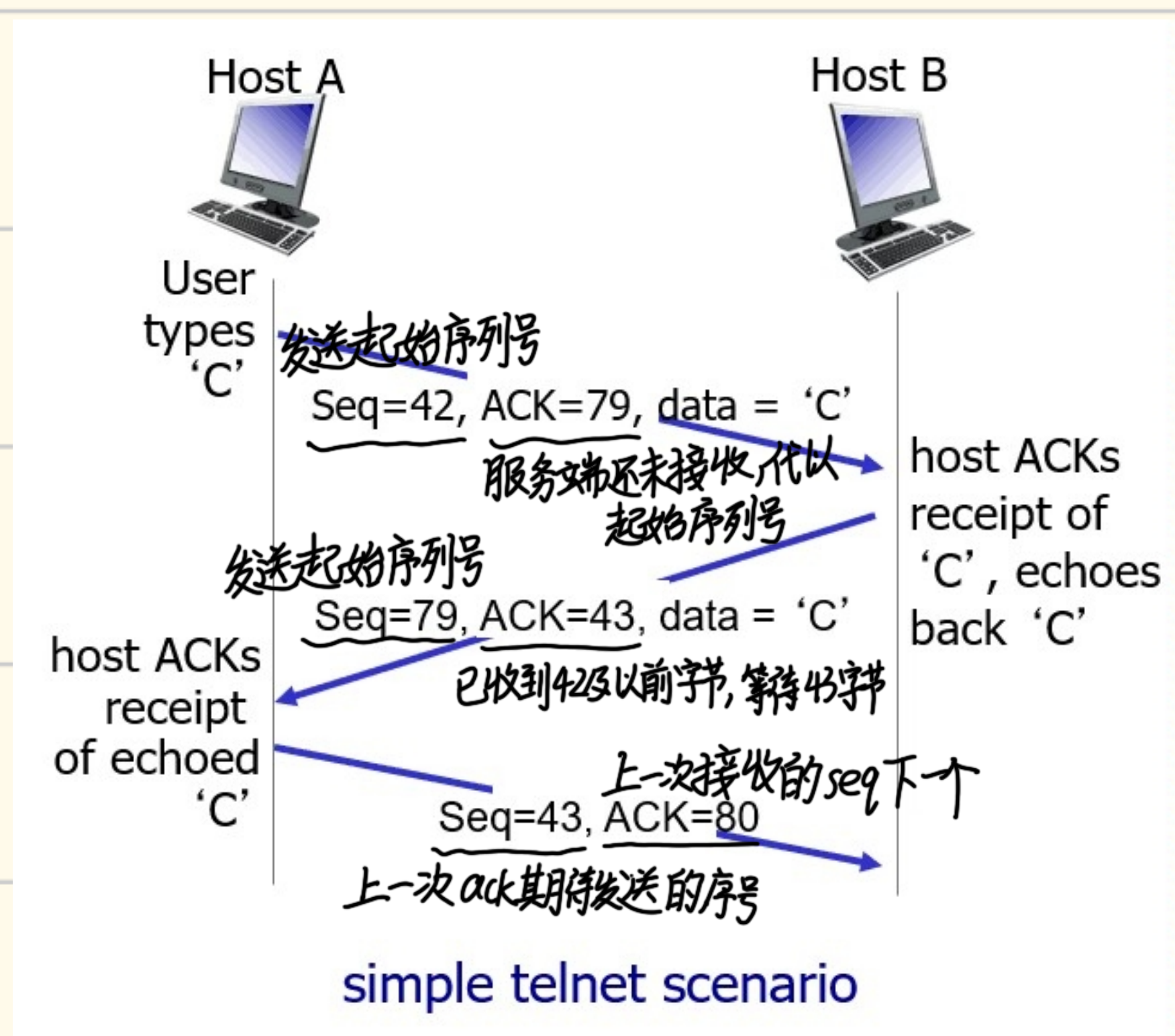
seq # of next byte expected



cumulative ACK



eg.



TCP round trip time

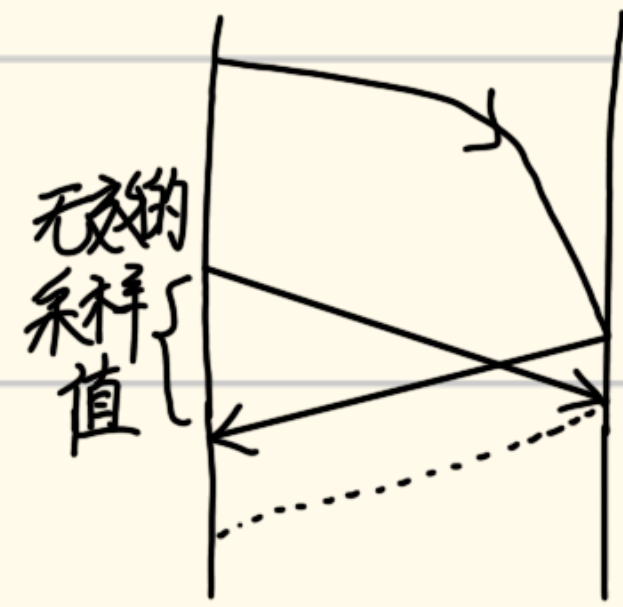
RTT + a bit longer time : exponential weighted moving average

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$

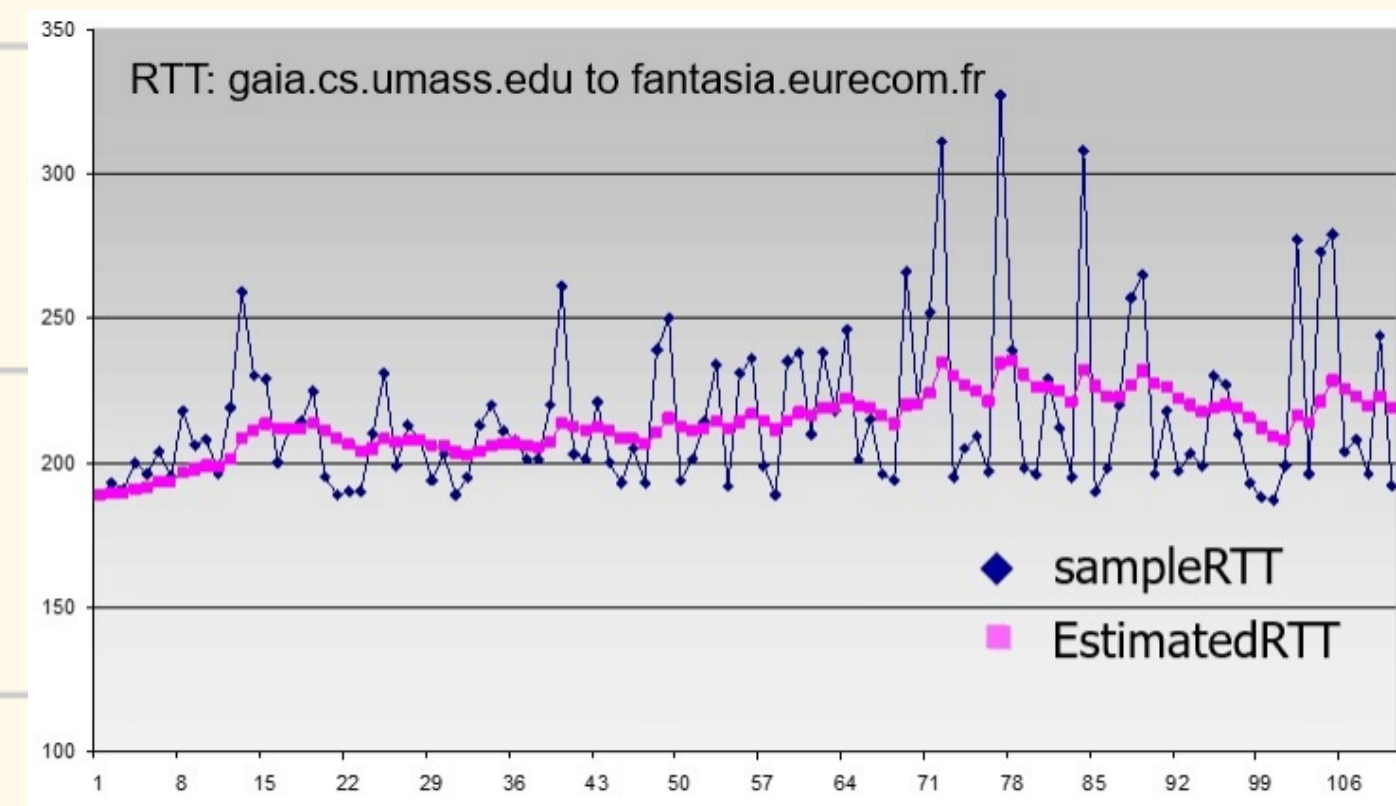
RTT: exponential weighted moving average

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

注:重传包不能计入 SampleRTT

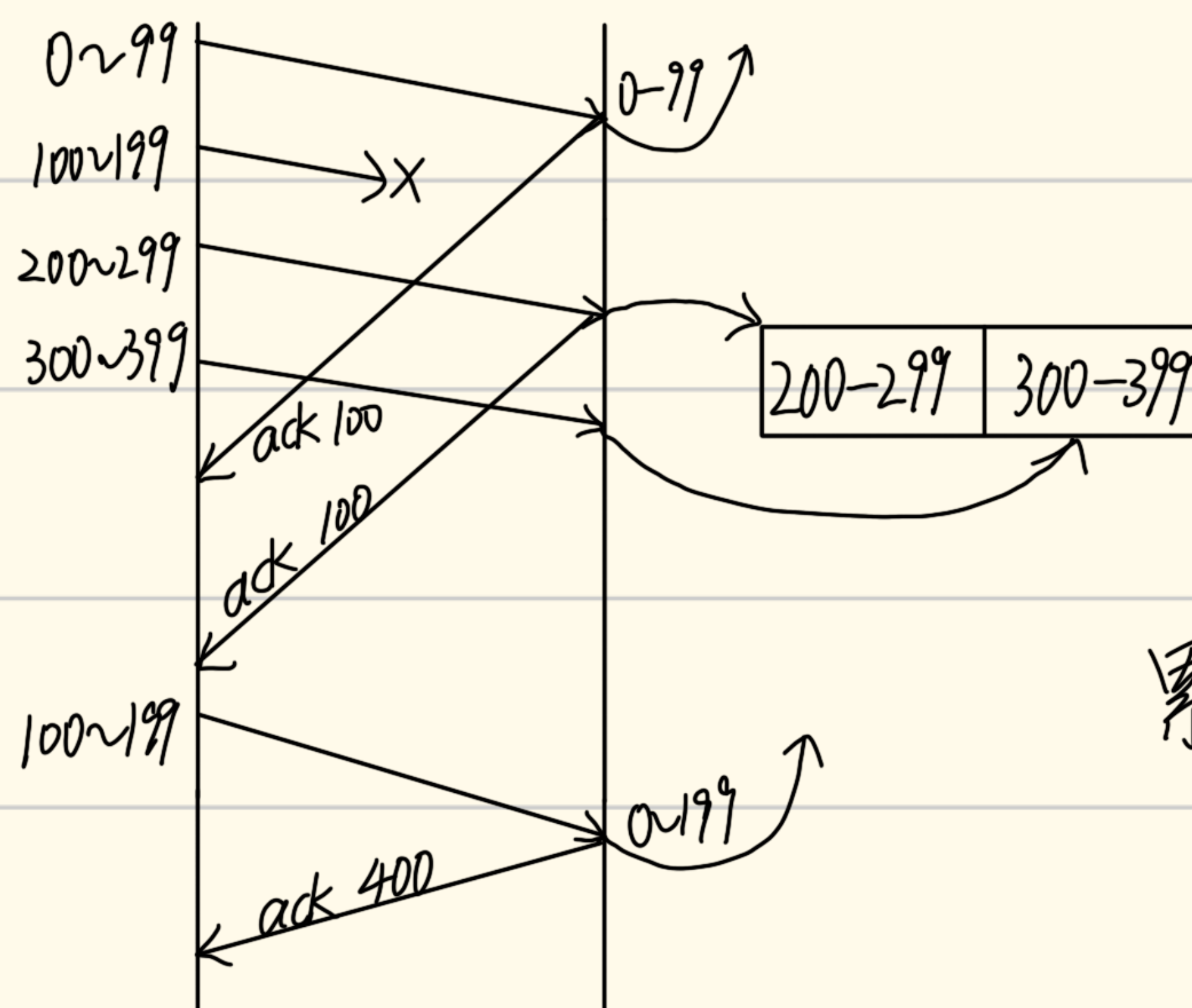
 $(\alpha = 0.125)$ 

A Bit Longer Time



$$\text{DevRTT} = (1-\beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}|$$

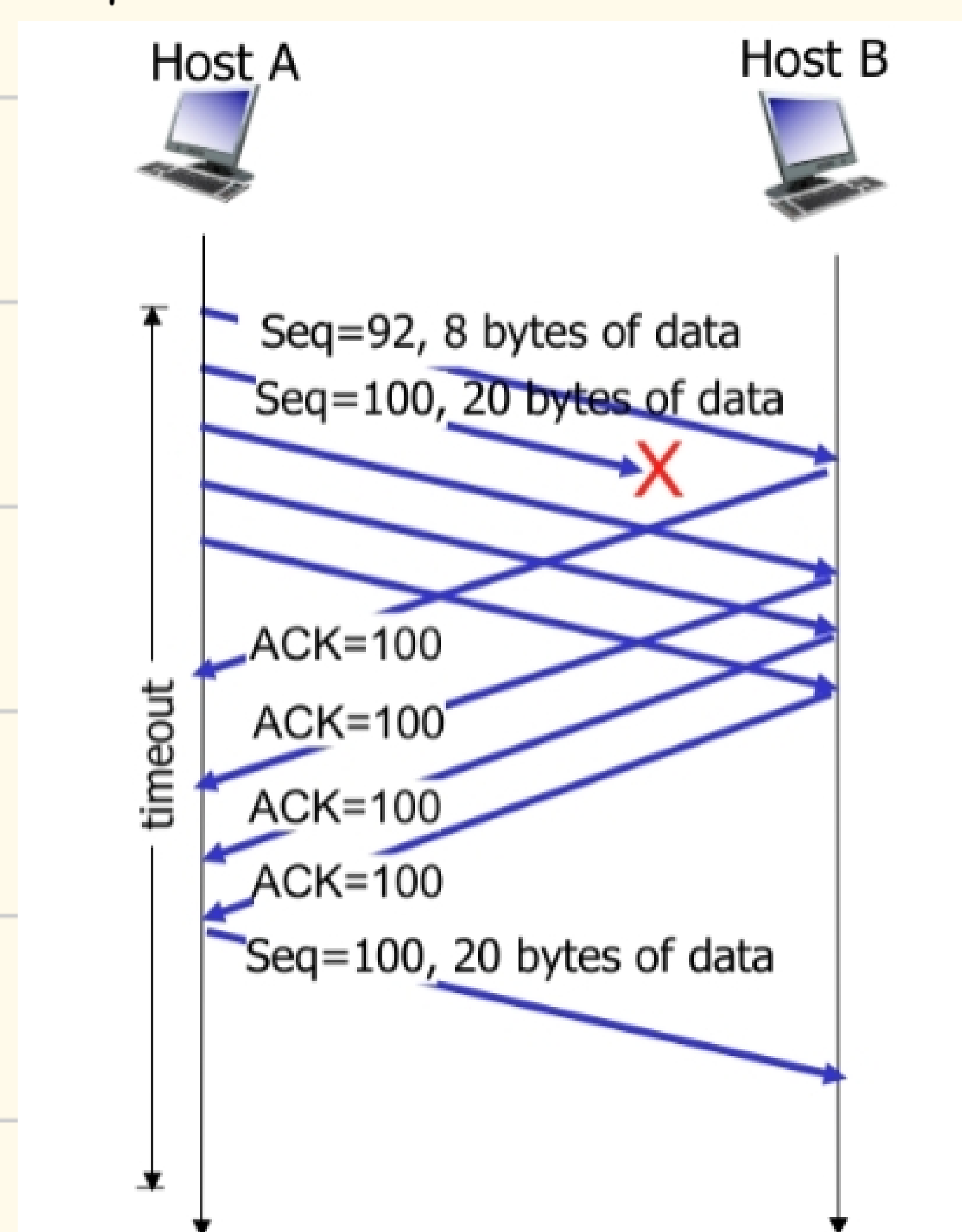
TCP: retransmission scenarios



累积ACK + 缓存

TCP fast retransmit

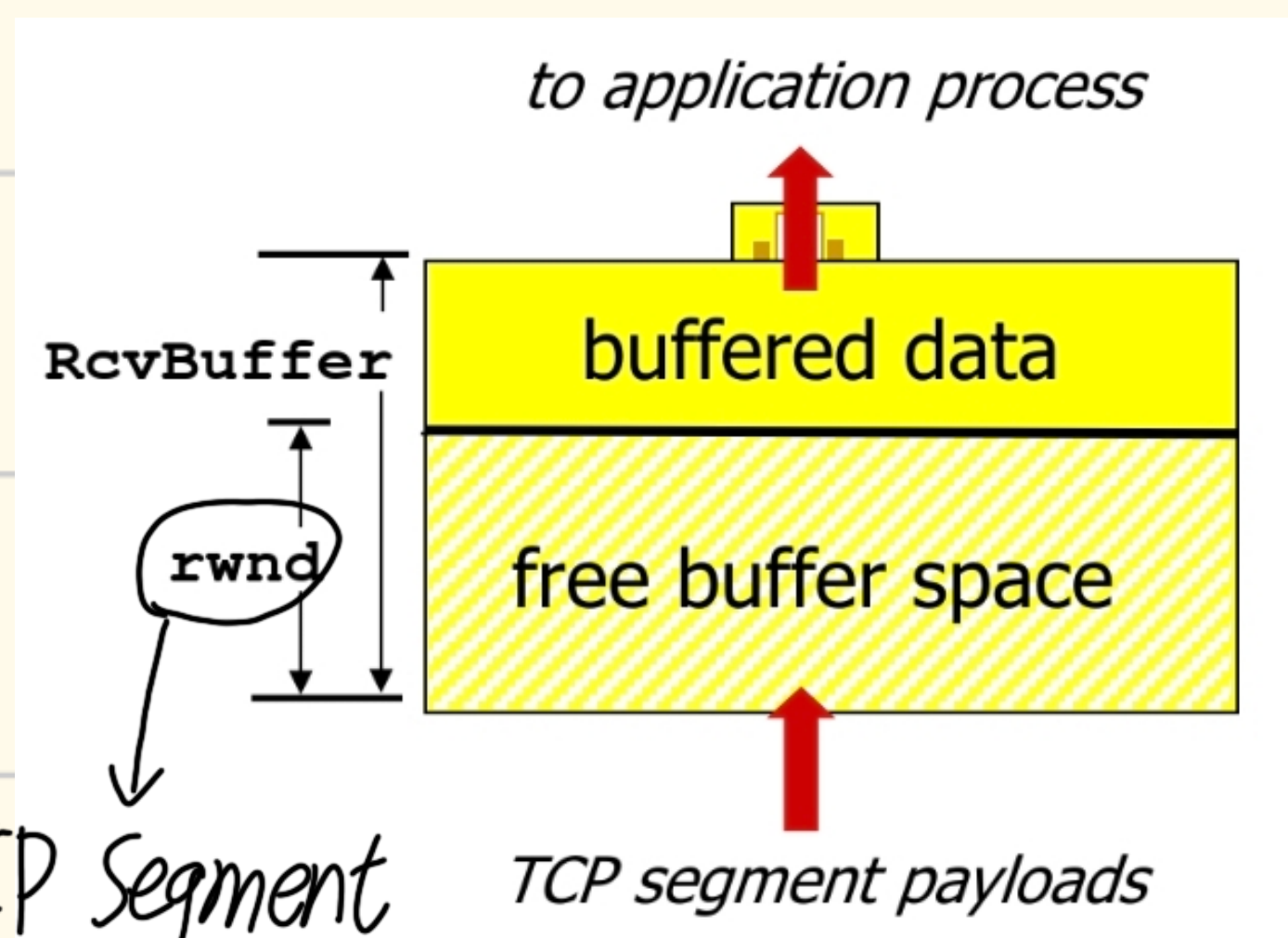
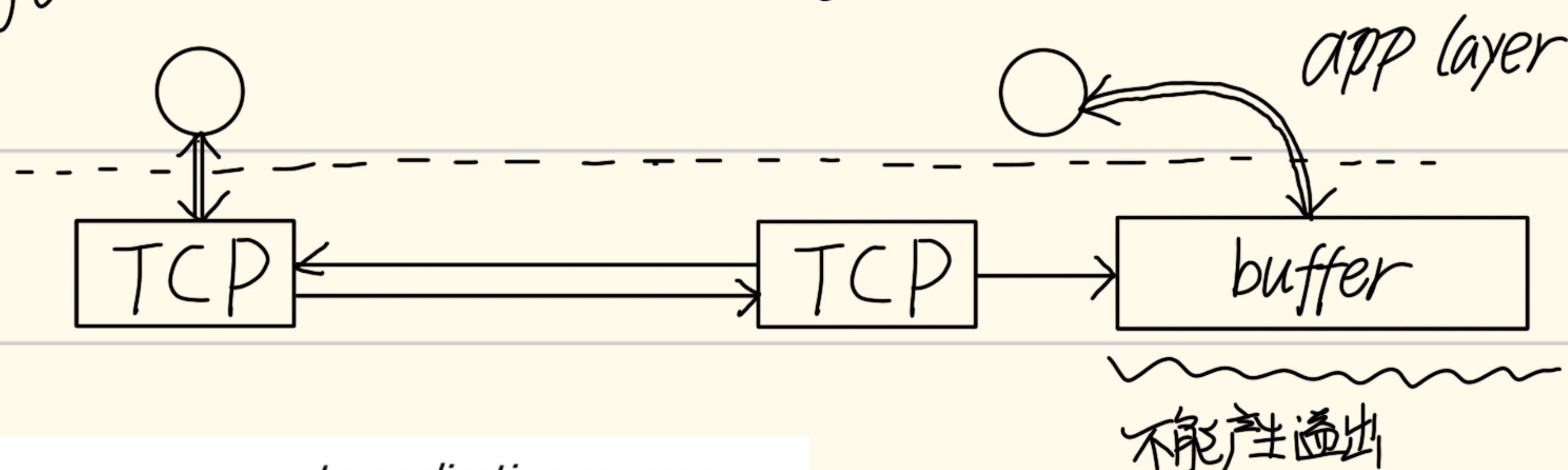
triple duplicate ACK $\rightarrow$ retransmit



TCP flow control

fast sender

slow rcvr



TCP Segment

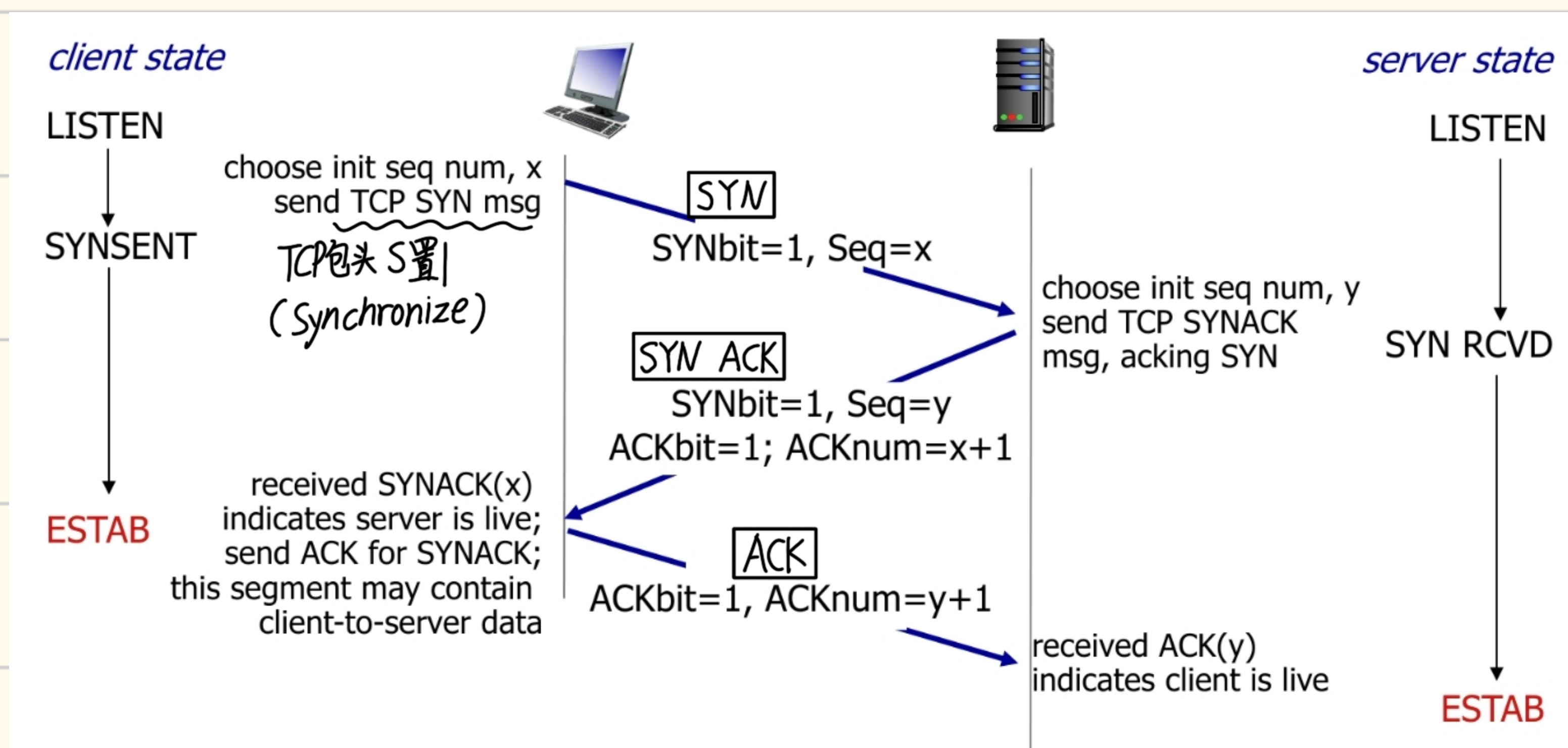
TCP segment payloads

中的receive window $\Rightarrow$ 决定了sender窗口大小, TCP segment包体大小

Sender window size = $\min(rwnd, cwnd)$

TCP segment包体大小 $\leq rwnd$ 且 $\leq MSS$

TCP connection management



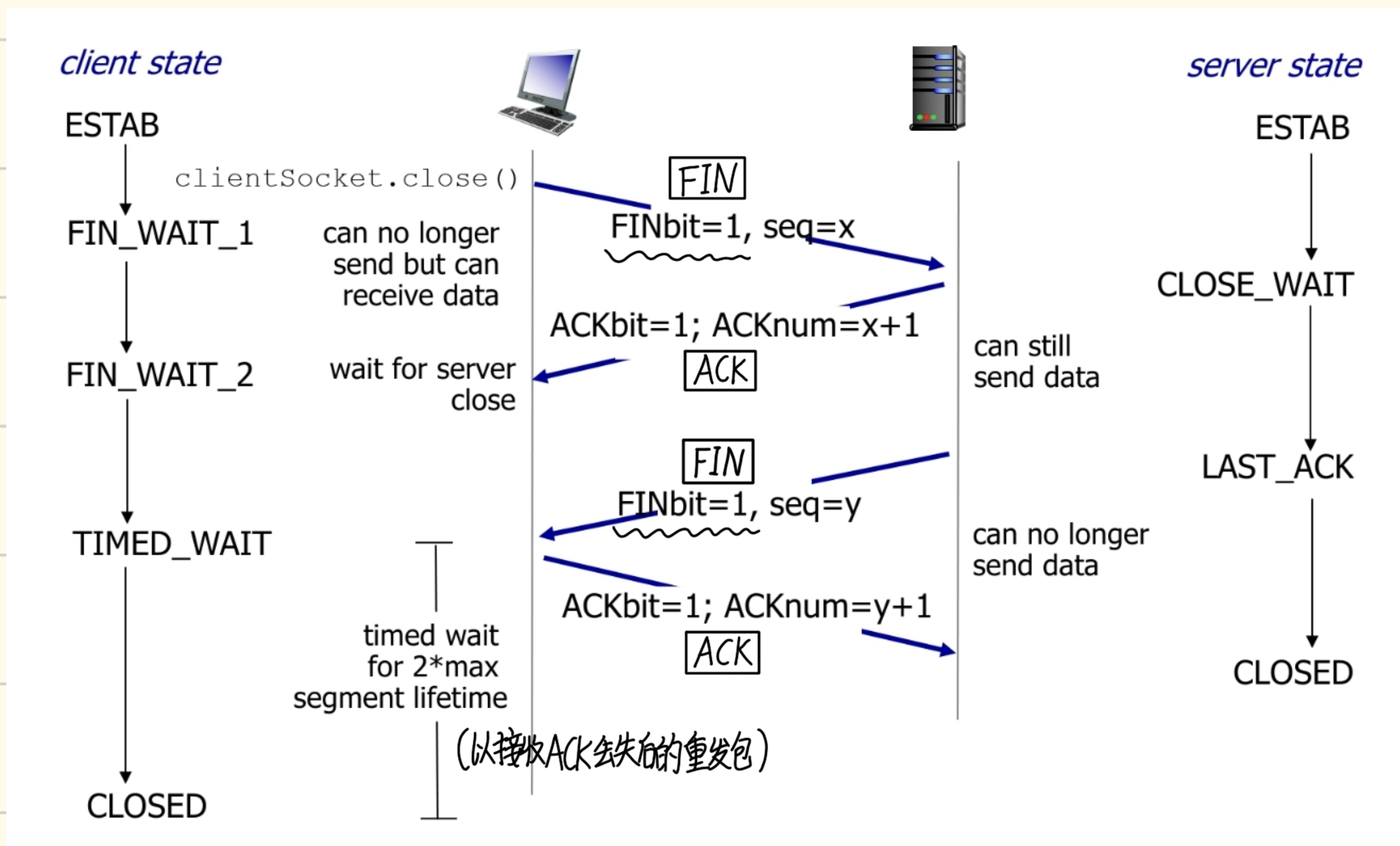
若SVR故障/未开始: 返回RST(R=1)

SYN (S=1) $\left\{ \begin{array}{l} \text{initial sequence number (ISN): } 0 \sim 2^{32} \text{ 随机} \\ \text{rwnd} \\ \text{MSS} \end{array} \right.$

SYN ACK (S=1, A=1) $\left\{ \begin{array}{l} \text{ISN} \\ \text{ACKnumber} = \text{SYN的Seq} + 1 \end{array} \right.$ (消耗第一个序列号, 供应用层首个序列号为 **SYN** 的Seq+1)

ACK (A=1) — ACKnumber = **SYN ACK** 的Seq+1

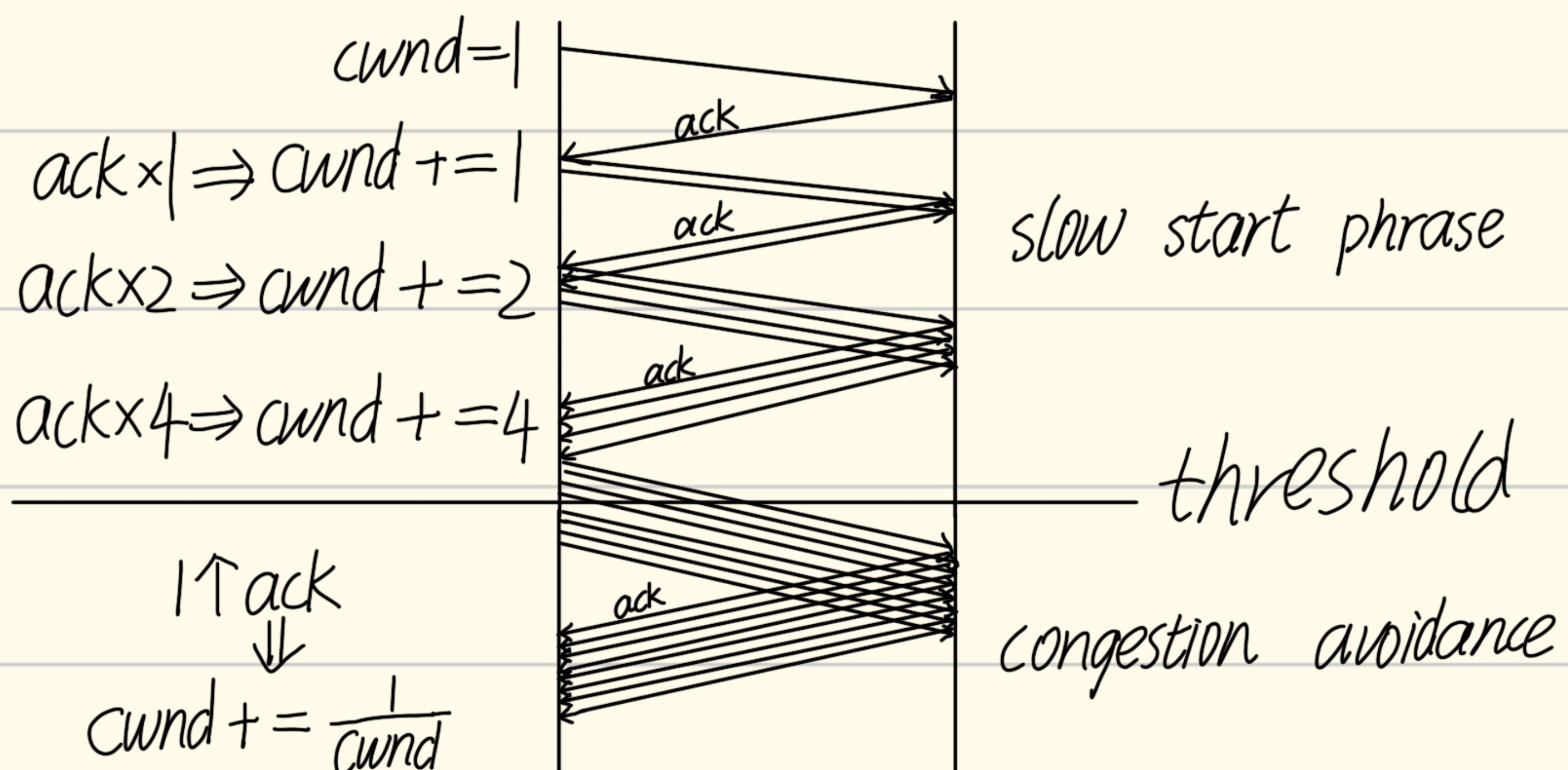
TCP = closing connection

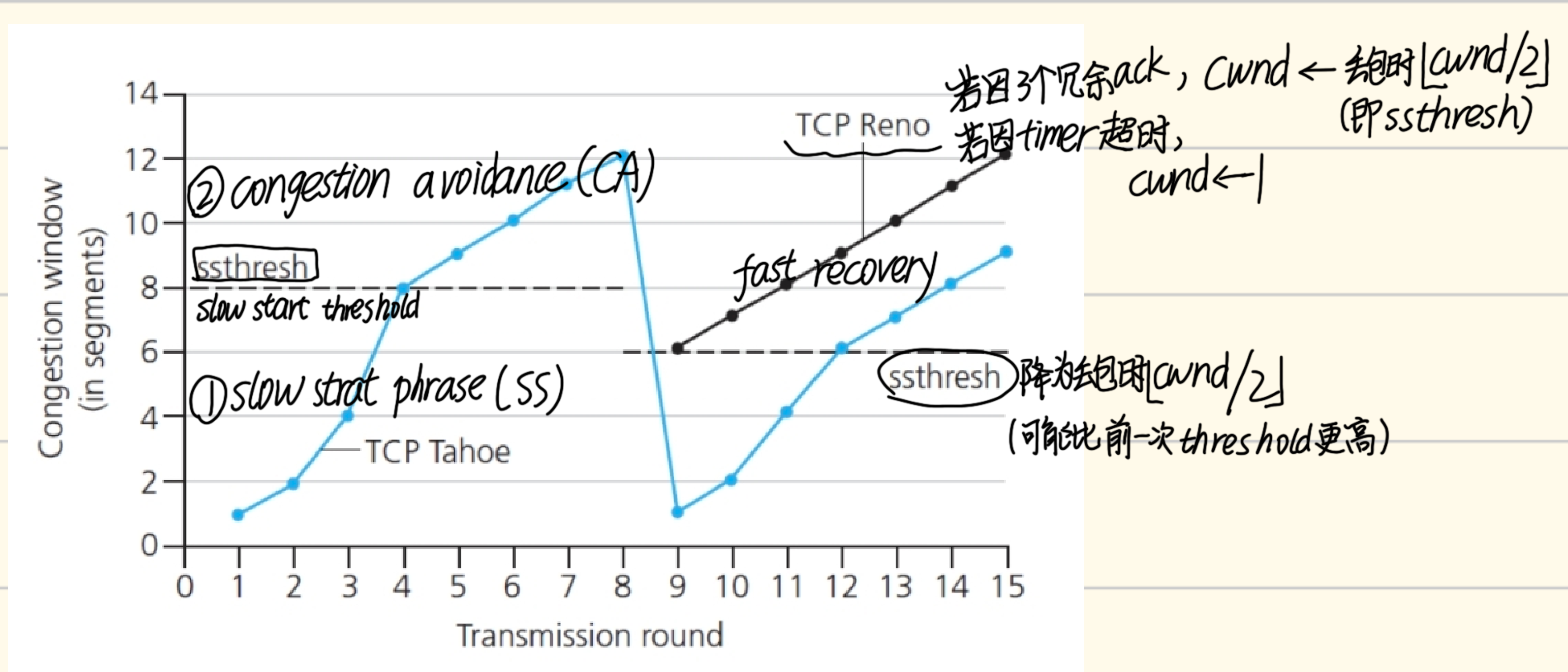


三路握手必为 Client → SVR, 四路挥手 Client, SVR 都可以发起
并非双方同时关闭

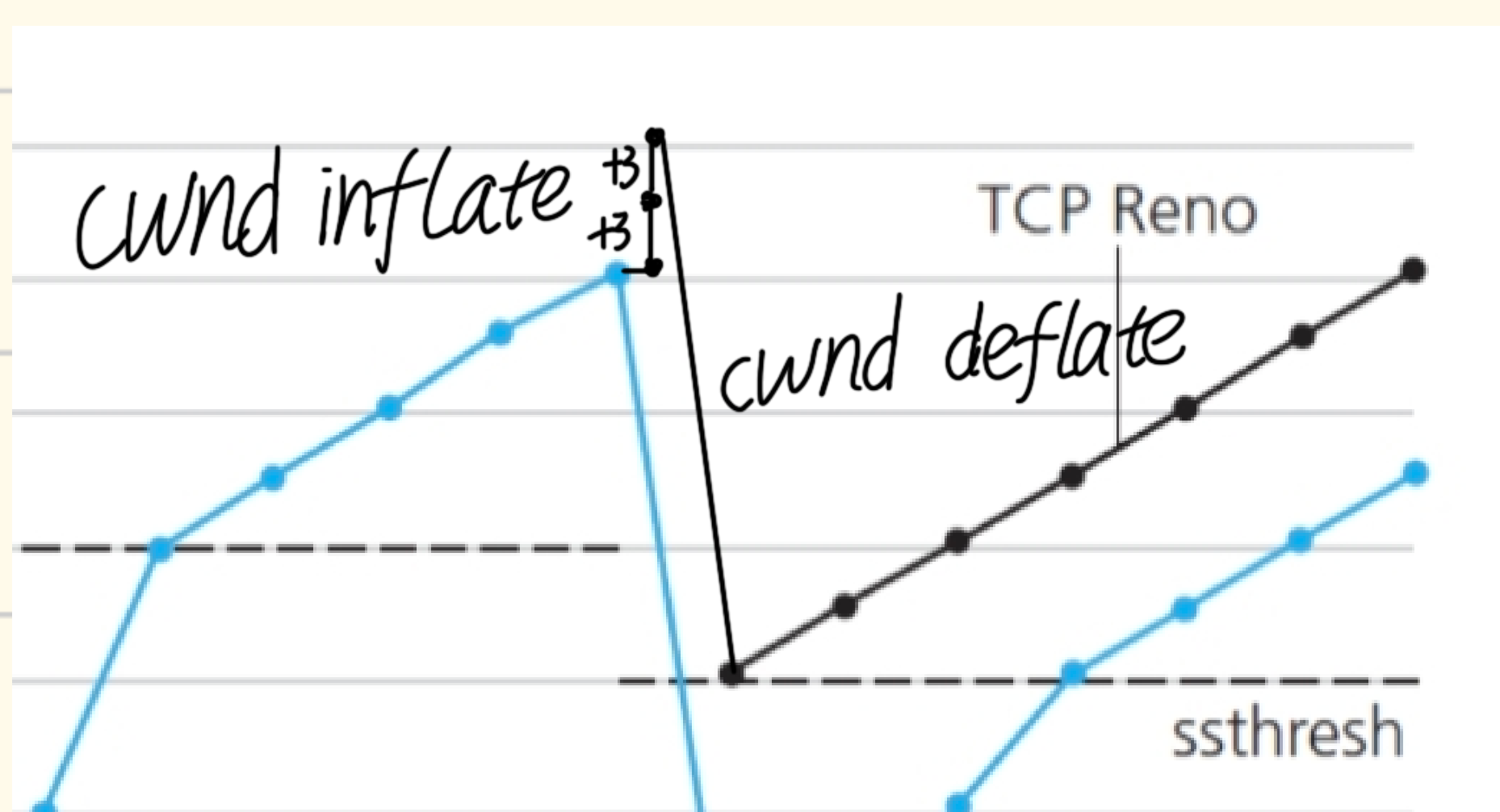
3.7 TCP congestion control

AIMD = additive increase & multiplicative decrease
cwnd (congestion window)



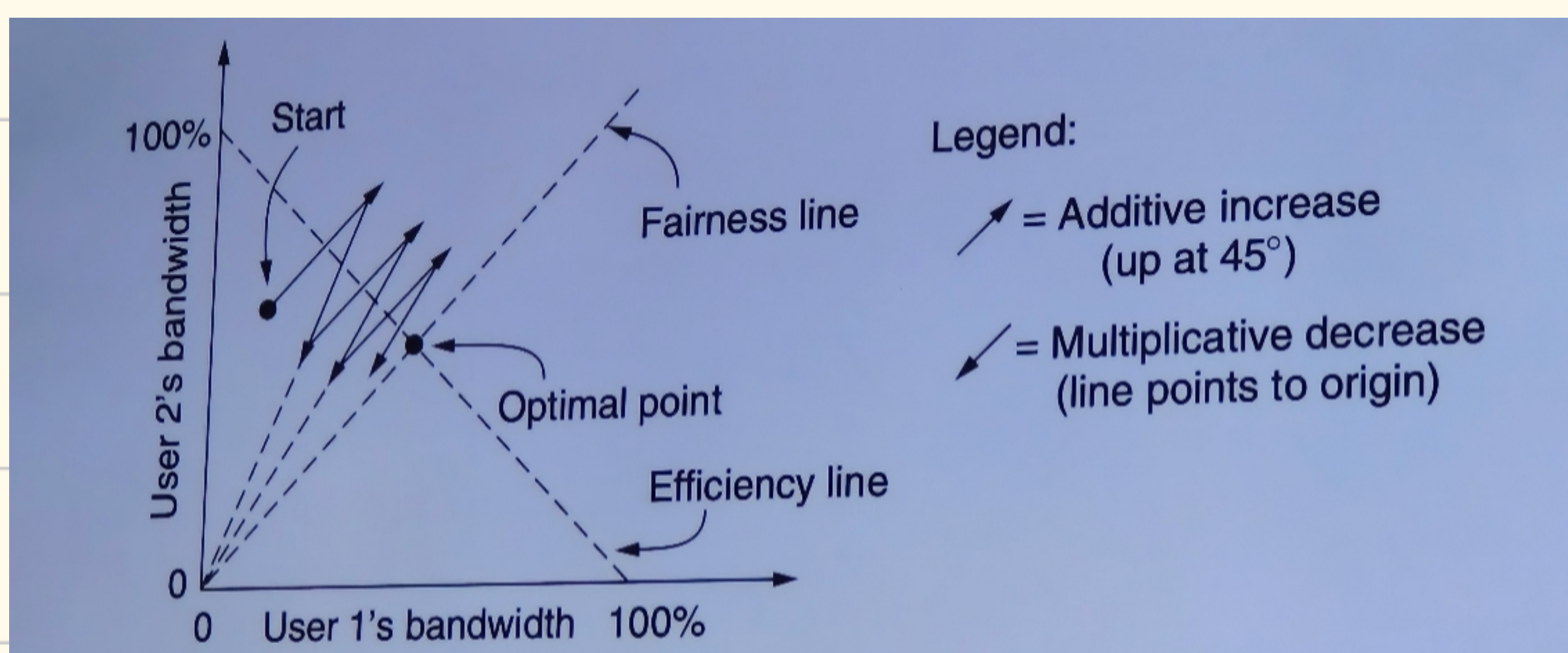
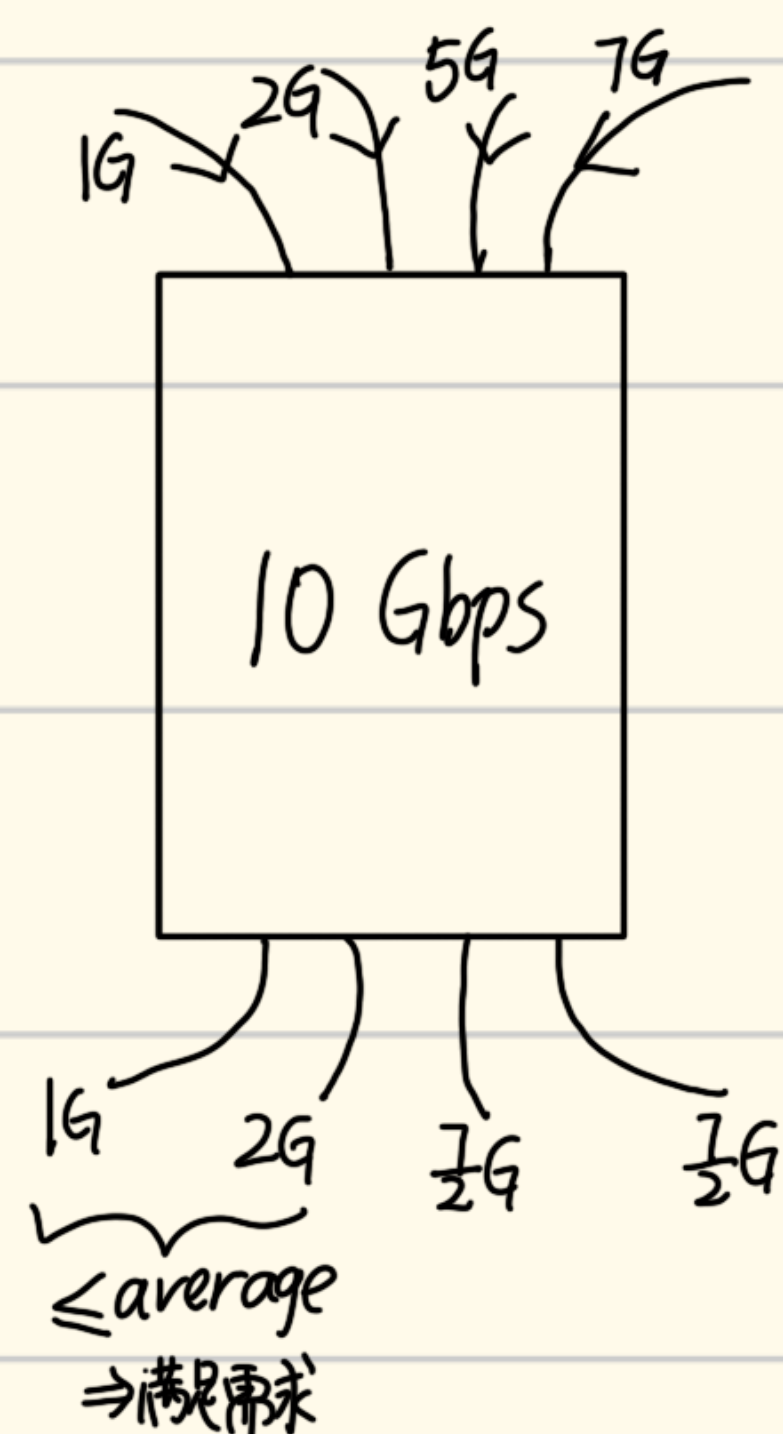


AIMD { slow start
congestion avoidance
fast recovery



TCP fairness:

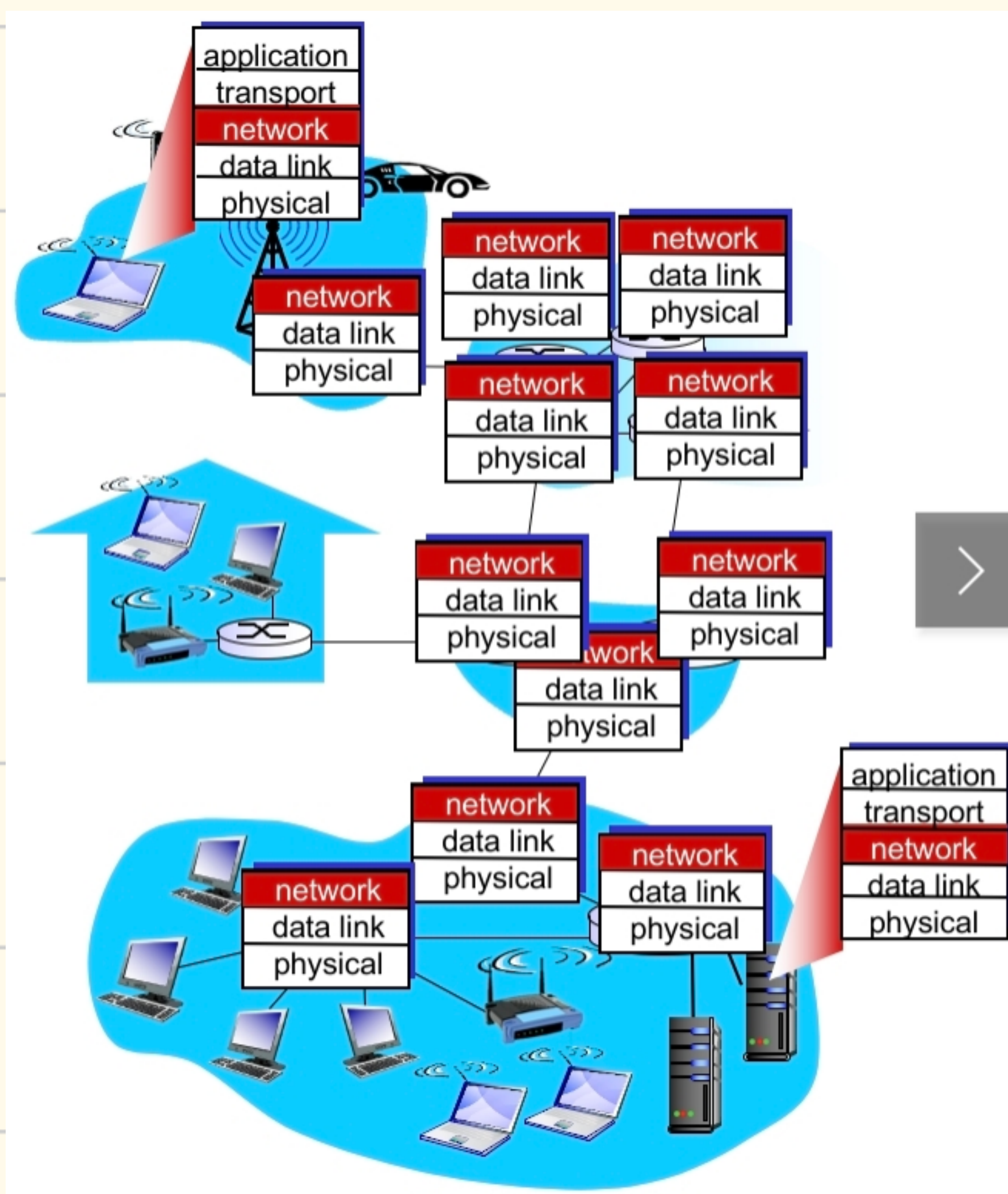
TCP max-min fairness (劫富济贫)



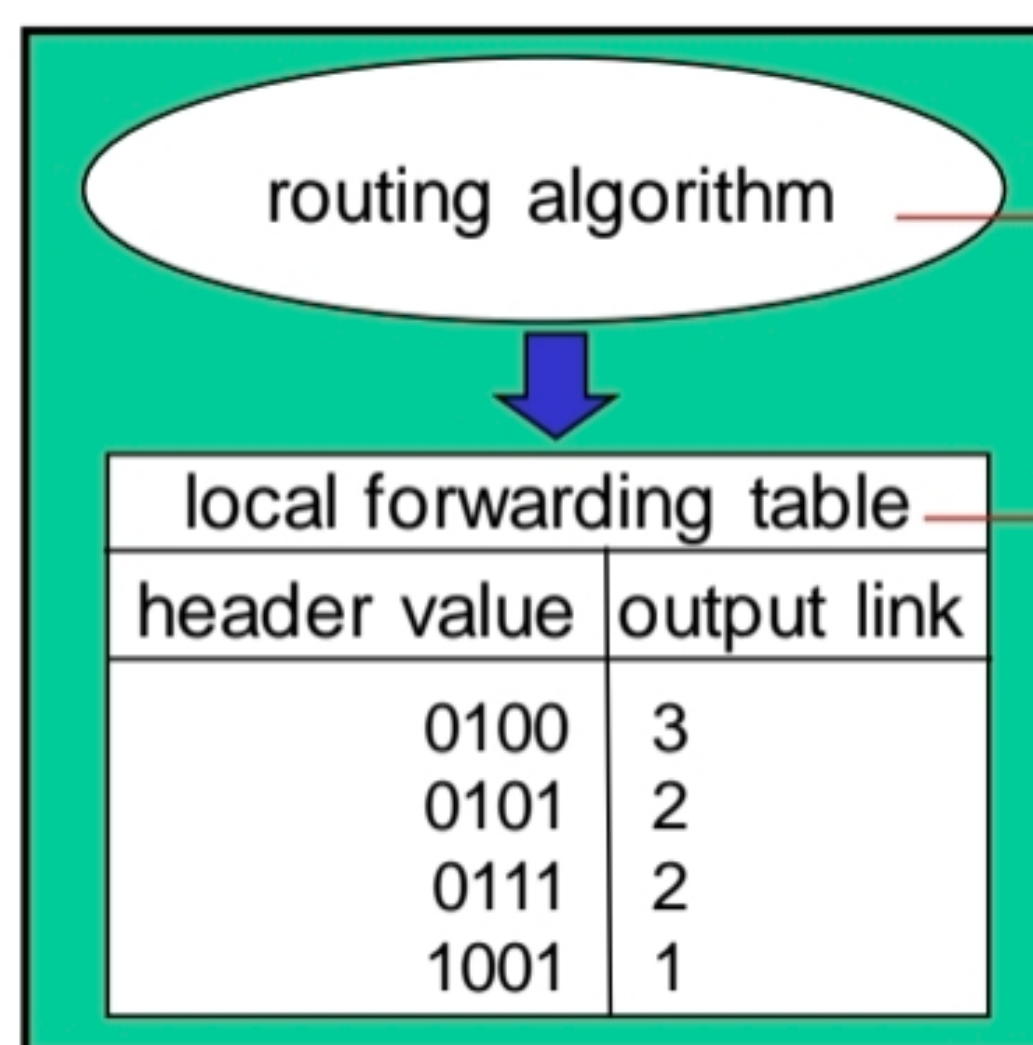
Chapter 4

Network Layer

4.1 Overview of Network Layer



network layer protocols
in every host, router

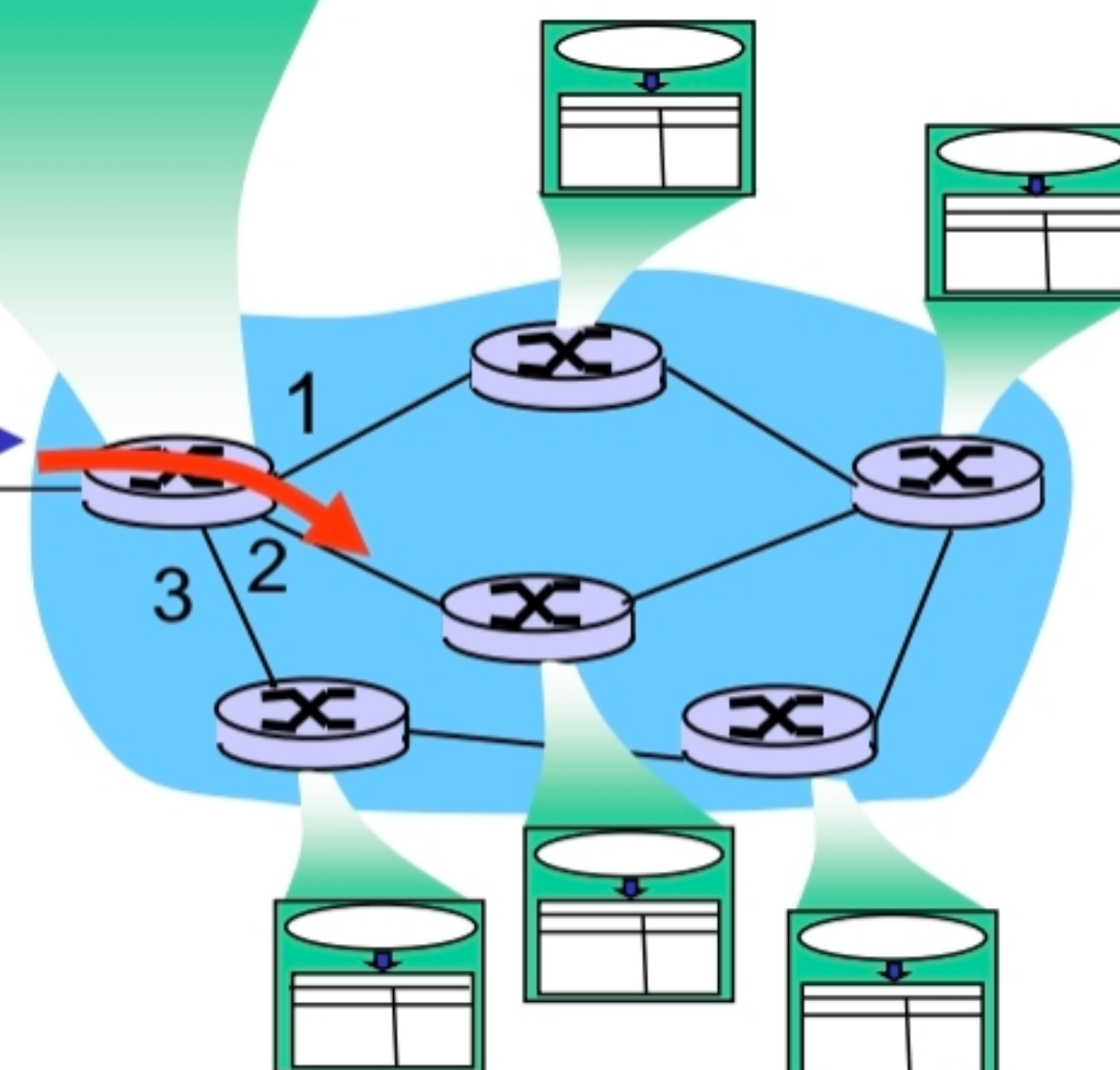


- ① routing algorithm determines end-end-path through network
- ② forwarding table determines local forwarding at this router

网络层的主要功能

value in arriving
packet's header

0111

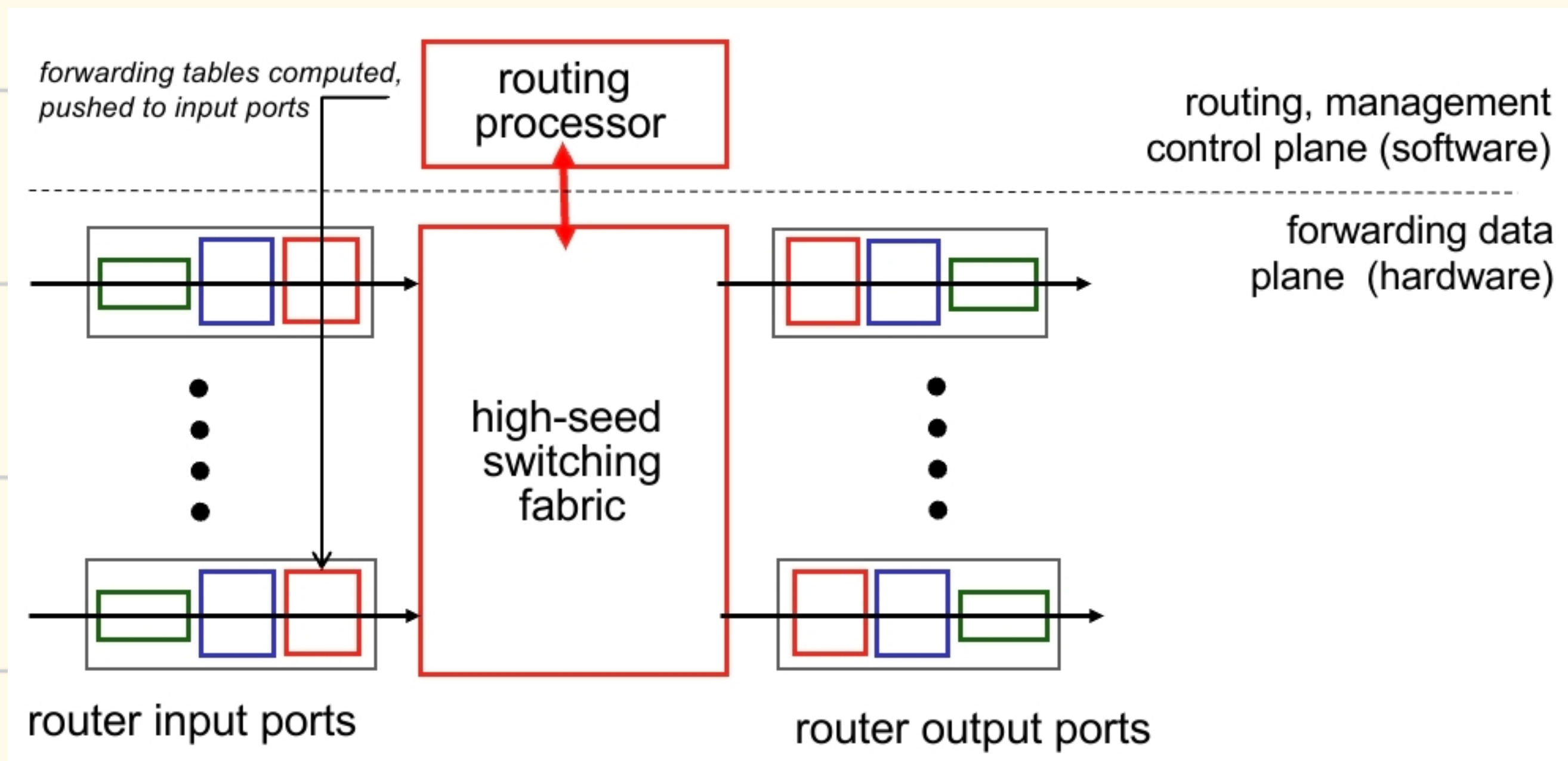


4.2 Virtual Circuit and Datagram networks

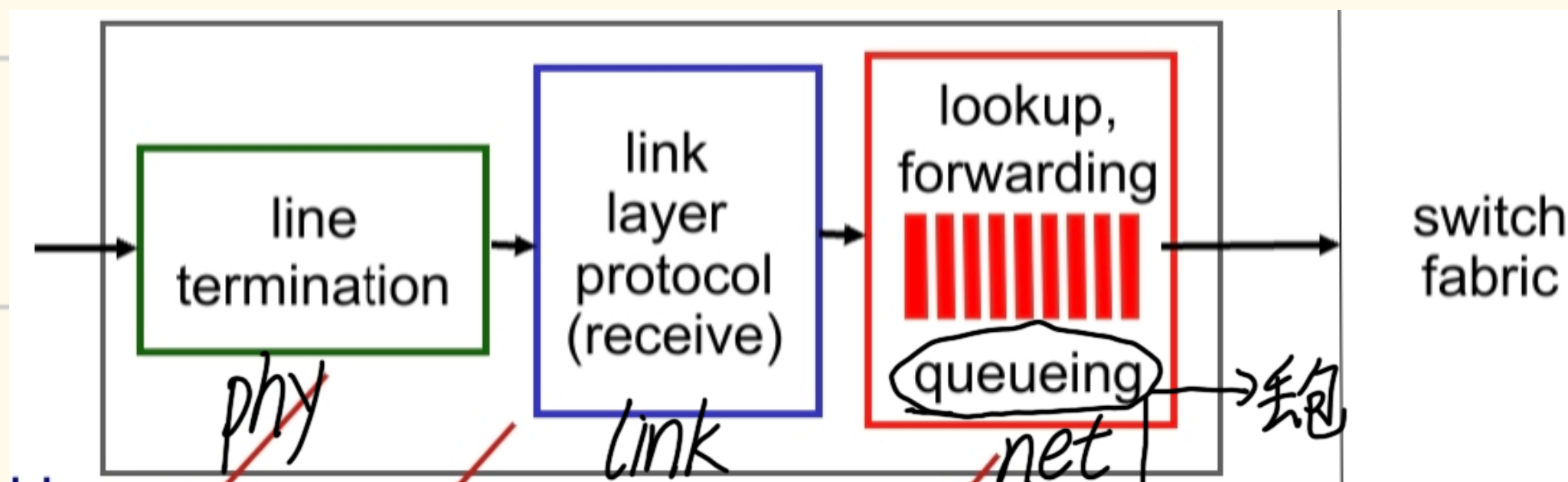
组网技术 { circuit switch : dedicated, static
datagram (packet) : dynamic → virtual circuit switch
(对 bandwidth & delay 要求, static)

4.3 What's inside a router

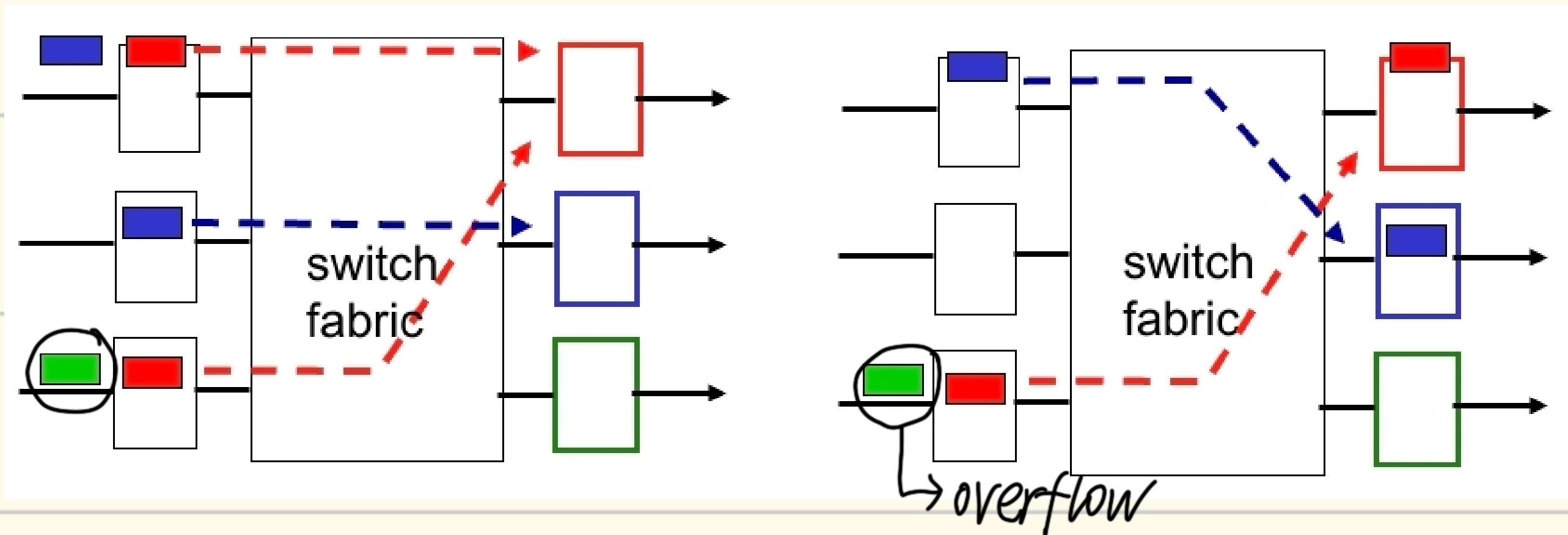
Router architecture overview



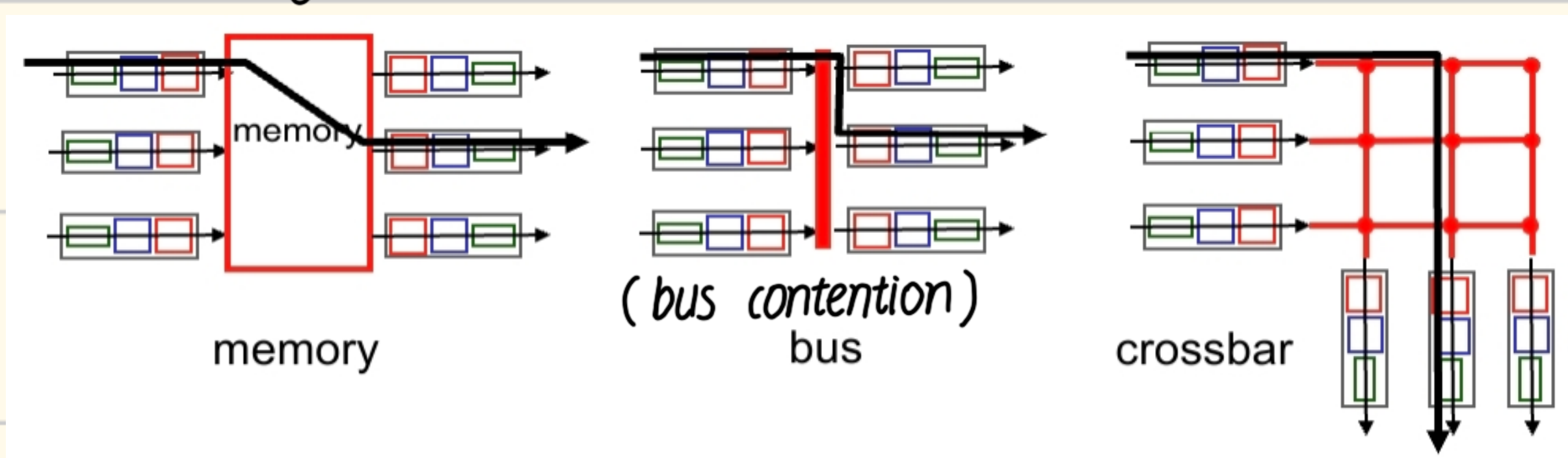
Input port



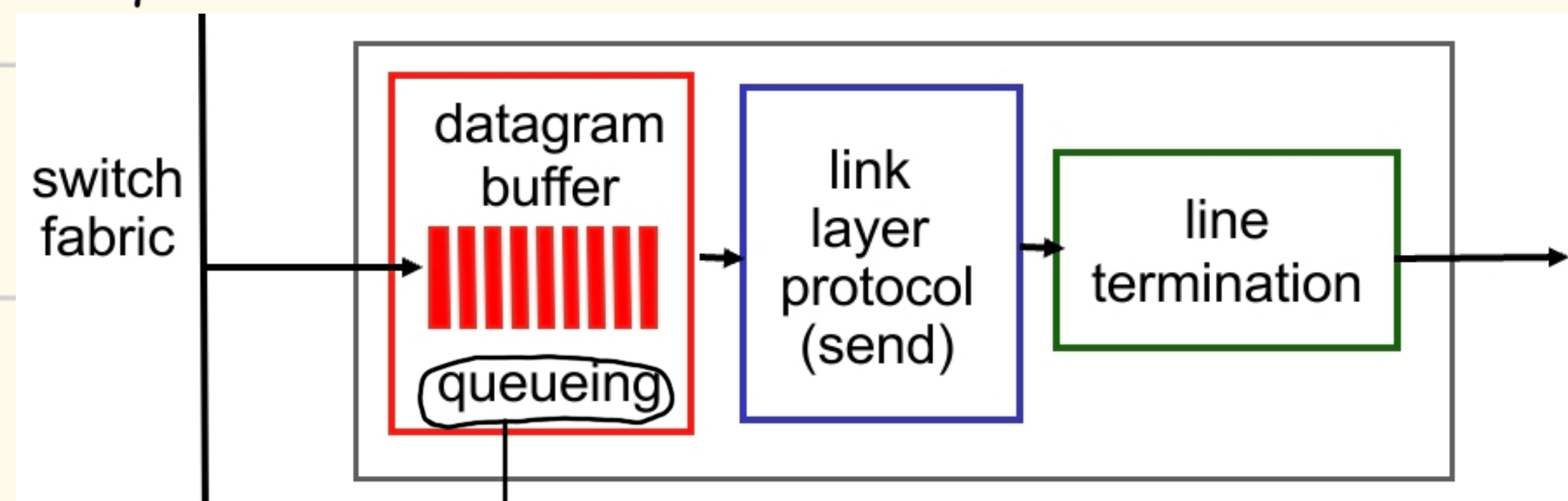
HOL: Head of Line Blocking



Switching fabrics

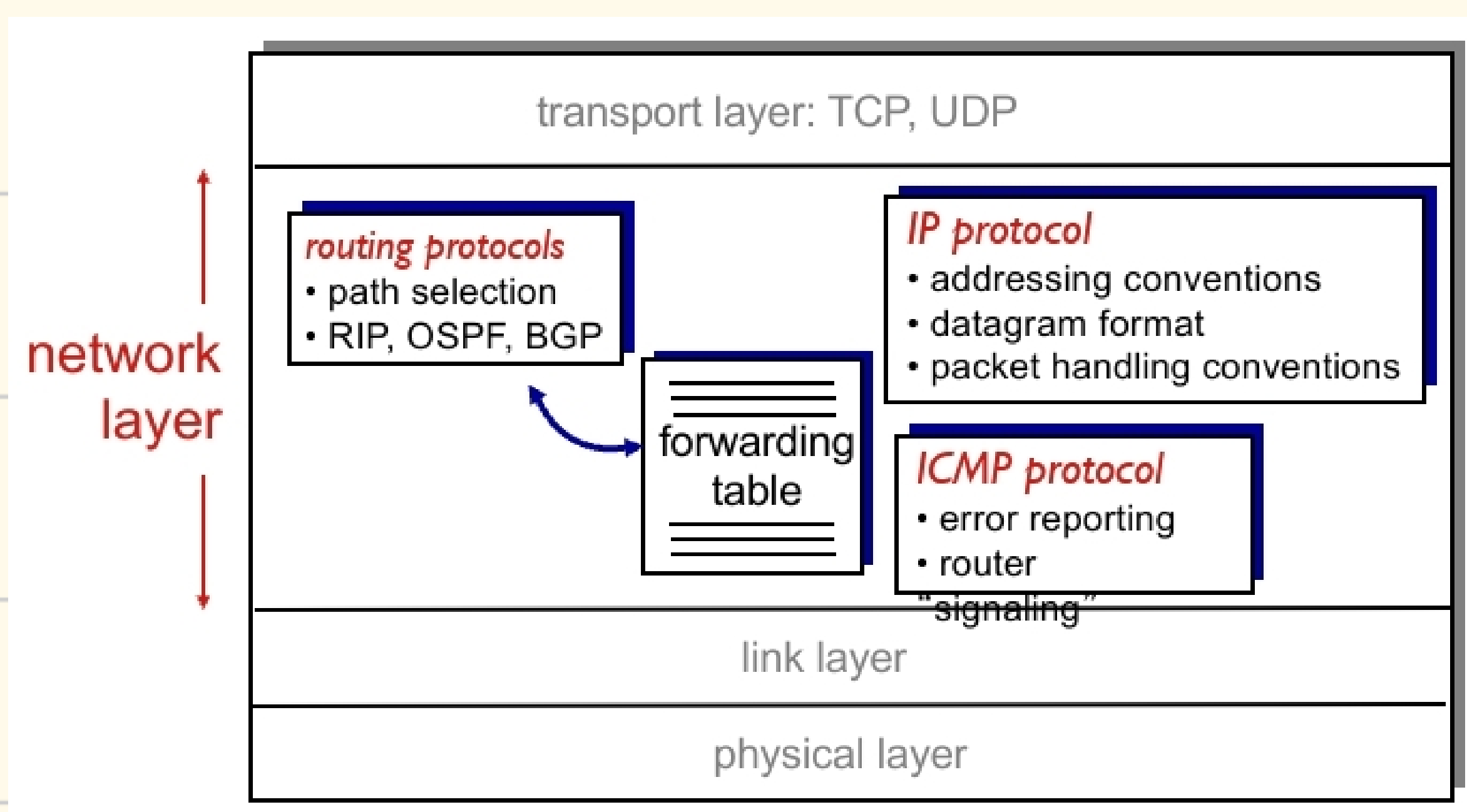


Output port



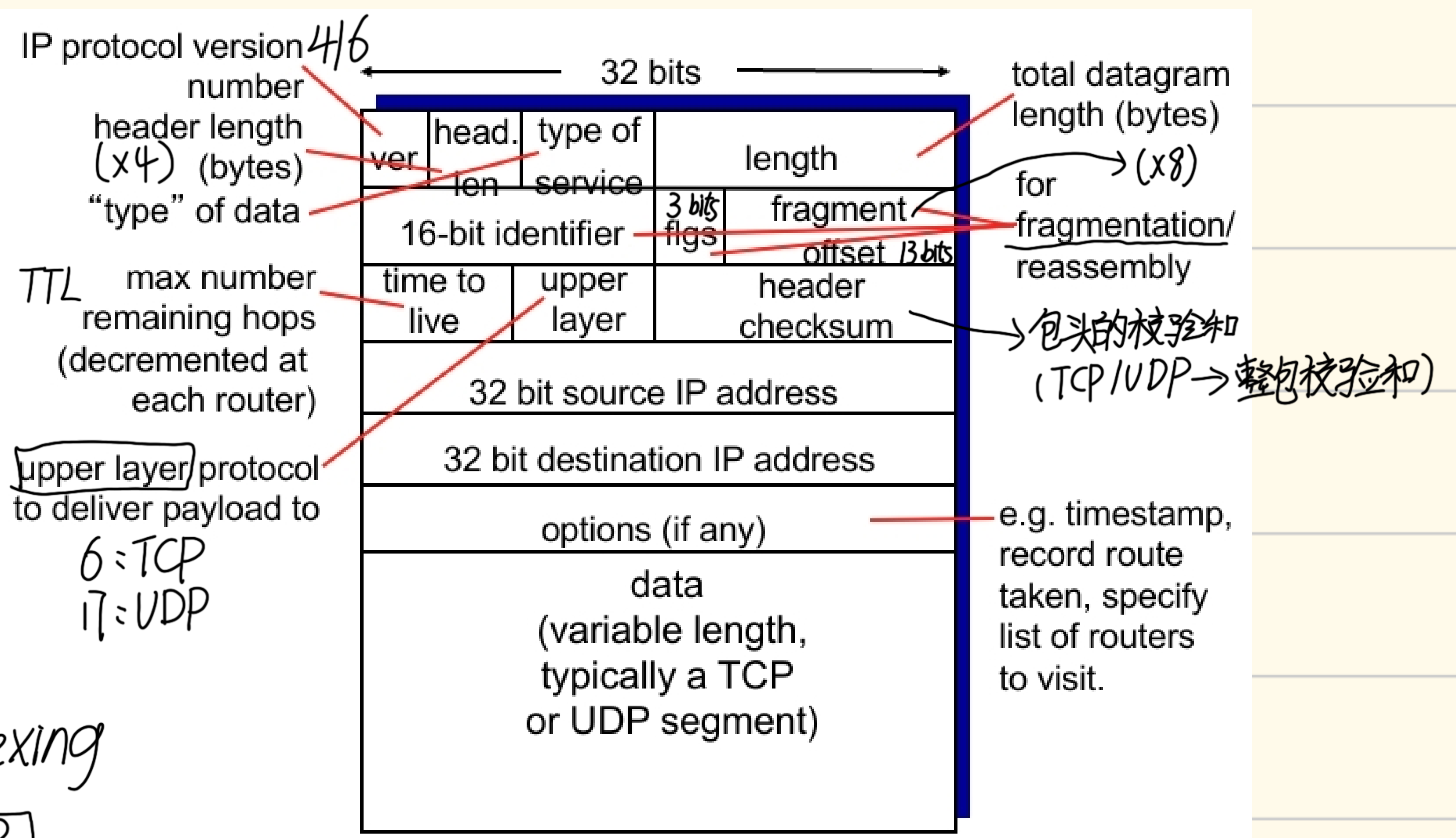
scheduling discipline: FIFO, WFS

4.4 IP: Internet Protocol



无连接
不可靠
互联网的窄腰
互通性

IP datagram format

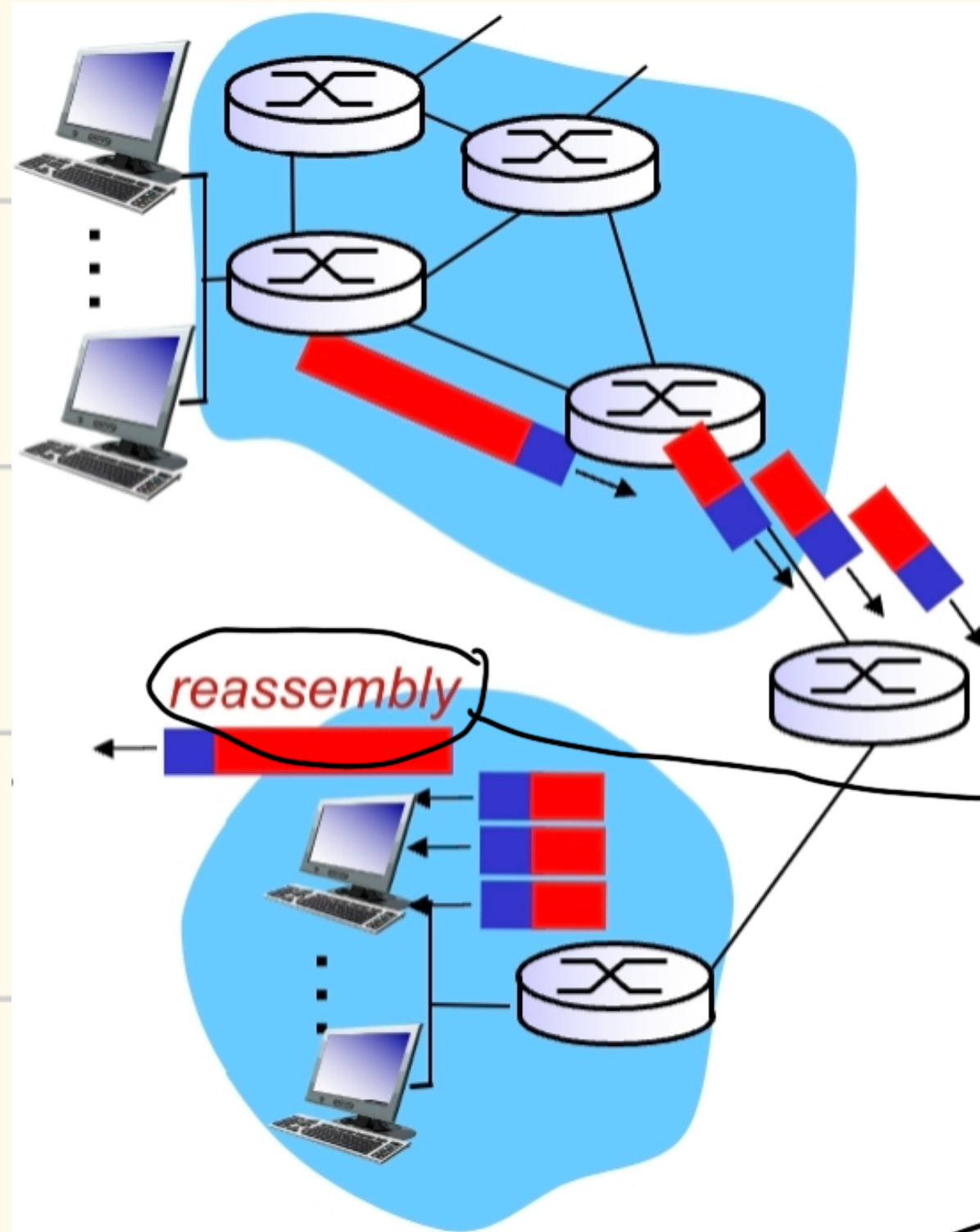


demultiplexing



ip: upper layer
udp: dest port
tcp: 4 tuple

IP fragmentation, reassembly



network links MTU

→ only at final destination

$$20 + 3980 = 1480 + 1480 + 1020$$

length	ID	fragflag	offset
=4000	=x	=0	=0

只切包体
均为8的倍数

one large datagram becomes several smaller datagrams

length	ID	fragflag	offset
=1500	=x	=1	=0
length	ID	fragflag	offset
=1500	=x	=1	=185
length	ID	fragflag	offset
=1040	=x	=0	=370

包头+包体 → 属于同一包 → 顺序

length < MTU

最后包

(除最后一个包外 包体字节数必须为8的倍数)

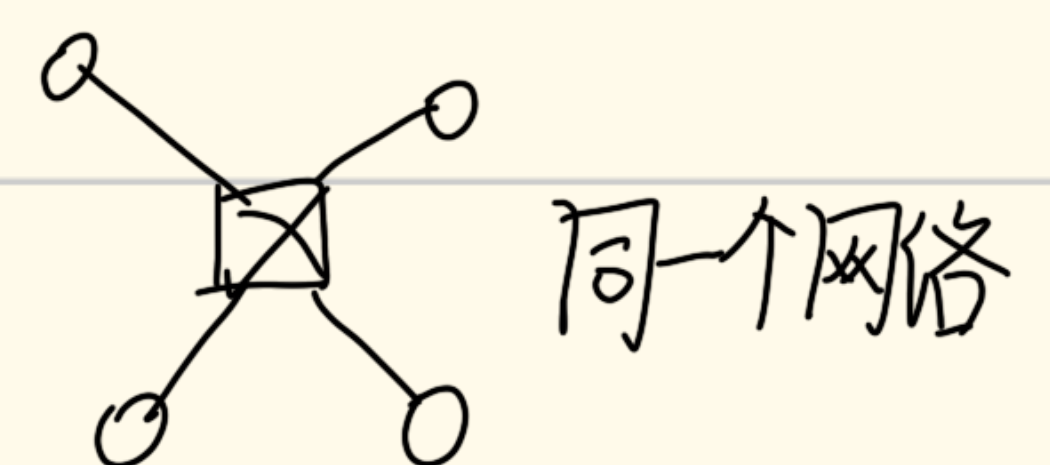
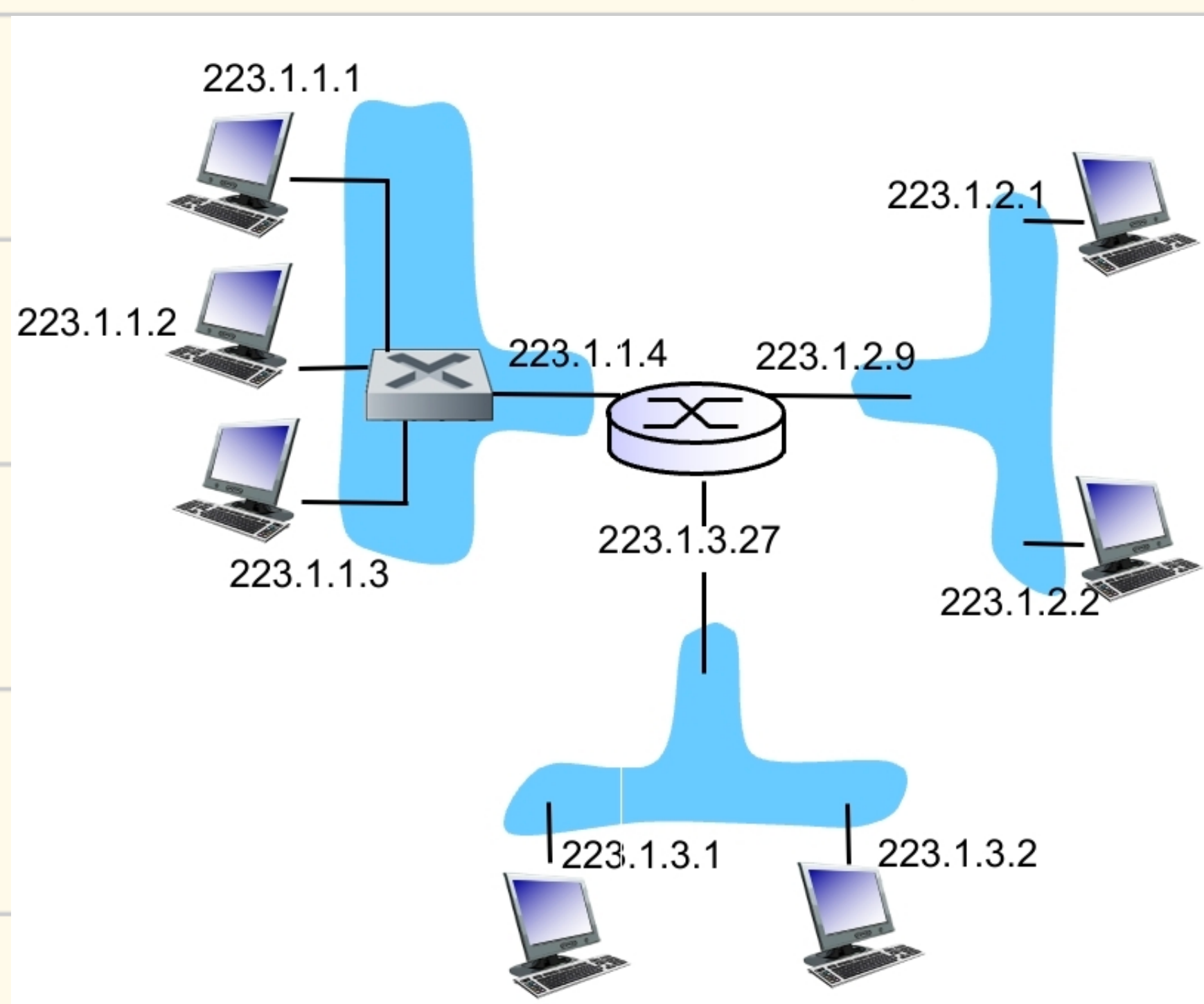
$1480/8$

IP addressing

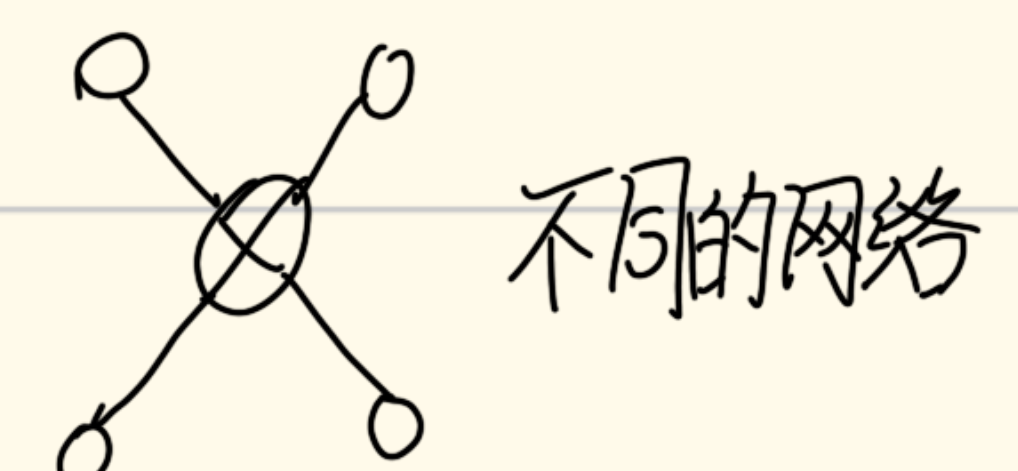
32 bits : 223.1.1.1 = 11011111 00000001 00000001 00000001

→ NIC
(network interface card)

interface = router 多个 interfaces
host 1~2个



同一个网络



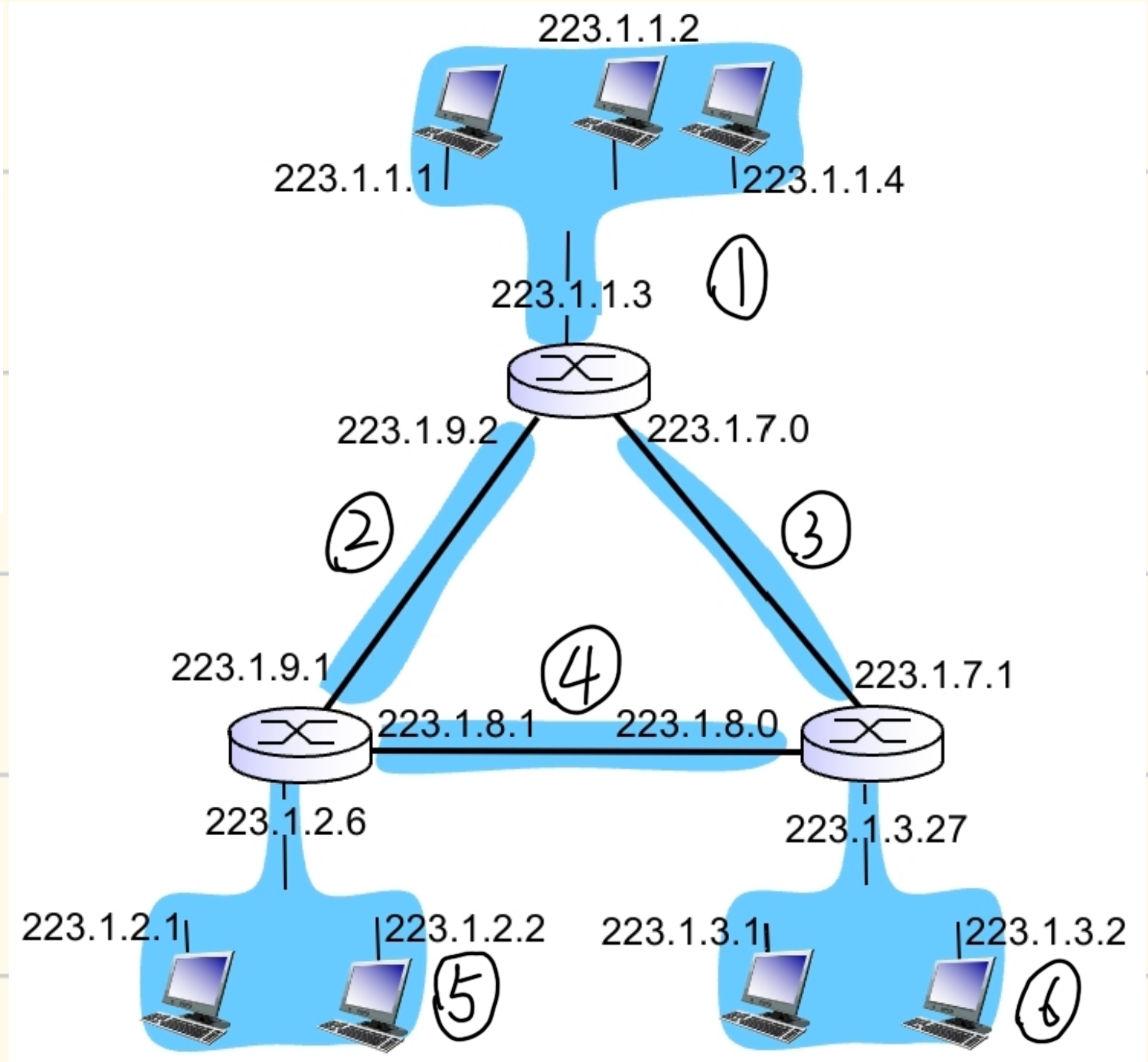
不同的网络

Subnets

what's a subnet?

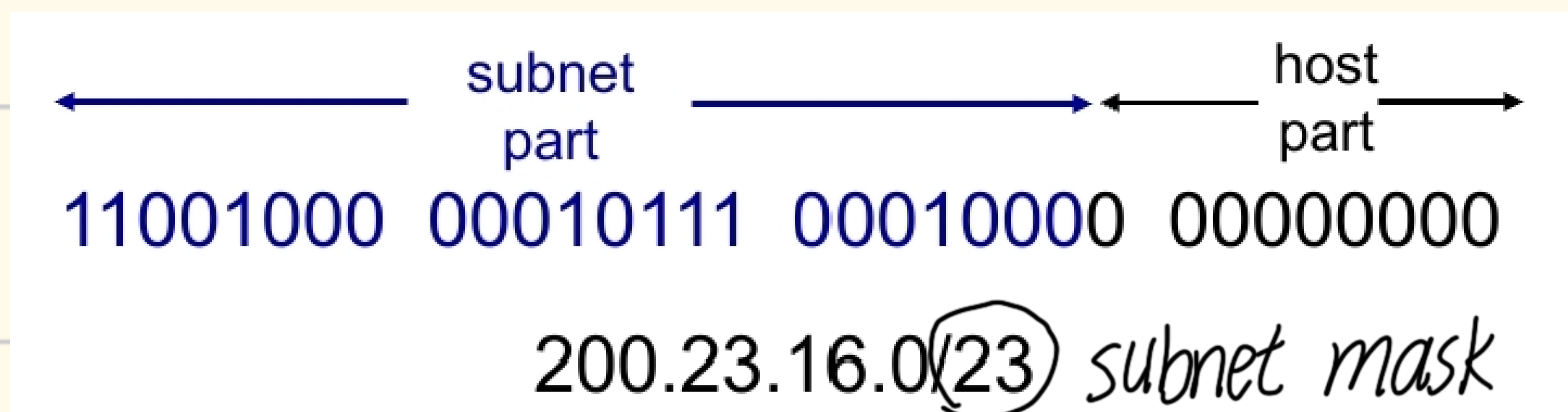
- device interfaces with same subnet part of IP address
- can physically reach each other **without intervening router**

⇒ 减少 broadcast domain, 避免广播风暴、出于安全考虑



CIDR

classless Inter Domain Routing (无类域间路由)



注意: 192.168.0.1 和 192.168.0.128 未必属于同一子网 (host id 划分位不确定)

如 192.168.0.00000000

25 bits

7 bits ⇒ 192.168.0.0 ~ 192.168.0.127
共128个

prefix	suffix
--------	--------

x位 → $2^x - 2$ 个 nic ip

(00...0 ⇒ net ip

11...1 ⇒ subnet broadcast addr.)

255.255.255.255 受限广播包

192.168.0.255 定向广播包

mask

192.168.0.0/24

255.255.255.0

192.168.0.0/25

255.255.255.10000000
7位
128

作用: dest IP AND mask
(脏link层) result =? net ip ⇒ 决定交换机向内网发 包还是发向路由器

eg. 将 128.119.40.128/26 分为等大小的四个子网

128.119.40.10 00000000 $\Rightarrow$ 128.119.40.64
01 128.119.40.80
10 128.119.40.96
11 128.119.40.112

eg. 将 223.1.17.0/24 分为至少 90, 60, 12 个 interface 子网

223.1.17. 0000 0000 } $2^7 = 128 > 90$
 ⋮
 0111 1111 }
 1000 0000 } $2^6 = 64 > 60$
 ⋮
 1011 1111 }
 1100 0000 } $2^4 = 16 > 12$
 ⋮
 1100 1111 }

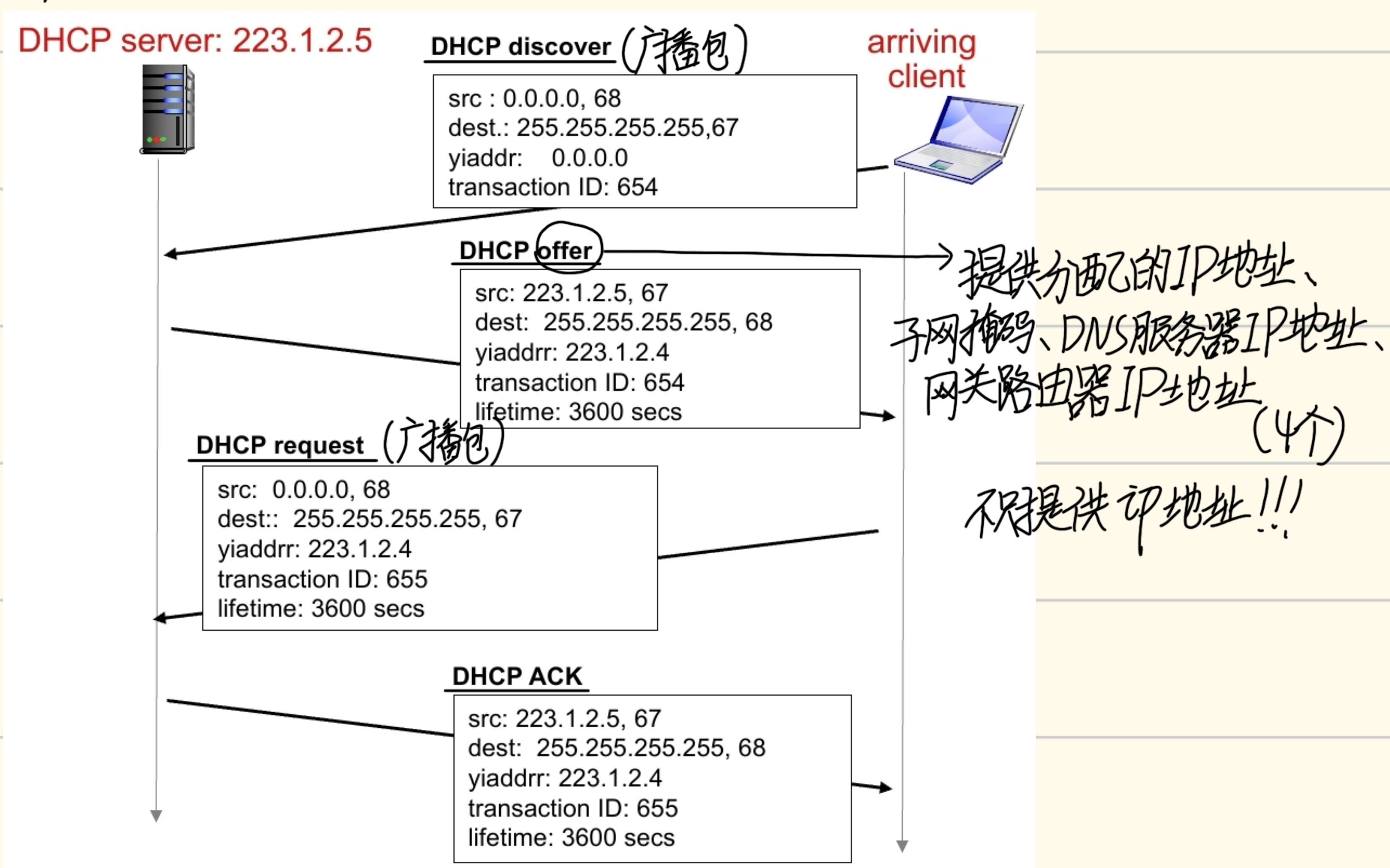
$\Rightarrow$ 223.1.17.0/25

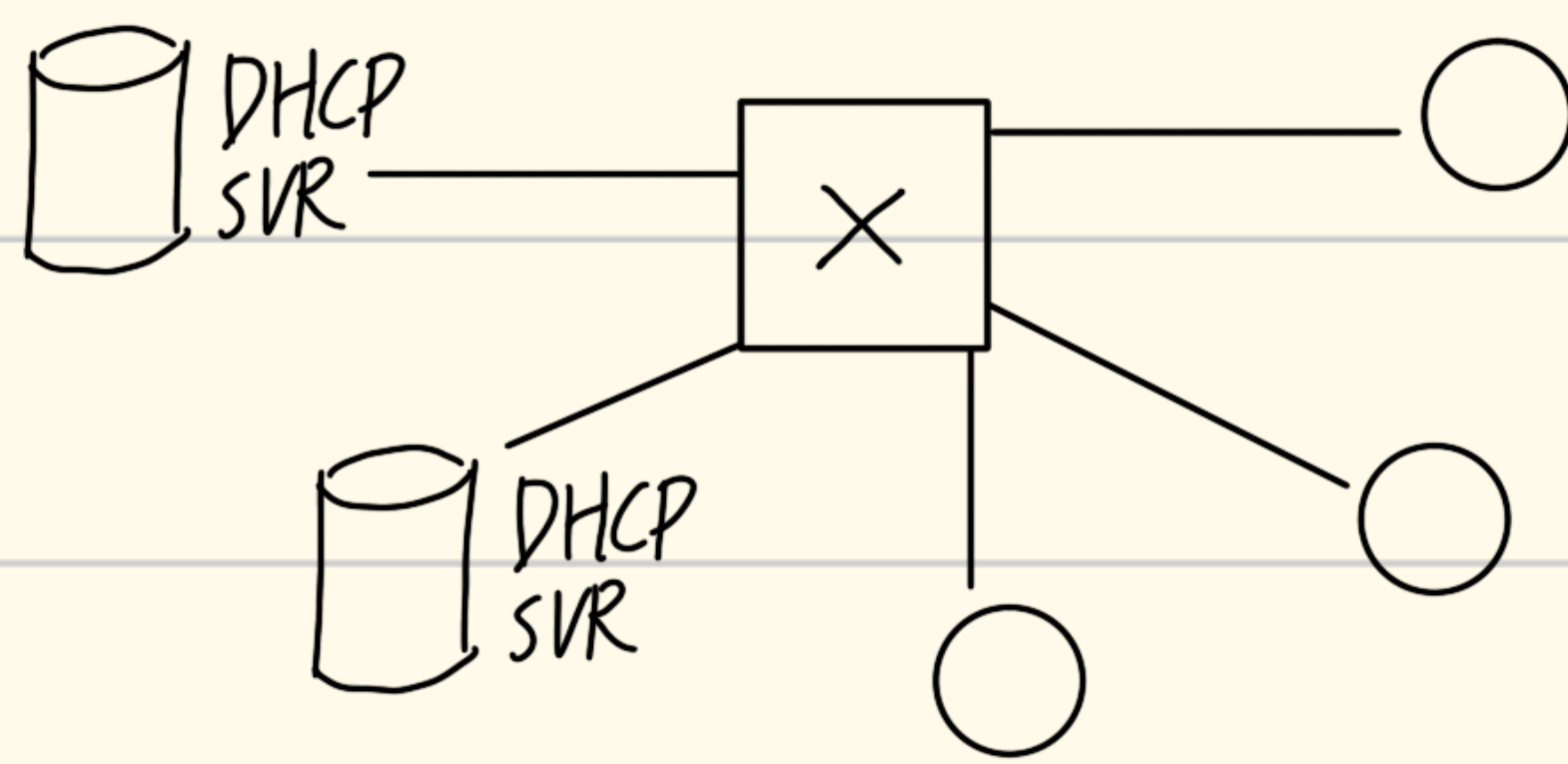
223.1.17.128/26

223.1.17.192/28

DHCP

Dynamic Host Configuration Protocol

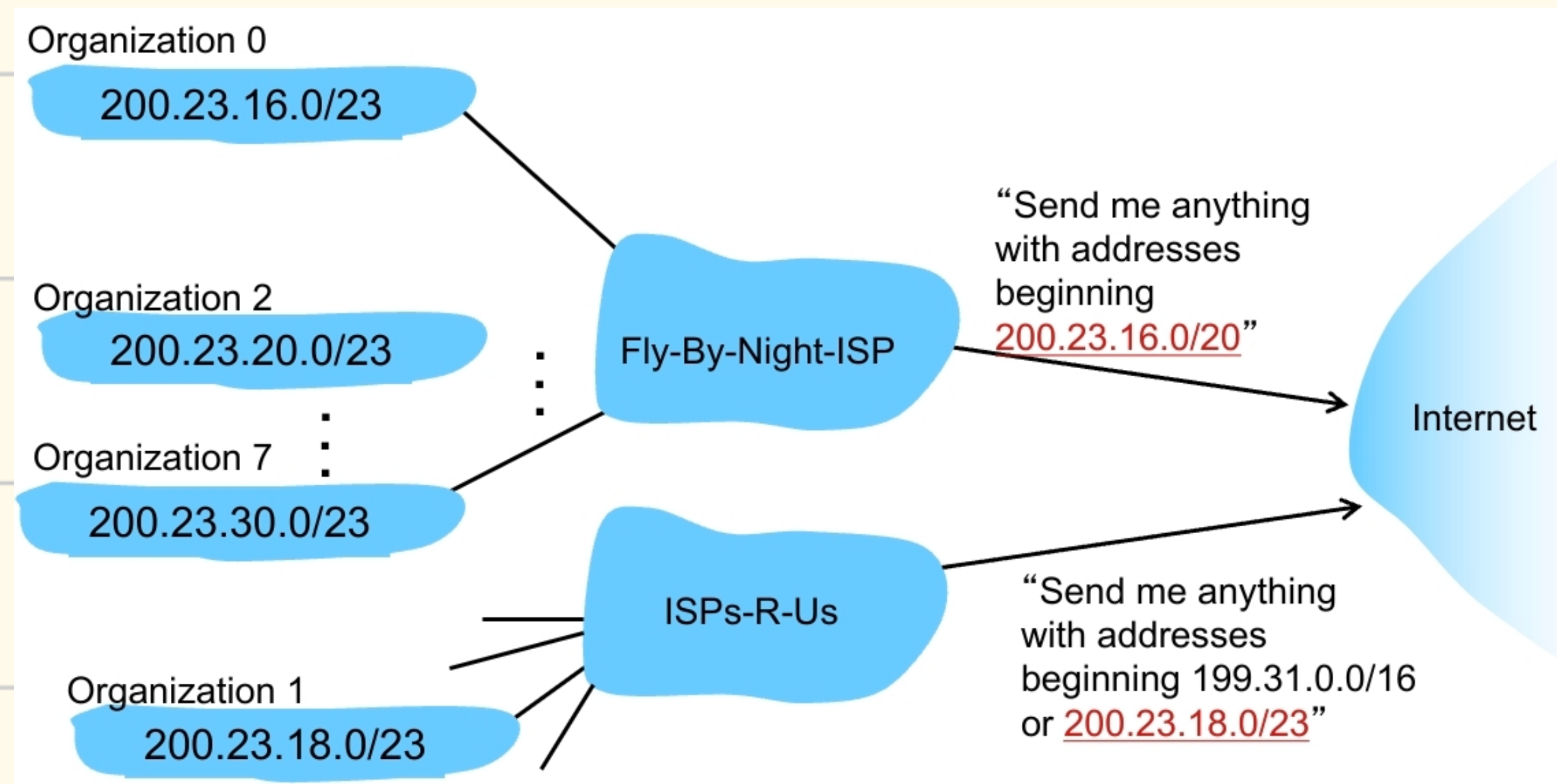




DHCP request + ACK:
应对多个DHCP服务器情况

DHCP SVR直接:应用层 ↔ 网络层, 封装在datagram的包体

Aggregation



NAT

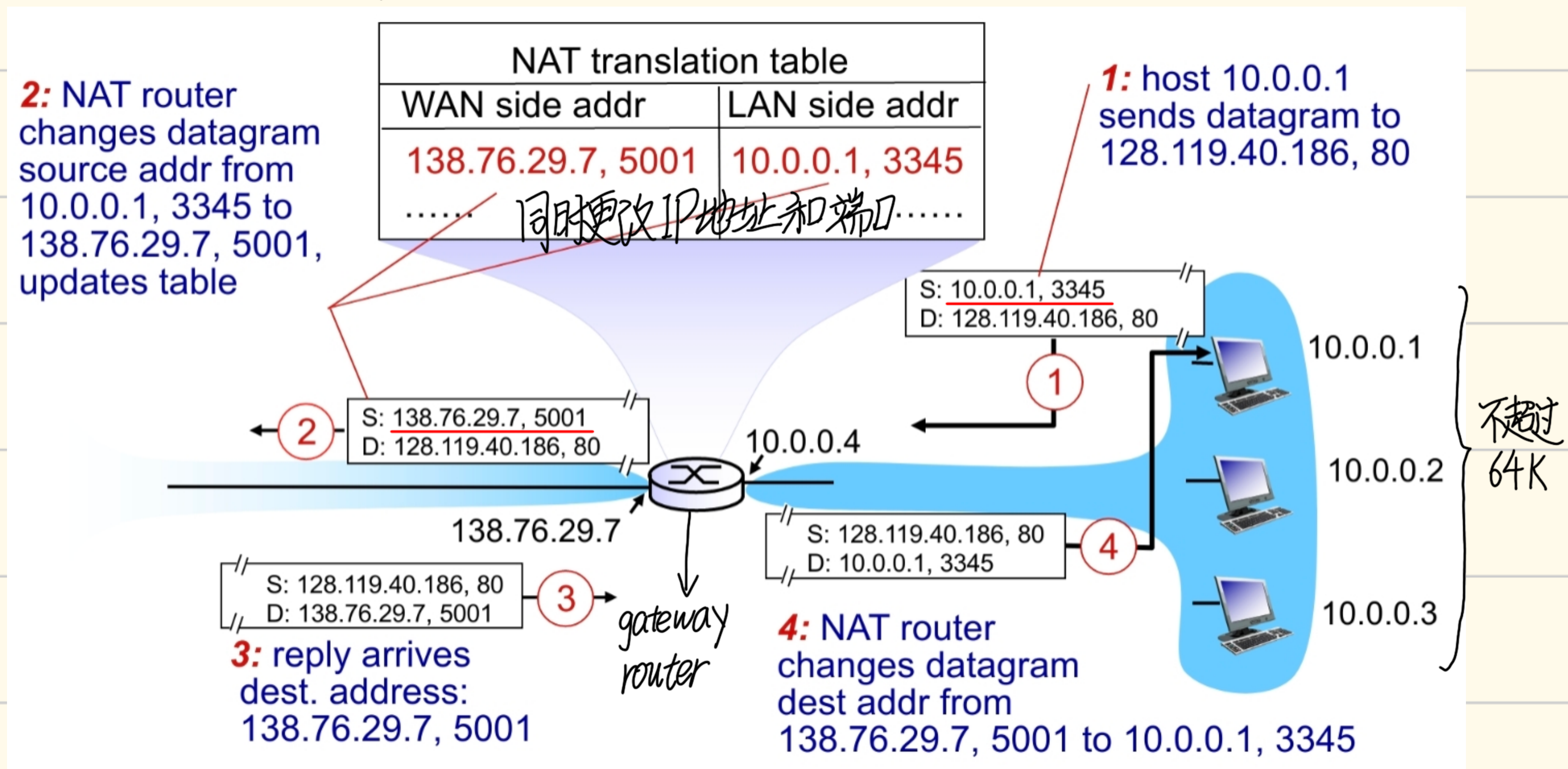
(network address translation)

内网地址: 10.

172.16 ~ 172.31

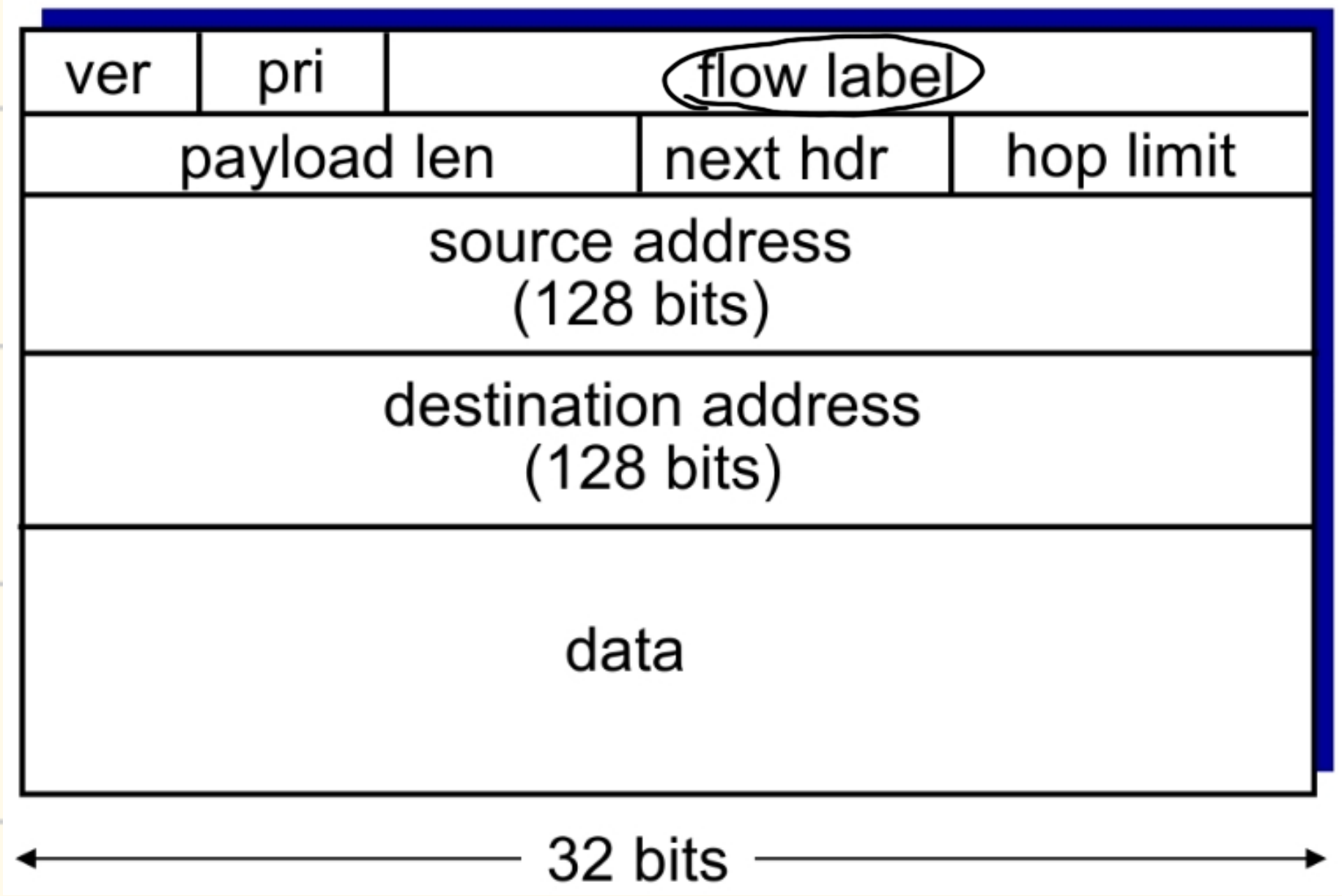
192.168

内网ip地址的数据包不会被路由器转发



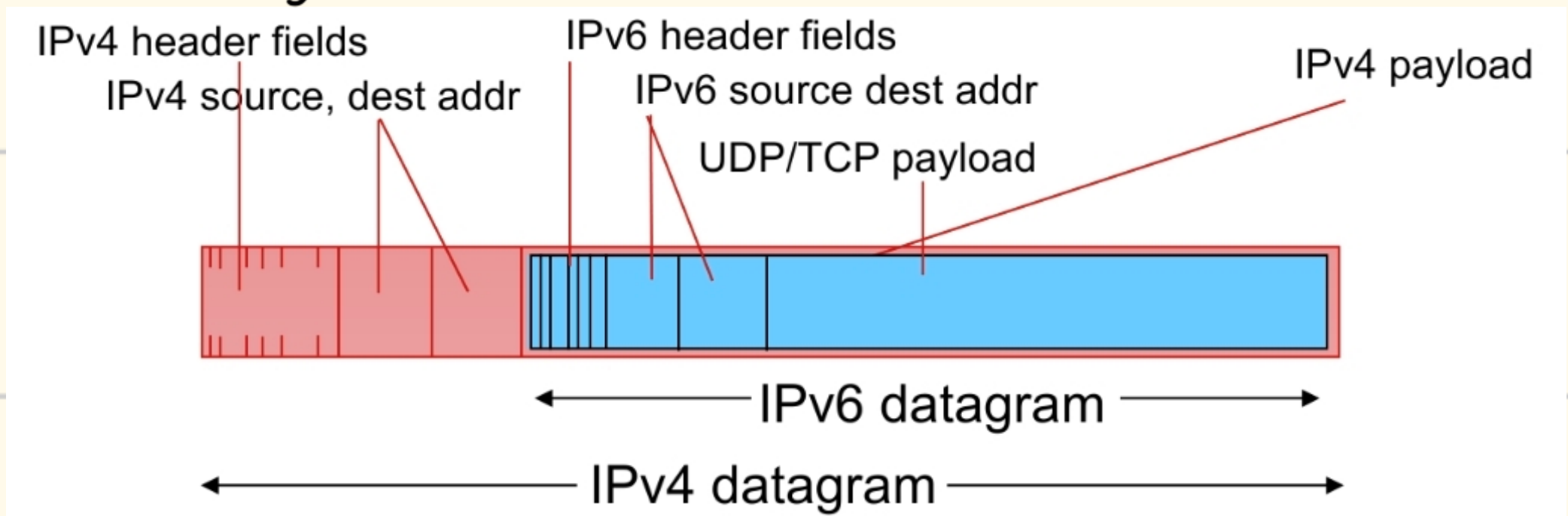
IPv6

包头: 定长 40 bytes
不允许切包、没有 checksum 检测



IPv4 与 IPv6 共存技术:

Tunneling



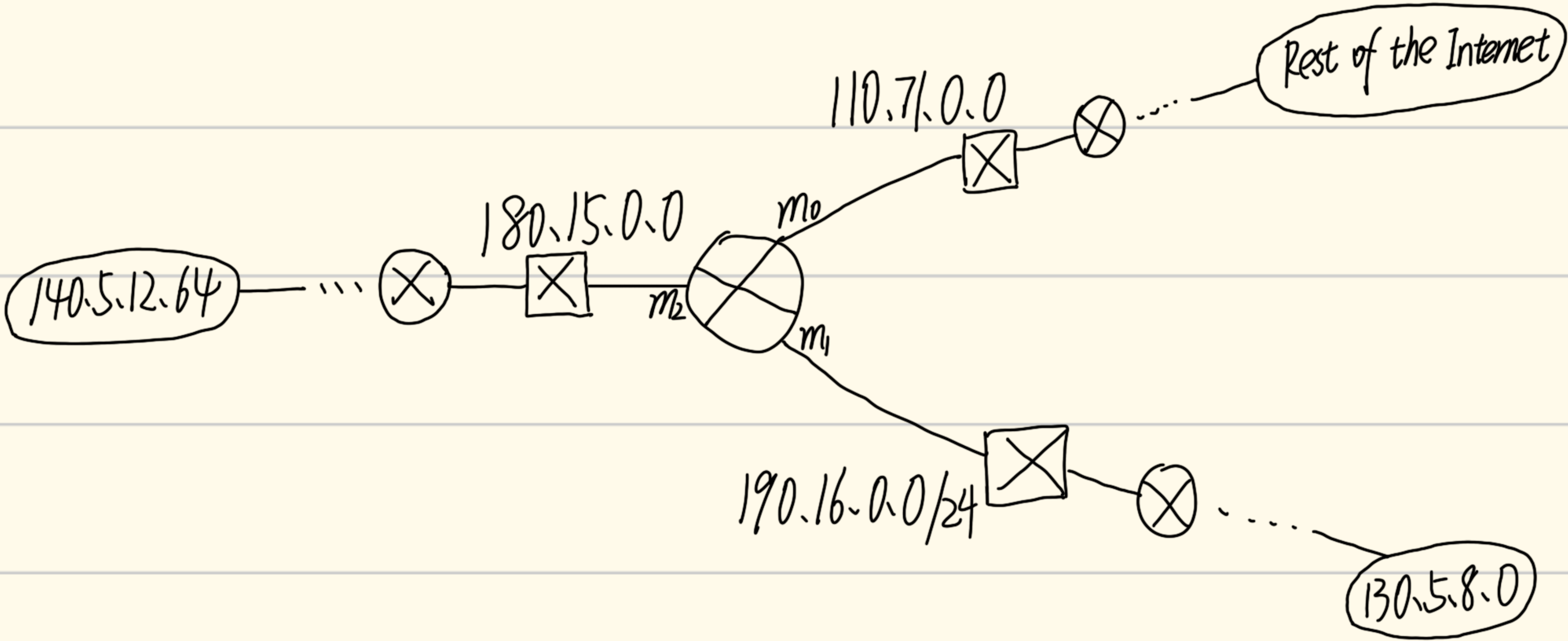
4.5 Routing Algorithms

eg.

Consider the routing table of router R1 shown below.

Subnet Mask	Destination Network Address	Next Hop Address	interface
/26	140.5.12.64	180.15.2.5	m2
/24	130.5.8.0	190.16.6.2	m1
/16	110.71.0.0	----	m0
/16	180.15.0.0	----	m2
/16	190.16.0.0	----	m1
Default	Default	110.71.4.5	m0

Draw the connection topology of the networks and routers mentioned above. IP addresses and interfaces must be labeled.



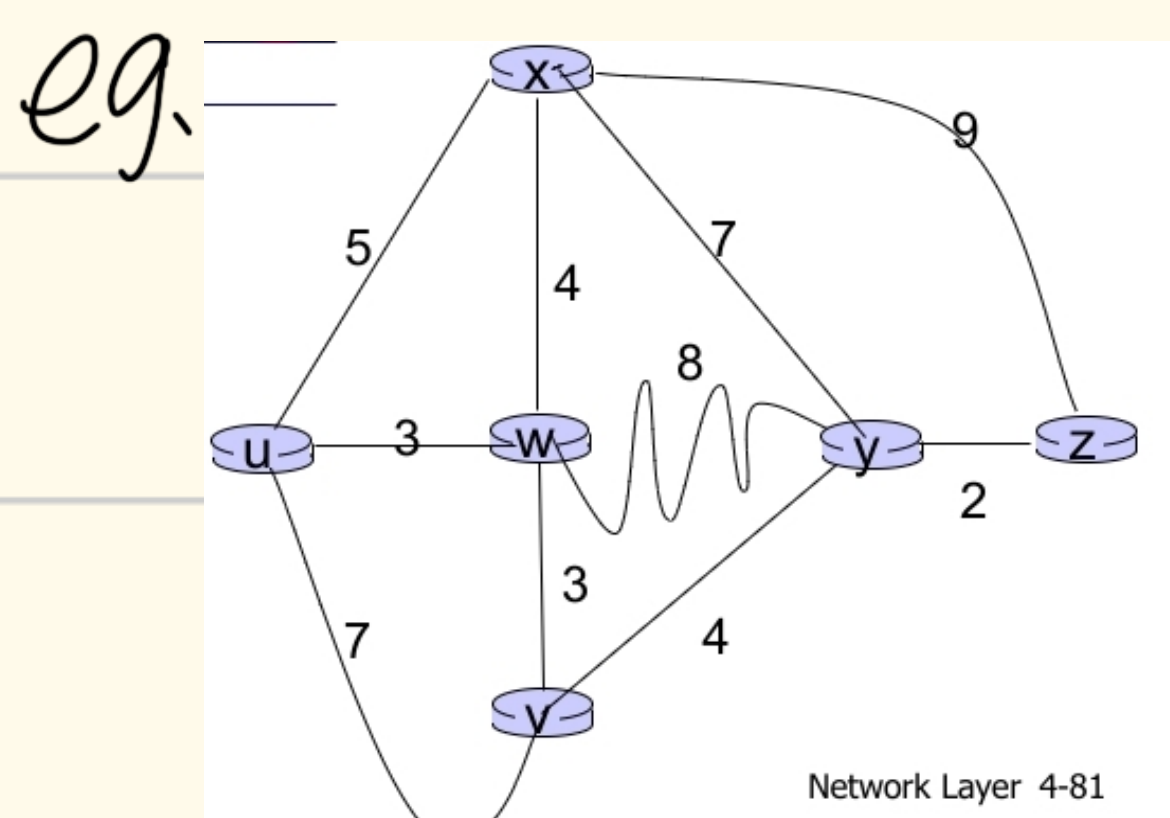
Routing Algorithm Classification

$\begin{cases} L-S & \text{: global、需事先知道网络拓扑图、dynamic (广播)} \\ D-V & \text{: decentralized、dynamic} \end{cases}$

Dijkstra's Algorithm ☆ (Link-State)

```

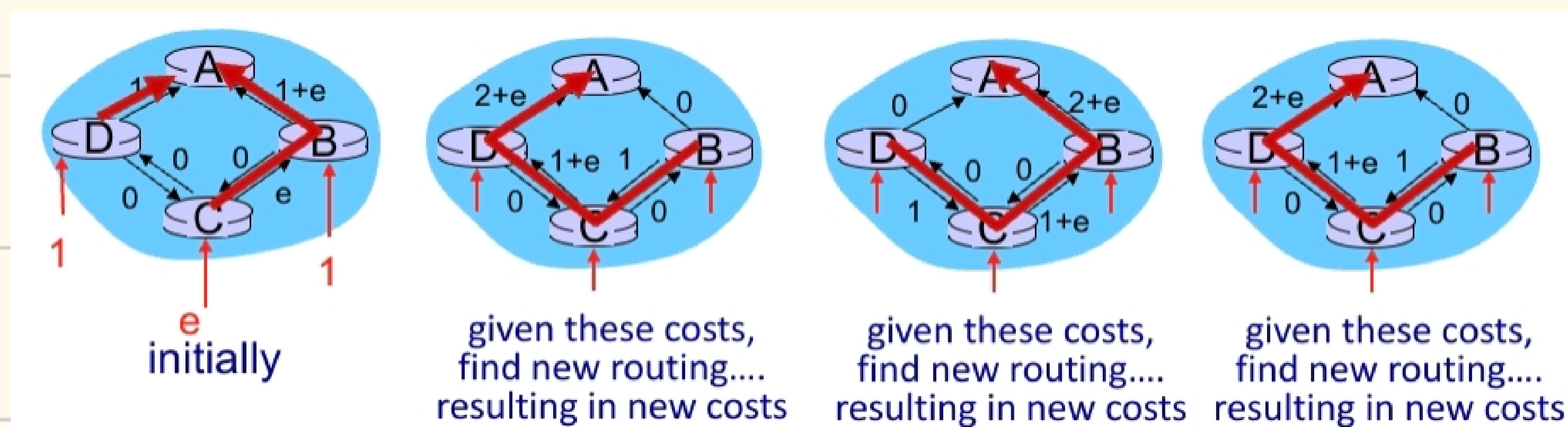
1 Initialization:
2  $N' = \{u\}$ 
3 for all nodes  $v$ 
4   if  $v$  adjacent to  $u$ 
5     then  $D(v) = c(u,v)$ 
6   else  $D(v) = \infty$ 
7
8 Loop
9   find  $w$  not in  $N'$  such that  $D(w)$  is a minimum
10  add  $w$  to  $N'$ 
11  update  $D(v)$  for all  $v$  adjacent to  $w$  and not in  $N'$ :
12     $D(v) = \min(D(v), D(w) + c(w,v))$ 
13    /* new cost to  $v$  is either old cost to  $v$  or known
14       shortest path cost to  $w$  plus cost from  $w$  to  $v$  */
15 until all nodes in  $N'$ 
    
```



Step	N'	$D(v)$ $p(v)$	$D(w)$ $p(w)$	$D(x)$ $p(x)$	$D(y)$ $p(y)$	$D(z)$ $p(z)$
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwX	6,w			11,w	14,x
3	uwXv				10,v	14,x
4	uwXvy					12,y
5	uwXvyz					

$u \rightarrow w \rightarrow v \rightarrow y \rightarrow z$
 $u \rightarrow x$

缺点: oscillations、广播链路状态信息



Distance Vector Algorithm

neighbor: 仅在邻居间交换距离矢量信息

Bellman-Ford equation

let
 $d_x(y) := \text{cost of least-cost path from } x \text{ to } y$
 then
 $d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$
 min taken over all neighbors v of x
 $c(x,v)$: cost to neighbor v
 $d_v(y)$: cost from neighbor v to destination y

特点: good news travels fast
 bad news travels slow

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} \\ = \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} \\ = \min\{2+1, 7+0\} = 3$$

node x table

	cost to		
	x	y	z
from x	0	2	7
from y	∞	∞	∞
from z	∞	∞	∞

	cost to		
	x	y	z
from x	0	2	3
from y	2	0	1
from z	7	1	0

	cost to		
	x	y	z
from x	0	2	3
from y	2	0	1
from z	3	1	0

node y table

	cost to		
	x	y	z
from x	∞	∞	∞
from y	2	0	1
from z	∞	∞	∞

	cost to		
	x	y	z
from x	0	2	7
from y	2	0	1
from z	7	1	0

	cost to		
	x	y	z
from x	0	2	3
from y	2	0	1
from z	3	1	0

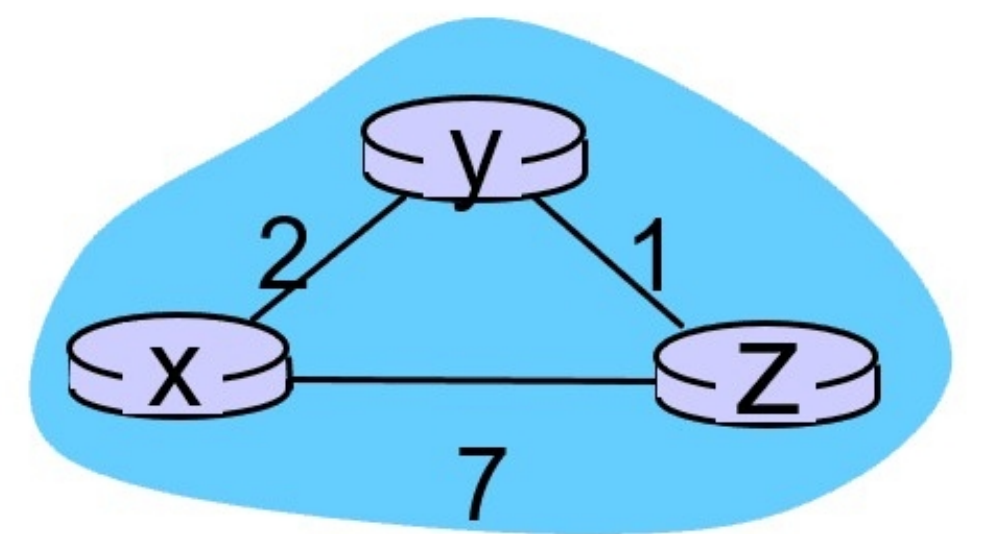
node z table

	cost to		
	x	y	z
from x	∞	∞	∞
from y	∞	∞	∞
from z	7	1	0

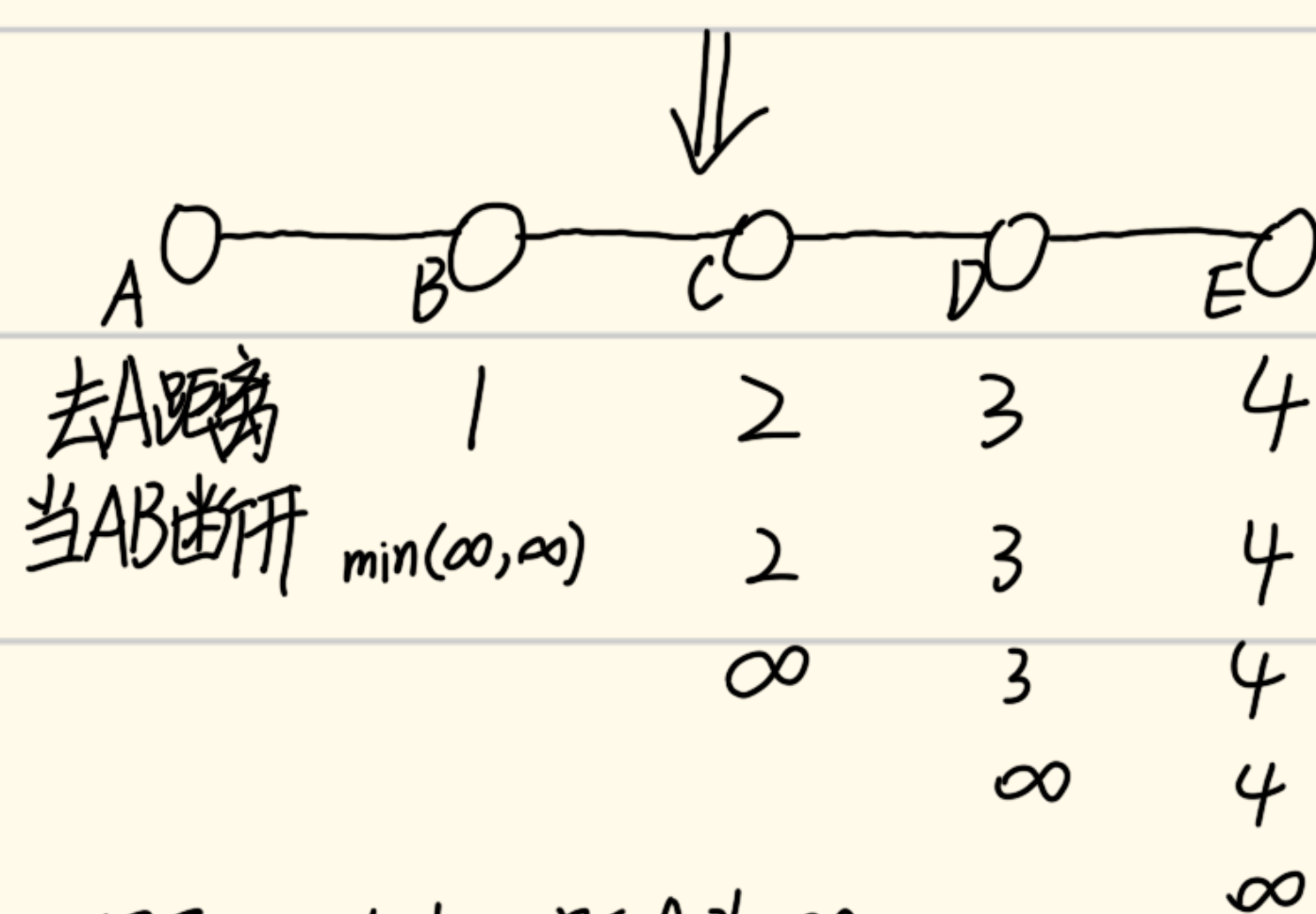
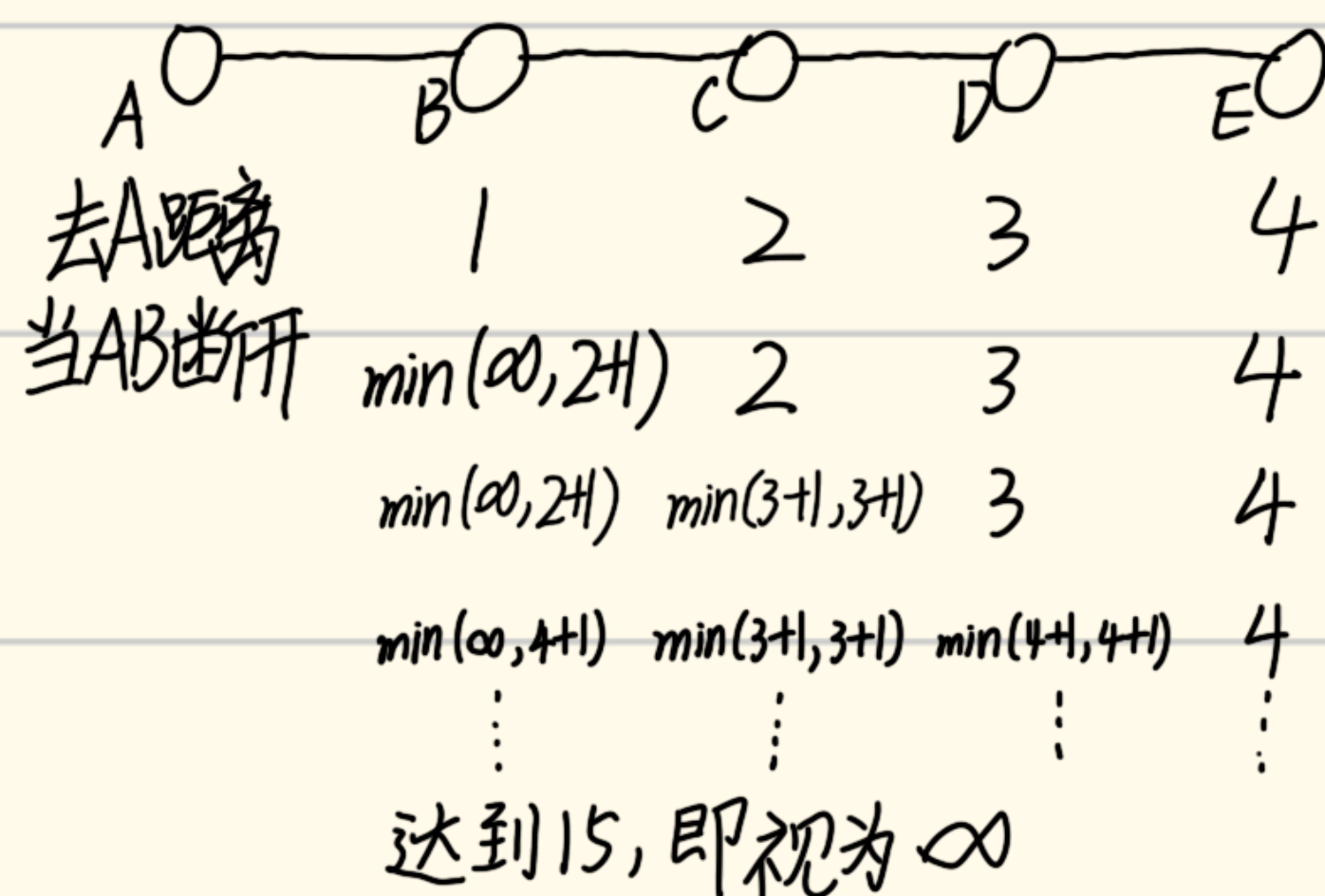
	cost to		
	x	y	z
from x	0	2	7
from y	2	0	1
from z	3	1	0

	cost to		
	x	y	z
from x	0	2	3
from y	2	0	1
from z	3	1	0

time

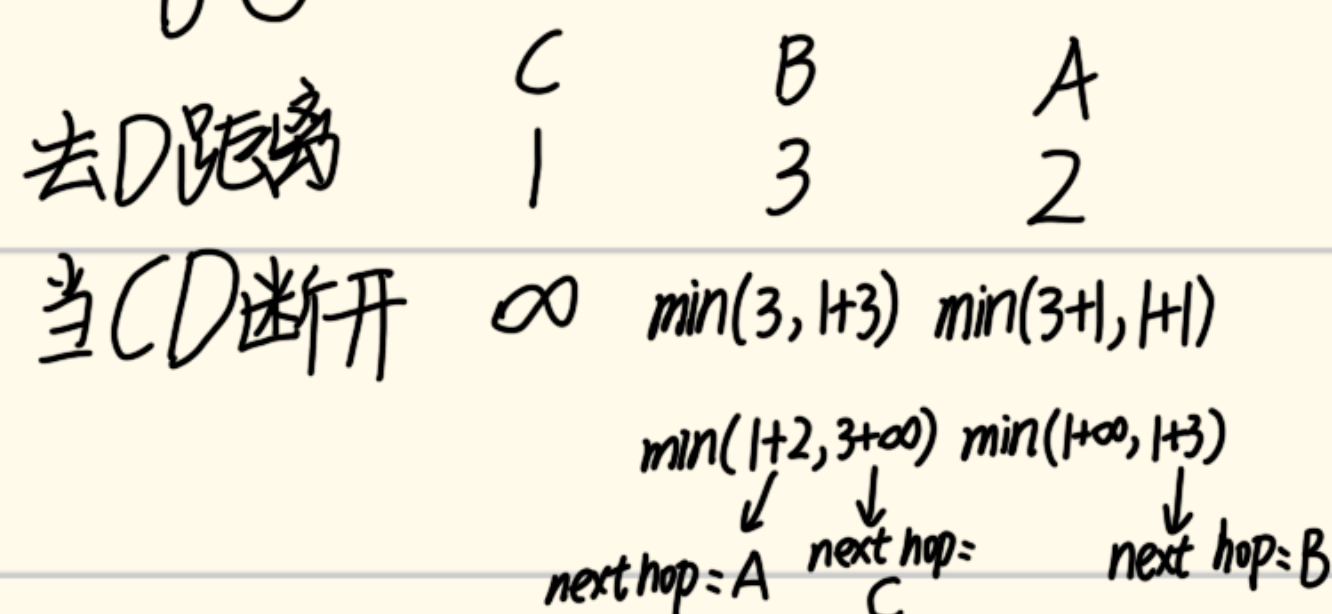
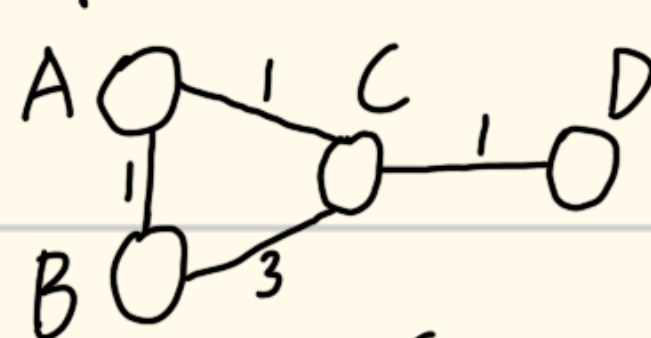


Poisoned Reverse (只对邻居有效)



只要 next hop 距A为 ∞ ,
则将距离置为 ∞

无效举例:



涉及3个及以上结点的环路
 $\Rightarrow$ 毒性反转失效

AS = autonomous system 自治系统

Intra-AS routing $\begin{cases} \text{OSPF (link state)} \\ \text{RIP (distance vector)} \end{cases}$
Inter-AS routing — BGP (不以最短路, 而以策略计)

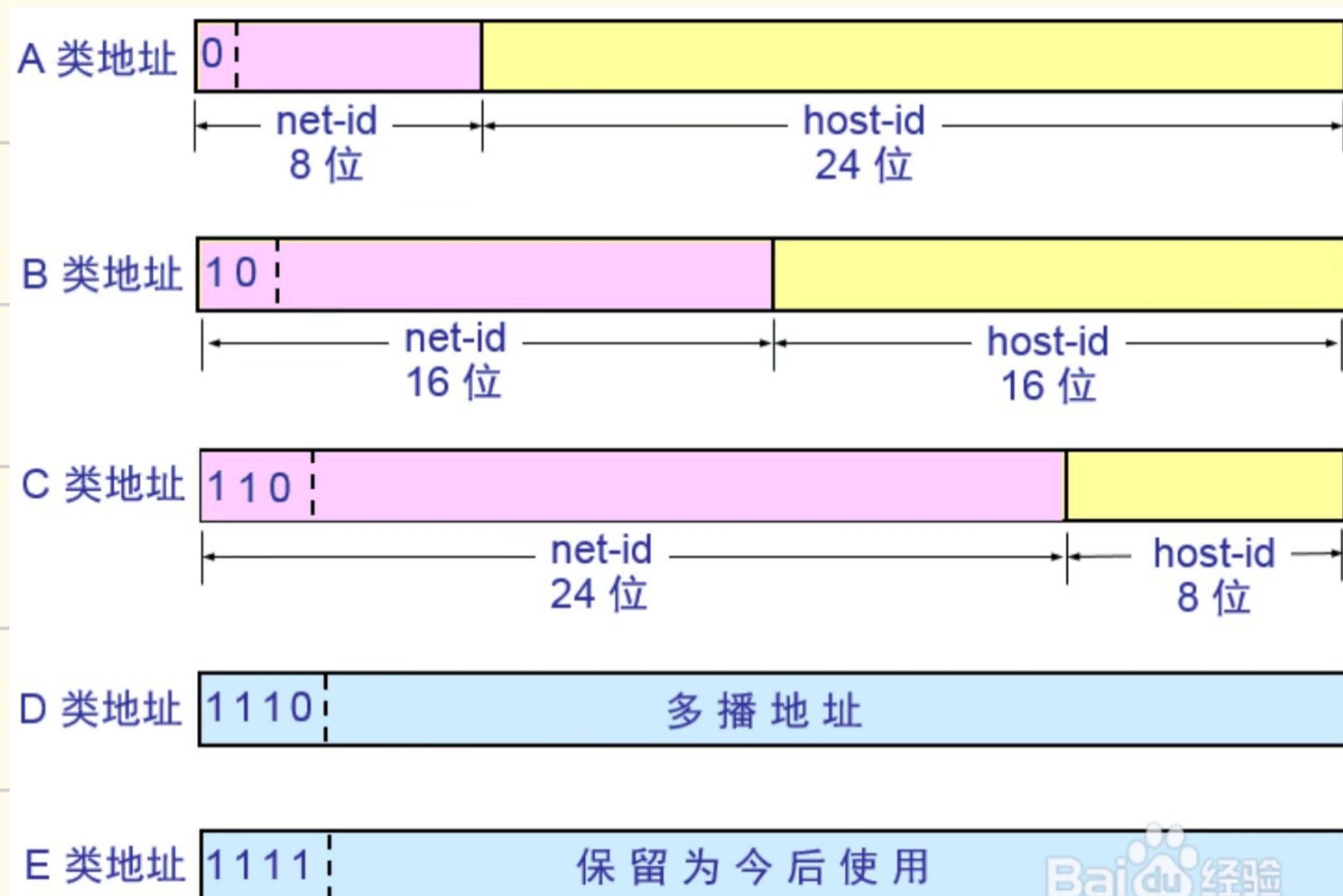
Interior Gateway Protocol: IGP

Open Shortest Path First: OSPF

Routing Information Protocol: RIP

Border Gateway Protocol: BGP

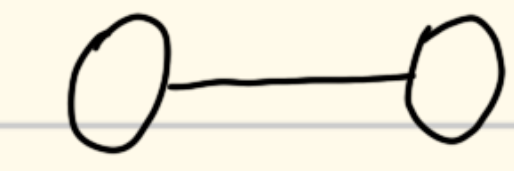
附: IP地址分类

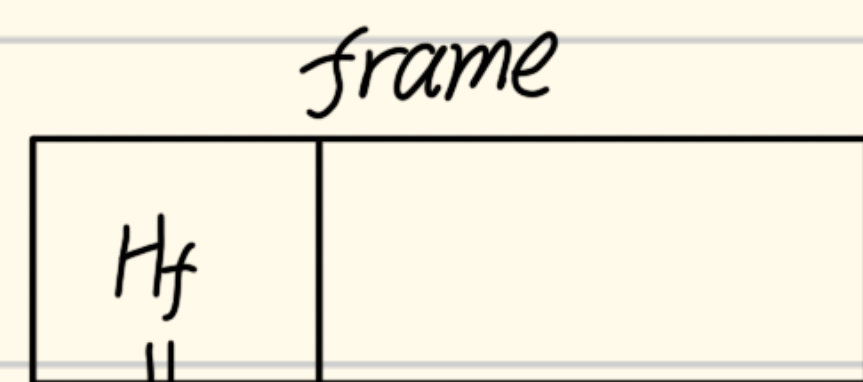


Chapter 5

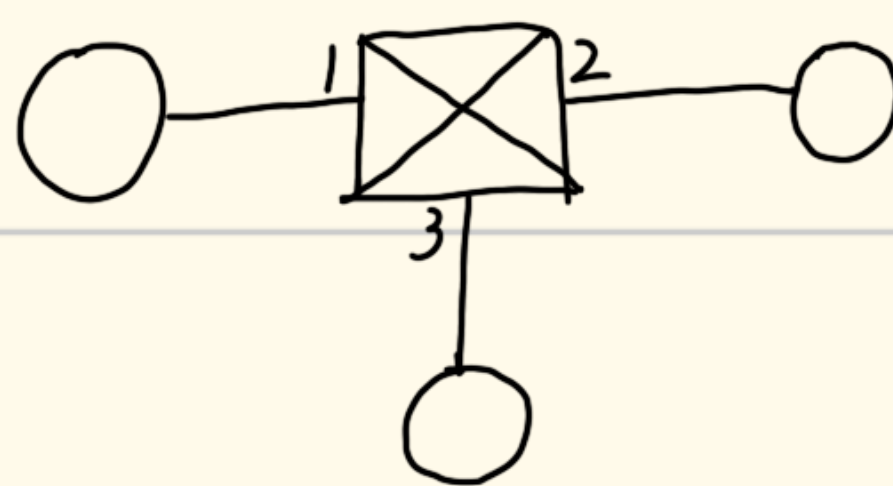
Link Layer

5.1 Link Layer: Introduction

physically adjacent < 



MAC地址
(不是IP地址)



dest	port
MAC ₁	1
MAG ₂	2
MAG ₃	3

Link layer service

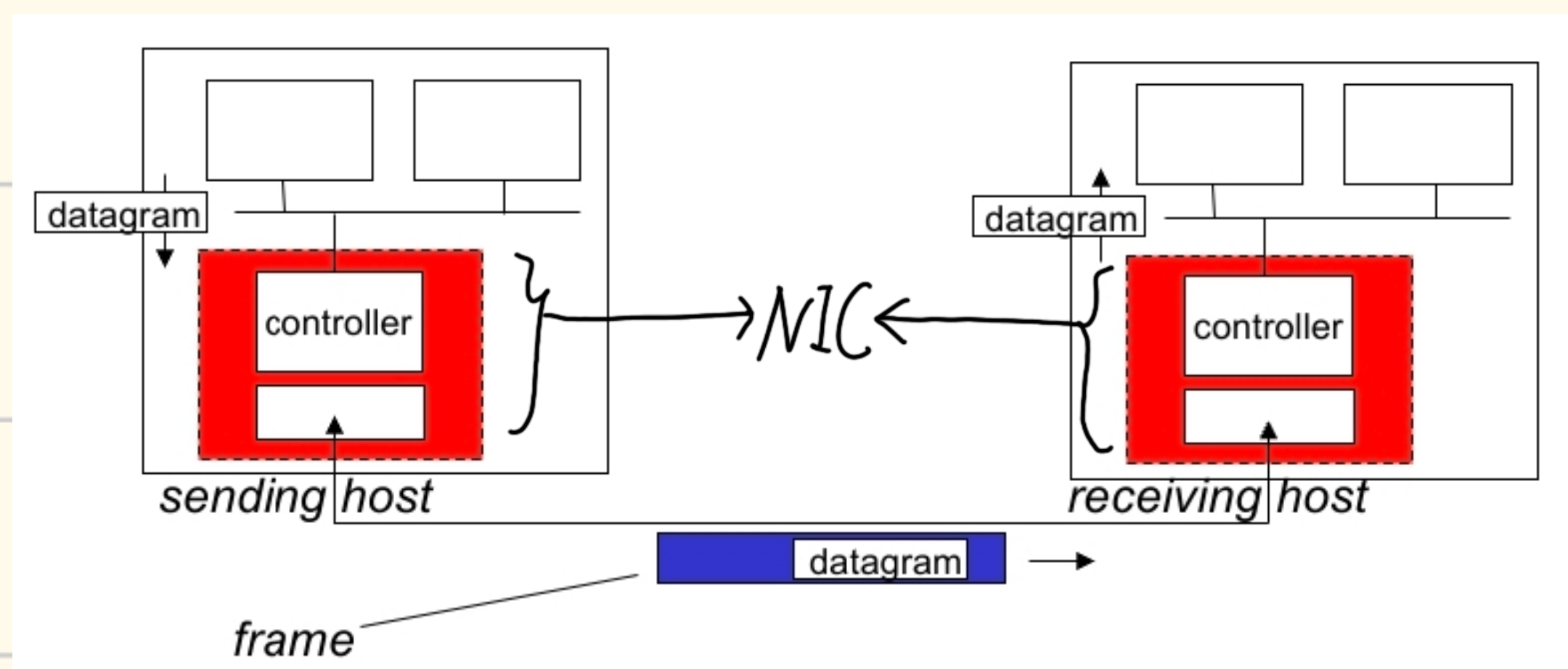
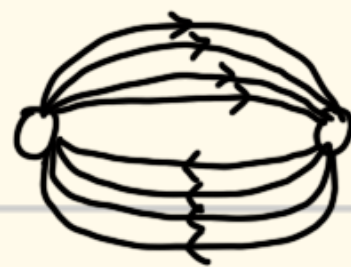
framing = MAC address

reliable = 未必有 (low-error link 无, wireless link 有)

flow control

error detection + correction

half-duplex, full-duplex



5.2 Error Detection, Correction

EDC: error detection and correction bits

Parity Checking

single bit parity:

- ❖ detect single bit errors

← d data bits
→ parity bit

0111000110101011
0

two-dimensional bit parity:

- ❖ detect and correct single bit errors

				row parity →
	$d_{1,1}$	...	$d_{1,j}$	$d_{1,j+1}$
	$d_{2,1}$	...	$d_{2,j}$	$d_{2,j+1}$
	...	...	...	...
	$d_{i,1}$	...	$d_{i,j}$	$d_{i,j+1}$
column parity ↓	$d_{i+1,1}$	...	$d_{i+1,j}$	$d_{i+1,j+1}$

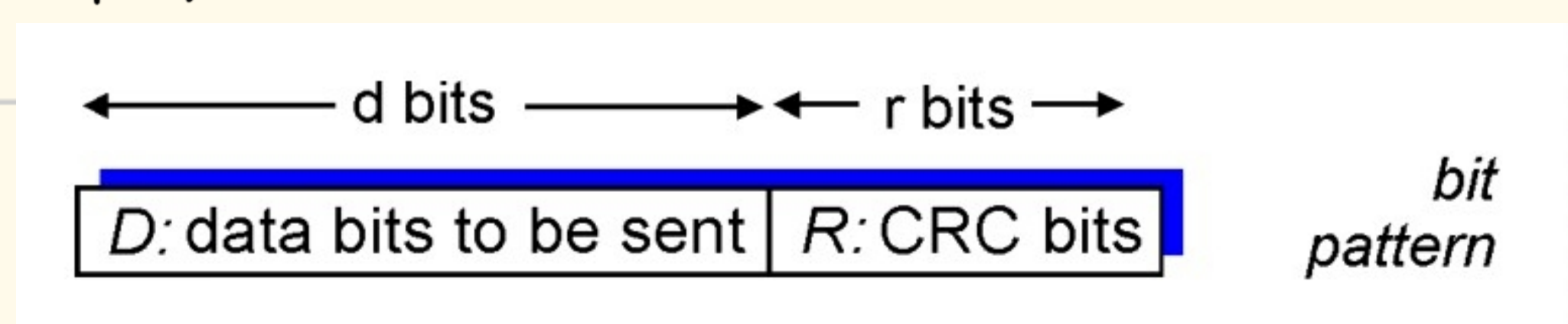
1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0
no errors					

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0
parity error					
correctable single bit error					

Checksum

每16bit相加,最后取反

Cyclic Redundancy Check (polynomial code)



生成多项式为 $x^3 + 1 \Rightarrow G = 1001$

若 $D = 101110$, $d=6$, $r=3$

1001

)

101110

000

左移位数 = G位数 - 1

XOR

1001

1001

1100

1001

1010

1001

011

011

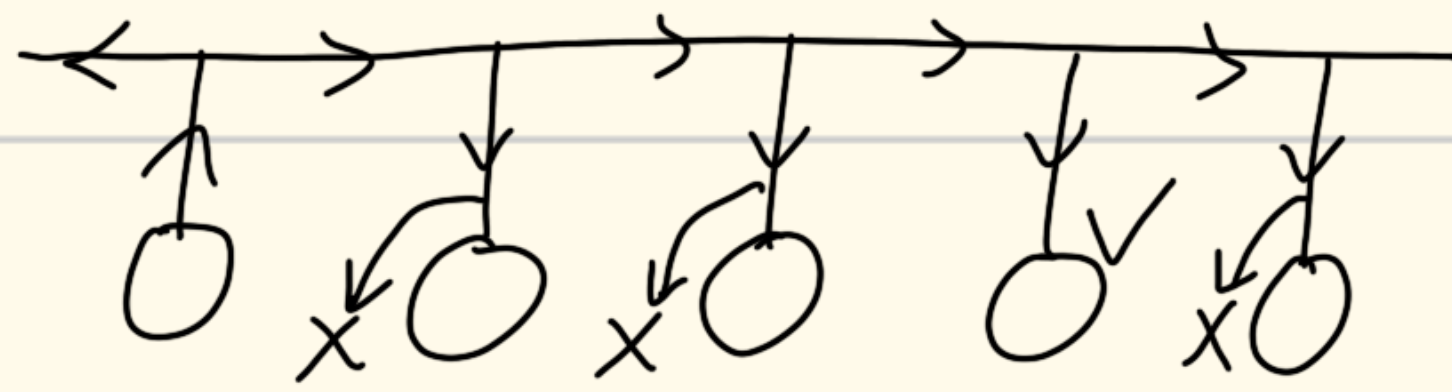
R

校验时 101110011 异或除 G , 余数为0则成功

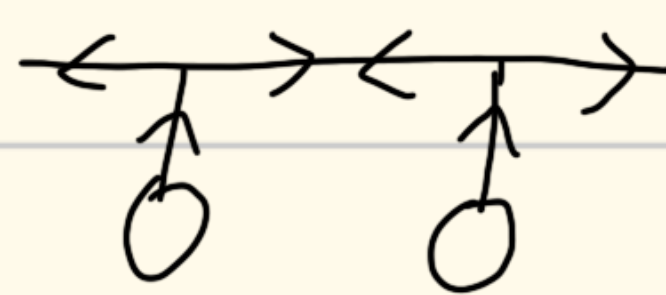
5.3 Multiple Access Protocols

links $\begin{cases} \text{point-to-point} \\ \text{broadcast} \end{cases}$

Multiple access protocol:
single shared broadcast channel



collision



理想多路传输协议 $\begin{cases} \text{fully-decentralized} \\ \text{R bps} \\ \text{R/M bps} \end{cases}$

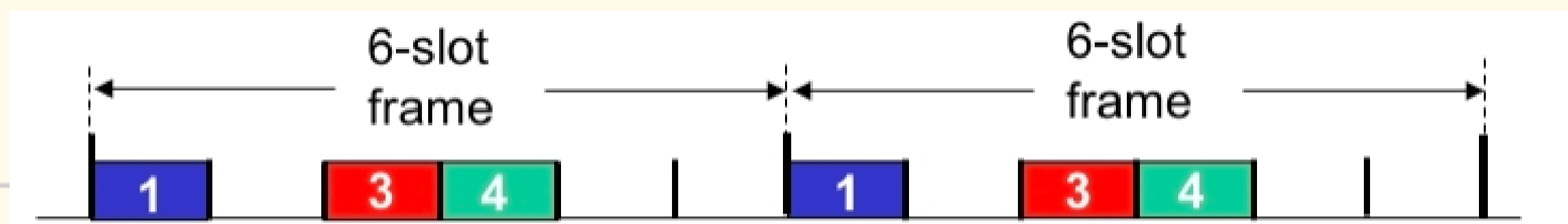
MAC Protocols

channel partitioning $\begin{cases} \text{TDMA (分时)} \\ \text{FDMA (分频)} \end{cases}$

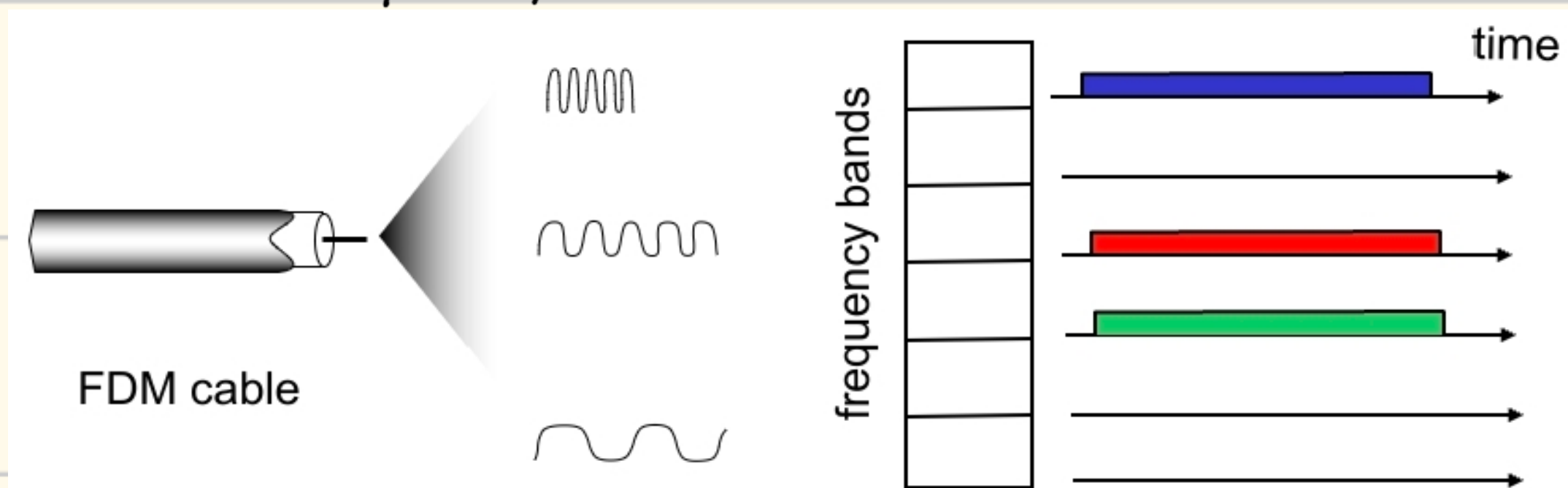
random access $\begin{cases} \text{slotted ALOHA} \\ \text{ALOHA} \\ \text{CSMA, CSMA/CD, CSMA/CA} \end{cases}$ 造成大量浪费

taking turns

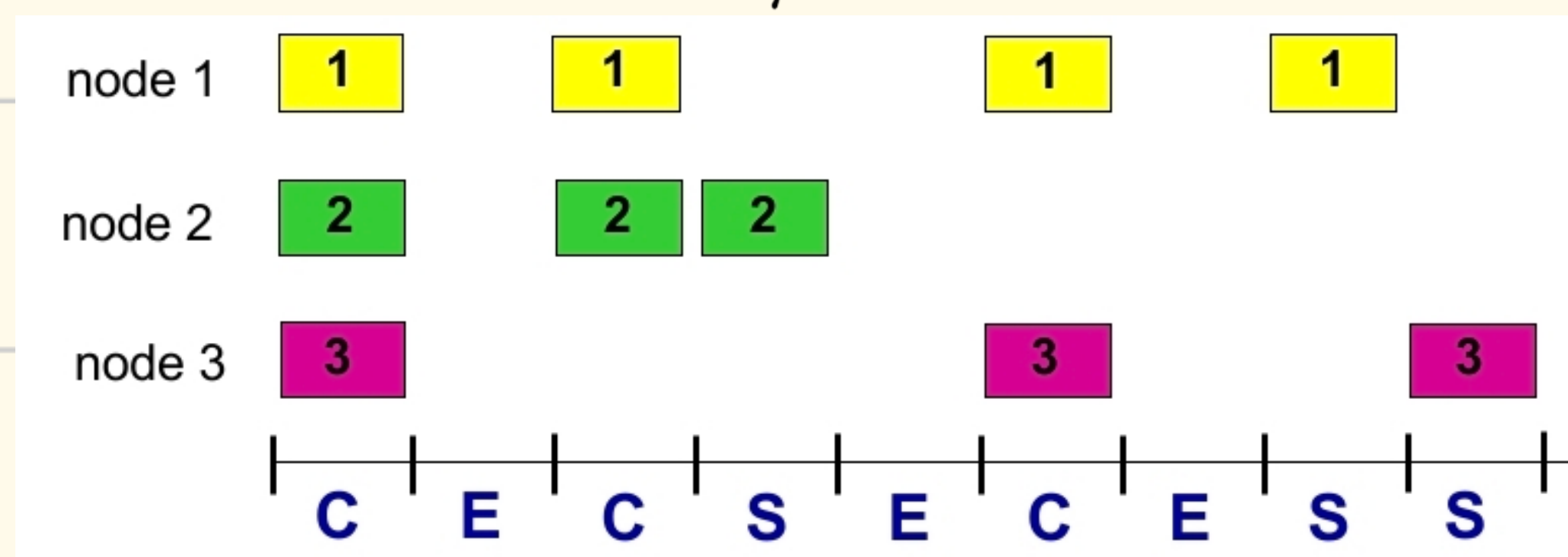
TDMA = time division multiple access (分时、分带宽)



FDMA = frequency division multiple access

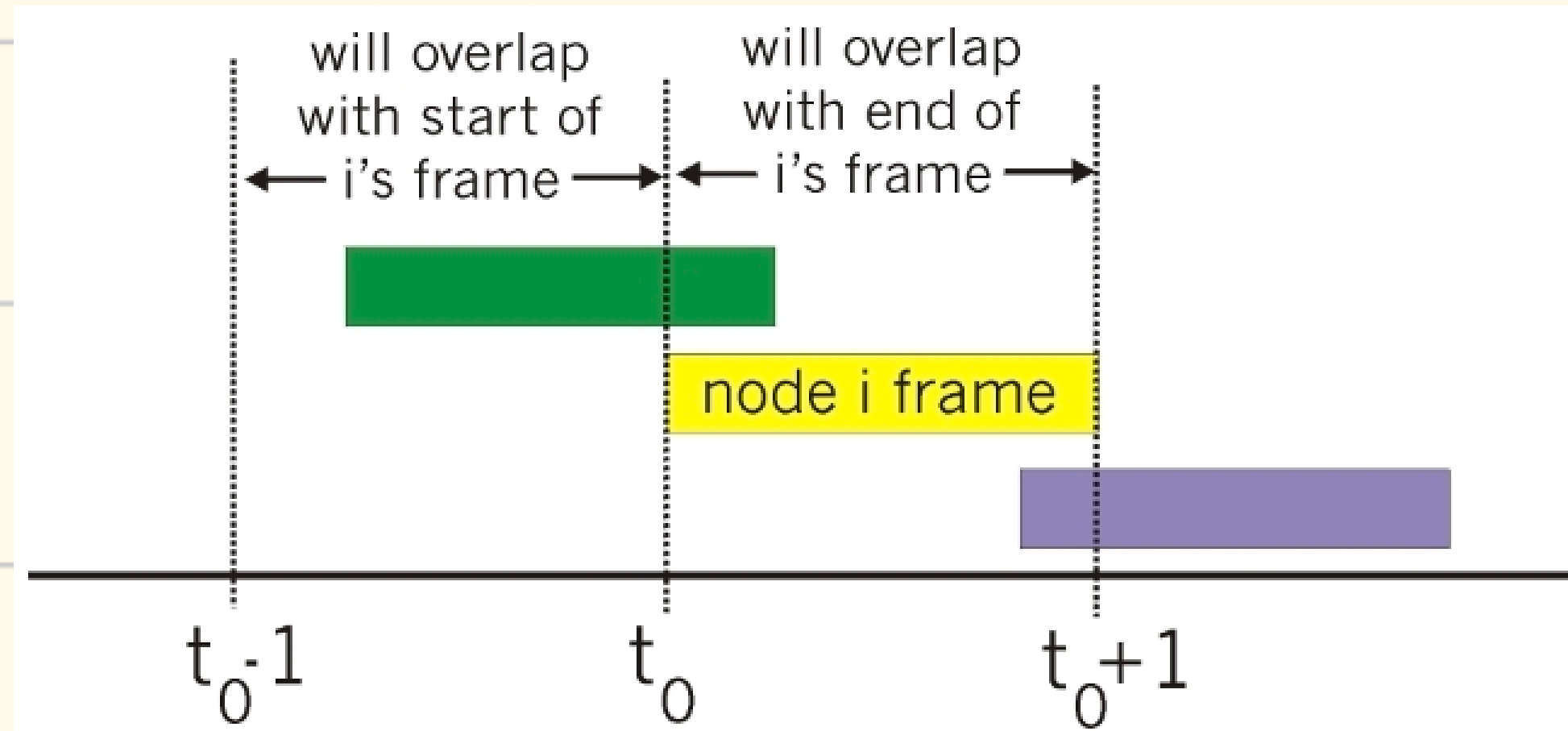


Slotted ALOHA (synchronized)



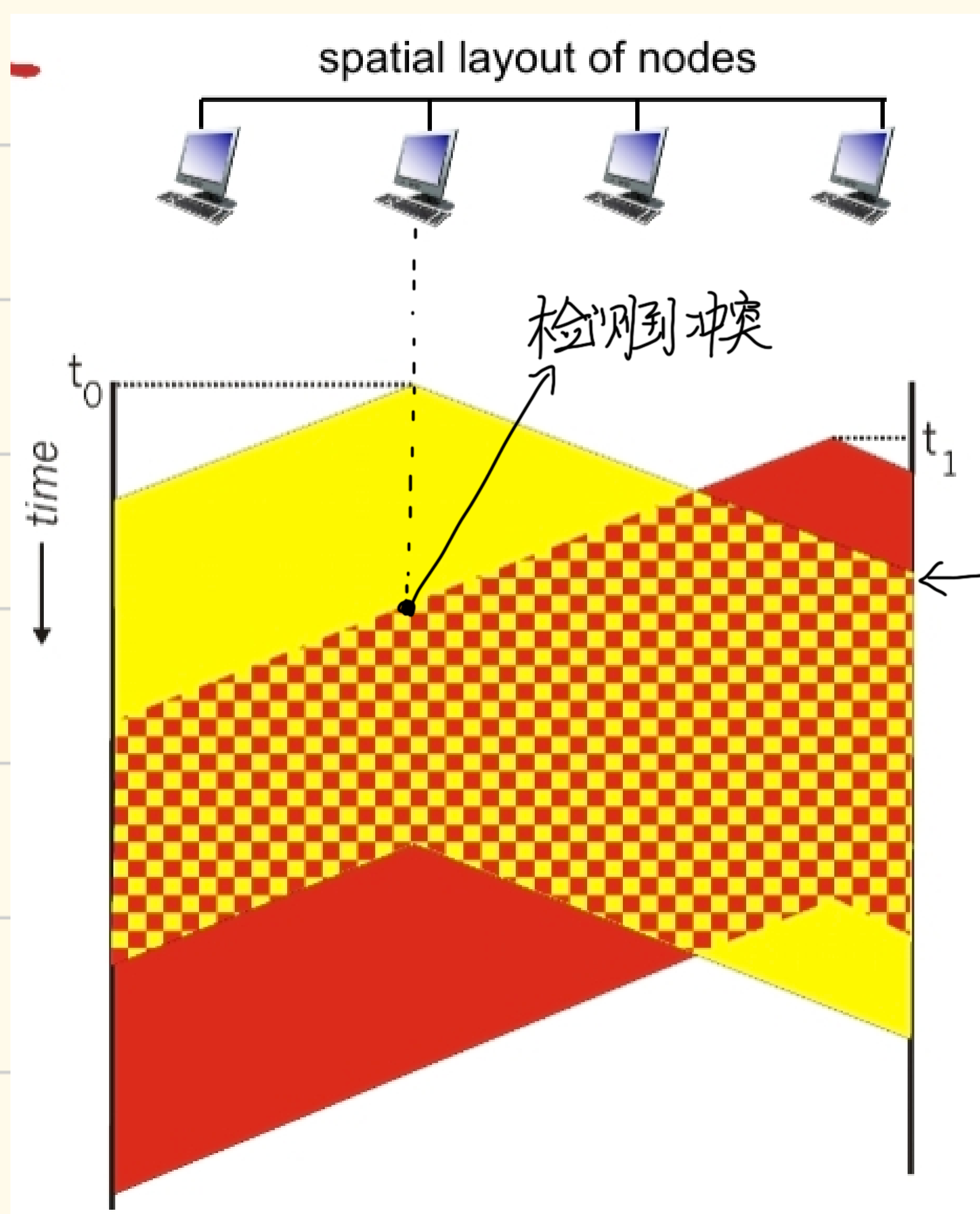
37%效率

Pure ALOHA



18%效率

CSMA = carrier sense multiple access



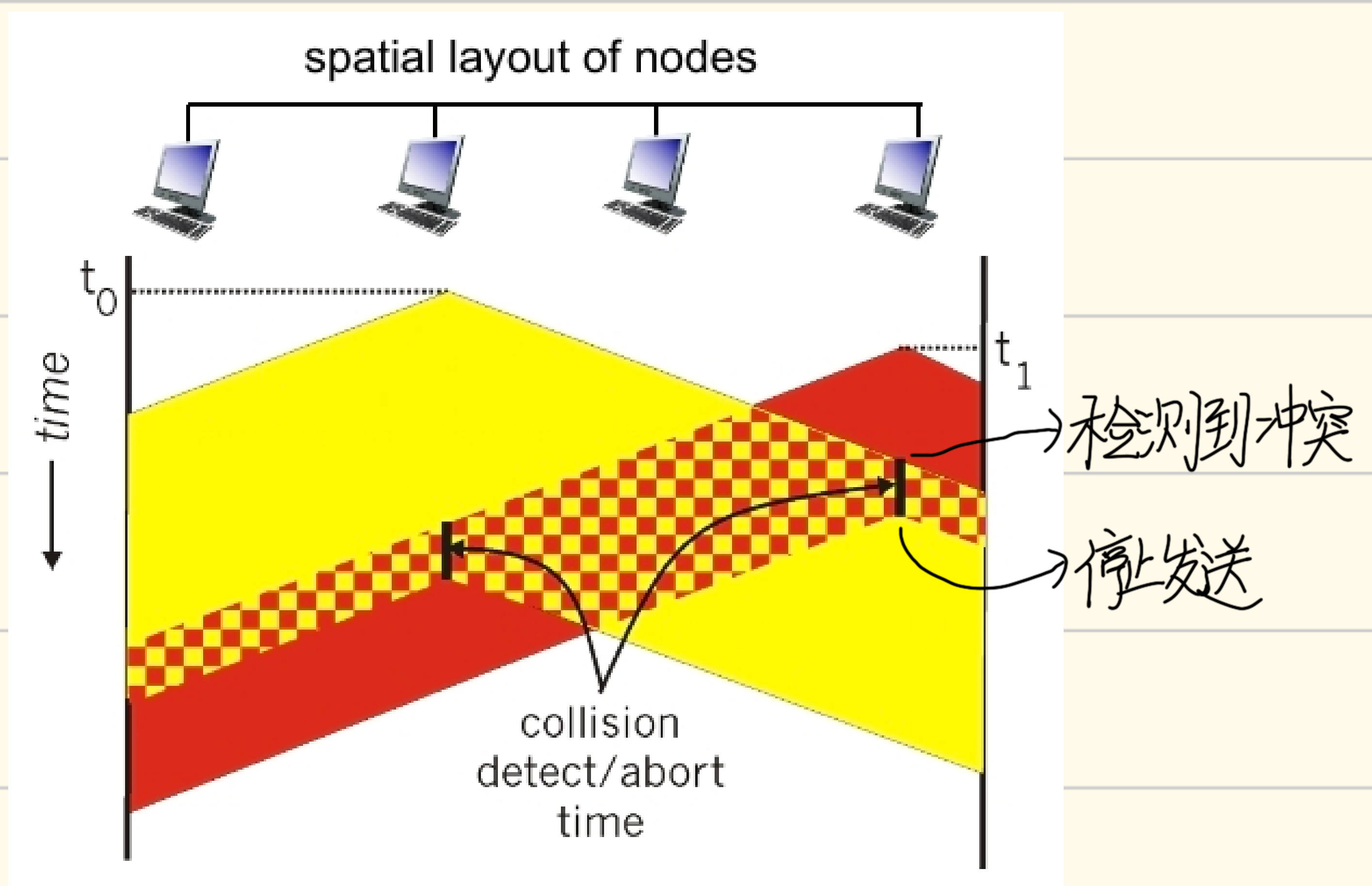
propagation delay $\Rightarrow$ collision

CSMA/CD

carrier sense multiple access = collision detection

Collision detected in short time

$\Downarrow$
transmissions aborted



CSMA/CD Algorithm

NIC senses channel idle

↓
transmits

↓
detect another transmission

↓
aborting

+
send jam signal

↓
binary (exponential) backoff

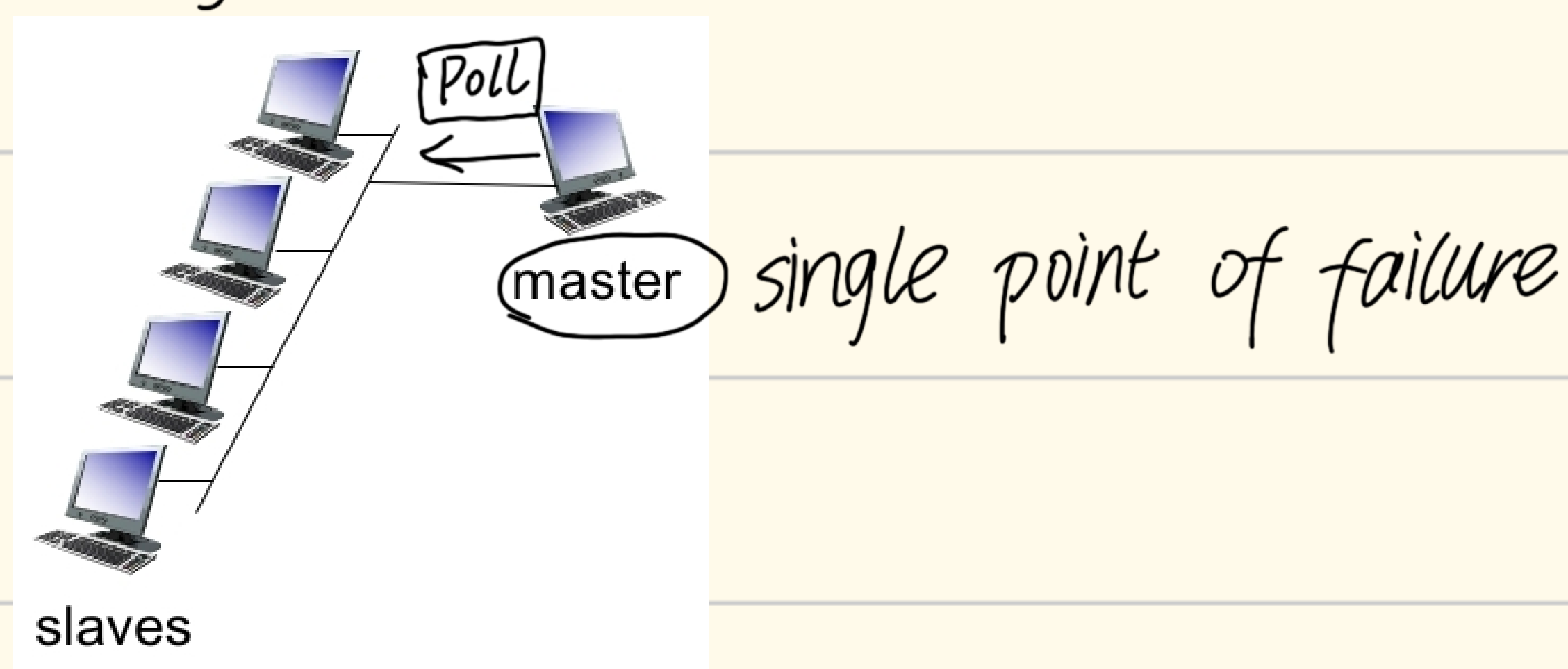
{ m次碰撞后
随机选-K值 {0, 1, 2, ..., 2^m-1}
NIC等待 K · 512 bit times

eg. K=100, R=10Mbps

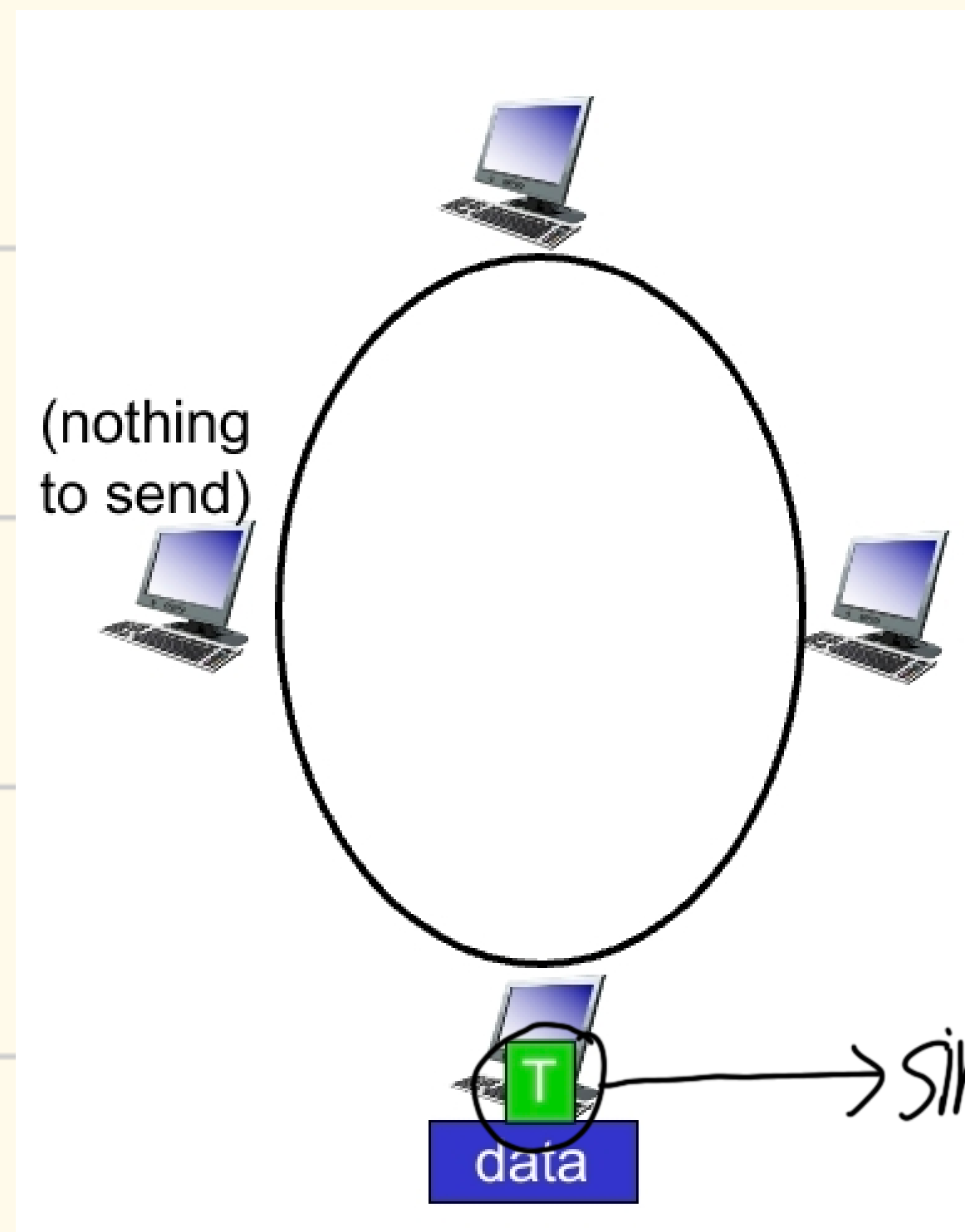
等待 $\frac{100 \times 512 \text{ bits}}{10 \times 10^6 \text{ bits/s}} = 5.12 \text{ ms}$

注: CSMA/CD in Ethernet
CSMA/CA in 802.11

Taking Turns = Polling



Taking Turns: Token Passing



channel partitioning MAC protocols: $\Rightarrow$ 总是有数据发送

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, $1/N$ bandwidth allocated even if only 1 active node!

random access MAC protocols $\Rightarrow$ 偶尔有数据发送、结点数量多

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

“taking turns” protocols

look for best of both worlds!

5.4 LANS

Mac Address

IP : 32-bit, 4字节, hierarchical $\Rightarrow$ layer 3 (network)
Mac : 48-bit, 6字节, flat (portability) $\Rightarrow$ layer 2 (link)

Mac : 1A-2F-BB-76-09-AD
burned in NIC ROM

Mac addr = LAN / physical / Ethernet addr
一般烧录后不会变化, 且每设备唯一不重复

ARP (address resolution protocol)

ARP table

IP	MAC
...	...
...	...

若表中没有, 发送 broadcasts ARP query packet (广播)

destMAC who is IP_B frame

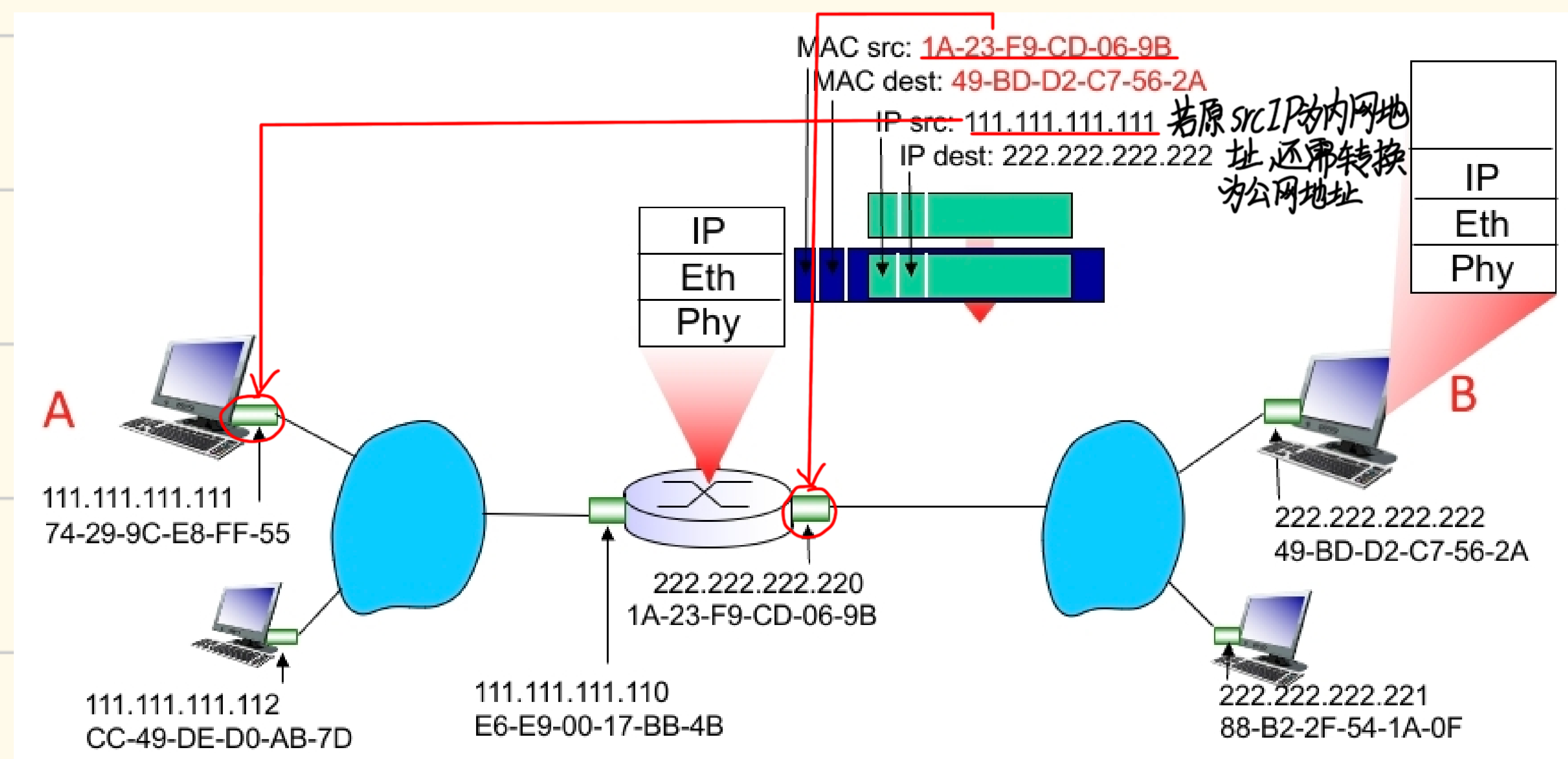
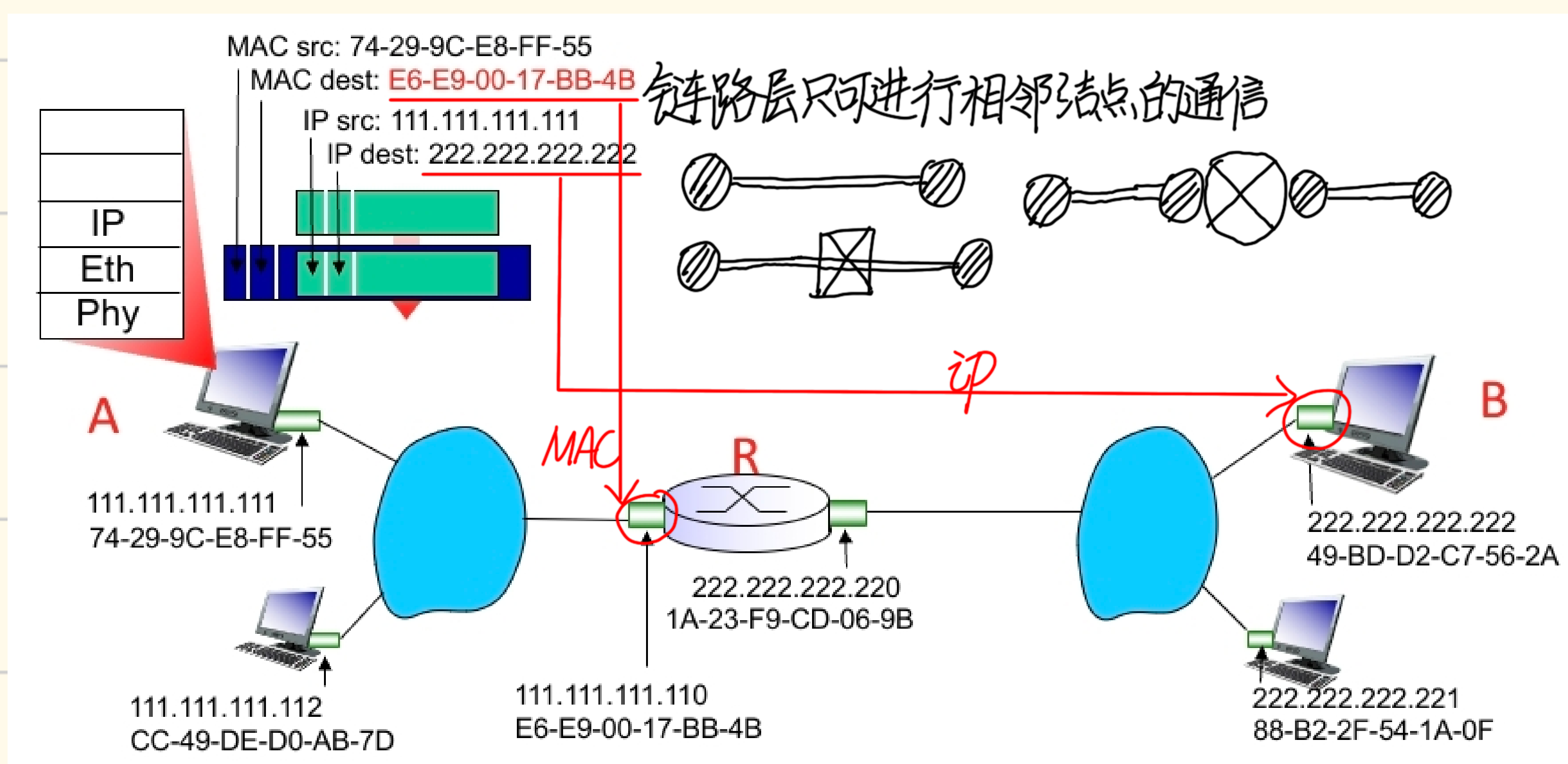
FF-FF-FF-FF-FF-FF

仅IP_B结点应答, 发送 Unicast replies (单点)

Routing To Another LAN

walkthrough: send datagram from A to B via R

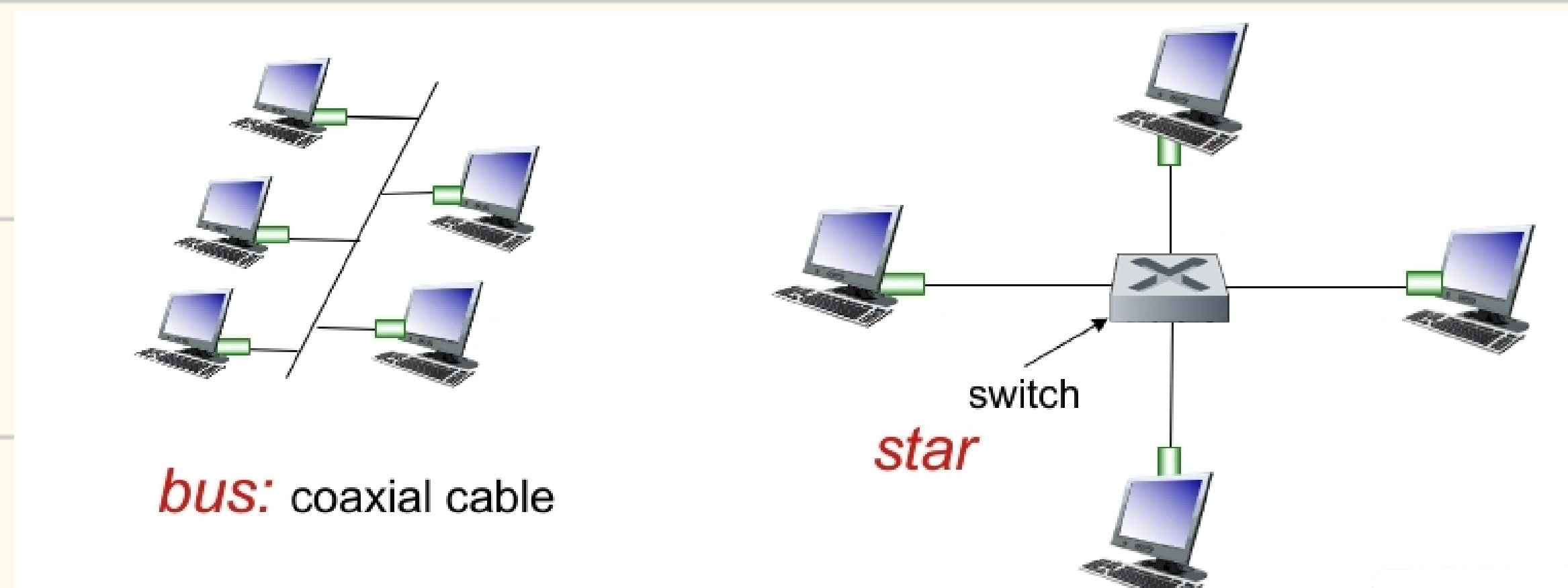
- focus on addressing – at IP (datagram) and MAC layer (frame)
 - assume A knows B's IP address
 - assume A knows IP address of first hop router, R (how?)
 - assume A knows R's MAC address (how?)
- 网关IP ⇒ DHCP offer
ARP (MAC不是通过DHCP获取!!!)



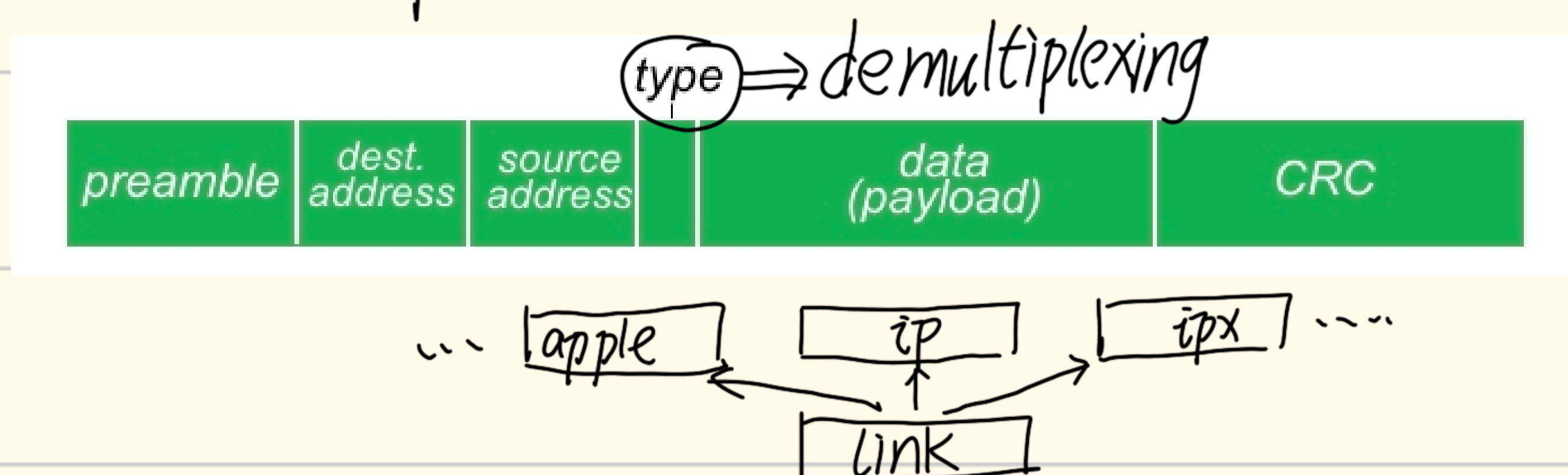
Ethernet

dominant wired LAN technology

bus
star = hub; switch; router



Ethernet frame



向上逐层demultiplexing的字段

Ethernet: type

network: upper layer

TCP: 四元组

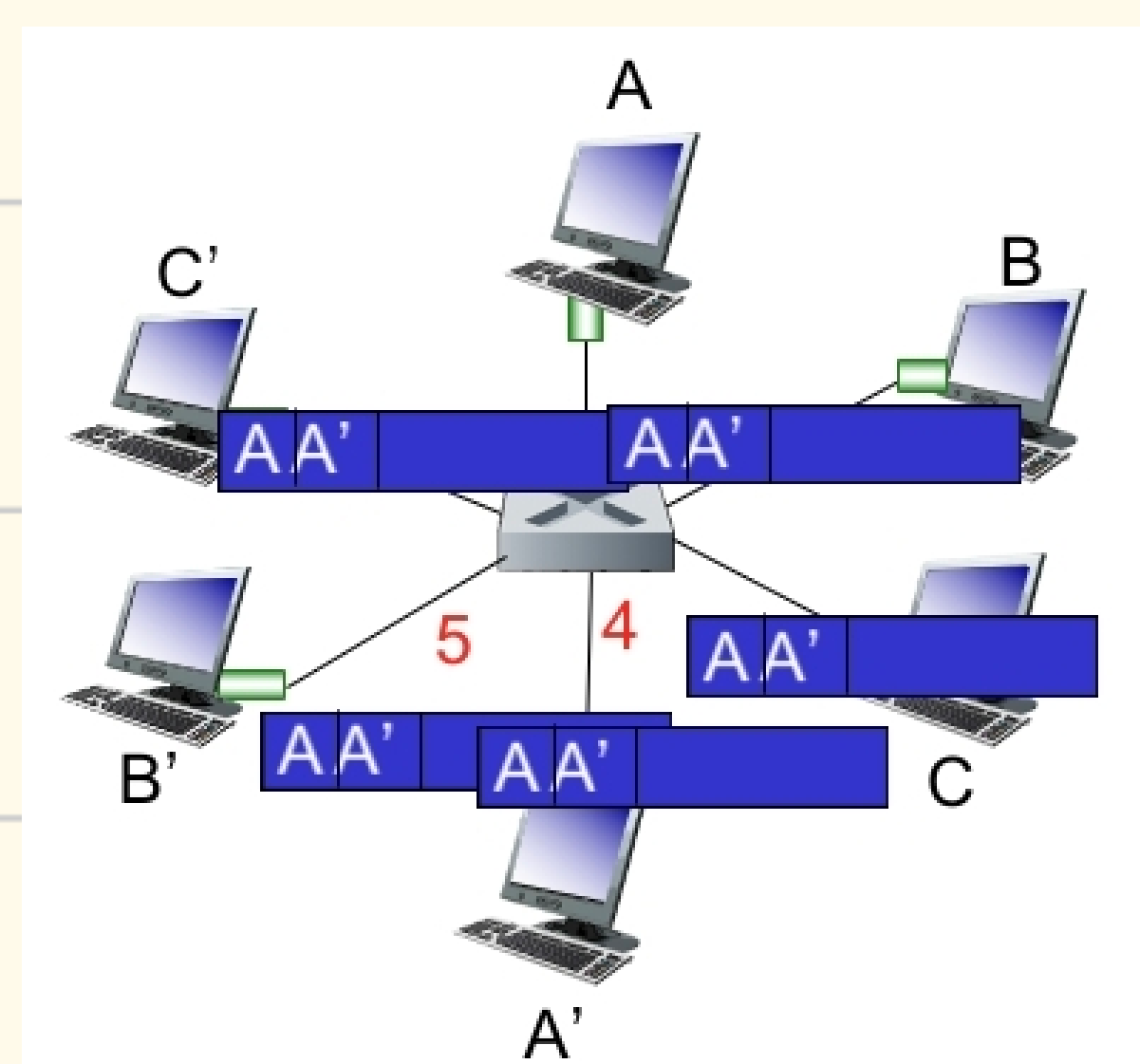
UDP: port

Ethernet Switch

link-layer device, CSMA/CD to forward

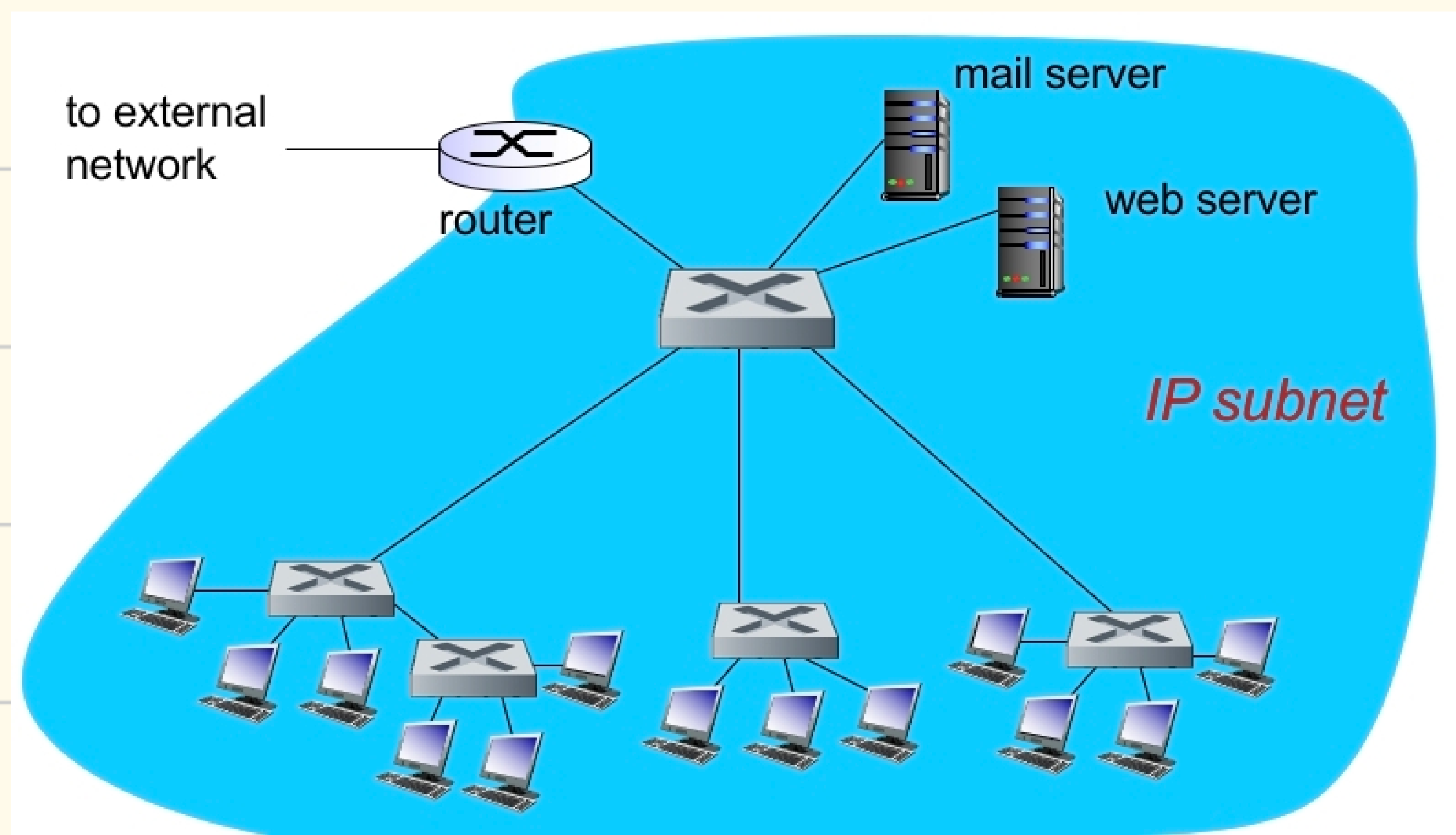
transparent: 无需手动配置, host unaware

self-learning:



广播

interconnecting switches
(级联)



对比：

Hub——转发的每一个数据包均广播

Switch——转发包通过ARP表定向转发至MAC地址

Router——多个网络间定向转发至IP地址

电话、视频会议和流式存储媒体。此外，还讨论如何设计分组交换网络以对音频和视频应用程序提供一致的服务质量。

课后习题和问题



复习题

1.1 节

- R1. “主机”和“端系统”之间有什么不同？列举几种不同类型的端系统。Web 服务器是一种端系统吗？
 R2. “协议”一词常被用于描述外交关系。维基百科是怎样描述外交协议的？
 R3. 标准对于协议为什么重要？

1.2 节

- R4. 列出 6 种接入技术。将它们分类为住宅接入、公司接入或广域无线接入。
 R5. HFC 传输速率在用户间是专用的还是共享的？在下行 HFC 信道中，可能出现碰撞吗？为什么？
 R6. 列出你所在城市中的可供使用的住宅接入技术。对于每种类型的接入方式，给出所宣称的下行速率、上行速率和每月的价格。
 R7. 以太 LAN 的传输速率是多少？
 R8. 能够运行以太网的一些物理媒体是什么？
 R9. 拨号调制解调器、HFC、DSL 和 FTTH 都用于住宅接入。对于这些技术，给出每种技术的传输速率的范围，并讨论它们的传输速率是共享的还是专用的。
 R10. 描述今天最为流行的无线因特网接入技术。对它们进行比较和对照。



1.3 节

- R11. 假定在发送主机和接收主机间只有一台分组交换机。发送主机和交换机间以及交换机和接收主机间的传输速率分别是 R_1 和 R_2 。假设该交换机使用存储转发分组交换方式，发送一个长度为 L 的分组的端到端总时延是什么？（忽略排队时延、传播时延和处理时延。）
 R12. 与分组交换网络相比，电路交换网络有哪些优点？在电路交换网络中，TDM 比 FDM 有哪些优点？
 R13. 假定用户共享一条 2Mbps 链路。同时假定当每个用户传输时连续以 1Mbps 传输，但每个用户仅传输 20% 的时间。

- a. 当使用电路交换时，能够支持多少用户？
 b. 作为该题的后继问题，假定使用分组交换。为什么如果两个或更少的用户同时传输的话，在链路前面基本上没有排队时延？为什么如果 3 个用户同时传输的话，将有排队时延？
 c. 求出某指定用户正在传输的概率。
 d. 假定现在有 3 个用户。求出在任何给定的时间，所有 3 个用户在同时传输的概率。求出队列增长的时间比率。

$$P = C_3^3 p^3 (1-p)^0 = 0.008$$

仅当 3 个用户同时传输，才会 queueing

- R14. 为什么等级结构中级别相同的两个 ISP 通常互相对等？某 IXP 是如何挣钱的？
 R15. 某些内容提供商构建了自己的网络。描述谷歌的网络。内容提供商构建这些网络的动机是什么？

1.4 节

- R16. 考虑从某源主机跨越一条固定路由向某目的主机发送一分组。列出端到端时延中的时延组成成分。这些时延中的哪些是固定的，哪些是变化的？
 R17. 访问配套 Web 网站上有关传输时延与传播时延的 Java 小程序。在速率、传播时延和可用的分组长度之中找出一种组合，使得该分组的第一个比特到达接收方之前发送方结束了传输。找出另一种组合，使得发送方完成传输之前，该分组的第一个比特到达了接收方。
 R18. 一个长度为 1000 字节的分组经距离为 2500km 的链路传播，传播速率为 2.5×10^8 m/s 并且传输速率为 2Mbps，它需要用多长时间？更为一般地，一个长度为 L 的分组经距离为 d 的链路传播

在链路上，已经完成 transmit

$$\frac{d}{s}$$

率为 s 并且传输速率为 R bps, 它需要用多长时间? 该时延与传输速率相关吗? **无关**

R19. 假定主机 A 要向主机 B 发送一个大文件。从主机 A 到主机 B 的路径上有 3 段链路, 其速率分别为 $R_1 = 500$ **kbps**, $R_2 = 2$ Mbps, $R_3 = 1$ Mbps。

a. 假定该网络中没有其他流量, 该文件传送的吞吐量是多少? **500 kbps (最慢为限制因素)**

b. 假定该文件为 4 MB **byte**, 用吞吐量除以文件长度, 将该文件传输到主机 B 大致需要多长时间?

c. 重复 (a) 和 (b), 只是这时 R_2 减小到 100 kbps。

$$\frac{4 \times 10^6 \text{ bytes} \times 8 \text{ bits}}{500 \times 10^3 \text{ bps/s}} = 64 \text{ s}$$

R20. 假定端系统 A 要向端系统 B 发送一个大文件。在一个非常高的层次上, 描述端系统怎样从该文件生成分组。当这些分组之一到达某分组交换机时, 该交换机使用分组中的什么信息来决定将该分组转发到哪一条链路上? 因特网中的分组交换为什么可以与驱车从一个城市到另一个城市并沿途询问方向相类比?

R21. 访问配套 Web 站点的排队和丢包 Java 小程序。最大发送速率和最小的传输速率是多少? 对于这些速率, 流量强度是多大? 用这些速率运行该 Java 小程序并确定出现丢包要花费多长时间? 然后第二次重复该实验, 再次确定出现丢包花费多长时间。这些值有什么不同? 为什么会有这种现象?

1.5 节

R22. 列出一个层次能够执行的 5 个任务。这些任务中的一个 (或两个) 可能由两个 (或更多) 层次执行吗?

R23. 因特网协议栈中的 5 个层次有哪些? 在这些层次中, 每层的主要任务是什么?

R24. 什么是应用层报文? 什么是运输层报文段? 什么是网络层数据报? 什么是链路层帧?

R25. **路由器**处理因特网协议栈中的哪些层次? **链路层**交换机处理的是哪些层次? **主机**处理的是哪些层次? **layer 1-3** **layer 1-2** **layer 1-5**

1.6 节

R26. 病毒和蠕虫之间有什么不同?

R27. 描述如何产生一个僵尸网络, 以及僵尸网络是怎样被用于 DDoS 攻击的。

R28. 假定 Alice 和 Bob 经计算机网络互相发送分组。假定 Trudy 将自己安置在网络中, 使得她能够俘获由 Alice 发送的所有分组, 并发送她希望给 Bob 的东西; 她也能够俘获由 Bob 发送的所有分组, 并发送她希望给 Alice 的东西。列出在这种情况下 Trudy 能够做的某些恶意的东西。



习题

P1. 设计并描述在自动柜员机和银行的中央计算机之间使用的一种应用层协议。你的协议应当允许验证用户卡和口令, 查询账目结算 (这些都在中央计算机中进行维护), 支取账目 (即向用户支付钱)。你的协议实体应当能够处理取钱时账目中钱不够的常见问题。通过列出自动柜员机和银行中央计算机在报文传输和接收过程中交换的报文和采取的动作来定义你的协议。使用类似于图 1-2 所示的图, 拟定在简单无差错取钱情况下该协议的操作。明确地阐述在该协议中关于底层端到端运输服务所做的假设。

单链路: $\frac{L}{R}$, N 段: $\frac{NL}{R}$, 等待剩余 $(P-1)$ 个包: $(P-1) \cdot \frac{L}{R}$

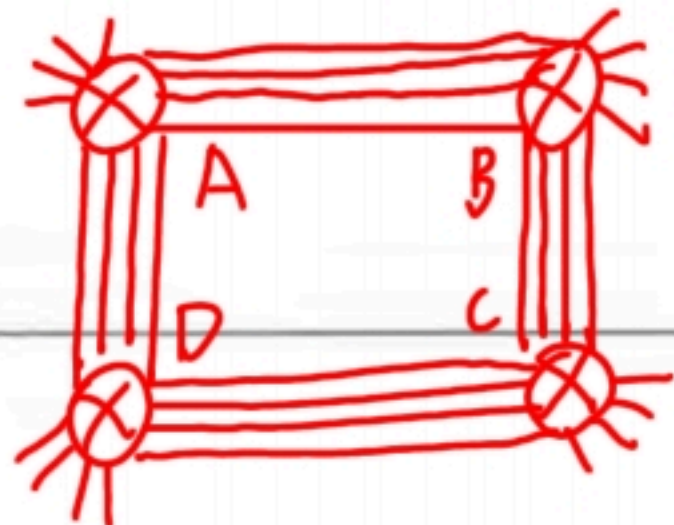
P2. 式 (1-1) 给出了经传输速率为 R 的 N 段链路发送长度 L 的一个分组的端到端时延。对于经过 N 段链路一个接一个地发送 P 个这样的分组, 一般化地表示出这个公式。 **$(N+P-1) \cdot \frac{L}{R}$**

P3. 考虑一个应用程序以稳定的速率传输数据 (例如, 发送方每 k 个时间单元产生一个 N 比特的数据单元, 其中 k 较小且固定)。另外, 当这个应用程序启动时, 它将连续运行相当长的一段时间。回答下列问题, 简要论证你的回答:

a. 是分组交换网还是电路交换网更为适合这种应用? 为什么? **电路交换**

b. 假定使用了分组交换网, 并且该网中的所有流量都来自如上所述的这种应用程序。此外, 假定该应用程序数据传输速率的总和小于每条链路的各自容量。需要某种形式的拥塞控制吗? 为什么?

P4. 考虑在图 1-13 中的电路交换网。回想在每条链路上有 4 条链路, 以顺时针方向标记四台交换机 A、



B、C 和 D。

- a. 在该网络中，任何时候能够进行同时连接的最大数量是多少？ $4+4+4+4=16$ 条
- b. 假定所有连接位于交换机 A 和 C 之间。能够进行同时连接的最大数量是多少？ $4+4=8$ 条
- c. 假定我们要在交换机 A 和 C 之间建立 4 条连接，在交换机 B 和 D 之间建立另外 4 条连接。我们能够让这些呼叫通过这 4 条链路建立路由以容纳所有 8 条连接吗？ 可以

P5. 回顾在 1.4 节中的车队的类比。假定传播速度为 100km/h。

- a. 假定车队旅行 150km：在一个收费站前面开始，通过第二个收费站，并且正好在第三个收费站后面结束。其端到端时延是多少？
- b. 重复 (a)，现在假定车队中有 8 辆汽车而不是 10 辆。

P6. 这个习题开始探讨传播时延和传输时延，这是数据网络中的两个重要概念。考虑两台主机 A 和 B 由一条速率为 R bps 的链路相连。假定这两台主机相隔 m 米，沿该链路的传播速率为 s m/s。主机 A 向主机 B 发送长度 L 比特的分组。

- a. 用 m 和 s 来表示传播时延 d_{prop} 。
- b. 用 L 和 R 来确定该分组的传输时间 d_{trans} 。
- c. 忽略处理和排队时延，得出端到端时延的表达式。
- d. 假定主机 A 在时刻 $t=0$ 开始传输该分组。在时刻 $t=d_{\text{trans}}$ ，该分组的最后一个比特在什么地方？
- e. 假定 d_{prop} 大于 d_{trans} 。在时刻 $t=d_{\text{trans}}$ ，该分组的第一个比特在何处？ $d_{\text{prop}} = \frac{m}{s} = d_{\text{trans}} = \frac{L}{R}$
- f. 假定 d_{prop} 小于 d_{trans} 。在时刻 $t=d_{\text{trans}}$ ，该分组的第一个比特在何处？ $\Rightarrow m = \frac{120 \times 2.5 \times 10^8}{56 \times 10^3} = 536 \text{ km}$
- g. 假定 $s = 2.5 \times 10^8$ ， $L = 120$ 比特， $R = 56 \text{ kbps}$ 。求出使 d_{prop} 等于 d_{trans} 的距离 m 。

P7. 在这个习题中，我们考虑从主机 A 向主机 B 通过分组交换网发送语音 (VoIP)。主机 A 将模拟语音转换为传输中的 64kbps 数字比特流。然后主机 A 将这些比特分为 56 字节的分组。A 和 B 之间有一条链路：它的传输速率是 2Mbps，传播时延是 10ms。一旦 A 收集了一个分组，就将它向主机 B 发送。一旦主机 B 接收到一个完整的分组，它将该分组的比特转换成模拟信号。从比特产生（从位于主机 A 的初始模拟信号起）的时刻起，到该比特被解码（在主机 B 上作为模拟信号的一部分），花了多少时间？

P8. 假定用户共享一条 3Mbps 的链路。又设每个用户传输时要求 150kbps，但是每个用户仅有 10% 的时间传输。（参见 1.3 节中关于“分组交换与电路交换的对比”的讨论。）

- a. 当使用电路交换时，能够支持多少用户？
- b. 对于本习题的后续小题，假定使用分组交换。求出某给定用户正在传输的概率。
- c. 假定有 120 个用户。求出在任何给定时刻，实际有 n 个用户同时在传输的概率。（提示：使用二项式分布。）
某用户在传输 $P=0.2$ ， n 个用户 $P = C_{120}^n p^n (1-p)^{120-n}$
- d. 求出有 21 个或更多用户同时传输的概率。

P9. 考虑在 1.3 节“分组交换与电路交换的对比”的讨论中，给出了一个具有一条 1Mbps 链路的例子。用户在忙时以 100kbps 速率产生数据，但忙时仅以 $p=0.1$ 的概率产生数据。假定用 1Gbps 链路替代 1Mbps 的链路。

- a. 当采用电路交换技术时，能被同时支持的最大用户数量 N 是多少？
- b. 现在考虑分组交换和有 M 个用户的情况。给出多于 N 用户发送数据的概率公式（用 p 、 M 、 N 表示）。

P10. 考虑一个长度为 L 的分组从端系统 A 开始，经 3 段链路传送到目的端系统。令 d_i 、 s_i 和 R_i 表示链路 i 的长度、传播速度和传输速率 ($i=1, 2, 3$)。该分组交换机对每个分组的时延为 d_{proc} 。假定没有排队时延，用 d_i 、 s_i 、 R_i ($i=1, 2, 3$) 和 L 表示，该分组总的端到端时延是什么？现在假定该分组是 1500 字节，在所有 3 条链路上的传播时延是 2.5×10^8 m/s，所有 3 条链路的传输速率是 2Mbps，分组交换机的处理时延是 3ms，第一段链路的长度是 5000km，第二段链路的长度是 4000km，并且最后一段链路的长度是 1000km。对于这些值，该端到端时延为多少？

P11. 在上述习题中，假定 $R_1 = R_2 = R_3 = R$ 且 $d_{\text{proc}} = 0$ 。进一步假定该分组交换机不存储转发分组，而是

P7. 在这个习题中，我们考虑从主机 A 向主机 B 通过分组交换网发送语音 (VoIP)。主机 A 将模拟语音转换为传输中的 64kbps 数字比特流。然后主机 A 将这些比特分为 56 字节的分组。A 和 B 之间有一条链路：它的传输速率是 2Mbps，传播时延是 10ms。一旦 A 收集了一个分组，就将它向主机 B 发送。一旦主机 B 接收到一个完整的分组，它将该分组的比特转换成模拟信号。从比特产生（从位于主机 A 的初始模拟信号起）的时刻起，到该比特被解码（在主机 B 上作为模拟信号的一部分），花了多少时间？

标定组内第一个 bit

模拟语音 $\rightarrow$ 比特流 = (等待组内其余比特产生) $\frac{56 \text{ bytes} \times 8 \text{ bits}}{64 \times 10^3 \text{ bits/s}} = 7 \times 10^{-3} \text{ s}$

主机 A $\rightarrow$ 主机 B: $d_{\text{prop}} + d_{\text{trans}} = 10 \times 10^{-3} \text{ s} + \frac{56 \times 8 \text{ bits}}{2 \times 10^6 \text{ bits/s}} = 10.224 \times 10^{-3} \text{ s}$

比特流 $\rightarrow$ 模拟语音: 0s

$\Rightarrow$ 共需 $17.224 \times 10^{-3} \text{ s}$

P10. 考虑一个长度为 L 的分组从端系统 A 开始，经 3 段链路传送到目的端系统。令 d_i 、 s_i 和 R_i 表示链路 i 的长度、传播速度和传输速率 ($i=1, 2, 3$)。该分组交换机对每个分组的时延为 d_{proc} 。假定没有排队时延，用 d_i 、 s_i 、 R_i ($i=1, 2, 3$) 和 L 表示，该分组总的端到端时延是什么？现在假定该分组是 1500 字节，在所有 3 条链路上的传播时延是 $2.5 \times 10^{-8} \text{ m/s}$ ，所有 3 条链路的传输速率是 2Mbps，分组交换机的处理时延是 3ms，第一段链路的长度是 5000km，第二段链路的长度是 4000km，并且最后一段链路的长度是 1000km。对于这些值，该端到端时延为多少？

P11. 在上述习题中，假定 $R_1 = R_2 = R_3 = R$ 且 $d_{\text{proc}} = 0$ 。进一步假定该分组交换机不存储转发分组，而是在等待分组到达前立即传输它收到的每个比特。这时端到端时延为多少？

P10 原先: $d = \frac{L}{R_1} + \frac{L}{R_2} + \frac{L}{R_3} + \frac{d_1}{s_1} + \frac{d_2}{s_2} + \frac{d_3}{s_3} + 2d_{\text{proc}}$

P11 后来: 2 个交换机不再存储, 无需等待其余 bits 到达

但接收端仍需存储、等待并检验

$$d = \frac{L}{R_3} + \frac{d_1}{s_1} + \frac{d_2}{s_2} + \frac{d_3}{s_3}$$

在等待分组到达前立即传输它收到的每个比特。这时端到端时延为多少？

- P12. 一台分组交换机接收一个分组并决定该分组应当转发的出链路。当某分组到达时，另一个分组正在该出链路上被发送到一半，还有 4 个其他分组正等待传输。这些分组以到达的次序传输。假定所有分组是 1500 字节并且链路速率是 2Mbps。该分组的排队时延是多少？在更一般的情况下，当所有分组的长度是 L ，传输速率是 R ，当前正在传输的分组已经传输了 x 比特，并且已经在队列中有 n 个分组，其排队时延是多少？
- P13. a. 假定有 N 个分组同时到达一条当前没有分组传输或排队的链路。每个分组长为 L ，链路传输速率为 R 。对 N 个分组而言，其平均排队时延是多少？
b. 现在假定每隔 LN/R 秒有 N 个分组同时到达链路。一个分组的平均排队时延是多少？
- P14. 考虑某路由器缓存中的排队时延。令 I 表示流量强度；即 $I = La/R$ 。假定排队时延的形式为 $IL/R(1-I)$ ，其中 $I < 1$ 。
a. 写出总时延即排队时延加上传输时延的公式。
b. 以 L/R 为函数画出总时延的图。
- P15. 令 a 表示在一条链路上分组的到达率（以分组/秒计），令 μ 表示一条链路上分组的传输率（以分组/秒计）。基于上述习题中推导出的总时延公式（即排队时延加传输时延），推导出以 a 和 μ 表示的总时延公式。
- P16. 考虑一台路由器缓存前面的一条出链路。在这个习题中，将使用李特尔（Little）公式，这是排队论中的一个著名公式。令 N 表示在缓存中的分组加上被传输的分组的平均数。令 a 表示到达该链路的分组速率。令 d 表示一个分组历经的平均总时延（即排队时延加传输时延）。李特尔公式是 $N = a \times d$ 。假定该缓存平均包含 10 个分组，并且平均分组排队时延是 10ms。该链路的传输速率是 100 分组/秒。使用李特尔公式，在没有丢包的情况下，平均分组到达率是多少？
- P17. a. 对于不同的处理速率、传输速率和传播时延，给出 1.4.3 节中式 (1-2) 的一般表达式。
b. 重复 (a)，不过此时假定在每个节点有平均排队时延 d_{queue} 。
- P18. 在一天的 3 个不同的小时内，在同一个大陆上的源和目的地之间执行 Traceroute。
a. 在这 3 个小时的每个小时中，求出往返时延的均值和方差。
b. 在这 3 个小时的每个小时中，求出路径上的路由器数量。在这些时段中，该路径发生变化了吗？
c. 试图根据源到目的地 Traceroute 分组通过的情况，辨别 ISP 网络的数量。具有类似名字和/或类似的 IP 地址的路由器应当被认为是同一个 ISP 的一部分。在你的实验中，在相邻的 ISP 间的对等接口处出现最大的时延了吗？
d. 对位于不同大陆上的源和目的地重复上述内容。比较大陆内部和大陆之间的这些结果。
- P19. a. 访问站点 www.traceroute.org，并从法国两个不同的城市向位于美国的相同的目的主机执行 Traceroute。在这两个 Traceroute 中，有多少条链路是相同的？大西洋沿岸国家的链路相同吗？
b. 重复 (a)，但此时选择位于法国的一个城市和位于德国的另一个城市。
c. 在美国挑选一个城市，然后向位于中国的两个不同城市的主机执行 Traceroute。在这两次 Traceroute 中有多少链路是相同的？在到达中国前这两个 Traceroute 分开了吗？



- P20. 考虑对应于图 1-20b 吞吐量的例子。现在假定有 M 对客户-服务器而不是 10 对。用 R_s 、 R_c 和 R 分别表示服务器链路、客户链路和网络链路的速率。假设所有的其他链路都有充足容量，并且除了由这 M 对客户-服务器产生的流量外，网络中没有其他流量。推导出由 R_s 、 R_c 、 R 和 M 表示的通用吞吐量表达式。 $\min\{R_s, R_c, R/M\}$

- P21. 考虑图 1-19b。现在假定在服务器和客户之间有 M 条路径。任两条路径都不共享任何链路。路径 k ($k=1, \dots, M$) 由传输速率为 $R_1^k, R_2^k, \dots, R_N^k$ 的 N 条链路组成。如果服务器仅能够使用一条路径向客户发送数据，则该服务器能够取得的最大吞吐量是多少？如果该服务器能够使用所有 M 条路径发送数据，则该服务器能够取得的最大吞吐量是多少？ $\min\{R_1^k, \dots, R_N^k\} = t_k$

- P22. 考虑图 1-19b。假定服务器与客户之间的每条链路的丢包概率为 p ，且这些链路的丢包率是独立的。一个（由服务器发送的）分组成功地被接收方收到的概率是多少？如果在从服务器到客户的路径上

用单条路径: $\max\{t_1, \dots, t_k\}$
用全部路径: $\sum t_k$

成功传输 $(1-p)^N = p_0$, 而传输次数 $\sim G(p_0)$ [几何分布]
 $E(\text{传输次数}) = \frac{1}{(1-p)^N} \Rightarrow \text{重传 } \frac{1}{(1-p)^N} - 1 \text{ 次}$

分组丢失了, 则服务器将重传该分组。平均来说, 为了使客户成功地接收该分组, 服务器将要重传该分组多少次?

P23. 考虑图 1-19a。假定我们知道沿着从服务器到客户的路径的瓶颈链路是速率为 R_s bps 的第一段链路。假定我们从服务器向客户发送紧密相连的一对分组, 且沿这条路径没有其他流量。假定每个分组的长度为 L 比特, 两条链路具有相同的传播时延 d_{prop} 。

a. 在目的地, 分组的到达间隔时间有多大? 也就是说, 从第一个分组的最后一个比特到达第二个分组最后一个比特到达所经过的时间有多长? L/R_s

b. 现在假定第二段链路是瓶颈链路 (即 $R_c < R_s$)。第二个分组在第二段链路输入队列中排队是可能的吗? 是 请解释原因。现在假定服务器在发送第一个分组 T 秒之后再发送第二个分组。为确保在第二段链路之前没有排队, T 必须要有多长? 试解释原因。 $T \geq \frac{L}{R_c} - \frac{L}{R_s}$

P24. 假设你希望从波士顿向洛杉矶紧急传送 40×10^{12} 字节数据。你有一条 100Mbps 专用链路可用于传输数据。你是愿意通过这条链路传输数据, 还是愿意使用 FedEx 夜间快递来交付? 解释你的理由。

P25. 假定两台主机 A 和 B 相隔 20 000km, 由一条直接的 $R = 2\text{Mbps}$ 的链路相连。假定跨越该链路的传播速率是 $2.5 \times 10^8 \text{ m/s}$ 。

a. 计算带宽-时延积 $R \cdot t_{\text{prop}}$ 。

b. 考虑从主机 A 到主机 B 发送一个 800 000 比特的文件。假定该文件作为一个大的报文连续发送。在任何给定的时间, 在链路上具有的比特数量最大值是多少?

c. 给出带宽-时延积的一种解释。

d. 在该链路上一个比特的宽度 (以米计) 是多少? 它比一个足球场更长吗?

e. 用传播速率 s 、带宽 R 和链路 m 的长度表示, 推导出一个比特宽度的一般表示式。

P26. 对于习题 P25, 假定我们能够修改 R 。对什么样的 R 值, 一个比特的宽度能与该链路的长度一样长?

P27. 考虑习题 P25, 但此时链路的速率是 $R = 1\text{Gbps}$ 。

a. 计算带宽-时延积 $R \cdot d_{\text{prop}}$ 。

b. 考虑从主机 A 到主机 B 发送一个 800 000 比特的文件。假定该文件作为一个大的报文连续发送。在任何给定的时间, 在链路上具有的比特数量最大值是多少?

c. 在该链路上一个比特的宽度 (以米计) 是多少? $\min\{\text{带宽时延积 } R \cdot t_{\text{prop}}, \text{文件总bits}\}$

P28. 再次考虑习题 P25。

a. 假定连续发送, 发送该文件需要多长时间?

b. 假定现在该文件被划分为 20 个分组, 每个分组包含 40 000 比特。假定每个分组被接收方确认, 确认分组的传输时间可忽略不计。最后, 假定前一个分组被确认后, 发送方才能发送分组。发送该文件需要多长时间?

c. 比较 (a) 和 (b) 的结果。

P29. 假定在同步卫星和它的地球基站之间有一条 10Mbps 的微波链路。每分钟该卫星拍摄一幅数字照片, 并将它发送到基站。假定传播速率是 $2.4 \times 10^8 \text{ m/s}$ 。

a. 该链路的传播时延是多少?

b. 带宽-时延积 $R \cdot d_{\text{prop}}$ 是多少?

c. 若 x 表示该照片的大小。对于这条微波链路, 能够连续传输的 x 最小值是多少?

P30. 考虑 1.5 节中我们在分层讨论中对航空旅行的类比, 随着协议数据单元向协议栈底层流动, 首部在增加。随着旅客和行李移动到航线协议栈底部, 有与上述首部信息等价的概念吗?

P31. 在包括因特网的现代分组交换网中, 源主机将长应用层报文 (如一个图像或音乐文件) 分段为较小的分组并向网络发送。接收方则将这些分组重新装配为初始报文。我们称这个过程为报文分段。图 1-27 显示了一个报文在报文不分段或报文分段情况下的端到端传输。考虑一个长度为 8×10^6 比特的报文, 它在图 1-27 中从源发送到目的地。假定在该图中的每段链路是 2Mbps。忽略传播、排队和处理时延。

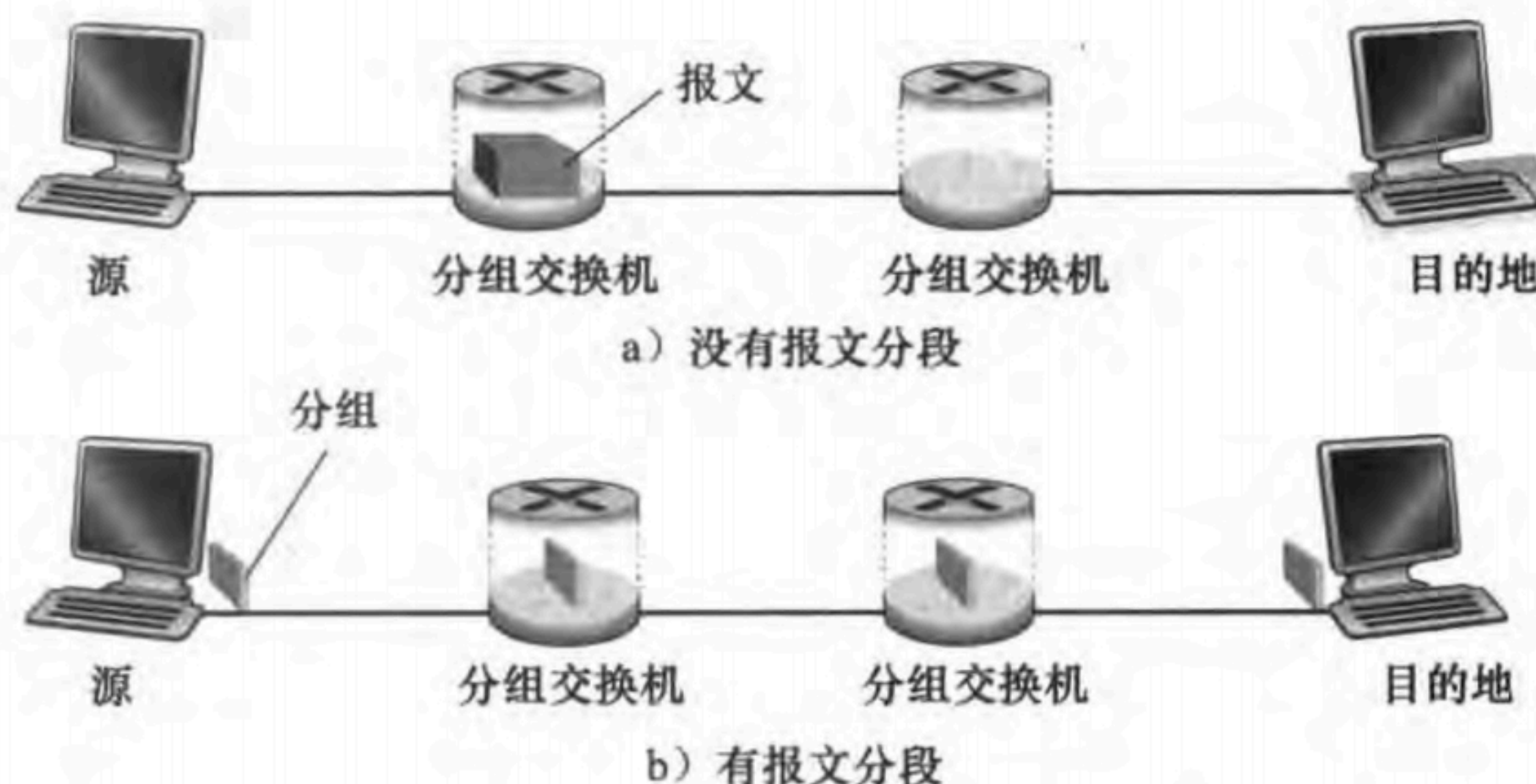


图 1-27 端到端报文传输

- a. 考虑从源到目的地发送该报文且没有报文分段。从源主机到第一台分组交换机移动报文需要多长时间？记住，每台交换机均使用存储转发分组交换，从源主机移动该报文到目的主机需要多长时间？ $3 \cdot \frac{L}{R} = 12s$
- b. 现在假定该报文被分段为 n 个分组，每个分组 L_0 比特长。从源主机移动第一个分组到第一台交换机需要多长时间？从第一台交换机发送第一个分组到第二台交换机，从源主机发送第二个分组到第一台交换机各需要多长时间？什么时候第二个分组能被第一台交换机全部收到？ $3 \cdot \frac{L_0}{R} + (n-1) \cdot \frac{L_0}{R} = 4.01s$
- c. 当进行报文分段时，从源主机向目的主机移动该文件需要多长时间？将该结果与 (a) 的答案进行比较并解释之。
- d. 除了减小时延外，使用报文分段还有什么原因？
- e. 讨论报文分段的缺点。

- P32. 用本书的 Web 网站上的报文分段小 Java 小程序进行实验。该 Java 程序中的时延与前一个习题中的时延相当吗？链路传播时延是怎样影响分组交换（有报文分段）和报文交换的端到端总时延的？
- P33. 考虑从主机 A 到主机 B 发送一个 F 比特的大文件。A 和 B 之间有三段链路（和两台交换机），并且该链路不拥塞（即没有排队时延）。主机 A 将该文件分为每个为 S 比特的报文段，并为每个报文段增加一个 80 比特的首部，形成 $L = 80 + S$ 比特的分组。每条链路的传输速率为 R bps。求出从 A 到 B 移动该文件时延最小的值 S 。忽略传播时延。 $delay = \frac{3L}{R} + (\frac{F}{S} - 1) \cdot \frac{L}{R}$
- P34. Skype 提供了一种服务，使你能用 PC 向普通电话打电话。这意味着语音呼叫必须通过因特网和电话网。讨论这是如何做到的。 $即 delay = \frac{2S}{R} + \frac{80F}{R \cdot S} + \frac{160+F}{R}$



Wireshark 实验

“不闻不若闻之，闻之不若见之，见之不若知之，知之不若行之。”

——中国谚语

一个人通常能够通过以下方法加深对网络协议的理解：观察它们的动作和经常摆弄它们，即观察两个协议实体之间交换的报文序列，钻研协议运行的细节，使协议执行某些动作，观察这些动作及其后果。这能够在仿真环境下或在如因特网这样的真实网络环境下完成。在本书配套 Web 站点上的 Java 小程序采用的是第一种方法。在 Wireshark 实验中，我们将采用后一种方法。你可以在家中或实验室中使用桌面计算机在各种情况下运行网络应用程序。在你的计算机上观察网络协议，它是如何与在因特网别处执行的协议实体交互和交换报文的。因此，你与你的计算机将是这些真实实验的有机组成部分。你将通过动手来观察和学习。

用来观察执行协议实体之间交换的报文的基本工具称为分组嗅探器（packet sniffer）。顾名思义，一个分组嗅探器被动地拷贝（嗅探）由你的计算机发送和接收的报文；它也能显示出这些被捕获报文的各个协议字段的内容。图 1-28 中显示了 Wireshark 分组嗅探器的屏幕快照。Wireshark 是一个运行在 Win-

POP3 从他的邮件服务器访问自己的邮件。讨论该报文是如何从 Alice 主机到 Bob 主机的。要列出在两台主机间移动该报文时所使用的各种应用层协议。

- R17. 将你最近收到的报文首部打印出来。其中有多少 Received: 首部行? 分析该报文的首部行中的每一行。
- R18. 从用户的观点看, POP3 协议中下载并删除模式和下载并保留模式有什么区别吗?
- R19. 一个机构的 Web 服务器和邮件服务器可以有完全相同的主机名别名 (例如, foo.com) 吗? 包含邮件服务器主机名的 RR 有什么样的类型?
- R20. 仔细检查收到的电子邮件, 查找由使用 .edu 电子邮件地址的用户发送的报文首部。从其首部, 能够确定发送该报文的主机的 IP 地址吗? 对于由 Gmail 账号发送的报文做相同的事。

2.5 节

- R21. 在 BitTorrent 中, 假定 Alice 向 Bob 提供一个 30 秒间隔的文件块吞吐量。Bob 将必须进行回报, 在相同的间隔中向 Alice 提供文件块吗? 为什么?
- R22. 考虑一个新对等方 Alice 加入 BitTorrent 而不拥有任何文件块。没有任何块, 因此她没有任何东西可上载, 她无法成为任何其他对等方的前 4 位上载者。那么 Alice 将怎样得到她的第一个文件块呢?
- R23. 覆盖网络是什么? 它包括路由器吗? 在覆盖网络中边是什么?

2.6 节

- R24. CDN 通常采用两种不同的服务器放置方法之一。列举并简单描述它们。
- R25. 除了如时延、丢包和带宽性能等网络相关的考虑外, 设计一种 CDN 服务器选择策略时还有其他重要因素。它们是什么?

2.7 节

- R26. 2.7 节中所描述的 UDP 服务器仅需要一个套接字, 而 TCP 服务器需要两个套接字。为什么? 如果 TCP 服务器支持 n 个并行连接, 每条连接来自不同的客户主机, 那么 TCP 服务器将需要多少个套接字?
- R27. 对于 2.7 节所描述的运行在 TCP 之上的客户-服务器应用程序, 服务器程序为什么必须先于客户程序运行? 对于运行在 UDP 之上的客户-服务器应用程序, 客户程序为什么可以先于服务器程序运行?



习题

P1. 是非判断题。

- a. 假设用户请求由一些文本和 3 幅图像组成的 Web 页面。对于这个页面, 客户将发送一个请求报文并接收 4 个响应报文。✗
- b. 两个不同的 Web 页面 (例如, www.mit.edu/research.html 及 www.mit.edu/students.html) 可以通过同一个持续连接发送。✓
- c. 在浏览器和初始服务器之间使用非持续连接的话, 一个 TCP 报文段是可能携带两个不同的 HTTP 服务请求报文的。✗
- d. 在 HTTP 响应报文中的 Date: 首部指出了该响应中对象最后一次修改的时间。✗
- e. HTTP 响应报文决不会具有空的报文体。

P2. SMS、iMessage 和 WhatsApp 都是智能手机即时通信系统。在因特网上进行一些研究后, 为这些系统分别写一段它们所使用协议的文字。然后撰文解释它们的差异所在。

P3. 考虑一个要获取给定 URL 的 Web 文档的 HTTP 客户。该 HTTP 服务器的 IP 地址开始时并不知道。在这种情况下, 除了 HTTP 外, 还需要什么运输层和应用层协议?
HTTP DNS
↓ ↓
TCP UDP

P4. 考虑当浏览器发送一个 HTTP GET 报文时, 通过 Wireshark 俘获到下列 ASCII 字符串 (即这是一个 HTTP GET 报文的实际内容)。字符 `<cr>` `<lf>` 是回车和换行符 (即下面文本中的斜体字符串 `<cr>` 表示了单个回车符, 该回车符包含在 HTTP 首部中的相应位置)。回答下列问题, 指出你在下面 HTTP GET 报文中找到答案的地方。


```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-
Encoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf><u>Connection:keep-alive<cr><lf><cr><lf>
```

- 由浏览器请求的文档的 URL 是什么? *http://gaia.cs.umass.edu/cs453/index.html*
- 该浏览器运行的是 HTTP 的何种版本? *HTTP 1.1*
- 该浏览器请求的是一条非持续连接还是一条持续连接? *持续*
- 该浏览器所运行的主机的 IP 地址是什么?
- 发起该报文的浏览器的类型是什么? 在一个 HTTP 请求报文中, 为什么需要浏览器类型?

P5. 下面文本中显示的是来自服务器的回答, 以响应上述问题中 HTTP GET 报文。回答下列问题, 指出你在下面报文中找到答案的地方。

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)
<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46
GMT<cr><lf>ETag: "526c3-f22-a88a4c80"<cr><lf>Accept-
Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>
Keep-Alive: timeout=max=100<cr><lf>Connection:
Keep-Alive<cr><lf>Content-Type: text/html; charset=
ISO-8859-1<cr><lf><cr><lf><!doctype html public "-
//w3c//dtd html 4.0transitional//en"><lf><html><lf>
<head><lf> <meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"><lf> <meta
name="GENERATOR" content="Mozilla/4.79 [en] (Windows NT
5.0; U) Netscape]"><lf> <title>CMPSCI 453 / 591 /
NTU-ST550ASpring 2005 homepage</title><lf></head><lf>
<much more document text following here (not shown)>
```

- 服务器能否成功地找到那个文档? *能* 该文档提供回答是什么时间? *2008, 5, 7 12:39:45*
- 该文档最后修改是什么时间? *2005, 12, 10 18:27:46*
- 文档中被返回的字节有多少? *3874 bytes*
- 文档被返回的前 5 个字节是什么? 该服务器同意一条持续连接吗? *同意持续连接*

P6. 获取 HTTP/1.1 规范 (RFC 2616)。回答下面问题:

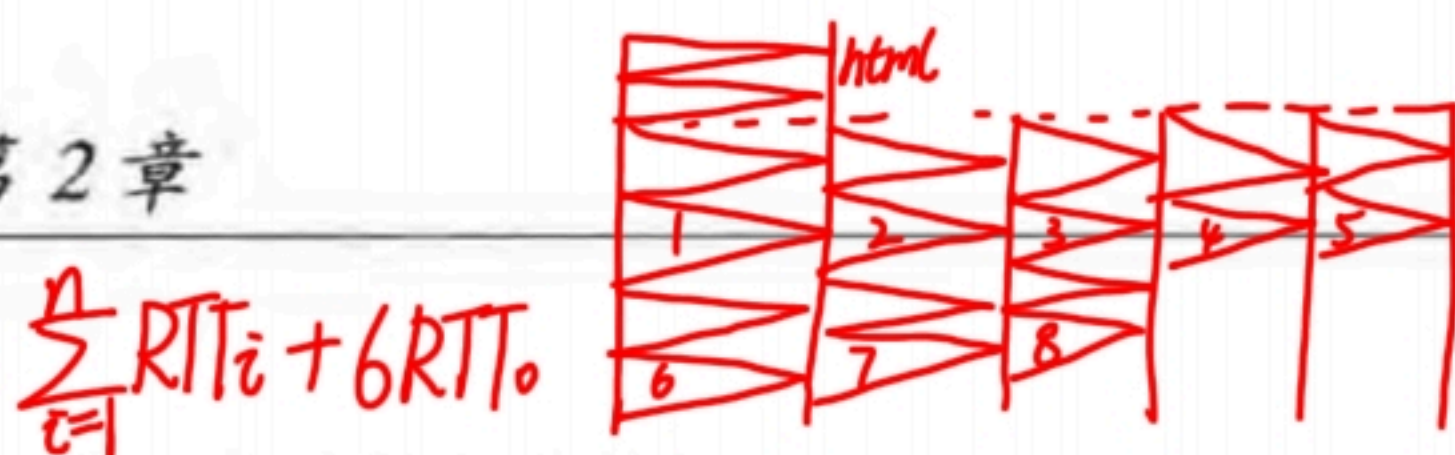
- 解释在客户和服务端之间用于指示关闭持续连接的信令机制。客户、服务器或两者都能发送信令通知连接关闭吗? *Client, SVR均可关闭*
- HTTP 提供了什么加密服务? *无*
- 一个客户能够与一个给定的服务器打开 3 条或更多条并发连接吗? *A single user client: 最多2条并发连接*
- 如果一个服务器或一个客户检测到连接已经空闲一段时间, 该服务器或客户可以关闭两者之间的传输连接。一侧开始关闭连接而另一侧通过该连接传输数据是可能的吗? *请解释。*

P7. 假定你在浏览器中点击一条超链接获得 Web 页面。相关联的 URL 的 IP 地址没有缓存在本地主机上, 因此必须使用 DNS lookup 以获得该 IP 地址。如果主机从 DNS 得到 IP 地址之前已经访问了 n 个 DNS 服务器; 相继产生的 RTT 依次为 $RTT_1, \dots, RTT_n$ 。进一步假定与链路相关的 Web 页面只包含一个对象, 即由少量的 HTML 文本组成。令 RTT_0 表示本地主机和包含对象的服务器之间的 RTT 值。假定该对象传输时间为零, 则从该客户点击该超链接到它接收到该对象需要多长时间? *$\sum_{i=1}^n RTT_i + 2RTT_0$*

P8. 参照习题 P7, 假定在同一服务器上某 HTML 文件引用了 8 个非常小的对象。忽略发送时间, 在下列情况下需要多长时间: *连接建立+获取html*

- 没有并行 TCP 连接的非持续 HTTP。

$$\sum_{i=1}^8 RTT_i + 18 RTT_0$$



b. 配置有 5 个并行连接的非持续 HTTP。

c. 持续 HTTP。 $\sum_{i=1}^5 RTT_i + 10RTT_0$

P9. 考虑图 2-12, 其中有一个机构的网络和因特网相连。假定对象的平均长度为 850 000 比特, 从这个机构网的浏览器到初始服务器的平均请求率是每秒 16 个请求。还假定从接入链路的因特网一侧的路由器转发一个 HTTP 请求开始, 到接收到其响应的平均时间是 3 秒 (参见 2.2.5 节)。将总的平均响应时间建模为平均接入时延 (即从因特网路由器到机构路由器的时延) 和平均因特网时延之和。对于平均接入时延, 使用 $\Delta/(1-\Delta\beta)$, 式中 Δ 是跨越接入链路发送一个对象的平均时间, β 是对象对该接入链路的平均到达率。

a. 求出总的平均响应时间。

b. 现在假定在这个机构 LAN 中安装了一个缓存器。假定命中率为 0.4, 求出总的响应时间。

P10. 考虑一条 10 米短链路, 某发送方经过它能够以 150bps 速率双向传输。假定包含数据的分组是 100 000 比特长, 仅包含控制 (如 ACK 或握手) 的分组是 200 比特长。假定 N 个并行连接每个都获得 $1/N$ 的链路带宽。现在考虑 HTTP 协议, 并且假定每个下载对象是 100Kb 长, 这些初始下载对象包含 10 个来自相同发送方的引用对象。在这种情况下, 经非持续 HTTP 的并行实例的并行下载有意义吗? 现在考虑持续 HTTP。你期待这比非持续的情况有很大增益吗? 评价并解释你的答案。

P11. 考虑在前一个习题中引出的情况。现在假定该链路由 Bob 和 4 个其他用户所共享。Bob 使用非持续 HTTP 的并行实例, 而其他 4 个用户使用无并行下载的非持续 HTTP。

a. Bob 的并行连接能够帮助他更快地得到 Web 页面吗?

b. 如果所有 5 个用户打开 5 个非持续 HTTP 并行实例, 那么 Bob 的并行连接仍将是有益处的吗? 为什么?

P12. 写一个简单的 TCP 程序, 使服务器接收来自客户的行并将其打印在服务器的标准输出上。(可以通过修改本书中的 TCPServer.py 程序实现上述任务。) 编译并执行你的程序。在另一台有浏览器的机器上, 设置浏览器的代理服务器为你正在运行服务器程序的机器, 同时适当地配置端口号。这时你的浏览器向服务器发送 GET 请求报文, 你的服务器应当在其标准输出上显示该报文。使用这个平台来确定你的浏览器是否对本地缓存的对象产生了条件 GET 报文。

P13. SMTP 中的 MAIL FROM 与该邮件报文自身中的 From: 之间有什么不同?

P14. SMTP 是怎样标识一个报文体结束的? HTTP 是怎样做的呢? HTTP 能够使用与 SMTP 标识一个报文体结束相同的方法吗? 试解释。

P15. 阅读用于 SMTP 的 RFC 5321。MTA 代表什么? 考虑下面收到的垃圾邮件 (从一份真实垃圾邮件修改得到)。假定这封垃圾邮件的唯一始作俑者是恶意的, 而其他主机是诚实的, 指出产生了这封垃圾邮件的恶意主机。

```
From - Fri Nov 07 13:41:30 2008
Return-Path: <tennis5@pp33head.com>
Received: from barmail.cs.umass.edu (barmail.cs.umass.edu
[128.119.240.3]) by cs.umass.edu (8.13.1/8.12.6) for
<hg@cs.umass.edu>; Fri, 7 Nov 2008 13:27:10 -0500
Received: from asusus-4b96 (localhost [127.0.0.1]) by
barmail.cs.umass.edu (Spam Firewall) for <hg@cs.umass.edu>; Fri, 7
Nov 2008 13:27:07 -0500 (EST)
Received: from asusus-4b96 ([58.88.21.177]) by barmail.cs.umass.edu
for <hg@cs.umass.edu>; Fri, 07 Nov 2008 13:27:07 -0500 (EST)
Received: from [58.88.21.177] by inbnd55.exchangedddd.com; Sat, 8
Nov 2008 01:27:07 +0700
From: "Jonny" <tennis5@pp33head.com>
To: <hg@cs.umass.edu>

Subject: How to secure your savings
```

P16. 阅读 POP3 的 RFC, 即 RFC 1939。UIDL POP3 命令的目的是什么?

P9. 考虑图 2-12，其中有一个机构的网络和因特网相连。假定对象的平均长度为 850 000 比特，从这个机构网的浏览器到初始服务器的平均请求率是每秒 16 个请求。还假定从接入链路的因特网一侧的路由器转发一个 HTTP 请求开始，到接收到其响应的平均时间是 3 秒（参见 2.2.5 节）。将总的平均响应时间建模为平均接入时延（即从因特网路由器到机构路由器的时延）和平均因特网时延之和。对于平均接入时延，使用 $\Delta/(1-\Delta\beta)$ ，式中 Δ 是跨越接入链路发送一个对象的平均时间， β 是对象对该接入链路的平均到达率。

a. 求出总的平均响应时间。

b. 现在假定在这个机构 LAN 中安装了一个缓存器。假定命中率为 0.4，求出总的响应时间。

a. 平均因特网时延: (RTT) $3s$

平均接入时延: 跨越一个链路 $d_{trans} = \frac{L}{R} s$

对象对链路平均到达率 $\beta = 16 \text{ request/s}$

总平均响应时间 $3s + \frac{\Delta}{1-\Delta\beta} = 3.6s$

b. $\beta' = 16 \times (1-0.4) \text{ request/s}$

对 cache 缺失部分, $0.6 \times (3s + \frac{\Delta}{1-\Delta\beta'}) s$

cache 中找到, $0.4 \times 0s$

P10. 考虑一条 10 米短链路，某发送方经过它能够以 150bps 速率双向传输。假定包含数据的分组是 100 000 比特长，仅包含控制（如 ACK 或握手）的分组是 200 比特长。假定 N 个并行连接每个都获得 $1/N$ 的链路带宽。现在考虑 HTTP 协议，并且假定每个下载对象是 100Kb 长，这些初始下载对象包含 10 个来自相同发送方的引用对象。在这种情况下，经非持续 HTTP 的并行实例的并行下载有意义吗？现在考虑持续 HTTP。你期待这比非持续的情况有很大增益吗？评价并解释你的答案。

对数据分组, $d_{trans_1} = \frac{10^5 \text{ bit}}{150 \text{ bit/s}}$ 、 d_{prop}

对控制分组, $d_{trans_2} = \frac{200 \text{ bit}}{150 \text{ bit/s}}$ 、 d_{prop}

非持续、非并行: $d = 11 \times (3d_{trans_2} + d_{trans_1})$

非持续、并行:



$d = 2 \times (3d_{trans_2} + d_{trans_1})$

持续、非并行: $d = (3d_{trans_2} + d_{trans_1}) + 10 \times (d_{trans_2} + d_{trans_1})$

- a. 第一个报文段中有多少数据?
- b. 假设第一个报文段丢失而第二个报文段到达主机 B。那么在主机 B 发往主机 A 的确认报文中, 确认号应该是多少?

R16. 考虑在 3.5 节中讨论的 Telnet 的例子。在用户键入字符 C 数秒之后, 用户又键入字符 R。那么在用户键入字符 R 之后, 总共发送了多少个报文段, 这些报文段中的序号和确认字段应该填入什么?

3.7 节

- R17. 假设两条 TCP 连接存在于一个带宽为 R bps 的瓶颈链路上。它们都要发送一个很大的文件 (以相同方向经过瓶颈链路), 并且两者是同时开始发送文件。那么 TCP 将为每条连接分配什么样的传输速率?
- R18. 是非判断题。考虑 TCP 的拥塞控制。当发送方定时器超时, 其 $ssthresh$ 的值将被设置为原来值的一半。
- R19. 在 3.7 节的“TCP 分岔”讨论中, 对于 TCP 分岔的响应时间, 断言大约是 $4 * RTT_{FE} + RTT_{BE} + \text{处理时间}$ 。评价该断言。

习题

P1. 假设客户 A 向服务器 S 发起一个 Telnet 会话。与此同时, 客户 B 也向服务器 S 发起一个 Telnet 会话。给出下面报文段的源端口号和目的端口号:

- a. 从 A 向 S 发送的报文段。
- b. 从 B 向 S 发送的报文段。
- c. 从 S 向 A 发送的报文段。
- d. 从 S 向 B 发送的报文段。
- e. 如果 A 和 B 是不同的主机, 那么从 A 向 S 发送的报文段的源端口号是否可能与从 B 向 S 发送的报文段的源端口号相同? ☒
- f. 如果它们是同一台主机, 情况会怎么样? ☒ 源端口号不同

P2. 考虑图 3-5。从服务器返回客户进程的报文流中的源端口号和目的端口号是多少? 在承载运输层报文段的网络层数据报中, IP 地址是多少? **80 26145/26145/7532**

P3. UDP 和 TCP 使用反码来计算它们的检验和。假设你有下面 3 个 8 比特字节: 01010011, 01100110, 01110100。这些 8 比特字节和的反码是多少? (注意到尽管 UDP 和 TCP 使用 16 比特的字来计算检验和, 但对于这个问题, 你应该考虑 8 比特和。) 写出所有工作过程。UDP 为什么要用该和的反码, 即为什么不直接使用该和呢? 使用该反码方案, 接收方如何检测出差错? 1 比特的差错将可能检测不出来吗? 2 比特的差错呢? ☒ **将3个数和checksum相加, 为全0则不出错** **11010001** ☒

- P4. a. 假定你有下列 2 个字节: 01011100 和 01100101。这 2 个字节之和的反码是什么? **00111110**
- b. 假定你有下列 2 个字节: 11011010 和 01100101。这 2 个字节之和的反码是什么? **01101101**
- c. 对于 (a) 中的字节, 给出一个例子, 使得这 2 个字节中的每一个都在一个比特反转时, 其反码不会改变。

P5. 假定某 UDP 接收方对接收到的 UDP 报文段计算因特网检验和, 并发现它与承载在检验和字段中的值相匹配。该接收方能够绝对确信没有出现过比特差错吗? ☒ **不能** 解释之。

P6. 考虑我们改正协议 rdt2.1 的动机。试说明如图 3-57 所示的接收方与如图 3-11 所示的发送方运行时, 接收方可能会引起发送方和接收方进入死锁状态, 即双方都在等待不可能发生的事件。

P7. 在 rdt3.0 协议中, 从接收方向发送方流动的 ACK 分组没有序号 (尽管它们具有 ACK 字段, 该字段包括了它们正在确认的分组的序号)。为什么这些 ACK 分组不需要序号呢?

P8. 画出协议 rdt3.0 中接收方的 FSM。

P9. 当数据分组和确认分组发生篡改时, 给出 rdt3.0 协议运行的轨迹。你画的轨迹应当类似于图 3-16 中所用的图。

P10. 考虑一个能够丢失分组但其最大时延已知的信道。修改协议 rdt2.1, 以包括发送方超时和重传机制。

非正式地论证：为什么你的协议能够通过该信道正确通信？



图 3-57 协议 rdt2.1 的一个不正确的接收方

- P11. 考虑在图 3-14 中的 rdt2.2 接收方，在状态“等待来自下层的 0”和状态“等待来自下层的 1”中的自转换（即从某状态转换回自身）中生成一个新分组： $\text{sndpk} = \text{make_pkt}(\text{ACK}, 1, \text{checksum})$ 和 $\text{sndpk} = \text{make_pkt}(\text{ACK}, 0, \text{checksum})$ 。如果这个动作从状态“等待来自下层的 1”中的自转换中删除，该协议将正确工作吗？评估你的答案。在状态“等待来自下层的 0”中的自转换中删除这个事件将会怎样？[提示：在后一种情况下，考虑如果第一个发送方到接收方的分组损坏的话，将会发生什么情况？]
- P12. rdt3.0 协议的发送方直接忽略（即不采取任何动作）接收到的所有出现差错和确认分组的确认号（acknum）字段中的值有差错的分组。假设在这种情况下，rdt3.0 只是重传当前的数据分组，该协议是否还能正常运行？（提示：考虑在下列情况下会发生什么情况：仅有一个比特差错时；报文没有丢失但能出现定时器过早超时。考虑到当 n 趋于无穷时，第 n 个分组将被发送多少次。）
- P13. 考虑 rdt3.0 协议。如果发送方和接收方的网络连接能够对报文重排序（即在发送方和接收方之间的媒体上传播的两个报文段能重新排序），那么比特交替协议将不能正确工作（确信你清楚地理解这时它不能正确工作的原因），试画图说明之。画图时把发送方放在左边，接收方放在右边，使时间轴朝下，标出交换的数据报文（D）和确认报文（A）。要标明与任何数据和确认报文段相关的序号。
- P14. 考虑一种仅使用否定确认的可靠数据传输协议。假定发送方只是偶尔发送数据。只用 NAK 的协议是否会比使用 ACK 的协议更好？为什么？现在我们假设发送方要发送大量的数据，并且该端到端连接很少丢包。在第二种情况下，只用 NAK 的协议是否会比使用 ACK 的协议更好？为什么？
- P15. 考虑显示在图 3-17 中的网络跨越国家的例子。窗口长度设置成多少时，才能使该信道的利用率超过 90%？假设分组的长度为 1500 字节（包括首部字段和数据）。

$$\text{发送方实际可发送比特进信道} / \text{单包发送时间} = \frac{L}{R} \cdot \frac{1}{RTT + t_o} \geq 90\%$$
- P16. 假设某应用使用 rdt3.0 作为其运输层协议。因为停等协议具有非常低的信道利用率（显示在网络跨越国家的例子中），该应用程序的设计者让接收方持续回送许多（大于 2）交替的 ACK 0 和 ACK 1，即使对应的数据未到达接收方。这个应用程序设计将能增加信道利用率吗？为什么？该方法存在某种潜在的问题吗？试解释之。
- P17. 考虑两个网络实体 A 和 B，它们由一条完善的双向信道所连接（即任何发送的报文将正确地收到；

信道将不会损坏、丢失或重排序分组)。A 和 B 将以交互的方式彼此交付报文：首先，A 必须向 B 交付一个报文，B 然后必须向 A 交付一个报文，接下来 A 必须向 B 交付一个报文，等等。如果一个实体处于它不试图向另一侧交付报文的状态，将存在一个来自上层的类似于 `rdt_send(data)` 调用的事件，它试图向下传送数据以向另一侧传输，来自上层的该调用能够直接忽略对于 `rdt_unable_to_send(data)` 调用，这通知较高层当前不能够发送数据。[注意：做出这种简化的假设，使你不必担心缓存数据。]

对该协议画出 FSM 说明（一个 FSM 用于 A，一个 FSM 用于 B）。注意你不必担心这里的可靠性机制，该问题的要点在于创建反映这两个实体的同步行为的 FSM 说明。应当使用与图 3-9 中协议 `rdt1.0` 有相同含义的下列事件和动作：`rdt_send(data)`，`packet = make_pkt(data)`，`udt_send(data)`，`rdt_rcv(packet)`，`extract(packet, data)`，`deliver_data(data)`。保证你的协议反映了 A 和 B 之间发送的严格交替。还要保证在你的 FSM 描述中指出 A 和 B 的初始状态。

- P18. 在 3.4.4 节我们学习的一般性 SR 协议中，只要报文可用（如果报文在窗口中），发送方就会不等待确认而传输报文。假设现在我们要求一个 SR 协议，一次发出一对报文，而且只有在知道第一对报文中的一个报文都正确到达后才发送第二对报文。

假设该信道中可能会丢失报文，但报文不会发生损坏和失序。试为报文的单向可靠传输而设计一个差错控制协议。画出发送方和接收方的 FSM 描述。描述在发送方和接收方之间两个方向发送的报文格式。如果你使用了不同于 3.4 节（例如 `udt_send()`、`start_timer()`、`rdt_rcv()` 等）中的任何其他过程调用，详细地阐述这些动作。举例说明（用发送方和接收方的时序踪迹图）你的协议是如何恢复报文丢失的。*Window size 为一对数据报 $\Rightarrow$ seq num 为 0, 1, 2, 3*

- P19. 考虑一种情况，主机 A 想同时向主机 B 和主机 C 发送分组。A 与 B 和 C 是经过广播信道连接的，即由 A 发送的分组通过该信道传送到 B 和 C。假设连接 A、B 和 C 的这个广播信道具有独立的报文丢失和损坏特性（例如，从 A 发出的报文可能被 B 正确接收，但没有被 C 正确接收）。设计一个类似于停等协议的差错控制协议，用于从 A 可靠地传输分组到 B 和 C。该协议使得 A 直到得知 B 和 C 已经正确接收到当前报文，才获取上层交付的新数据。给出 A 和 C 的 FSM 描述。（提示：B 的 FSM 大体上应当与 C 的相同。）同时，给出所使用的报文格式的描述。
- P20. 考虑一种主机 A 和主机 B 要向主机 C 发送报文的情况。主机 A 和 C 通过一条报文能够丢失和损坏（但不重排序）的信道相连接。主机 B 和 C 由另一条（与连接 A 和 C 的信道独立）具有相同性质的信道连接。在主机 C 上的运输层，在向上层交付来自主机 A 和 B 的报文时应当交替进行（即它应当首先交付来自 A 的分组中的数据，然后是来自 B 的分组中的数据，等等）。设计一个类似于停等协议的差错控制协议，以可靠地向 C 传输来自 A 和 B 的分组，同时以前面描述的方式在 C 处交替地交付。给出 A 和 C 的 FSM 描述。（提示：B 的 FSM 大体上应当与 A 的相同。）同时，给出所使用的报文格式的描述。
- P21. 假定我们有两个网络实体 A 和 B。B 有一些数据报文要通过下列规则传给 A。当 A 从其上层得到一个请求，就从 B 获取下一个数据（D）报文。A 必须通过 A—B 信道向 B 发送一个请求（R）报文。仅当 B 收到一个 R 报文后，它才会通过 B—A 信道向 A 发送一个数据（D）报文。A 应当准确地将每份 D 报文的副本交付给上层。R 报文可能会在 A—B 信道中丢失（但不会损坏）；D 报文一旦发出总是能够正确交付。两个信道的时延未知且是变化的。

设计一个协议（给出 FSM 描述），它能够综合适当的机制，以补偿会丢包的 A—B 信道，并且实现在 A 实体中向上层传递报文。只采用绝对必要的机制。

- P22. 考虑一个 GBN 协议，其发送方窗口为 4，序号范围为 1024。假设在时刻 t ，接收方期待的下一个有序分组的序号是 k 。假设媒体不会对报文重新排序。回答以下问题：
- 在 t 时刻，发送方窗口内的报文序号可能是多少？论证你的回答。
 - 在 t 时刻，在当前传播回发送方的所有可能报文中，ACK 字段的所有可能值是多少？论证你的回答。

- ★ P23. 考虑 GBN 协议和 SR 协议。假设序号空间的长度为 k ，那么为了避免出现图 3-27 中的问题，对于这

P22. 考虑一个 GBN 协议，其发送方窗口为 4，序号范围为 1024。假设在时刻 t ，接收方期待的下一个有序分组的序号是 k 。假设媒体不会对报文重新排序。回答以下问题：

- 在 t 时刻，发送方窗口内的报文序号可能是多少？论证你的回答。
- 在 t 时刻，在当前传播回发送方的所有可能报文中，ACK 字段的所有可能值是多少？论证你的回答。

a. 若 rcvr 的 ack 被 sender 收到， $k+1$ 号包被确认

$$\Rightarrow \text{sender window} = [k, k+3]$$

若 rcvr 的 ack 未被收到，

rcvr: $k-4 \quad k-3 \quad k-2 \quad k-1 \quad \boxed{k} \quad \dots$

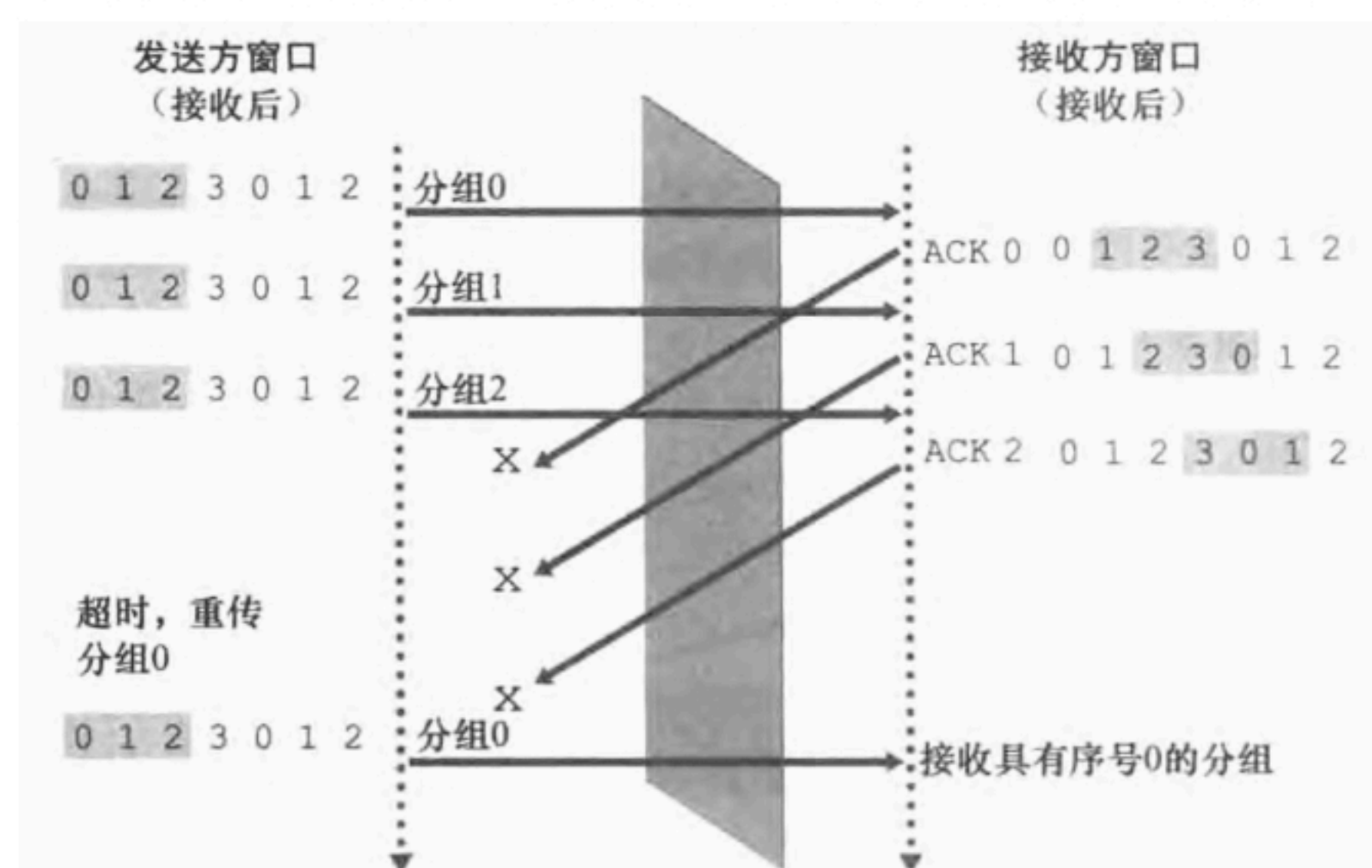
sender: $\boxed{k-4 \quad k-3 \quad k-2 \quad k-1} \quad k$

$$\Rightarrow \text{sender window} = [k-4, k-1]$$

综上， $[k-4, k+3]$

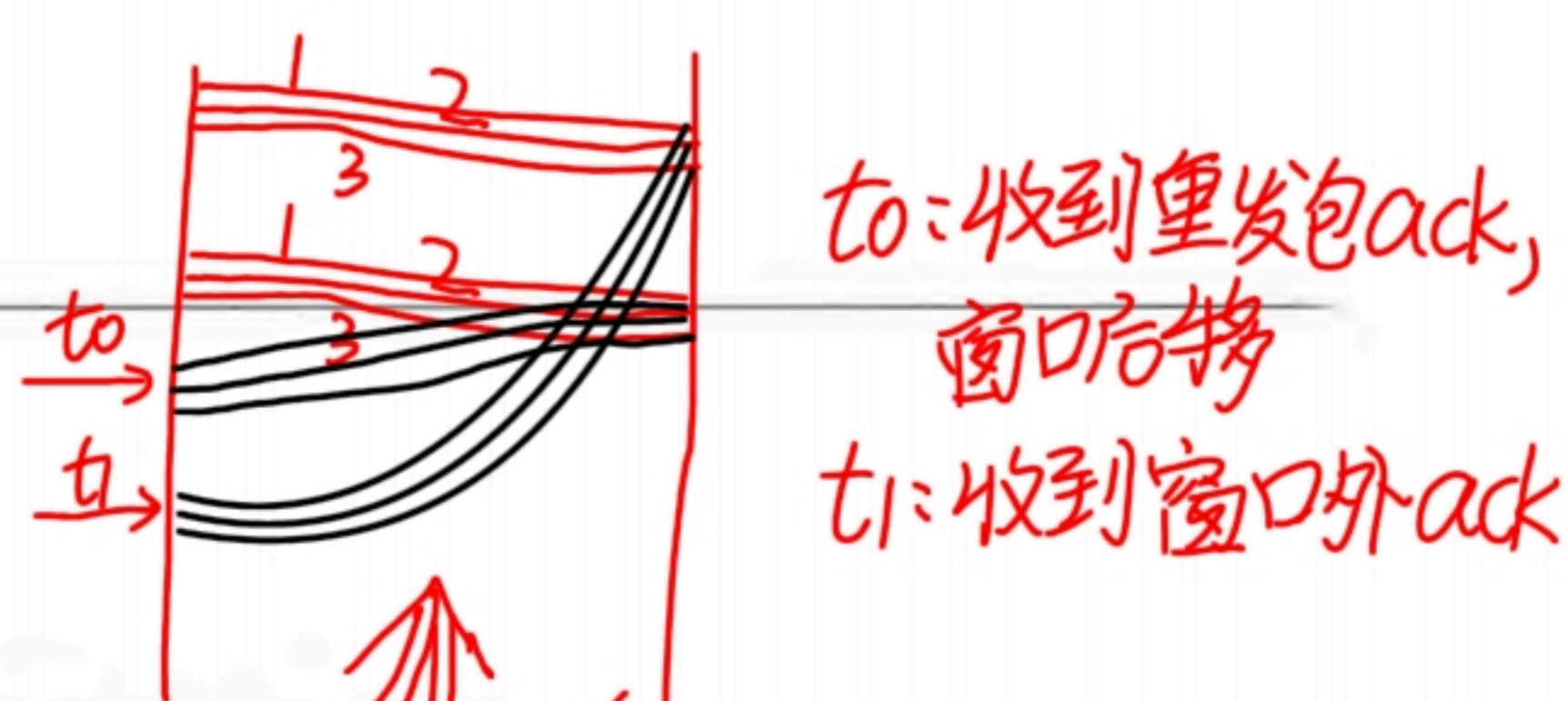
b. ack seq: $[k-4, k-1]$

P23. 考虑 GBN 协议和 SR 协议。假设序号空间的长度为 k ，那么为了避免出现图 3-27 中的问题，对于这两种协议中的每一种，允许的发送方窗口最大为多少？



对 SR，接收方窗口为 $[m, m+W-1]$ ，
若 send 未收到任何 ack，则窗口为 $[m-W, m-1]$ ，为不出现窗口序号重叠（避免视重发包为新包） $\Rightarrow$ 序号可覆盖 2 窗口，不重复 $\Rightarrow (m+W-1) - (m-W) + 1 \leq k$
即 $k \geq 2W$

对 GBN，发送窗口 $[m-W, m-1]$ ，接收窗口 $[m]$ (大小为 1) $\Rightarrow$ 序号可覆盖 2 窗口，不重复 $\Rightarrow m - (m-W) + 1 \leq k$ ，即 $k \geq W+1$



两种协议中的每一种，允许的发送方窗口最大为多少？

P24. 对下面的问题判断是非，并简要地证实你的回答：

- 对于 SR 协议，发送方可能会收到落在其当前窗口之外的分组的 ACK。✓
- 对于 GBN 协议，发送方可能会收到落在其当前窗口之外的分组的 ACK。✓
- 当发送方和接收方窗口长度都为 1 时，比特交替协议与 SR 协议相同。
- 当发送方和接收方窗口长度都为 1 时，比特交替协议与 GBN 协议相同。

P25. 我们曾经说过，应用程序可能选择 UDP 作为运输协议，因为 UDP 提供了（比 TCP）更好的应用层控制，以决定在报文段中发送什么数据和发送时机。

- 应用程序为什么对在报文段中发送什么数据有更多的控制？UDP 不会切包、重组
- 应用程序为什么对何时发送报文段有更多的控制？没有 flow/congestion control 造成的 delay

P26. 考虑从主机 A 向主机 B 传输 1 字节的大文件，假设 MSS 为 536 字节。

- 为了使得 TCP 序号 ~~对每个字节编号~~ 的最大值是多少？前面讲过 TCP 的序号字段为 4 字节。
- 对于你在 (a) 中得到的 L ，求出传输此文件要用多长时间？假定运输层、网络层和数据链路层首部总共为 66 字节，并加在每个报文段上，然后经 155Mbps 链路发送得到的分组。忽略流量控制和拥塞控制，使主机 A 能够一个接一个和连续不断地发送这些报文段。 $\frac{MSS+66}{R} \times \frac{256}{MSS}$

P27. 主机 A 和 B 经一条 TCP 连接通信，并且主机 B 已经收到了来自 A 的最长为 126 字节的所有字节。假定主机 A 随后向主机 B 发送两个紧接着的报文段。第一个和第二个报文段分别包含了 80 字节和 40 字节的数据。在第一个报文段中，序号是 127，源端口号是 302，目的地端口号是 80。无论何时主机 B 接收到来自主机 A 的报文段，它都会发送确认。

- 在从主机 A 发往 B 的第二个报文段中，序号 ~~207~~、源端口号 ~~302~~ 和目的端口号 ~~80~~ 各是什么？
- 如果第一个报文段在第二个报文段之前到达，在第一个到达报文段的确认中，确认号 ~~207~~、源端口号 ~~302~~ 和目的端口号 ~~80~~ 各是什么？
- 如果第二个报文段在第一个报文段之前到达，在第一个到达报文段的确认中，确认号是什么？~~127~~
- 假定由 A 发送的两个报文段按序到达 B。第一个确认丢失了而第二个确认在第一个超时间间隔之后到达。画出时序图，显示这些报文段和发送的所有其他报文段和确认。（假设没有其他分组丢失。）对于图上每个报文段，标出序号和数据的字节数量；对于你增加的每个应答，标出确认号。

P28. 主机 A 和 B 直接经一条 100Mbps 链路连接。在这两台主机之间有一条 TCP 连接。主机 A 经这条连接向主机 B 发送一个大文件。主机 A 能够向它的 TCP 套接字以高达 120Mbps 的速率发送应用数据，而主机 B 能够以最大 50Mbps 的速率从它的 TCP 接收缓存中读出数据。描述 TCP 流量控制的影响。

P29. 在 3.5.6 节中讨论了 SYN cookie。

- 服务器在 SYNACK 中使用一个特殊的初始序号，这为什么是必要的？避免 SYN FLOOD 攻击
- 假定某攻击者得知了一台目标主机使用了 SYN cookie。该攻击者能够通过直接向目标发送一个 ACK 分组创建半开或全开连接吗？为什么？
- 假设某攻击者收集了由服务器发送的大量初始序号。该攻击者通过发送具有初始序号的 ACK，能够引起服务器产生许多全开连接吗？为什么？

P30. 考虑在 3.6.1 节中显示在第二种情况下的网络。假设发送主机 A 和 B 具有某些固定的超时值。

- 证明增加路由器有限缓存的长度可能减小吞吐量 (λ_{out})。
- 现在假设两台主机基于路由器的缓存时延，动态地调整它们的超时值（像 TCP 所做的那样）。增加缓存长度将有助于增加吞吐量吗？为什么？

P31. 假设测量的 5 个 SampleRTT 值（参见 3.5.3 节）是 106ms、120ms、140ms、90ms 和 115ms。在获得了每个 SampleRTT 值后计算 EstimatedRTT，使用 $\alpha = 0.125$ 并且假设在刚获得前 5 个样本之后 EstimatedRTT 的值为 100ms。在获得每个样本之后，也计算 DevRTT，假设 $\beta = 0.25$ ，并且假设在刚获得前 5 个样本之后 DevRTT 的值为 5ms。最后，在获得这些样本之后计算 TCP TimeoutInterval。

P32. 考虑 TCP 估计 RTT 的过程。假设 $\alpha = 0.1$ ，令 SampleRTT_1 设置为最新样本 RTT，令 SampleRTT_2 设置为下一个最新样本 RTT，等等。

- a. 对于一个给定的 TCP 连接, 假定 4 个确认报文相继到达, 带有 4 个对应的 RTT 值: SampleRTT_4 、 SampleRTT_3 、 SampleRTT_2 和 SampleRTT_1 。根据这 4 个样本 RTT 表示 EstimatedRTT 。
- b. 将你得到的公式一般化到 n 个 RTT 样本的情况。
- c. 对于在 (b) 中得到的公式, 令 n 趋于无穷。试说明为什么这个平均过程被称为指数移动平均。
- P33. 在 3.5.3 节中, 我们讨论了 TCP 的往返时间的估计。TCP 避免测量重传报文段的 SampleRTT , 对此你有什么看法?
- P34. 3.5.4 节中的变量 SendBase 和 3.5.5 节中的变量 LastByteRcvd 之间有什么关系?
- P35. 3.5.5 节中的变量 LastByteRcvd 和 3.5.4 节中的变量 y 之间有什么关系?
- P36. 在 3.5.4 节中, 我们看到 TCP 直到收到 3 个冗余 ACK 才执行快速重传。你对 TCP 设计者没有选择在收到对报文段的第一个冗余 ACK 后就快速重传有何看法?
- P37. 比较 GBN、SR 和 TCP (无延时的 ACK)。假设对所有 3 个协议的超时值足够长, 使得 5 个连续的数据报文段及其对应的 ACK 能够分别由接收主机 (主机 B) 和发送主机 (主机 A) 收到 (如果在信道中无丢失)。假设主机 A 向主机 B 发送 5 个数据报文段, 并且第二个报文段 (从 A 发送) 丢失。最后, 所有 5 个数据报文段已经被主机 B 正确接收。
- a. 主机 A 总共发送了多少报文段和主机 B 总共发送了多少 ACK? 它们的序号是什么? 对所有 3 个协议回答这个问题。
- b. 如果对所有 3 个协议超时值比 5RTT 长得多, 则哪个协议在最短的时间间隔中成功地交付所有 5 个数据报文段?
- P38. 在图 3-52 中的 TCP 描述中, 阈值 ssthresh 的值在几个地方被设置为 $\text{ssthresh} = \text{cwnd}/2$, 并且当出现一个丢包事件时, ssthresh 的值被设置为窗口长度的一半。当出现丢包事件时, 发送方发送的速率, 每个 RTT 必须大约等于 cwnd 报文段吗? 解释你的答案。如果你的回答是没有, 你能建议一种不同的方式, 进行 ssthresh 设置吗? $\checkmark$ 发送速率 $\sim \frac{\text{cwnd}}{\text{RTT}}$
- P39. 考虑图 3-46b。如果 λ'_{in} 增加超过了 $R/2$, λ_{out} 能够增加超过 $R/3$ 吗? 试解释之。现在考虑图 3-46c。假定一个分组从路由器到接收方平均转发两次的话, 如果 λ'_{in} 增加超过 $R/2$, λ_{out} 能够增加超过 $R/4$ 吗? 试解释之。
- P40. 考虑图 3-58。假设 TCP Reno 是一个经历如上所示行为的协议, 回答下列问题。在各种情况中, 简要地论证你的回答。[1,6], [23,26]
- a. 指出 TCP 慢启动运行时的时间间隔。[6,16], [17,22]
- b. 指出 TCP 拥塞避免运行时的时间间隔。
- c. 在第 16 个传输轮回之后, 报文段的丢失是根据 3 个冗余 ACK 还是根据超时检测出来的? 3个冗余ACK
- d. 在第 22 个传输轮回之后, 报文段的丢失是根据 3 个冗余 ACK 还是根据超时检测出来的? 超时
- e. 在第 1 个传输轮回里, ssthresh 的初始值设置为多少? 32
- f. 在第 18 个传输轮回里, ssthresh 的值设置为多少? 21
- g. 在第 24 个传输轮回里, ssthresh 的值设置为多少? $\lfloor \frac{29}{2} \rfloor = 14$
- h. 在哪个传输轮回内发送第 70 个报文段? 7
- i. 假定在第 26 个传输轮回后, 通过收到 3 个冗余 ACK 检测出有分组丢失, 拥塞的窗口长度和 ssthresh 的值应当是多少? 4
- j. 假定使用 TCP Tahoe (而不是 TCP Reno), 并假定在第 16 个传输轮回收到 3 个冗余 ACK。在第 19 个传输轮回, ssthresh 和拥塞窗口长度是什么? 21, 1

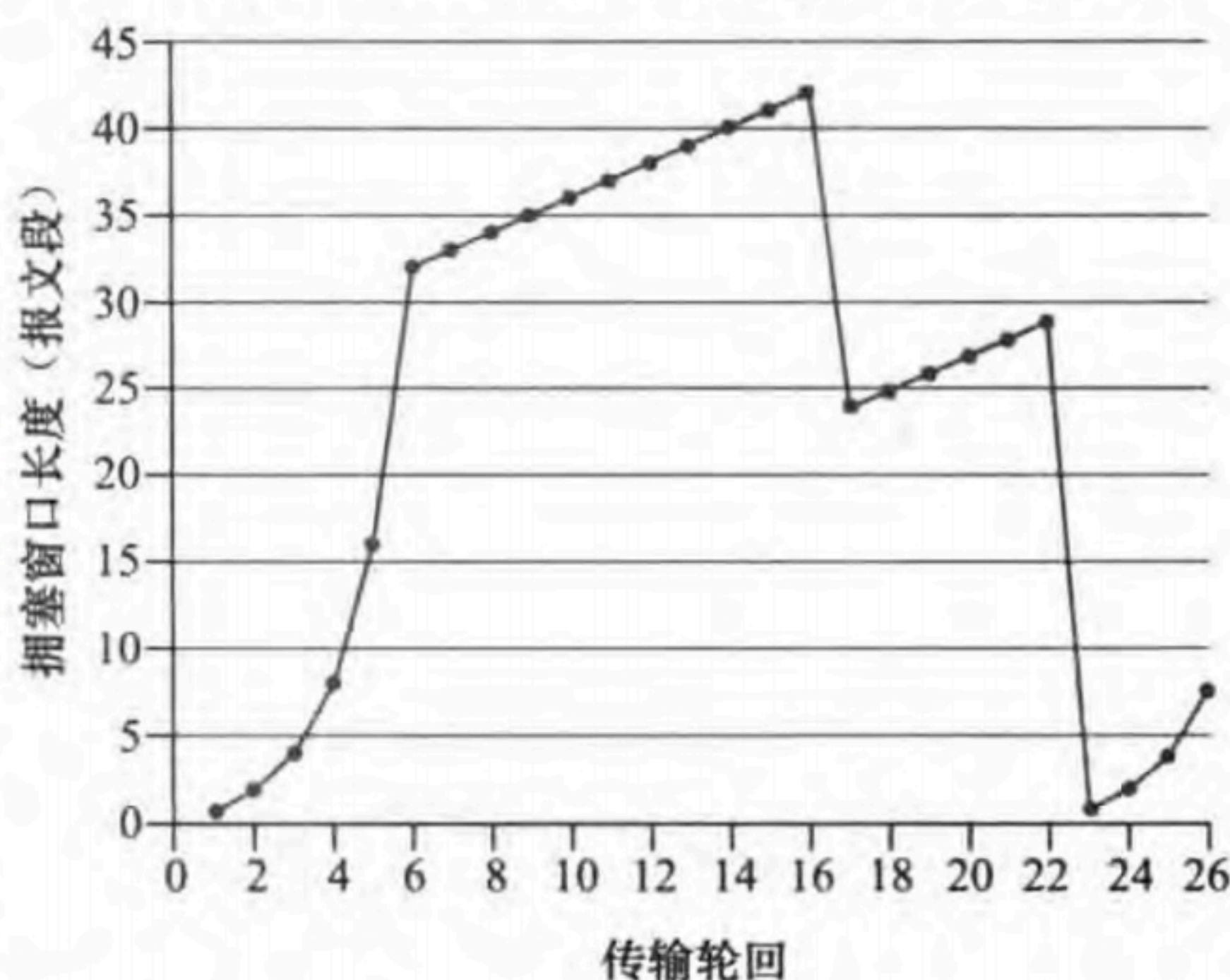
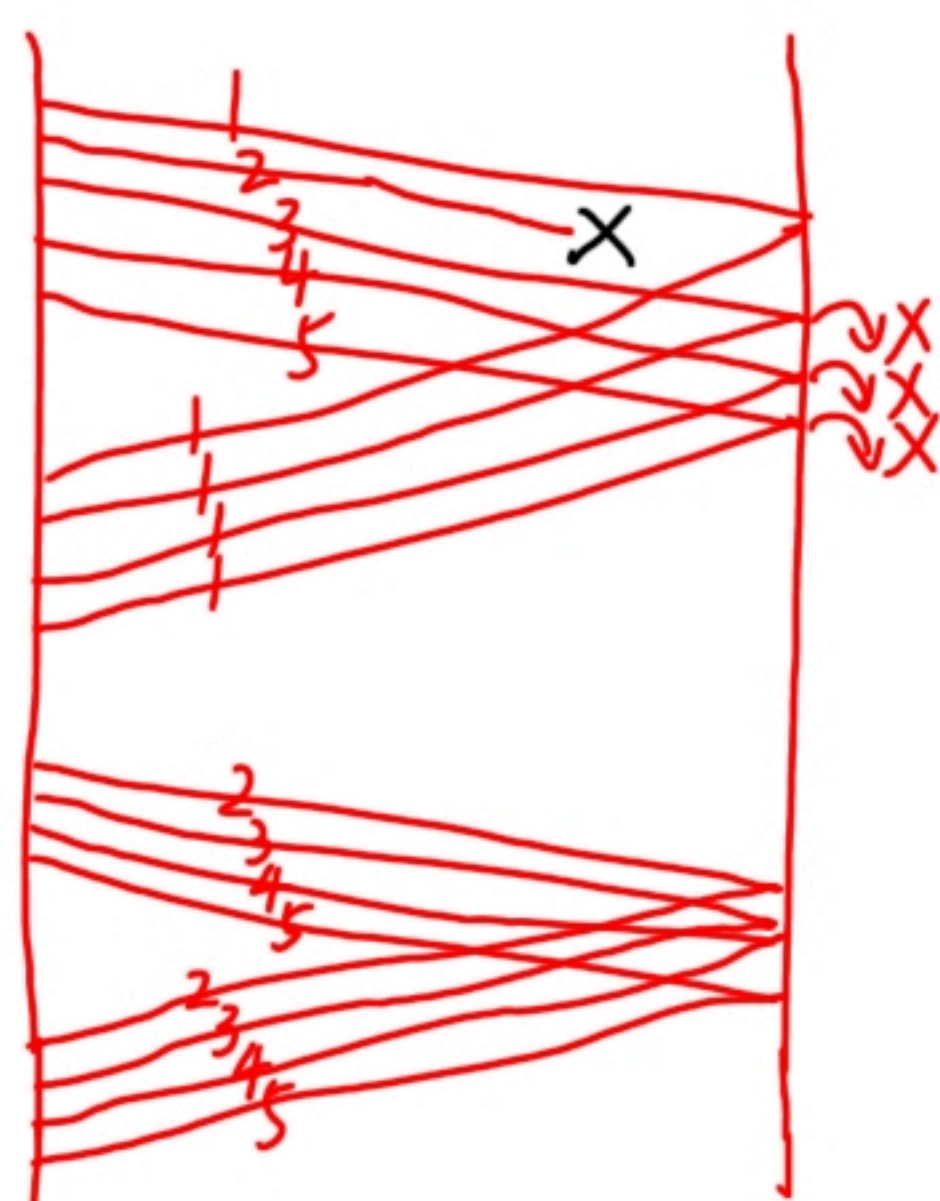


图 3-58 TCP 窗口长度作为时间的函数

P37. 比较 GBN、SR 和 TCP (无延时的 ACK)。假设对所有 3 个协议的超时值足够长, 使得 5 个连续的数据报文段及其对应的 ACK 能够分别由接收主机 (主机 B) 和发送主机 (主机 A) 收到 (如果在信道中无丢失)。假设主机 A 向主机 B 发送 5 个数据报文段, 并且第二个报文段 (从 A 发送) 丢失。最后, 所有 5 个数据报文段已经被主机 B 正确接收。

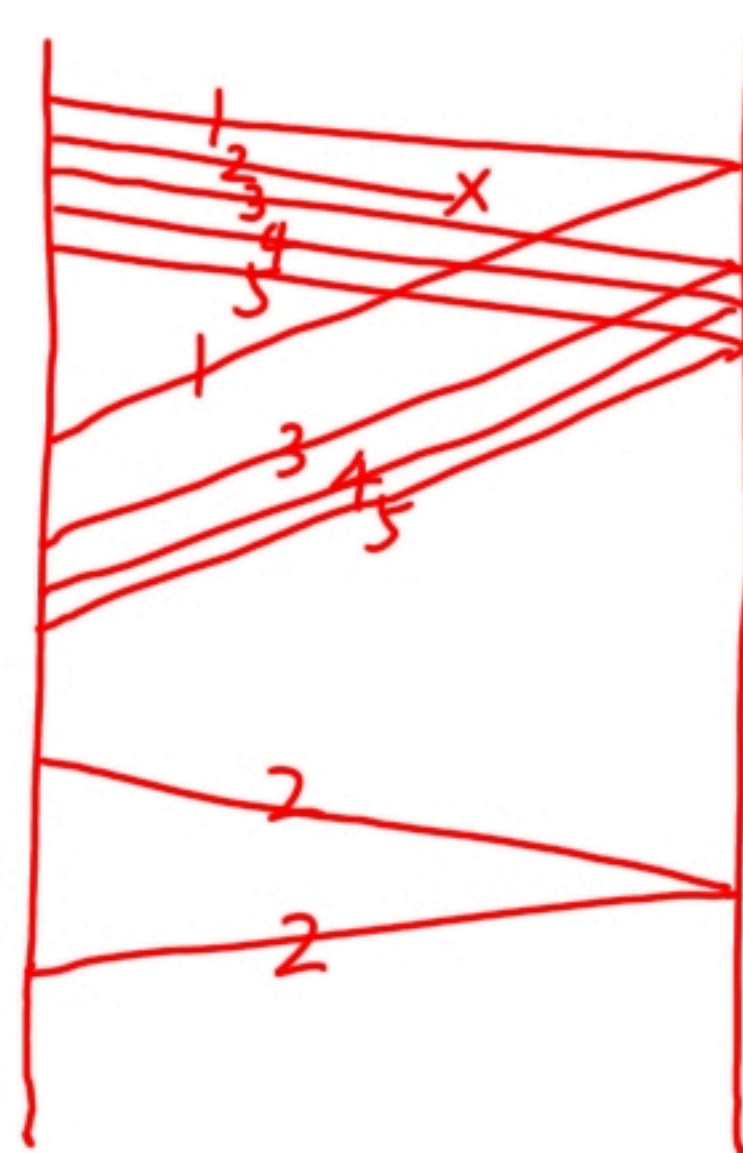
- 主机 A 总共发送了多少报文段和主机 B 总共发送了多少 ACK? 它们的序号是什么? 对所有 3 个协议回答这个问题。
- 如果对所有 3 个协议超时值比 5RTT 长得多, 则哪个协议在最短的时间间隔中成功地交付所有 5 个数据报文段?

a. 对 GBN,



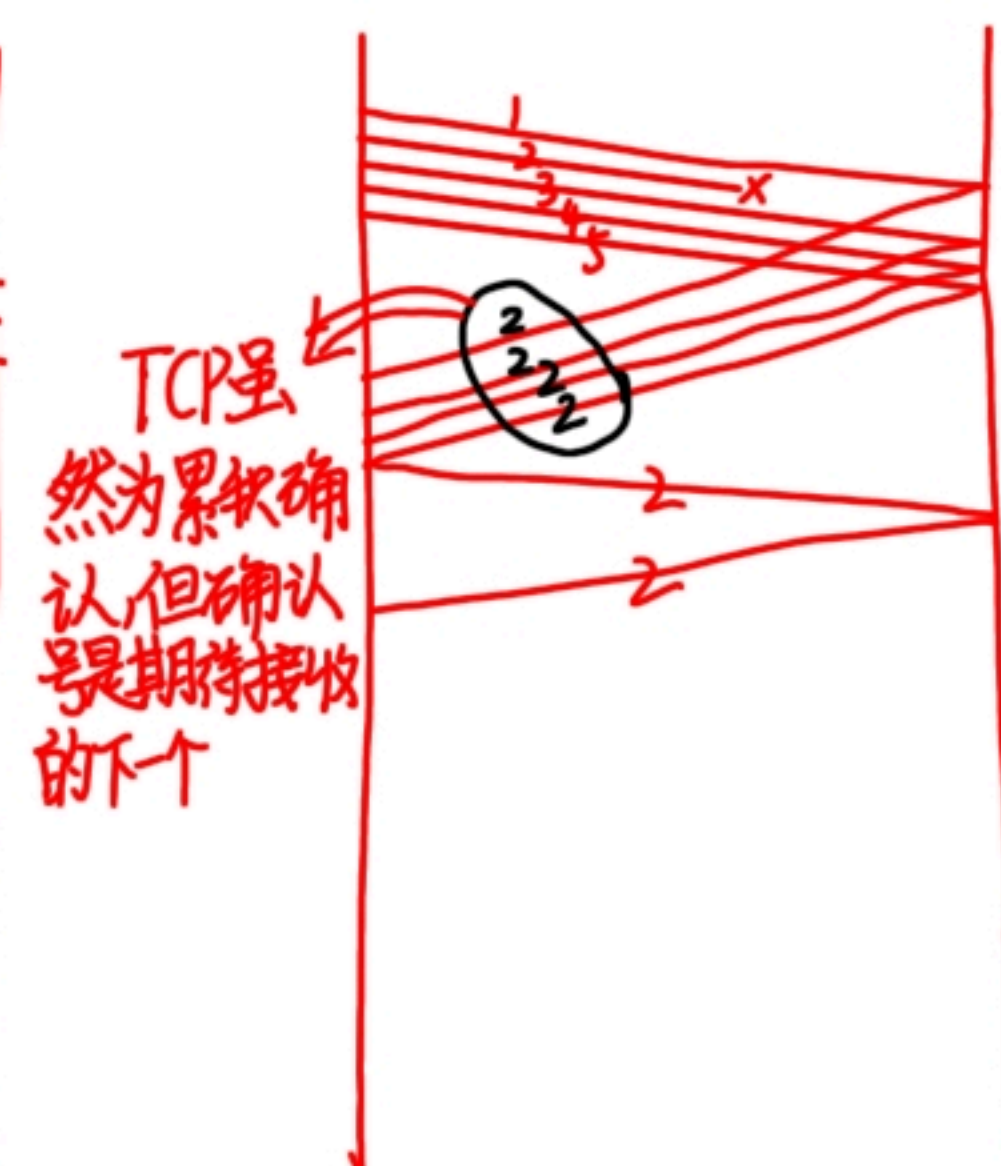
segment 9
ack 8

对 SR



6
5

对 TCP



6
5

b. TCP 快速重传

- k. 再次假设使用 TCP Tahoe, 在第 22 个传输轮回有一个超时事件。从第 17 个传输轮回回到第 22 个传输轮回 (包括这两个传输轮回), 一共发送了多少分组? $1+2+4+8+16+2=52$
- P41. 参考图 3-55, 该图描述了 TCP 的 AIMD 算法的收敛特性。假设 TCP 不采用乘性减, 而是采用按某一常量减小窗口。所得的 AIAD 算法将收敛于一种平等共享算法吗? 使用类似于图 3-55 中的图来证实你的结论。
- P42. 在 3.5.4 节中, 我们讨论了在发生超时事件后将超时间隔加倍。为什么除了这种加倍超时间隔机制外, TCP 还需要基于窗口的拥塞控制机制 (如在 3.7 节中学习的那种机制) 呢? *pipelined 仍一次发大量包*
- P43. 主机 A 通过一条 TCP 连接向主机 B 发送一个很大的文件。在这条连接上, 不会出现任何分组丢失和定时器超时。主机 A 与因特网连接链路的传输速率表示为 R bps。假设主机 A 上的进程能够以 S bps 的速率向 TCP 套接字发送数据, 其中 $S = 10 \times R$ 。进一步假设 TCP 的接收缓存足够大, 能够容纳整个文件, 并且发送缓存只能容纳这个文件的百分之一。如何防止主机 A 上的进程连续地向 TCP 套接字以速率 S bps 传送数据呢? 还是用 TCP 流量控制呢? 还是用 TCP 拥塞控制? 或者用其他措施? 阐述其理由。
- P44. 考虑从一台主机经一条没有丢包的 TCP 连接向另一台主机发送一个大文件。
- 假定 TCP 使用不具有慢启动的 AIMD 进行拥塞控制。假设每当收到一批 ACK 时, cwnd 增加 1 个 MSS, 并且假设往返时间大约恒定, cwnd 从 6MSS 增加到 12MSS 要花费多长时间 (假设没有丢包事件)? $6RTT$
 - 对于该连接, 到时间 $= 6RTT$, 其平均吞吐量是多少 (根据 MSS 和 RTT)? $\frac{6+7+8+9+10+11}{6} MSS$
- P45. 回想 TCP 吞吐量的宏观描述。在连接速率从 $W/(2 \times RTT)$ 变化到 W/RTT 的周期内, 只丢失了一个分组 (在该周期的结束)。
- 证明其丢包率 (分组丢失的比率) 等于:

$$L = \text{丢包率} = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$
 - 如果一条连接的丢包率为 L , 使用上面的结果, 则它的平均速率近似由下式给出:

$$\text{平均速率} \approx \frac{1.22 * MSS}{RTT \sqrt{L}}$$
- P46. 考虑仅有一条单一的 TCP (Reno) 连接使用一条 10Mbps 链路, 且该链路没有缓存任何数据。假设这条链路是发送主机和接收主机之间的唯一拥塞链路。假定某 TCP 发送方向接收方有一个大文件要发送, 而接收方的接收缓存比拥塞窗口要大得多。我们也做下列假设: 每个 TCP 报文段长度为 1500 字节; 该连接的双向传播时延是 150ms; 并且该 TCP 连接总是处于拥塞避免阶段, 即忽略了慢启动。
- $10Mbps = \frac{x \cdot 1500 \text{ bytes (MSS)}}{RTT}, x = 125 \text{ segments}$
- 这条 TCP 连接能够取得的最大窗口长度 (以报文段计) 是多少?
 - 这条 TCP 连接的平均窗口长度 (以报文段计) 和平均吞吐量 (以 bps 计) 是多少?
 - 这条 TCP 连接在从丢包恢复后, 再次到达其最大窗口要经历多长时间?
- P47. 考虑在前面习题中所描述的场景。假设 10Mbps 链路能够缓存有限个报文段。试论证为了使该链路总是忙于发送数据, 我们将要选择缓存长度, 使得其至少为发送方和接收方之间链路速率 C 与双向传播时延之积。
- P48. 重复习题 46, 但用一条 10Gbps 链路代替 10Mbps 链路。注意到在对 c 部分的答案中, 应当认识到在从丢包恢复后, 拥塞窗口长度到达最大窗口长度将需要很长时间。给出解决该问题的基本思路。
- P49. 令 T (用 RTT 度量) 表示一条 TCP 连接将拥塞窗口从 $W/2$ 增加到 W 所需的时间间隔, 其中 W 是最大的拥塞窗口长度。论证 T 是 TCP 平均吞吐量的函数。
- ★ P50. 考虑一种简化的 TCP 的 AIMD 算法, 其中拥塞窗口长度用报文段的数量来度量, 而不是用字节度量。在加性增中, 每个 RTT 拥塞窗口长度增加一个报文段。在乘性减中, 拥塞窗口长度减小一半 (如果结果不是一个整数, 向下取整到最近的整数)。假设两条 TCP 连接 C1 和 C2, 它们共享一条速

率为每秒 30 个报文段的单一拥塞链路。假设 C1 和 C2 均处于拥塞避免阶段。连接 C1 的 RTT 是 50ms，连接 C2 的 RTT 是 100ms。假设当链路中的数据速率超过了链路的速率时，所有 TCP 连接经受数据报文段丢失。

- a. 如果在时刻 t_0 ，C1 和 C2 具有 10 个报文段的拥塞窗口，在 1000ms 后它们的拥塞窗口为多长？
- b. 经长时间运行，这两条连接将取得共享该拥塞链路的相同的带宽吗？

P51. 考虑在前面习题中描述的网络。现在假设两条 TCP 连接 C1 和 C2，它们具有相同的 100ms RTT。假设在时刻 t_0 ，C1 的拥塞窗口长度为 15 个报文段，而 C2 的拥塞窗口长度是 10 个报文段。

- a. 在 2200ms 后，它们的拥塞窗口长度为多长？
- b. 经长时间运行，这两条连接将取得共享该拥塞链路的相同的带宽吗？
- c. 如果这两条连接在相同时间达到它们的最大窗口长度，并在相同时间达到它们的最小窗口长度，我们说这两条连接是同步的。经长时间运行，这两条连接将最终变得同步吗？如果是，它们的最大窗口长度是多少？
- d. 这种同步将有助于改善共享链路的利用率吗？为什么？给出打破这种同步的某种思路。

P52. 考虑修改 TCP 的拥塞控制算法。不使用加性增，使用乘性增。无论何时某 TCP 收到一个合法的 ACK，就将其窗口长度增加一个小正数 a ($0 < a < 1$)。求出丢包率 L 和最大拥塞窗口 W 之间的函数关系。论证：对于这种修正的 TCP，无论 TCP 的平均吞吐量如何，一条 TCP 连接将其拥塞窗口长度从 $W/2$ 增加到 W ，总是需要相同的时间。

P53. 在 3.7 节对 TCP 未来的讨论中，我们注意到为了取得 10Gbps 的吞吐量，TCP 仅能容忍 2×10^{-10} 的报文段丢失率（或等价于每 5 000 000 000 个报文段有一个丢包事件）。给出针对 3.7 节中给定的 RTT 和 MSS 值的对 2×10^{-10} 值的推导。如果 TCP 需要支持一条 100Gbps 的连接，所能容忍的丢包率是多少？

P54. 在 3.7 节中对 TCP 拥塞控制的讨论中，我们隐含地假定 TCP 发送方总是有数据要发送。现在考虑下列情况，某 TCP 发送方发送大量数据，然后在 t_1 时刻变得空闲（因为它没有更多的数据要发送）。TCP 在相对长的时间内保持空闲，然后在 t_2 时刻要发送更多的数据。当 TCP 在 t_2 开始发送数据时，让它使用在 t_1 时刻的 cwnd 和 ssthresh 值，将有什么样的优点和缺点？你建议使用什么样的方法？为什么？

P55. 在这个习题中我们研究是否 UDP 或 TCP 提供了某种程度的端点鉴别。

- a. 考虑一台服务器接收到在一个 UDP 分组中的请求并对该请求进行响应（例如，如由 DNS 服务器所做的那样）。如果一个具有 IP 地址 X 的客户用地址 Y 进行哄骗的话，服务器将向何处发送它的响应？**正常响应**
- b. 假定一台服务器接收到具有 IP 源地址 Y 的一个 SYN，在用 SYNACK 响应之后，接收一个具有 IP 源地址 Y 和正确确认号的 ACK。假设该服务器选择了一个随机初始序号并且没有“中间人”，该服务器能够确定该客户的确位于 Y 吗？**✓** 并且不在某个其他哄骗为 Y 的地址 X。）

P56. 在这个习题中，我们考虑由 TCP 慢启动阶段引入的时延。考虑一个客户和一个 Web 服务器直接连接到速率 R 的一条链路。假定该客户要取回一个对象，其长度正好等于 $15S$ ，其中 S 是最大段长度（MSS）。客户和服务器的往返时间表示为 RTT（假设为常数）。忽略协议首部，确定在下列情况下取回该对象的时间（包括 TCP 连接创建）：

- a. $4S/R > S/R + RTT > 2S/R$
- b. $S/R + RTT > 4S/R$
- c. $S/R > RTT$



编程作业

实现一个可靠传输协议

在这个编程作业实验中，你将要编写发送和接收运输层的代码，以实现一个简单的可靠数据运输协

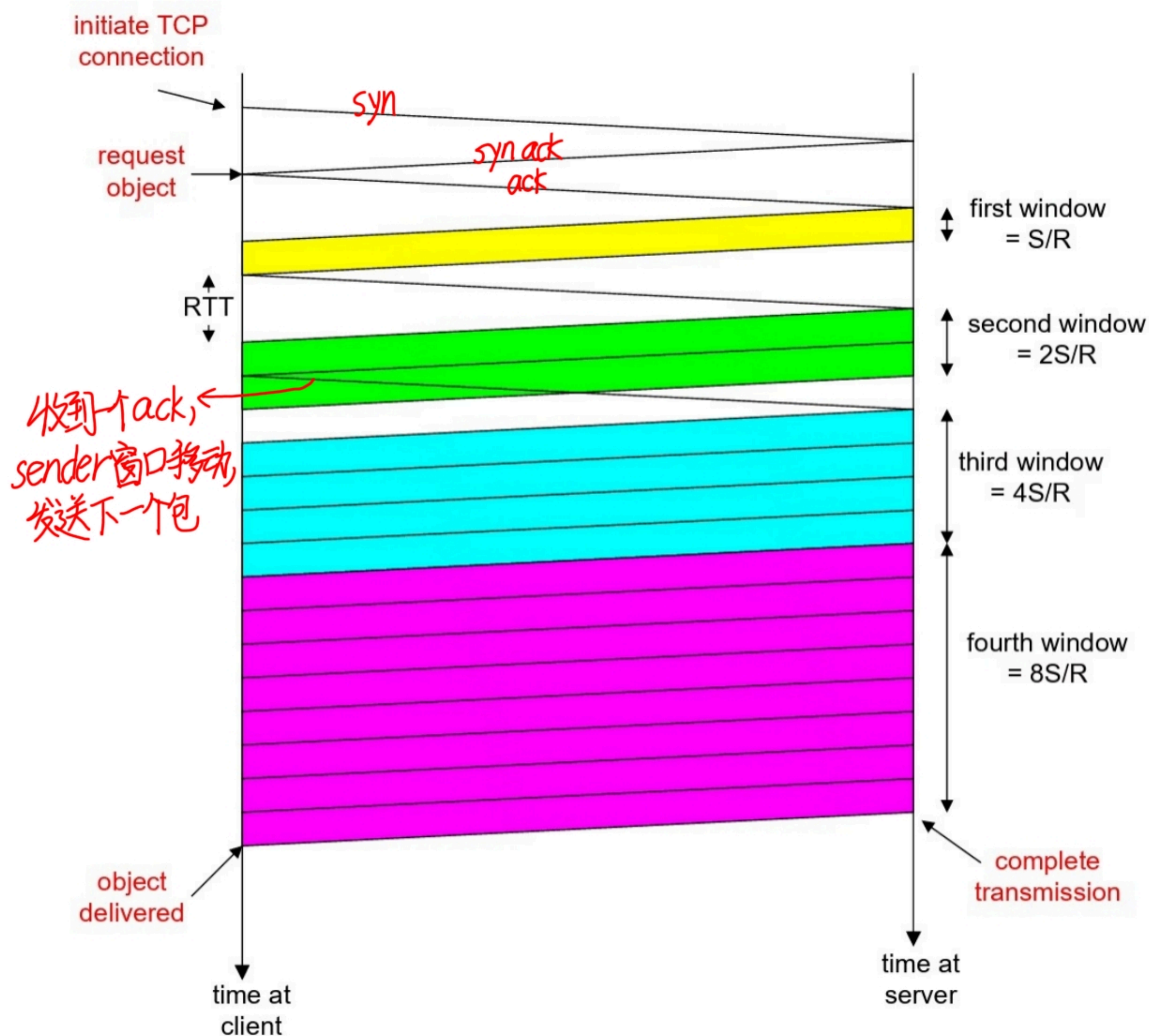
P56. 在这个习题中，我们考虑由 TCP 慢启动阶段引入的时延。考虑一个客户和一个 Web 服务器直接连接到速率 R 的一条链路。假定该客户要取回一个对象，其长度正好等于 $15S$ ，其中 S 是最大段长度 (MSS)。客户和服务端之间的往返时间表示为 RTT (假设为常数)。忽略协议首部，确定在下列情况下取回该对象的时间 (包括 TCP 连接创建)：

a. $4S/R > S/R + RTT > 2S/R$

b. $S/R + RTT > 4S/R$

c. $S/R > RTT$

wnd: $1 \rightarrow 2 \rightarrow 4 \rightarrow 8$



R34. 当一台主机加入一个多播组时，它必须将其 IP 地址改变为它所加入的多播组的地址吗？

R35. IGMP 和广域多播路由选择协议所起的作用是什么？

R36. 在多播路由选择场合中，一棵组共享的树与一棵基于源的树之间有什么区别？



习题

P1. 在本题中，考虑虚电路网络和数据报网络的某些优缺点。

a. 假设路由器遇到了可能经常会引起它无法正常运转的情况。提出理由说明是虚电路体系结构还是数据报体系结构更好，为什么？**数据报**

b. 对于源和目的结点间传输流量的排他性使用，假设该源和目的结点要求，在沿源到目的地的路径上的所有路由器总能提供固定的容量。提出理由说明是虚电路体系结构还是数据报体系结构更好，为什么？**虚电路**

c. 假设网络中的链路和路由器从不出故障，并且所有源/目的地对之间所使用的路径保持不变。在这种情况下，虚电路或数据报体系结构哪个控制流量开销会更多？为什么？**数据报**

P2. 考虑一个虚电路网络。假定其 VC 号是一个 8 比特字段。

a. 链路能够承载的虚电路的最大数量是多少？

b. 假定某中心结点在连接建立时确定了路径和 VC 号。假定沿着某虚电路的路径在每段链路使用相同的 VC 号。描述在连接建立时中心结点如何确定 VC 号。进行中虚电路比在 (a) 中确定的最大值要少，也没有相同的未用 VC 号，这种情况可能出现吗？

c. 假定沿着某条虚电路的路径允许不同的 VC 号。在连接建立期间，在端到端路径确定以后，描述链路如何以分散方式而不依赖中心结点选择它们的 VC 号并配置它们的转发表。

P3. 在虚电路网络中的基本转发表具有 4 列。在这些列中的值的含义是什么？在数据报网络中的基本转发表有两列。在这些列中的值的含义是什么？

P4. 考虑下列网络。

dest | port

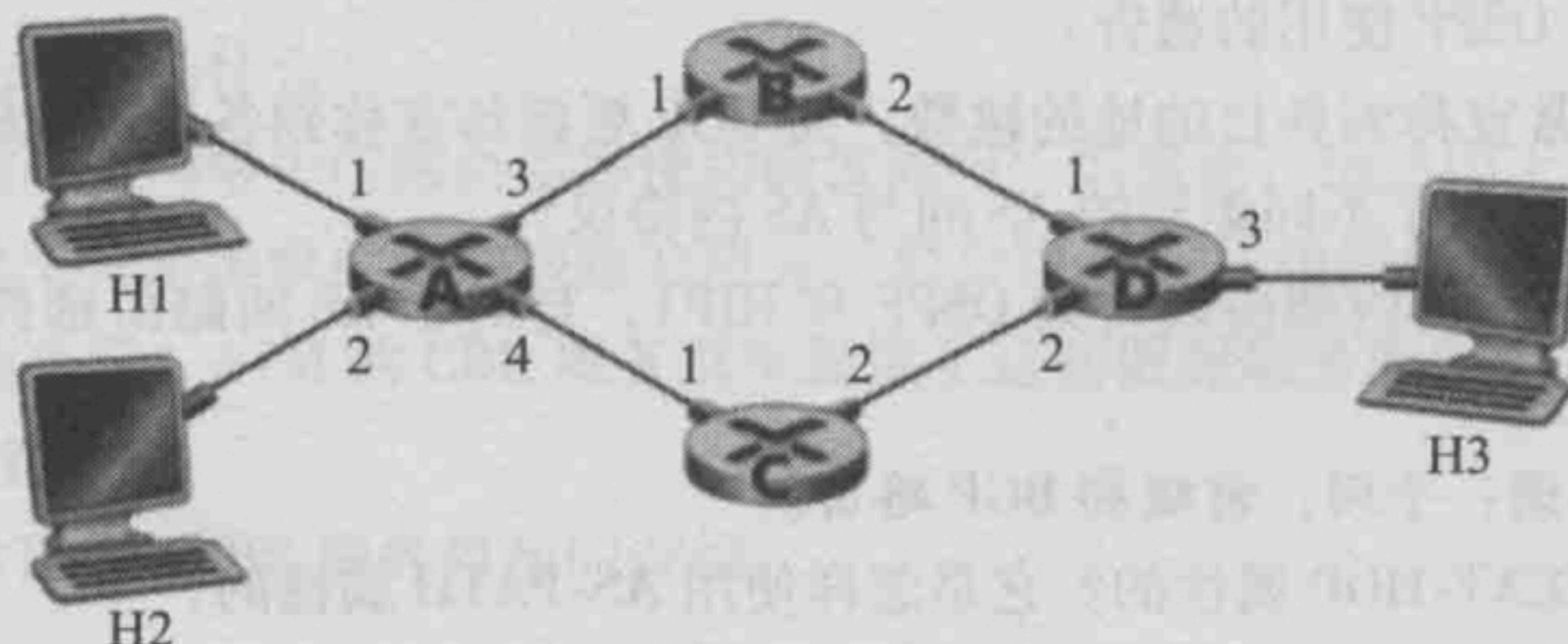
in-port | dest | out-port

a. 假定网络是一个数据报网络。显示路由器 A 中的转发表，其中所有从 H1 指向主机 H3 的流量通过接口 3 转发。

b. 假定网络是一个数据报网络。你能写出路由器 A 中的转发表吗？其中所有从 H1 指向主机 H3 的流量通过接口 3 转发，而所有从 H2 指向主机 H3 的流量通过接口 4 转发。（提示：这是一个技巧问题。）

c. 现在假定网络是虚电路网络，在 H1 和 H3 之间有一个进行中的呼叫，H2 和 H3 之间有另一个进行中的呼叫。写出路由器 A 中的转发表，其中所有从 H1 指向主机 H3 的流量通过接口 3 转发，而所有从 H2 指向主机 H3 的流量通过接口 4 转发。

d. 假设场景与 (c) 中相同，写出在结点 B、C 和 D 中的转发表。

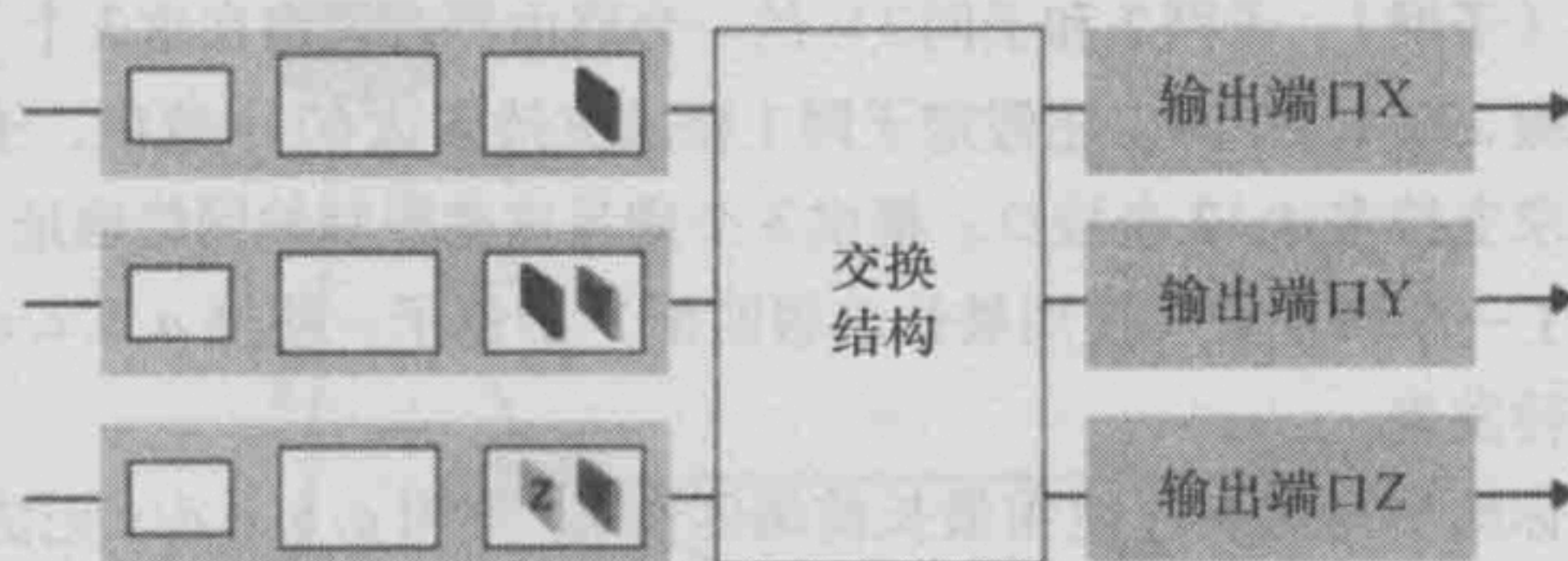


P5. 考虑一个用 2 比特字段表示 VC 号的虚电路网络。假定该网络要通过 4 条链路（链路 A、链路 B、链路 C 和链路 D）建立一条虚电路。假定这些链路中的每条当前都承载两条其他的虚电路，这些其他虚电路的 VC 号如下：

链路 A	链路 B	链路 C	链路 D
00	01	10	11
01	10	11	00

在回答下列问题，记住每个现有的虚电路可能通过这 4 条链路之一。

- a. 如果每条虚电路要求沿着其路径使用相同的 VC 号，能为该新的虚电路分配什么样的 VC 号？
 - b. 如果每条虚电路沿其路径允许不同的链路使用不同的 VC 号（因此转发表必须执行 VC 号转换），能够使用 4 个 VC 号的多少种不同的组合（一种组合用于这 4 条链路之一）？
- P6. 在本书中我们使用术语面向连接服务来描述运输层服务，使用术语连接服务描述网络层服务。在术语中为何有这种微妙的差异？
- P7. 假设两个分组在完全相同的时刻到达一台路由器的两个不同输入端口。同时假设在该路由器中没有其他分组。
- a. 假设这两个分组朝着两个不同的输出端口转发。当交换结构使用一条共享总线时，这两个分组可能在相同时刻通过交换结构转发吗？~~✗~~
 - b. 假设这两个分组朝着两个不同的输出端口转发。当交换结构使用纵横方式时，这两个分组可能在相同时刻通过交换结构转发吗？☒
 - c. 假设这两个分组朝着相同的输出端口转发。当交换结构使用纵横方式时，这两个分组可能在相同时刻通过交换结构转发吗？~~✗~~
- P8. 在 4.3 节中，我们注意到如果交换结构比输入线路速率快 n 倍，其最大的排队时延为 $(n-1)D$ 。假设所有分组有相同长度，在相同时刻 n 个分组到达 n 个输出端口，同时所有 n 个分组要转发到不同的输出端口。对于内存、总线和纵横式交换结构，一个分组的最大时延是多少？
- P9. 考虑下列交换机。假设所有数据报具有相同长度，交换机以一种分时隙、同步的方式运行，在一个时隙中一个数据报能够从某输入端口传送到某输出端口。其交换结构是纵横式的，因此在一个时隙中至多一个数据报能够传送到一个给定输出端口，但在一个单一时隙中不同的输出端口能够接收到来自不同输入端口的数据报，从输入端口到它们的输出端口传送所示的分组，所需的时隙数量最少是多少？此时假定使用你所需要的任何输入排队调度方法（即此时没有 HOL 阻塞）。假定采用你能够设计的最差情况下的调度方案，假定非空输入队列不会空闲，所需的时隙数量最大是多少？



- P10. 考虑使用 32 比特主机地址的某数据报网络。假定一台路由器具有 4 条链路，编号为 0~3，分组能被转发到如下的各链路接口：

目的地址范围	链路接口
11100000 00000000 00000000 00000000 到 11100000 00111111 11111111 11111111	0
11100000 01000000 00000000 00000000 到 11100000 01000000 11111111 11111111	1
11100000 01000001 00000000 00000000 到 11100001 01111111 11111111 11111111	2
其他	3

dest	port
224.0.0.0/22	0
224.64.0.0/16	1
224.64.0.0/22	2
others	3

- a. 提供一个具有 4 个表项的转发表，使用最长前缀匹配，转发分组到正确的链路接口。
 b. 描述你的转发表是如何为具有下列目的地址的数据报决定适当链路接口的。

11001000 10010001 01010001 01010101 3

11100001 01000000 11000011 00111100 2

11100001 10000000 00010001 01110111 2

P11. 考虑使用 8 比特主机地址的某数据报网络。假定一台路由器使用最长前缀匹配并具有下列转发表：

前缀匹配	接口
00 26	0
010 25	1
011 25	2
10 26	2
11 26	3

对这 4 个接口中的每个，给出相应的目的主机地址的范围和在该范围中的地址数量。

P12. 考虑使用 8 比特主机地址的数据报网络。假定一台路由器使用最长前缀匹配并具有下列转发表：

前缀匹配	接口
1 27	0
10 26	1
111 25	2
其他 27	3

对这 4 个接口中的每个，给出相应的目的主机地址的范围和在该范围中的地址数量。

P13. 考虑互联 3 个子网（子网 1、子网 2 和子网 3）的一台路由器。假定在这 3 个子网的每个子网中的所有接口要求具有前缀 223. 1. 17/24。还假定子网 1 要求支持多达 60 个接口，子网 2 要求支持多达 90 个接口和子网 3 要求支持多达 12 个接口。提供 3 个满足这些限制的网络地址（形式为 $a.b.c.d/x$ ）。

P14. 在 4.2.2 节中给出了一个转发表（使用最长前缀匹配）的例子。使用 $a.b.c.d/x$ 记法代替二进制字符串记法，重写该转发表。

P15. 在习题 P10 中要求你给出转发表（使用最长前缀匹配）。使用 $a.b.c.d/x$ 记法代替二进制字符串记法，重写该转发表。

P16. 考虑具有前缀 128. 119. 40. 128/26 的一个子网。给出能被分配给该网络的一个 IP 地址（形式为 xxx. xxx. xxx. xxx）的例子。假定一个 ISP 拥有形式为 128. 119. 40. 64/26 的地址块。假定它要从该地址块生成 4 个子网，每块具有相同数量的 IP 地址。这 4 个子网（形式为 $a.b.c.d/x$ ）的前缀是什么？

P17. 考虑图 4-17 中显示的拓扑。（在 12:00 以顺时针开始）标记具有主机的 3 个子网为网络 A、B 和 C。标记没有主机的子网为网络 D、E 和 F。

- a. 为这 6 个子网分配网络地址，要满足下列限制：所有地址必须从 214. 97. 254/23 起分配；子网 A 应当具有足够地址以支持 250 个接口；子网 B 应当具有足够地址以支持 120 个接口；子网 C 应当具有足够地址以支持 120 个接口。当然，子网 D、E 和 F 应当支持两个接口。对于每个子网，分配采用的形式是 $a.b.c.d/x$ 或 $a.b.c.d/x \sim e.f.g.h/y$ 。

b. 使用你对（a）部分的答案，为这 3 台路由器中的每台都提供转发表（使用最长前缀匹配）。

P18. 使用美国因特网编码注册机构（<http://www.arin.net/whois>）的 whois 服务来确定三所大学所用的 IP 地址块。whois 服务能被用于确定某个特定的 IP 地址的确定地理位置吗？使用 www.maxmind.com 来确定位于这三所大学的 Web 服务器的位置。

P19. 考虑向具有 700 字节 MTU 的一条链路发送一个 2400 字节的数据报。假定初始数据报标有标识号

P13. 考虑互联3个子网（子网1、子网2和子网3）的一台路由器。假定在这3个子网的每个子网中的所有接口要求具有前缀 223.1.17/24。还假定子网1要求支持多达60个接口，子网2要求支持多达90个接口和子网3要求支持多达12个接口。提供3个满足这些限制的网络地址（形式为 $a.b.c.d/x$ ）。

$$\begin{array}{l}
 223.1.17.00000000 \\
 \quad \vdots \\
 \quad 01111111 \\
 \left. \vphantom{\begin{array}{l} 223.1.17.00000000 \\ \vdots \\ 01111111 \end{array}} \right\} 2^7 = 128 > 90 \\
 10000000 \\
 \quad \vdots \\
 \quad 10111111 \\
 \left. \vphantom{\begin{array}{l} 10000000 \\ \vdots \\ 10111111 \end{array}} \right\} 2^6 = 64 > 60 \\
 11000000 \\
 \quad \vdots \\
 \quad 11001111 \\
 \left. \vphantom{\begin{array}{l} 11000000 \\ \vdots \\ 11001111 \end{array}} \right\} 2^4 = 16 > 12
 \end{array}$$

子网1: 223.1.17.128/26

子网2: 223.1.17.0/25

子网3: 223.1.17.192/28

P16. 考虑具有前缀 128.119.40.128/26 的一个子网。给出能被分配给该网络的一个 IP 地址（形式为 $xxx.xxx.xxx.xxx$ ）的例子。假定一个 ISP 拥有形式为 128.119.40.64/26 的地址块。假定它要从该地址块生成4个子网，每块具有相同数量的 IP 地址。这4个子网（形式为 $a.b.c.d/x$ ）的前缀是什么？

$$128.119.40.0 \mid \underbrace{00000000}_{\text{分为 } 2^2 = 4 \text{ 个子网}} / 26$$

分为 $2^2 = 4$ 个子网

$$\begin{array}{c}
 00 \ 0000 \\
 \quad \vdots \ \vdots \\
 \quad \vdots \ \vdots
 \end{array}
 \Rightarrow 128.119.40.64/28$$

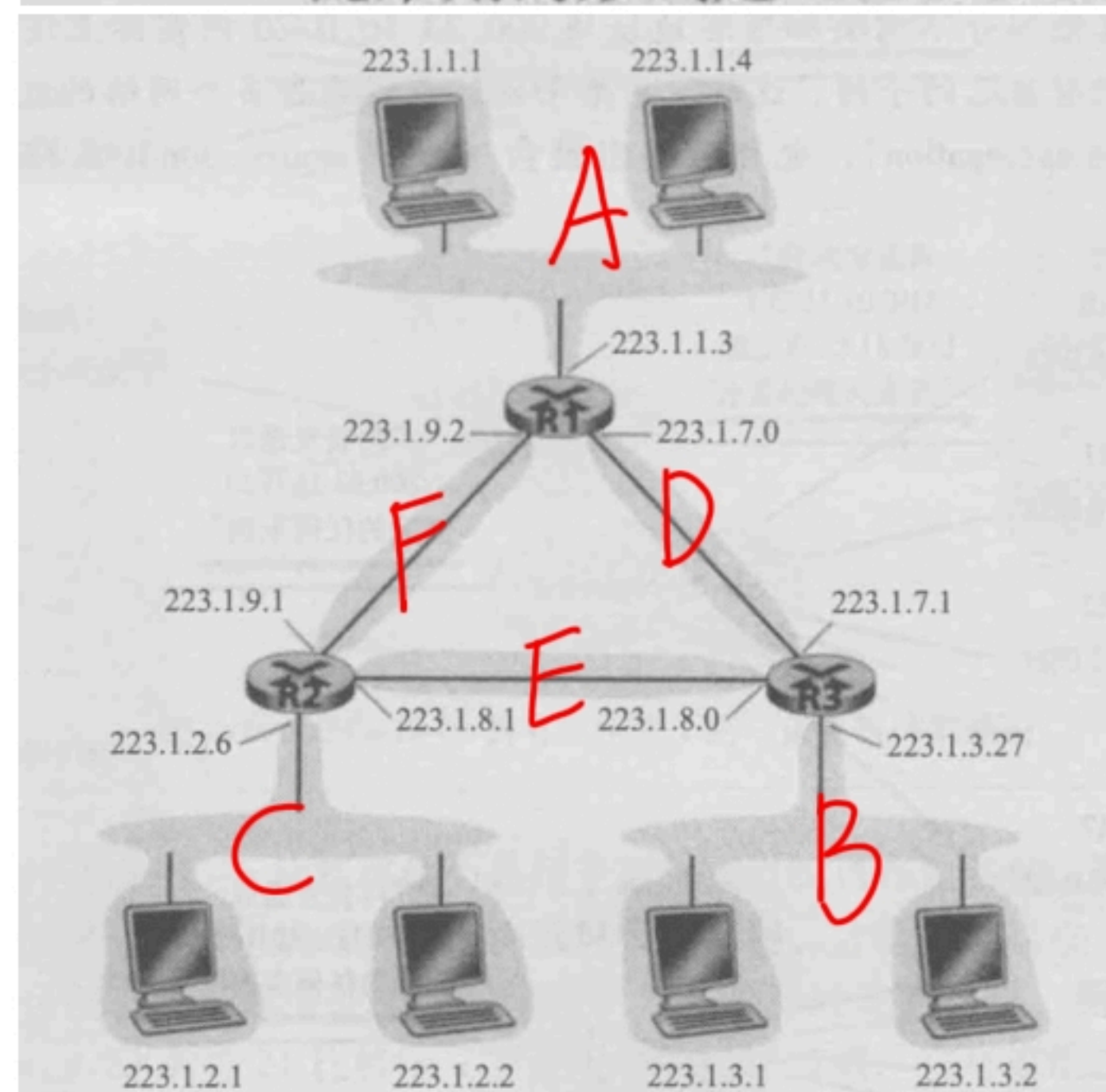
$$\begin{array}{c}
 01 \ 0000 \\
 \quad \vdots \ \vdots \\
 \quad \vdots \ \vdots
 \end{array}
 \Rightarrow 128.119.40.80/28$$

$$\begin{array}{c}
 10 \ 0000 \\
 \quad \vdots \ \vdots \\
 \quad \vdots \ \vdots
 \end{array}
 \Rightarrow 128.119.40.96/28$$

$$\begin{array}{c}
 11 \ 0000 \\
 \quad \vdots \ \vdots \\
 \quad \vdots \ \vdots
 \end{array}
 \Rightarrow 128.119.40.112/28$$

P17. 考虑图 4-17 中显示的拓扑。(在 12:00 以顺时针开始) 标记具有主机的 3 个子网为网络 A、B 和 C。标记没有主机的子网为网络 D、E 和 F。

- a. 为这 6 个子网分配网络地址，要满足下列限制：所有地址必须从 214.97.254/23 起分配；子网 A 应当具有足够地址以支持 250 个接口；子网 B 应当具有足够地址以支持 120 个接口；子网 C 应当具有足够地址以支持 120 个接口。当然，子网 D、E 和 F 应当支持两个接口。对于每个子网，分配采用的形式是 $a.b.c.d/x$ 或 $a.b.c.d/x \sim e.f.g.h/y$ 。



错解:

214.97.11111110.0000 0000 } $2^8 = 256 > 250$
 1111 1111 }
 1 0000 0000 } $2^7 = 128 > 120$
 1 0111 1111 }
 1 1000 0000 } $2^7 = 128 > 120$
 1111 1111 }

⇒ 无法再为 D、E、F 分配，
 IP 已用尽

正解:

网络层 $\Rightarrow$ MTU: 700 datagram包体: $2380 = 680 + 680 + 680 + 340$
 $\Rightarrow$ length = 700, 700, 700, 360 id = 422, 422, 422, 422 283
 flag = 1, 1, 1, 0 offset = 0, $\frac{680}{8} = 85, 170, 255$

422. 将会生成多少个分片? 在生成相关分片的数据报中的各个字段中的值是多少? $1500 - \text{datagram包头} - \text{segment包头}$

P20. 假定在源主机 A 和目的主机 B 之间的数据报被限制为 1500 字节 (包括首部)。假设 IP 首部为 20 字节, 要发送一个 5MB 组成的 MP3 需要多少个数据报? 解释你的答案是如何计算的。

P21. 考虑在图 4-22 中建立的网络。假定 ISP 此时为路由器分配地址 24. 34. 112. 235 以及家庭网络的网络地址是 192. 168. 1/24。

a. 在家庭网络中为所有接口分配地址。

b. 假定每台主机具有两个进行中的 TCP 连接, 所有都是对主机 128. 119. 40. 86 的 80 端口的。在 NAT 转换表中提供 6 个对应表项。

P22. 假设你有兴趣检测 NAT 后面的主机数量。你观察到在每个 IP 分组上 IP 层顺序地标出一个标识号。由一台主机生成的第一个 IP 分组的标识号是一个随机数, 后继 IP 分组的标识号是顺序分配的。假设由 NAT 后面主机产生的所有 IP 分组都发往外部。

a. 基于这个观察, 假定你能够俘获由 NAT 向外部发送的所有分组, 你能概要给出一种简单的技术来检测 NAT 后面不同主机的数量吗? 评估你的答案。

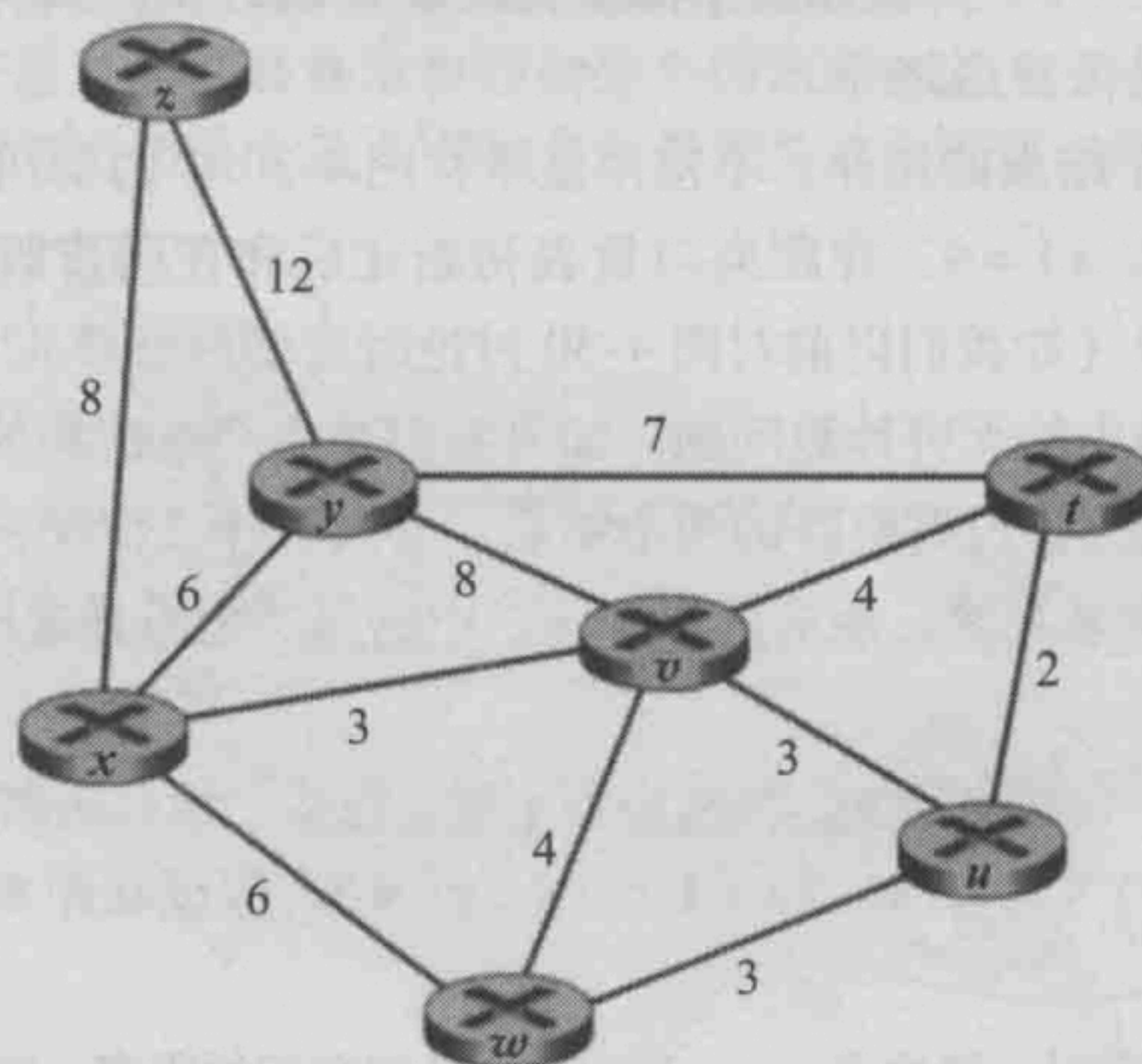
b. 如果标识号不是顺序分配而是随机分配的, 这种技术还能正常工作吗? 评估你的答案。

P23. 在这个习题中, 我们将探讨 NAT 对 P2P 应用程序的影响。假定具有用户名 Arnold 的对等方通过查询发现, 具有用户名 Bernard 的对等方有一个要下载的文件。同时假定 Bernard 和 Arnold 都位于 NAT 后面。尝试设计一种技术, 使得 Arnold 与 Bernard 创建一条 TCP 连接, 而不对 NAT 做应用特定的配置。如果你难以设计这样的技术, 试讨论其原因。

P24. 观察图 4-27, 列举从 y 到 u 不包含任何环路的路径。

P25. 重复习题 P24, 列举从 x 到 z 、 z 到 u 以及 z 到 w 的不包含任何环路的路径。

P26. 考虑下面的网络。对于标明的链路费用, 用 Dijkstra 的最短路径算法计算出从 x 到所有网络结点的最短路径。通过计算一个类似于表 4-3 的表, 说明该算法是如何工作的。



P27. 考虑习题 P26 中所示的网络。使用 Dijkstra 算法和一个类似于表 4-3 的表来说明你做的工:

a. 计算出从 t 到所有网络结点的最短路径。

b. 计算出从 u 到所有网络结点的最短路径。

c. 计算出从 v 到所有网络结点的最短路径。

d. 计算出从 w 到所有网络结点的最短路径。

e. 计算出从 y 到所有网络结点的最短路径。

f. 计算出从 z 到所有网络结点的最短路径。

P28. 考虑下图所示的网络, 假设每个结点初始时知道到它的每个邻居的费用。考虑距离向量算法, 并显示在结点 z 中的距离表表项。

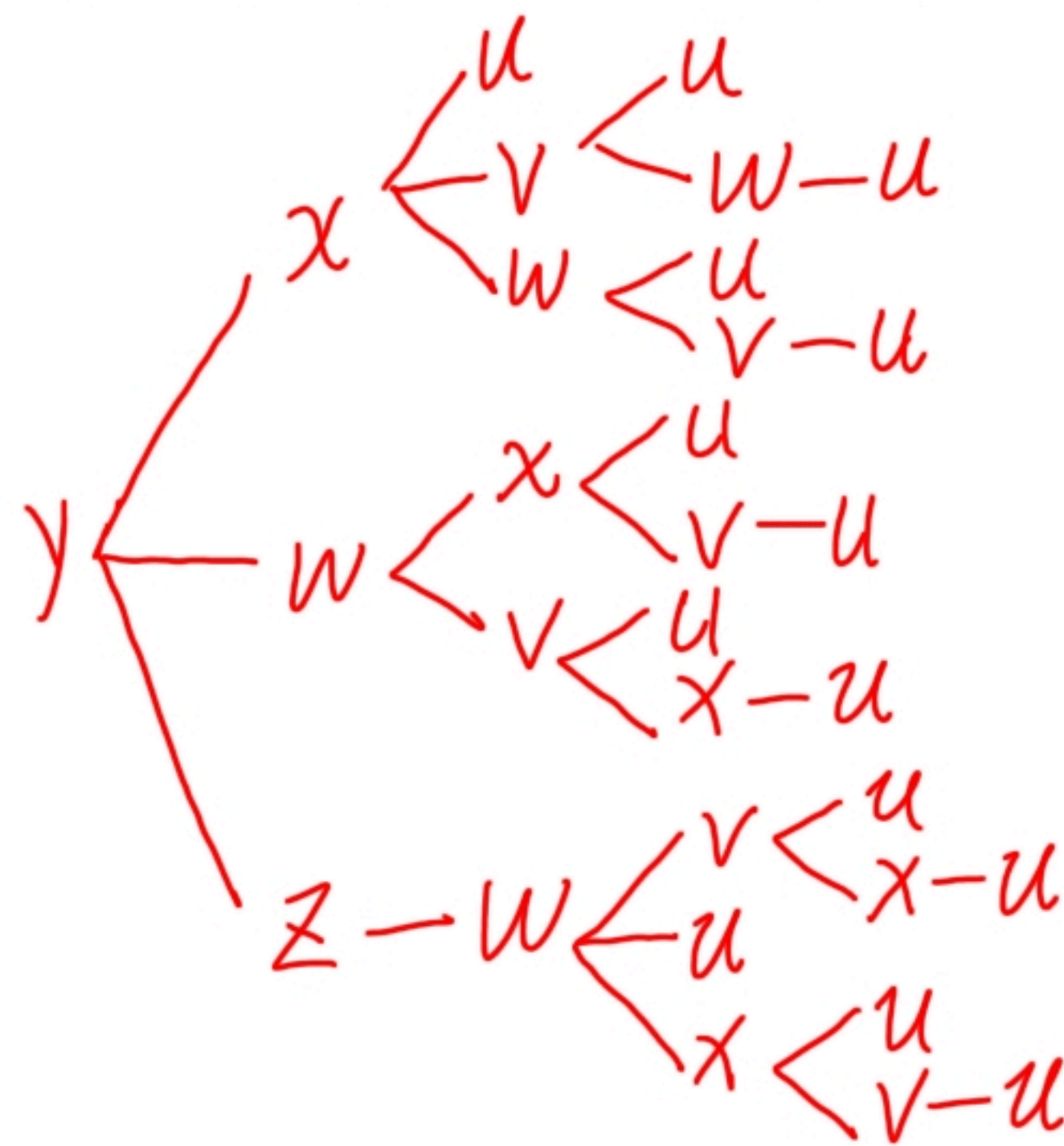
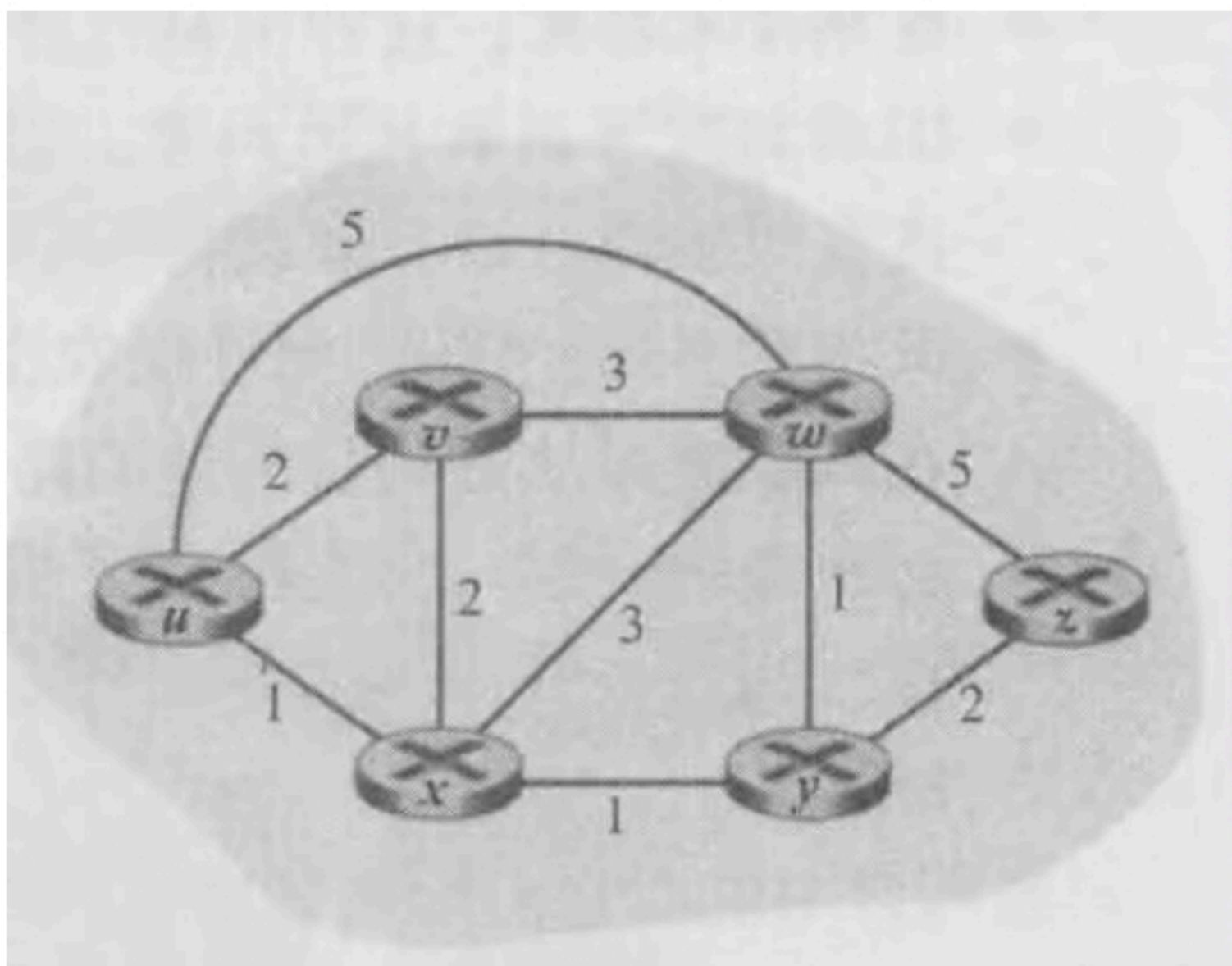
P21. 考虑在图 4-22 中建立的网络。假定 ISP 此时为路由器分配地址 24.34.112.235 以及家庭网络的网络地址是 192.168.1/24。

a. 在家庭网络中为所有接口分配地址。

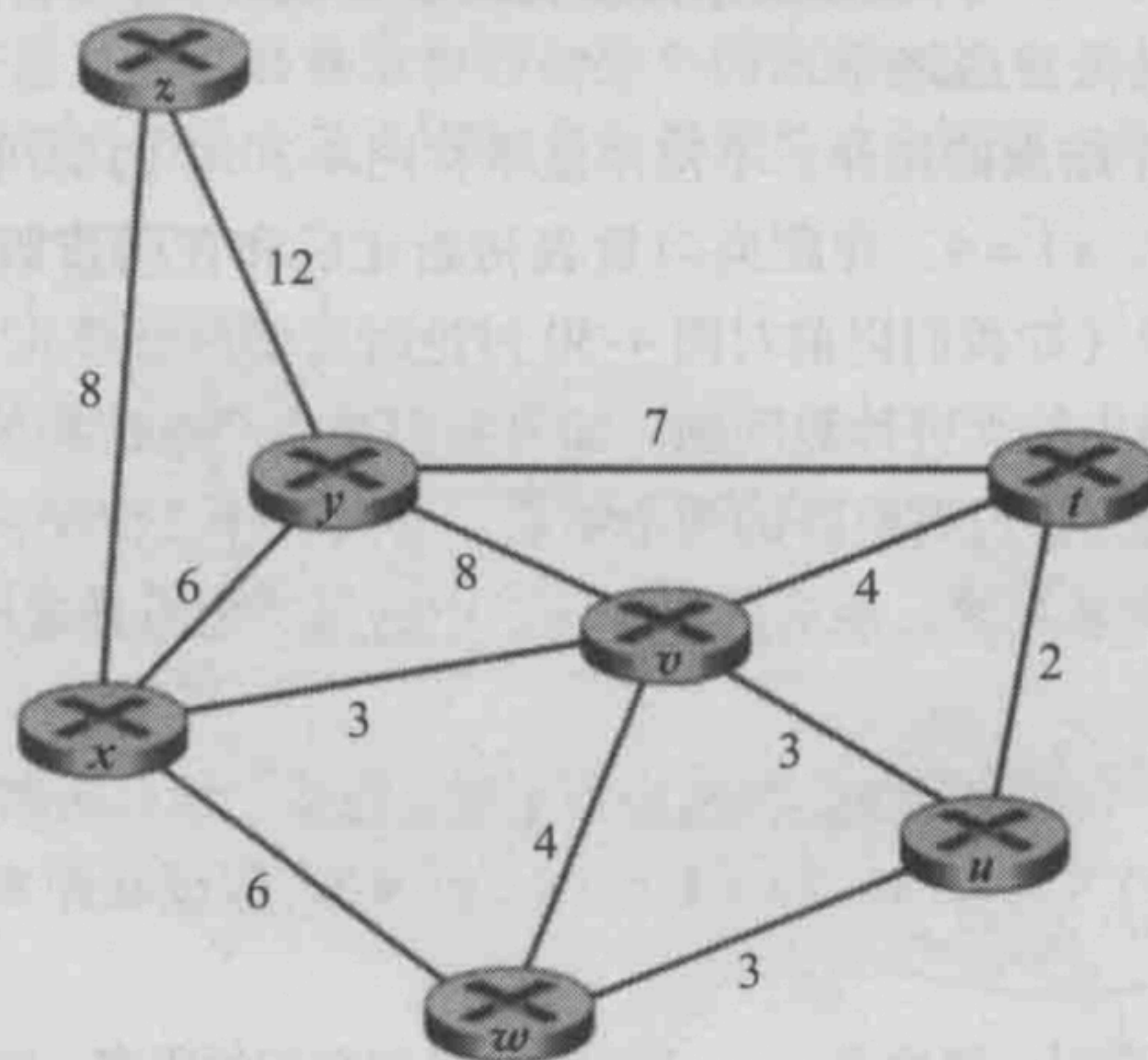
b. 假定每台主机具有两个进行中的 TCP 连接，所有都是对主机 128.119.40.86 的 80 端口的。在 NAT 转换表中提供 6 个对应表项。

a. 192.168.1.1/24, 192.168.1.2/24, 192.168.1.3/24	
b. WAN	LAN
24.34.112.235, 4000	192.168.1.1, 3345
24.34.112.235, 4001	192.168.1.1, 3346
24.34.112.235, 4002	192.168.1.2, 3345
24.34.112.235, 4003	192.168.1.2, 3346
24.34.112.235, 4004	192.168.1.3, 3345
24.34.112.235, 4005	192.168.1.3, 3346

P24. 观察图 4-27，列举从 y 到 u 不包含任何环路的路径。

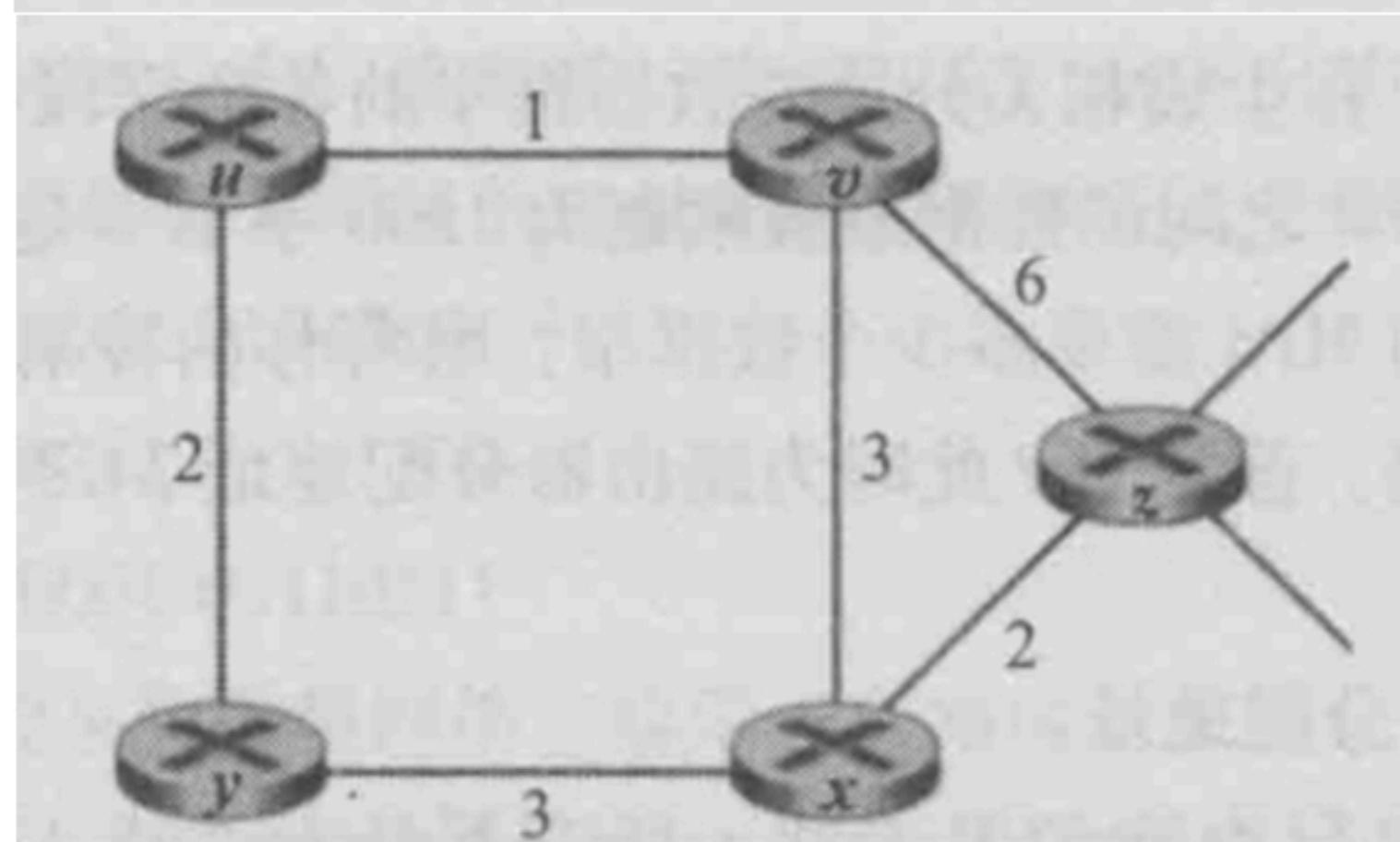


P26. 考虑下面的网络。对于标明的链路费用，用 Dijkstra 的最短路径算法计算出从 x 到所有网络结点的最短路径。通过计算一个类似于表 4-3 的表，说明该算法是如何工作的。



Step	N^p	y	z	w	v	u	t
0	x	6	8	6	(3)	∞	∞
1	xv	(6)	8	6		6	7
2	xvy		8	(6)		6	7
3	xvyw		8			(6)	7
4	xvywu		8				(7)
5	xvywut		(8)				
6	xvywutz						

P28. 考虑下图所示的网络，假设每个结点初始时知道到它的每个邻居的费用。考虑距离向量算法，并显示在结点z中的距离表表项。



	u	v	x	y	z
v	∞	0	3	∞	6
x	∞	3	0	3	2
z	∞	5	2	5	0

↓ step 1

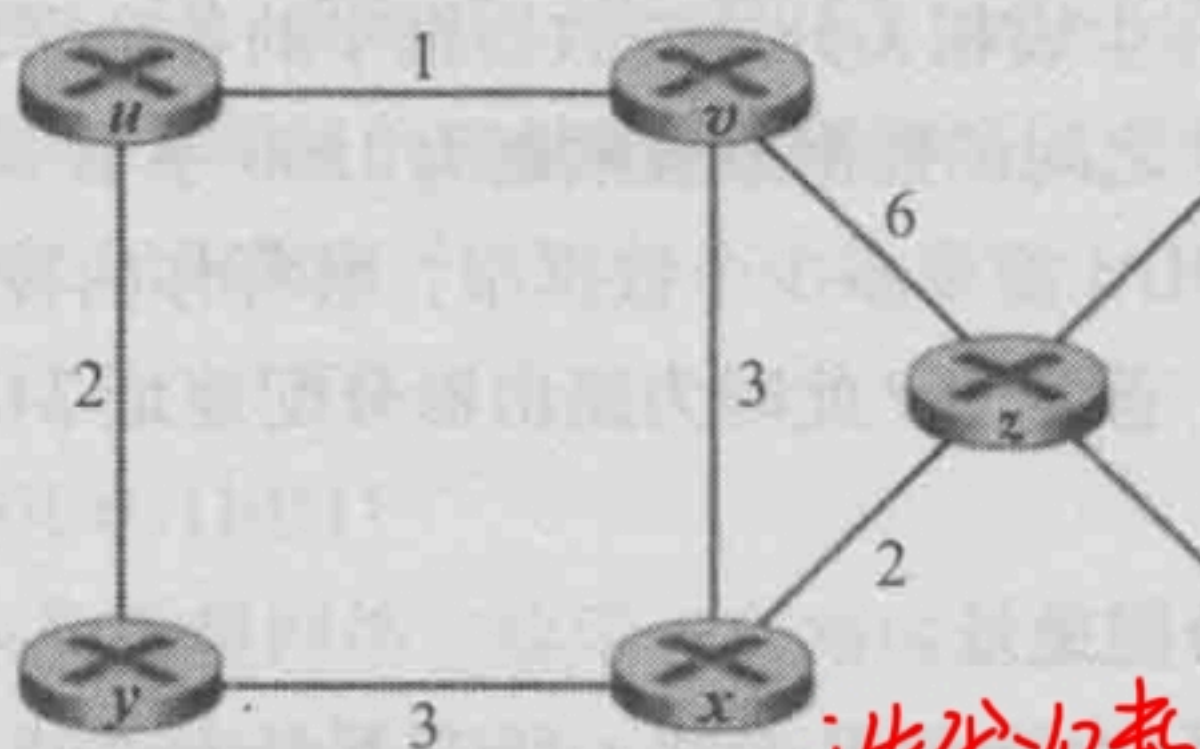
	u	v	x	y	z
v	1	0	3	∞	6
x	∞	3	0	3	2
z	7	5	2	5	0

↓ step 2

	u	v	x	y	z
v	1	0	3	3	5
x	4	3	0	3	2
z	6	5	2	5	0

↓ step 3

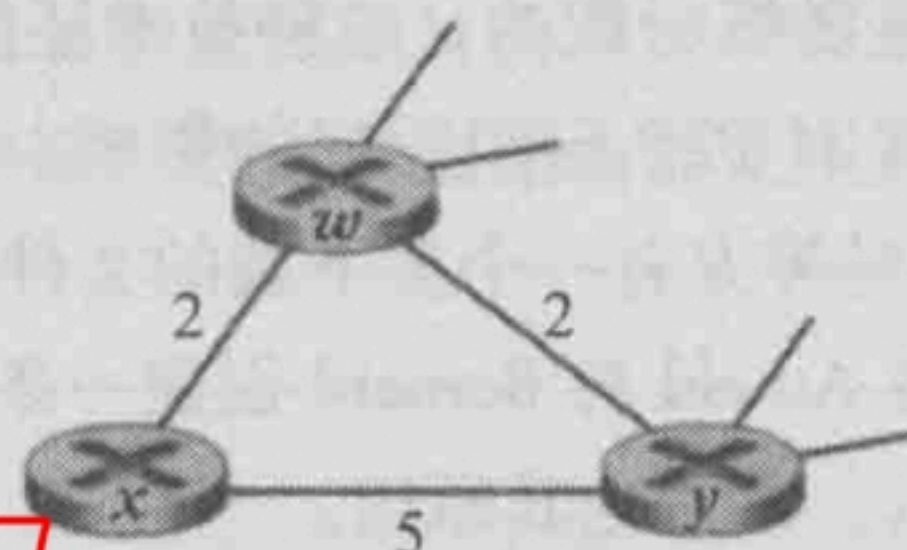
	u	v	x	y	z
v	1	0	3	3	5
x	4	3	0	3	2
z	6	5	2	5	0



迭代次数 $\leq$ 最长无环路径的边数 - 1

P29. 考虑一个一般性拓扑（即不是以上所显示的特定网络）和一个同步版本的距离向量算法。假设每次迭代时，一个结点与其邻居交换其距离向量并接收它们的距离向量。假定算法开始时，每个结点只知道到其直接邻居的费用，在该分布式算法收敛前所需的最大迭代次数是多少？评估你的答案。

P30. 考虑下图所示的网络段。 x 只有两个相连邻居 w 与 y 。 w 有一条通向目的地 u （没有显示）的最低费用路径，其值为 5； y 有一条通向目的地 u 的最低费用路径，其值为 6。从 w 与 y 到 u （以及 w 与 y 之间）的完整路径未显示出来。网络中所有链路费用皆为正整数值。



$D_x(w)=2, D_x(y)=5, D_x(u)=7$

- 给出 x 对目的地 w 、 y 和 u 的距离向量。 $C(x,w) < 2$ 或 $C(x,y) < 1$
- 给出对 $c(x, w)$ 或 $c(x, y)$ 的链路费用的变化，使得执行了距离向量算法后， x 将通知其邻居有一条通向 u 的新最低费用路径。 $C(x,w) \equiv 2, 1 \leq C(x,y)$
- 给出对 $c(x, w)$ 或 $c(x, y)$ 的链路费用的变化，使得执行了距离向量算法后， x 将不通知其邻居有一条通向 u 的新最低费用路径。

P31. 考虑如图 4-30 中所示 3 个结点的拓扑。不使用显示在图 4-30 中的费用值，链路费用值现在是 $c(x, y) = 3$, $c(y, z) = 6$, $c(z, x) = 4$ 。在距离向量表初始化后和在同步版本的距离向量算法每次迭代后，计算它的距离向量表（如我们以前对图 4-30 讨论时所做的那样）。

P32. 考虑在距离向量路由选择中的无穷计数问题。如果我们减小一条链路费用，将会出现无穷计数问题吗？~~增加 $D_x(y)$ 可能出现~~ 如果我们连接没有链路的两个结点，会出现什么情况？

P33. 讨论在图 4-30 中的距离向量算法，距离向量 $D(x)$ 中的每个值不是递增的并且最终将在有限步中稳定下来。

P34. 考虑图 4-31。假定有另一台路由器 w ，与路由器 y 和 z 连接。所有链路费用给定如下： $c(x, y) = 4$, $c(x, z) = 50$, $c(y, w) = 1$, $c(z, w) = 1$, $c(y, z) = 3$ 。假设在距离向量路由选择算法中使用了毒性逆转。

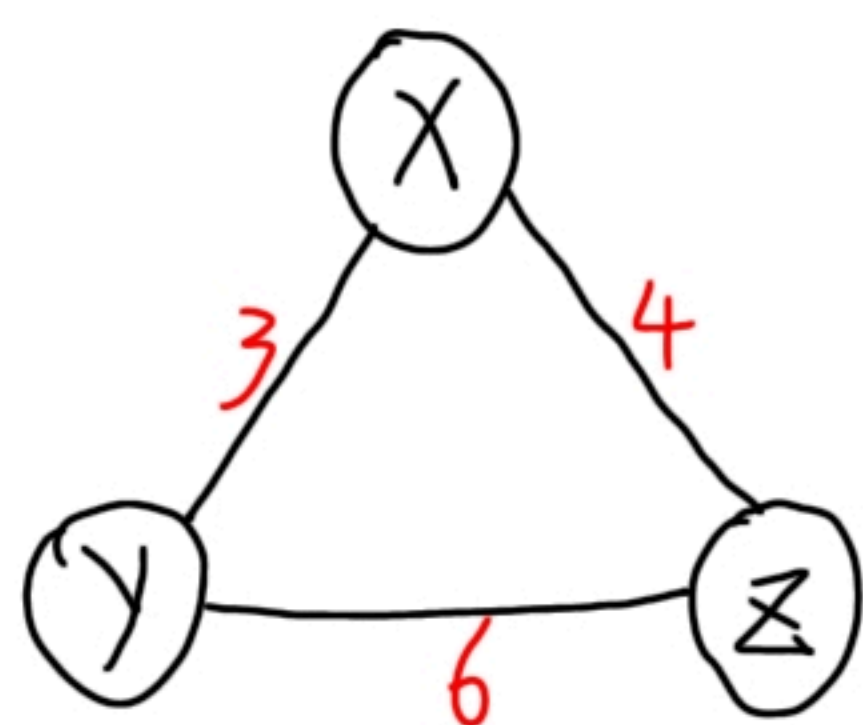
- 当距离向量路由选择稳定时，路由器 w 、 y 和 z 向 x 通知它们的距离。它们告诉彼此什么样的距离值？
- 现在假设 x 和 y 之间的链路成本增加到 60。即使使用了毒性逆转，将会存在无穷计数问题吗？为什么？如果存在无穷计数问题，距离向量路由选择需要多少次迭代才能再次到达稳定状态？评估你的答案。
- 如果 $c(y, x)$ 从 4 变化到 60，怎样修改 $c(y, z)$ 使得不存在无穷计数问题。

P35. 描述在 BGP 中是如何检测路径中的环路的。

P36. 一台 BGP 路由器将总是选择具有最短 AS 路径长度的无环路由吗？评估你的答案。

P37. 考虑下图所示的网络。假定 AS3 和 AS2 正在运行 OSPF 作为其 AS 内部路由选择协议。假定 AS1 和 AS4 正在运行 RIP 作为其 AS 内部路由选择协议。假定 AS 间路由选择协议使用的是 eBGP 和 iBGP。假定最初在 AS2 和 AS4 之间不存在物理链路。

P31. 考虑如图 4-30 中所示 3 个结点的拓扑。不使用显示在图 4-30 中的费用值，链路费用值现在是 $c(x, y) = 3$, $c(y, z) = 6$, $c(z, x) = 4$ 。在距离向量表初始化后和在同步版本的距离向量算法每次迭代后，计算它的距离向量表（如我们以前对图 4-30 讨论时所做的那样）。



Node x

	x	y	z
x	0	3	4
y	∞	∞	∞
z	∞	∞	∞

$\Rightarrow$

	x	y	z
x	0	3	4
y	3	0	6
z	4	6	0

Node y

	x	y	z
x	∞	∞	∞
y	3	0	6
z	∞	∞	∞

$\Rightarrow$

	x	y	z
x	0	3	4
y	3	0	6
z	4	6	0

Node z

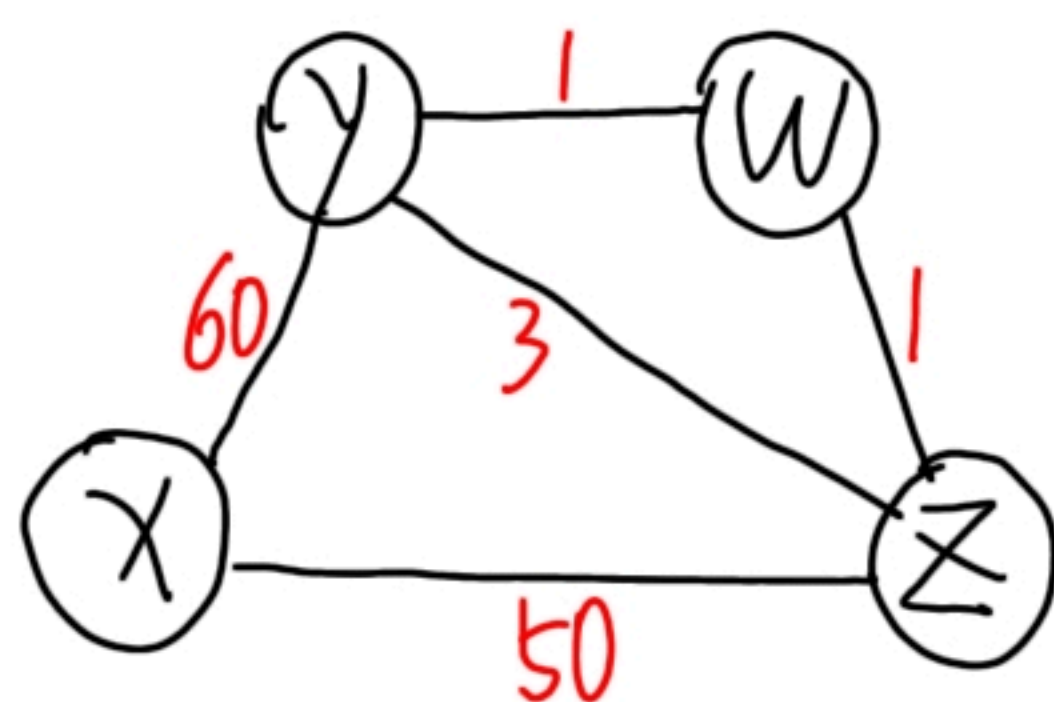
	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	4	6	0

$\Rightarrow$

	x	y	z
x	0	3	4
y	3	0	6
z	4	6	0

P34. 考虑图 4-31。假定有另一台路由器 w ，与路由器 y 和 z 连接。所有链路费用给定如下： $c(x, y) = 4$, $c(x, z) = 50$, $c(y, w) = 1$, $c(z, w) = 1$, $c(y, z) = 3$ 。假设在距离向量路由选择算法中使用了毒性逆转。

- 当距离向量路由选择稳定时，路由器 w 、 y 和 z 向 x 通知它们的距离。它们告诉彼此什么样的距离值？
- 现在假设 x 和 y 之间的链路成本增加到 60。即使使用了毒性逆转，将会存在无穷计数问题吗？为什么？如果存在无穷计数问题，距离向量路由选择需要多少次迭代才能再次到达稳定状态？评估你的答案。
- 如果 $c(y, x)$ 从 4 变化到 60，怎样修改 $c(y, z)$ 使得不存在无穷计数问题。



a. $w \rightarrow D_w(y) = 1$
 $w \rightarrow D_w(z) = 1$
 $z \rightarrow D_z(x) = 50$
 $z \rightarrow D_z(y) = 3$
 $z \rightarrow D_z(w) = 1$

$x \rightarrow D_x(y) = 4$
 $x \rightarrow D_x(z) = 50$
 $y \rightarrow D_y(x) = 4$
 $y \rightarrow D_y(w) = 1$
 $y \rightarrow D_y(z) = 3$

b. 到 x 距离

	y	z	w
	4	6	5

成本增加

	y	z	w
	60	50	

$\min(1+60, 1+5) \Rightarrow$ 无穷大

课后习题和问题



复习题

5.1~5.2 节

- R1. 考虑在 5.1.1 节中的运输类比。如果一个乘客类比为数据报，什么类比为链路层帧？
- R2. 如果在因特网中的所有链路都提供可靠的交付服务，TCP 可靠传输服务将是多余的吗？为什么？
- R3. 链路层协议能够向网络层提供哪些可能的服务？在这些链路层服务中，哪些在 IP 中有对应的服务？哪些在 TCP 中有对应的服务？

5.3 节

- R4. 假设两个结点同时经一个速率为 R 的广播信道开始传输一个长度为 L 的分组。用 d_{prop} 表示这两个结点之间的传播时延。如果 $d_{\text{prop}} < L/R$ ，会出现碰撞吗？为什么？
- R5. 在 5.3 节中，我们列出了广播信道的 4 种希望的特性。这些特性中的哪些是时隙 ALOHA 所具有的？令牌传递具有这些特性中的哪些？
- R6. 在 CSMA/CD 中，在第 5 次碰撞后，结点选择 $K=4$ 的概率有多大？结果 $K=4$ 在 10Mbps 以太网上对应于多少秒的时延？
- R7. 使用人类在鸡尾酒会交互的类比来描述轮询和令牌传递协议。
- R8. 如果局域网有很大的周长时，为什么令牌环协议将是低效的？

5.4 节

- R9. MAC 地址空间有多大？IPv4 的地址空间呢？IPv6 的地址空间呢？
- R10. 假设结点 A、B 和 C（通过它们的适配器）都连接到同一个广播局域网。如果 A 向 B 发送数千个 IP 数据报，每个封装帧都有 B 的 MAC 地址，C 的适配器会处理这些帧吗？如果会，C 的适配器将会把这些帧中的 IP 数据报传递给 C 的网络层吗？如果 A 用 MAC 广播地址来发送这些帧，你的回答将有怎样的变化呢？
- R11. ARP 查询为什么要在广播帧中发送呢？ARP 响应为什么要在一个具有特定目的 MAC 地址的帧中发送呢？
- R12. 对于图 5-19 中的网络，路由器有两个 ARP 模块，每个都有自己的 ARP 表。同样的 MAC 地址可能在两张表中都出现吗？
- R13. 比较 10BASE-T、100BASE-T 和吉比特以太网的帧结构。它们有什么不同吗？
- R14. 考虑图 5-15。在 4.4 节的寻址意义下，有多少个子网呢？
- R15. 在一个支持 802.1Q 协议交换机上能够配置的 VLAN 的最大数量是多少？
- R16. 假设支持 K 个 VLAN 组的 N 台交换机经过一个干线协议连接起来。连接这些交换机需要多少端口？评价你的答案。



习题

```

1110 | 1
0110 | 0
1001 | 0
1101 | 1
-----
1100 | 0

```

- P1. 假设某分组的信息内容是比特模式 1110 0110 1001 1101，并且使用了偶校验方案。在采用二维奇偶校验方案的情况下，包含该检验比特的字段的值是什么？你的回答应该使用最小长度检验和字段。
- P2. 说明（举一个不同于图 5-5 中那个的例子）二维奇偶校验能够纠正和检测单比特差错。说明（举一个例子）某些双比特差错能够被检测但不能纠正。
- P3. 假设某分组的信息部分（图 5-3 中的 D ）包含 10 字节，它由字符串 “Networking” 的 8 比特无符号二进制 ASCII 表示组成。对该数据计算因特网检验和。
- P4. 考虑前一个习题，但此时假设这 10 字节包含：
- 数字 1 到 10 的二进制表示。

P2. 说明（举一个不同于图 5-5 中那个的例子）二维奇偶校验能够纠正和检测单比特差错。说明（举一个例子）某些双比特差错能够被检测但不能纠正。

$$\begin{array}{c}
 \begin{array}{cccc|c}
 1 & 1 & 1 & 0 & 1 \\
 0 & 1 & 1 & 0 & 0 \\
 1 & 0 & 0 & 1 & 0 \\
 1 & 1 & 0 & 1 & 1 \\
 \hline
 1 & 1 & 0 & 0 & 0
 \end{array} \quad (\text{正确})
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{cccc|c}
 1 & 1 & 1 & 0 & 1 \\
 0 & 1 & 1 & 0 & 0 \\
 1 & 0 & 0 & 1 & 1 \\
 1 & 1 & 0 & 1 & 1 \\
 \hline
 1 & 0 & 0 & 0 & 0
 \end{array} \quad (\text{一位错})
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{cccc|c}
 1 & 1 & 1 & 0 & 1 \\
 0 & 0 & 1 & 0 & 1 \\
 1 & 1 & 0 & 1 & 1 \\
 1 & 1 & 0 & 1 & 1 \\
 \hline
 1 & 1 & 0 & 0 & 0
 \end{array} \quad (\text{两位错})
 \end{array}$$

P3. 假设某分组的信息部分（图 5-3 中的 D）包含 10 字节，它由字符串 “Networking” 的 8 比特无符号二进制 ASCII 表示组成。对该数据计算因特网检验和。

Networking: 01001100, 01101001, 00100000, 01001100,
01100001, 01111001, 01100101, 01110010

注意：变为 16 bits 一组!!! 最终取反码!

$\Rightarrow \text{checksum} = 010110111110011$

P5. 考虑 5 比特生成多项式， $G = 10011$ ，并且假设 D 的值为 1010101010。 R 的值是什么？

$$\begin{array}{r}
 \begin{array}{ccccccccc}
 & & & 1 & 0 & 1 & 0 & 1 & 0 \\
 \hline
 1 & 0 & 0 & 1 & 1 &) & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \hline
 & & & 1 & 0 & 0 & 1 & 1 & & & & & & & & & & \\
 \hline
 \text{XOR} & & & & & & & & & & & & & & & & & \\
 & & & & & & 1 & 1 & 0 & 0 & 1 & & & & & & & \\
 & & & & & & 1 & 0 & 0 & 1 & 1 & & & & & & & \\
 \hline
 & & & & & & & & & & & 1 & 0 & 1 & 0 & 0 & & \\
 & & & & & & & & & & & 1 & 0 & 0 & 1 & 1 & & \\
 \hline
 & & & & & & & & & & & & 1 & 1 & 1 & 0 & & \\
 & & & & & & & & & & & & 1 & 0 & 0 & 1 & 1 & \\
 \hline
 & & & & & & & & & & & & & 1 & 1 & 0 & 1 & 0 & \\
 & & & & & & & & & & & & & 1 & 0 & 0 & 1 & 1 & \\
 \hline
 & & & & & & & & & & & & & & 1 & 0 & 0 & 1 & 0 & \\
 & & & & & & & & & & & & & & 1 & 0 & 0 & 1 & 1 & \\
 \hline
 & & & & & & & & & & & & & & & & 0 & 1 & 0 & 0
 \end{array}
 \end{array}$$

$\Rightarrow R = 0100$

b. 字母 B 到 K (大写) 的 ASCII 表示。

c. 字母 b 到 k (小写) 的 ASCII 表示。

计算这些数据的因特网检验和。

P5. 考虑 5 比特生成多项式, $G = 10011$, 并且假设 D 的值为 1010101010。 R 的值是什么?

P6. 考虑上一个习题, 这时假设 D 具有值:

a. 1001010101。

b. 0101101010。

c. 1010100000。

P7. 在这道习题中, 我们探讨 CRC 的某些性质。对于在 5.2.3 节中给出的生成多项式 $G (= 1001)$, 回答下列问题:

a. 为什么它能够检测数据 D 中的任何单比特差错?

b. 上述 G 能够检测任何奇数比特差错吗? 为什么?

G 可被 11 (binary) 整除, 而奇数个 1 的数 / G, 余数不为 0

P8. 在 5.3 节中, 我们提供了时隙 ALOHA 效率推导的概要。在本习题中, 我们将完成这个推导。

a. 前面讲过, 当有 N 个活跃结点时, 时隙 ALOHA 的效率是 $Np(1-p)^{N-1}$ 。求出使这个表达式最大化的 p 值。

b. 使用在 (a) 中求出的 p 值, 令 N 接近于无穷, 求出时隙 ALOHA 的效率。(提示: 当 N 接近于无穷时, $(1 - 1/N)^N$ 接近于 $1/e$ 。)

P9. 说明纯 ALOHA 的最大效率是 $1/(2e)$ 。注意: 如果你完成了上面的习题, 本习题就很简单了。

P10. 考虑两个结点 A 和 B, 它们都使用时隙 ALOHA 协议来竞争一个信道。假定结点 A 比结点 B 有更多的数据要传输, 并且结点 A 的重传概率 p_A 比结点 B 的重传概率 p_B 要大。

a. 给出结点 A 的平均吞吐量的公式。具有这两个结点的协议的总体效率是多少?

b. 如果 $p_A = 2p_B$, 结点 A 的平均吞吐量比结点 B 的要大两倍吗? 为什么? 如果不是, 你能够选择什么样的 p_A 和 p_B 使得其成立?

c. 一般而言, 假设有 N 个结点, 其中的结点 A 具有重传概率 $2p$ 并且所有其他结点具有重传概率 p 。给出表达式来计算结点 A 和其他任何结点的平均吞吐量。

P11. 假定 4 个活跃结点 A、B、C 和 D 都使用时隙 ALOHA 来竞争访问某信道。假设每个结点有无限个分组要发送。每个结点在每个时隙中以概率 p 尝试传输。第一个时隙编号为时隙 1, 第二个时隙编号为时隙 2, 等等。

a. 结点 A 在时隙 5 中首先成功的概率是多少?

b. 某个结点 (A、B、C 或 D) 在时隙 4 中成功的概率是多少?

c. 在时隙 3 中出现首个成功的概率是多少?

d. 这个 4 结点系统的效率是多少?

P12. 对 N 的下列值, 画出以 p 为函数的时隙 ALOHA 和纯 ALOHA 的效率。

a. $N = 15$ 。

b. $N = 25$ 。

c. $N = 35$ 。

P13. 考虑具有 N 个结点和传输速率为 R bps 的一个广播信道。假设该广播信道使用轮询进行多路访问 (有一个附加的轮询结点)。假设从某结点完成传输到后续结点允许传输之间的时间量 (即轮询时延) 是 d_{poll} 。假设在一个轮询周期中, 一个给定的结点允许至多传输 Q 比特。该广播信道的最大吞吐量是多少?

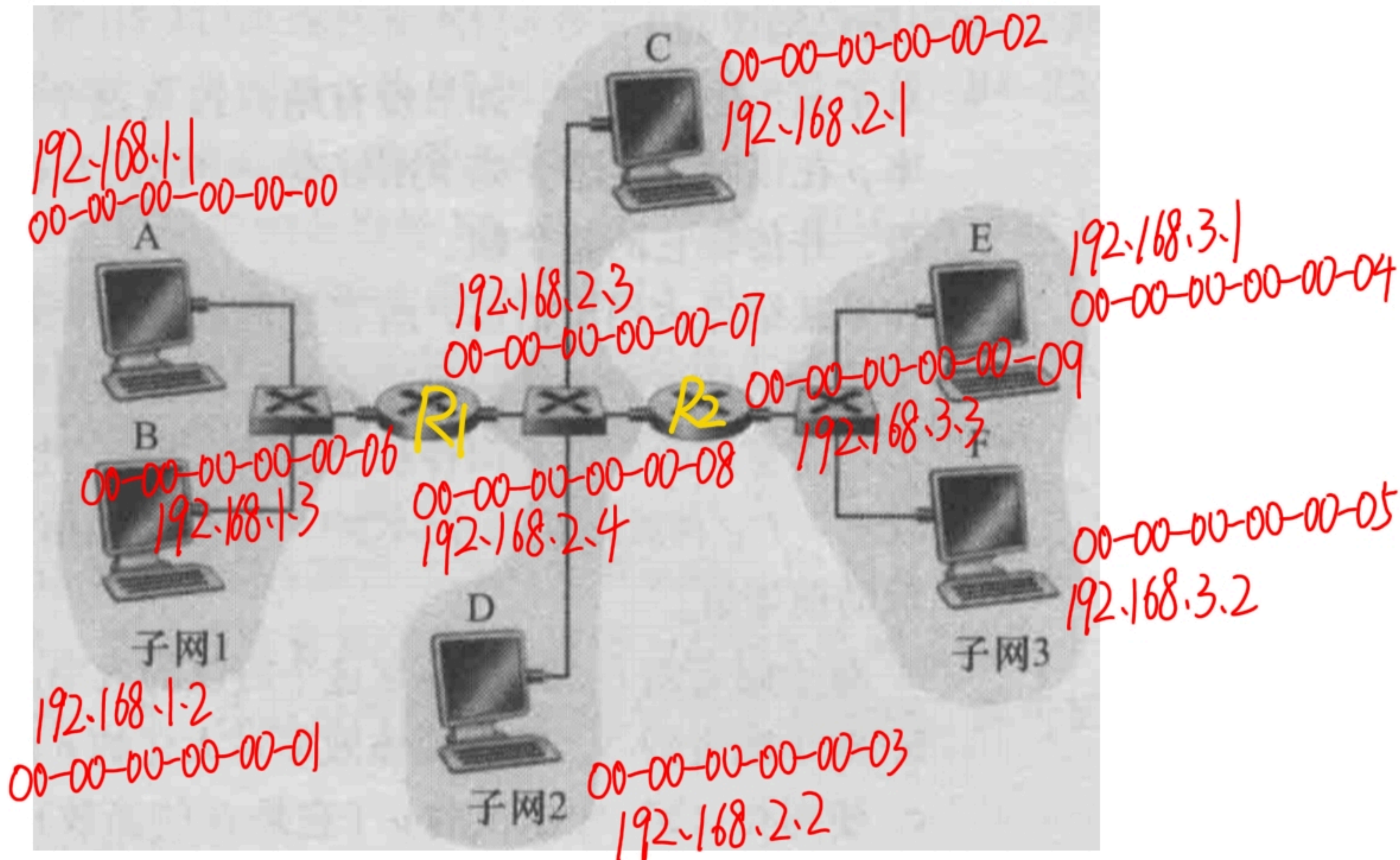
P14. 如图 5-33 所示, 考虑通过两台路由器互联的 3 个局域网。

a. 对所有的接口分配 IP 地址。对子网 1 使用形式为 192.168.1.xxx 的地址, 对子网 2 使用形式为 192.168.2.xxx 的地址, 对子网 3 使用形式为 192.168.3.xxx 的地址。

b. 为所有的适配器分配 MAC 地址。

P14. 如图 5-33 所示, 考虑通过两台路由器互联的 3 个局域网。

- 对所有的接口分配 IP 地址。对子网 1 使用形式为 192.168.1.xxx 的地址, 对子网 2 使用形式为 192.168.2.xxx 的地址, 对子网 3 使用形式为 192.168.3.xxx 的地址。
- 为所有的适配器分配 MAC 地址。→ 只要有网卡(路由器两端), 就需分配 !!!
- 考虑从主机 E 向主机 B 发送一个 IP 数据报。假设所有的 ARP 表都是最新的。就像在 5.4.1 节中对单路由器例子所做的那样, 列举出所有步骤。
- 重复 (c), 现在假设在发送主机中的 ARP 表为空 (并且其他表都是最新的)。



d. 均在内网中, 地址不变

E发ARP广播包, R2应答, 告知R2的MAC(后面均如此)

E → R2: src MAC: 00-00-00-00-00-04
dest MAC: 00-00-00-00-00-09

R2 → R1: src MAC: 00-00-00-00-00-08
dest MAC: 00-00-00-00-00-07

R1 → B: src MAC: 00-00-00-00-00-06
dest MAC: 00-00-00-00-00-01

c. 考虑从主机 E 向主机 B 发送一个 IP 数据报。假设所有的 ARP 表都是最新的。就像在 5.4.1 节中对单路由器例子所做的那样，列举出所有步骤。

d. 重复 (c)，现在假设在发送主机中的 ARP 表为空（并且其他表都是最新的）。

P15. 考虑图 5-33。现在我们用一台交换机代替子网 1 和子网 2 之间的路由器，并且将子网 2 和子网 3 之间的路由器标记为 R1。

a. 考虑从主机 E 向主机 F 发送一个 IP 数据报。主机 E 将请求路由器 R1 帮助转发该数据报吗？为什么？在包含 IP 数据报的以太网帧中，源和目的 IP 和 MAC 地址分别是什么？
E Mac F Mac

b. 假定 E 希望向 B 发送一个 IP 数据报，假设 E 的 ARP 缓存中不包含 B 的 MAC 地址。E 将执行 ARP 查询来发现 B 的 MAC 地址吗？为什么？在交付给路由器 R1 的以太网帧（包含发向 B 的 IP 数据报）中，源和目的 IP 和 MAC 地址分别是什么？
E Mac R1 右端 Mac

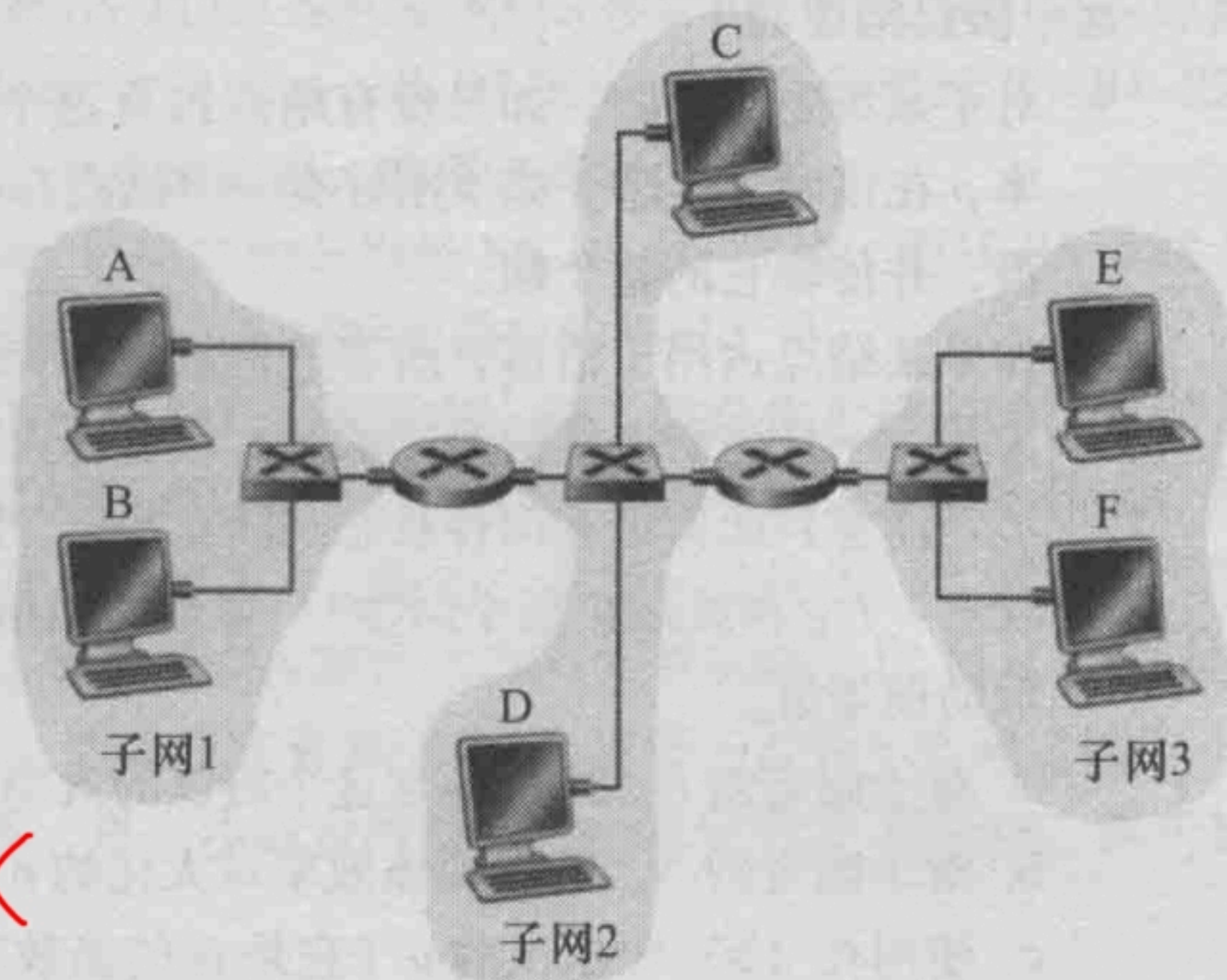


图 5-33 由路由器互联的 3 个子网

c. 假定主机 A 希望向主机 B 发送一个 IP 数据报，A 的 ARP 缓存不包含 B 的 MAC 地址，B 的 ARP 缓存也不包含 A 的 MAC 地址。进一步假定交换机 S1 的转发表仅包含主机 B 和路由器 R1 的表项。因此，A 将广播一个 ARP 请求报文。一旦交换机 S1 收到 ARP 请求报文将执行什么动作？路由器 R1 也会收到这个 ARP 请求报文吗？如果收到的话，R1 将向子网 3 转发该报文吗？一旦主机 B 收到这个 ARP 请求报文，它将向主机 A 回发一个 ARP 响应报文。但是它将发送一个 ARP 查询报文来请求 A 的 MAC 地址吗？为什么？一旦交换机 S1 收到来自主机 B 的一个 ARP 响应报文，它将做什么？
广播

P16. 考虑前面的习题，但是现在假设用一台交换机代替子网 2 和子网 3 之间的路由器，在这种新的场景中回答前面习题中的问题 (a) ~ (c)。

P17. 前面讲过，使用 CSMA/CD 协议，适配器在碰撞之后等待 $K \cdot 512$ 比特时间，其中 K 是随机选取的。对于 $K=100$ ，对于一个 10Mbps 的广播信道，适配器返回到第二步要等多长时间？对于 100Mbps 的广播信道来说呢？
 $\frac{100 \times 512 \text{ bit}}{10 \times 10^6 \text{ bit/s}} = 5.12 \times 10^{-3} \text{ s}$

P18. 假设结点 A 和结点 B 在同一个 10Mbps 广播信道上，这两个结点的传播时延为 325 比特时间。假设对这个广播信道使用 CSMA/CD 和以太网分组。假设结点 A 开始传输一帧，并且在它传输结束之前结点 B 开始传输一帧。在 A 检测到 B 已经传输之前，A 能完成传输吗？为什么？如果回答是可以，则 A 错误地认为它的帧已成功传输而无碰撞。提示：假设在 $t=0$ 比特时刻，A 开始传输一帧。在最坏的情况下，A 传输一个 $512 + 64$ 比特时间的最小长度的帧。因此 A 将在 $t=512 + 64$ 比特时刻完成帧的传输。如果 B 的信号在比特时间 $t=512 + 64$ 比特之前到达 A，则答案是否定的。在最坏的情况下，B 的信号什么时候到达 A？
 $t_{\text{trans}} = 512 + 64$ 比特时间，B 最晚 324 开始，324 + 324 到达 A 处，此时 A 已发送完

P19. 假设结点 A 和结点 B 在相同的 10Mbps 广播信道上，并且这两个结点的传播时延为 245 比特时间。假设 A 和 B 同时发送以太网帧，帧发生了碰撞，然后 A 和 B 在 CSMA/CD 算法中选择不同的 K 值。假设没有其他结点处于活跃状态，来自 A 和 B 的重传会碰撞吗？为此，完成下面的例子就足以说明问题了。假设 A 和 B 在 $t=0$ 比特时间开始传输。它们在 $t=245$ 比特时间都检测到了碰撞。假设 $K_A=0$ ， $K_B=1$ 。B 会将它的重传调整到什么时候？A 在什么时候开始发送？（注意：这些结点在返回第 2 步之后，必须等待一个空闲信道，参见协议。）A 的信号在什么时候到达 B 呢？B 在它预定的时刻抑制传输吗？

P20. 在这个习题中，你将对一个类似于 CSMA/CD 的多路访问协议的效率进行推导。在这个协议中，时间分为时隙，并且所有适配器都与时隙同步。然而，和时隙 ALOHA 不同的是，一个时隙的长度

P19. 假设结点 A 和结点 B 在相同的 10Mbps 广播信道上，并且这两个结点的传播时延为 245 比特时间。假设 A 和 B 同时发送以太网帧，帧发生了碰撞，然后 A 和 B 在 CSMA/CD 算法中选择不同的 K 值。假设没有其他结点处于活跃状态，来自 A 和 B 的重传会碰撞吗？为此，完成下面的例子就足以说明问题了。假设 A 和 B 在 $t=0$ 比特时间开始传输。它们在 $t=245$ 比特时间都检测到了碰撞。假设 $K_A=0$, $K_B=1$ 。B 会将它的重传调整到什么时间？A 在什么时间开始发送？（注意：这些结点在返回第 2 步之后，必须等待一个空闲信道，参见协议。）A 的信号在什么时间到达 B 呢？B 在它预定的时刻抑制传输吗？

$t=0$, A、B 开始发包

$t=245$, 测得碰撞, 发送 jam 包

$t=293$ (?), jam 发送完毕

$t=293+245=538$, 另一端的 jam 到达

$t=538+96=634$, 测得链路空闲, A 开始发送

$t=634+245=879$, A 到达

对于 B, $t=293+512 \times 1=805$ 后,

$t=805+96=901$, 测得链路空闲, 开始发包

$t=901+245=1146$, B 到达

(以秒计) 比一帧的时间 (即传输一帧的时间) 小得多。令 S 表示一个时隙的长度。假设所有帧都有恒定长度 $L = kRS$, 其中 R 是信道的传输速率, k 是一个大整数。假定有 N 个结点, 每个结点都有无穷多帧要发送。我们还假设 $d_{\text{prop}} < S$, 以便所有结点在一个时隙时间结束之前能够检测到碰撞。这个协议描述如下:

- 对于某给定的时隙, 如果没有结点占有这个信道, 所有结点竞争该信道; 特别是每个结点以概率 p 在该时隙传输。如果刚好有一个结点在该时隙中传输, 该结点在后续的 $k-1$ 个时隙占有信道, 并传输它的整个帧。
- 如果某结点占用了信道, 所有其他结点抑制传输, 直到占有信道的这个结点完成了该帧的传输为止。一旦该结点传输完它的帧, 所有结点竞争该信道。

注意到此信道在两种状态之间交替: “生产性状态” (它恰好持续 k 个时隙) 和 “非生产性状态” (它持续随机数个时隙)。显然, 该信道的效率是 $k/(k+x)$, 其中 x 是连续的非生产性时隙的期望值。

- a. 对于固定的 N 和 p , 确定这个协议的效率。
 - b. 对于固定的 N , 确定使该效率最大化的 p 值。
 - c. 使用在 (b) 中求出的 p (它是 N 的函数), 确定当 N 趋向无穷时的效率。
 - d. 说明随着帧长度变大, 该效率趋近于 1。
- P21. 现在考虑习题 P14 中的图 5-33。对主机 A、两台路由器和主机 F 的各个接口提供 MAC 地址和 IP 地址。假定主机 A 向主机 F 发送一个数据报。当在下列场合传输该帧时, 给出在封装该 IP 数据报的帧中的源和目的 MAC 地址: (i) 从 A 到左边的路由器; (ii) 从左边的路由器到右边的路由器; (iii) 从右边的路由器到 F。还要给出到达每个点时封装在该帧中的 IP 数据报中的源和目的 IP 地址。
- P22. 现在假定在图 5-33 最左边的路由器被一台交换机替换。主机 A、B、C 和 D 和右边的路由器以星形方式与这台交换机相连。当在下列场合传输该帧时, 给出在封装该 IP 数据报的帧中的源和目的 MAC 地址: (i) 从 A 到左边路由器; (ii) 从左边路由器到右边的路由器; (iii) 从右边的路由器到 F。还要给出到达每个点时封装在该帧中的 IP 数据报中源和目的 IP 地址。
- P23. 考虑图 5-15。假定所有链路都是 100Mbps。在该网络中的 9 台主机和两台服务器之间, 能够取得的最大总聚合吞吐量是多少? 你能够假设任何主机或服务器能够向任何其他主机或服务器发送分组。为什么?
- P24. 假定在图 5-15 中的 3 台连接各系的交换机用集线器来代替。所有链路是 100Mbps。现在回答习题 P23 中提出的问题。
- P25. 假定在图 5-15 中的所有交换机用集线器来代替。所有链路是 100Mbps。现在回答在习题 23 中提出的问题。
- P26. 在某网络中标识为 A 到 F 的 6 个结点以星形与一台交换机连接。考虑在该网络环境中某个正在学习的交换机的运行情况。假定: (i) B 向 E 发送一个帧; (ii) E 向 B 回答一个帧; (iii) A 向 B 发送一个帧; (iv) ~~B 向 A 回答一个帧~~ ^{不变}。该交换机表初始为空。显示在这些事件的前后该交换机表的状态。对于每个事件, 指出在其上面转发传输的帧的链路, 并简要地评价你的答案。
- P27. 在这个习题中, 我们探讨用于 IP 语音应用的小分组。小分组长度的一个主要缺点是链路带宽的较大比例被首部字节所消耗。基于此, 假定分组是由 P 字节和 5 字节首部组成。
- a. 考虑直接发送一个数字编码语音源。假定该源以 128kbps 的恒定速率进行编码。假设每个源向网络发送分组之前每个分组被完全填充。填充一个分组所需的时间是分组化时延 (packetization delay)。根据 L , 确定分组化时延 (以毫秒计)。
 - b. 大于 20 毫秒的分组化时延会导致一个明显的、令人不快的回音。对于 $L = 1500$ 字节 (大致对应于一个最大长度的以太网分组) 和 $L = 50$ 字节 (对应于一个 ATM 信元), 确定该分组化时延。
 - c. 对 $R = 622\text{Mbps}$ 的链路速率以及 $L = 1500$ 字节和 $L = 50$ 字节, 计算单台交换机的存储转发时延。
 - d. 对使用小分组长度的优点进行评述。

- P28. 考虑图 5-25 中的单个交换 VLAN，假定一台外部路由器与交换机端口 1 相连。为 EE 和 CS 的主机和路由器接口分配 IP 地址。跟踪从 EE 主机向 CS 主机传送一个数据报时网络层和链路层所采取的步骤（提示：重读课文中对图 5-19 的讨论）。
- P29. 考虑显示在图 5-29 中的 MPLS 网络，假定路由器 R5 和 R6 现在是 MPLS 使能的。假定我们要执行流量工程，使从 R6 发往 A 的分组要经 R6-R4-R3-R1 交换到 A，从 R5 发向 A 的分组要过 R5-R4-R2-R1 交换。给出 R5 和 R6 中的 MPLS 表以及在 R4 中修改的表，使得这些成为可能。
- P30. 再次考虑上一个习题中相同的场景，但假定从 R6 发往 D 的分组经 R6-R4-R3 交换，而从 R5 发往 D 的分组经 R4-R2-R1-R3 交换。说明为使这些成为可能在所有路由器中的 MPLS 表。
- P31. 在这个习题中，你将把已经学习过的因特网协议的许多东西拼装在一起。假设你走进房间，与以太网连接，并下载一个 Web 页面。从打开 PC 电源到得到 Web 网页，发生的所有协议步骤是什么？假设当你给 PC 加电时，在 DNS 或浏览器缓存中什么也没有。（提示：步骤包括使用以太网、DHCP、ARP、DNS、TCP 和 HTTP 协议。）明确指出在这些步骤中你如何获得网关路由器的 IP 和 MAC 地址。
- P32. 考虑在图 5-30 中具有等级拓扑的数据中心网络。假设现在有 80 对流，在第 1 和第 9 机架之间有 10 个流，在第 2 和第 10 机架之间有 10 个流，等等。进一步假设网络中的所有链路是 10Gbps，而主机和 TOR 交换机之间的链路是 1Gbps。
- 每条流具有相同的数据率；确定一条流的最大速率。
 - 对于相同的流量模式，对于图 5-31 中高度互联的拓扑，确定一条流的最大速率。
 - 现在假设有类似的流量模式，但在每个机架上涉及 20 台主机和 160 对流。确定对这两个拓扑的最大流速率。
- P33. 考虑图 5-30 中所示的等级网络，并假设该数据中心需要在其他应用程序之间支持电子邮件和视频分发。假定 4 个服务器机架预留用于电子邮件，4 个服务器机架预留用于视频。对于每个应用，所有 4 个机架必须位于某单一二层交换机之下，因为二层到一层链路没有充足的带宽来支持应用内部的流量。对于电子邮件应用，假定 99.9% 时间仅使用 3 个机架，并且视频应用具有相同的使用模式。
- 电子邮件应用需要使用第 4 个机架的时间比例有多大？视频应用需要使用第 4 个机架的时间比例有多大？
 - 假设电子邮件使用和视频使用是独立的，这两个应用需要其第 4 个机架的时间比例有多大（等价地，概率有多大）？
 - 假设对于一个应用服务器短缺的时间具为 0.001% 或更少（引起用户在极短时间内性能恶化）。讨论在图 5-31 中的拓扑能够怎样使用，使得仅 7 个机架被共同地分配给两个应用（假设拓扑能够支持所有流量）。



Wireshark 实验

在与本教科书配套的 Web 站点 (<http://www.awl.com/kurose-rose>) 上，你将找到一个 Wireshark 实验，该实验研究了 IEEE 802.3 协议的操作和以太网帧格式。第二，Wireshark 实验研究了在家庭网络场景下所获取的分组踪迹。

人物专访

Simon S. Lam 是位于奥斯丁的 T 得克萨斯大学计算机科学系的教授和董事会主席。1971~1974 年，他在 UCLA 的 ARPA 网络测量中心工作，从事卫星和无线电分组交换方面的工作。他领导的研究组于 1993 年发明了安全套接字及其原型，这是第一个安全套接字层（称为安全网络编程），为此赢得了 2004 年 ACM 软件系统奖。他的研究兴趣在于网络协议和安全服务的设计和分析。他从华盛顿州立大学获得了电子工程学士学位（BSEE），从 UCLA 获得了硕士和博士学位。他于 2007 年当选美国国家工程院院士。



Simon S. Lam

WLAN

P2P

HTTP-80

RTT

200

301

304

400

404

CDN

DNS (实现UDP)

TLID

RR

A

CNAME

C/S

FTP < 21

20

SMTP-25

POP-110

Demux { UDP
TCP
IP
以太网 }

UDP

best effort service

GBN > 区别 < ack
SR < timer
window

TCP

MSS

MTU

ISN

SYN

FIN

AIMD

round

cwnd

SS

CA

HOL Blocking

IP

TTL

IP fragmentation

frag flag = 1, 标志
offset 需 ÷ 8

NIC

CIDR

DHCP { discover (广播)
offer.
request (广播)
ack

NAT

内网 < 192.168
172.16-172.31
10.

L-S

step	N'	$D(\cdot)/p$	$D(\cdot)/p$	$D(\cdot)/p$
------	------	--------------	--------------	--------------

path oscillations

D-V

	x	y	z
node x	x	x	x
node y	x	y	y
node z	x	y	z

AS

RIP (L-S)

OSPF (D-V)

IGP

BGP

ip地址族

- A/8
- B/16
- C/24
- D 广播
- E 保留

MAC

EDC bits

CRC

- 左移
- XOR除
- 余数拼在后面

TDMA

FDMA

Slotted ALOHA

pure ALOHA

CSMA

CSMA/CD

二进制退避: m次碰撞

$\{0, 2^0, 2^1, \dots, 2^{m-1}\}$

$\Rightarrow$ wait $\downarrow$ $\times 512$ bit 时间

Polling

Token Passing

ARP

- query 广播
- reply 单点

ICMP