简答/计算/大题押题

概述

服务与协议的比较。

- 服务是垂直的，是协议栈中下层提供给上层的。
- 协议是水平的，是通信双方协议栈中两个对等实体相互通信的规则约定。
- 实体通过协议实现自己定义的服务，又通过调用下层服务来实现协议定义的交互动作。

分组交换和电路交换的比较。

- 分组交换：将报文划分成小的分组，将小分组存储转发。
  - 优点：
    - 适合突发流量场景，灵活；
    - 更高的带宽利用率，支持更多用户，更高性能；
    - 不需共享带宽，不需预留带宽，按需使用带宽；
    - 更简单，成本低。
  - 缺点：
    - 包可能会乱序到达；
    - 资源竞争 starvation，可能会拥塞，可能会丢包；
    - 存储转发时延。
- 电路交换：在通信双方之间建立一条物理连接，预留通信资源（带宽 缓存等），直接传输。
  - 优点：适合稳定流量场景。
  - 缺点：建立/撤销连接的时间，更昂贵，支持用户少，带宽利用率低，预留带宽的浪费。

解释存储转发、尽力而为。

- 存储转发：中间每个节点接收、存储完整的分组，然后将这个分组转发给下一个节点。
- 尽力而为：尽最大努力将数据传输给对方，但服务不可靠，没有保证不丢包 / 正确 / 按序 / 保证时延 / 流量控制 / 拥塞控制。

算存储转发 / 电路交换时延。

应用层

网络应用程序的体系结构：CS和P2P的比较。

- cs：主从的一对多关系，信息的存储管理集中稳定，容易实现，有server的单点失效问题。
- P2P：宏观上每个主机都是对等的，强调peer的对等性，但具体到一次通信过程仍存在cs。可扩展性好，适用于文件分发，时间短，带宽少（带宽利用率高，利用了每一个peer的带宽）。

填seq# ack#。

- ack# 是接下来想要的包的序号，seq# 是这个包的序号。

三次握手 & 四次挥手。

- SYN → SYN ACK → ACK。
- FIN → ACK → FIN → ACK。

cookie是什么，有什么作用。

- HTTP连接是无状态的，cookie用来改进这一情况，记录用户状态信息（不需要注册账号）。
- cookie的传输与存储：
    - 请求报文的cookie首部行。
    - 响应报文的cookie首部行。
    - 用户端系统存放cookie文件，浏览器进行管理。
    - web站点维护后端数据库。
- 应用：认证、推荐、用户会话状态。
- 隐私泄露的风险：拿到别人的cookie，就可以假装成那个人。https是加密的，可以避免cookie泄露。

【email整个过程】

- A给B发邮件，AB两方都拥有user agent和mail server。
- A在user agent写好邮件，ua通过 SMTP TCP 25 / HTTP TCP 80(浏览器写邮件) 把邮件push给A的mail server。
- A的ms通过SMTP连接把邮件push给B的ms。
- 等B收邮件的时候，B的ua使用 POP3 TCP 110 / IMAP TCP 143 / HTTP TCP 80 把邮件从B的ms pull到B的ua，B就可以看邮件了。
- 大家都是TCP：邮件过程需要数据按序、无差错地传输。

SMTP与HTTP的比较。

- 相同点：都是主机间传输文件，都调用TCP，持续的HTTP和SMTP都使用持久连接，都使用状态码。
- 不同点：
    - SMTP把所有对象封装在一个报文里，HTTP对每个对象分别发一个报文。
    - SMTP是push，HTTP是pull。
    - SMTP要求报文用7位ASCII码编码，HTTP没有这个限制。

DNS迭代查询和递归查询的比较。

- 迭代查询：
    - user 的 DNS client 请求 local name server；
    - local server 向 root DNS server 询问顶级域名 com 的 Top-Level Domain DNS server；
    - 然后 local server 再询问 TLD server 得到abc.com的 authoritative DNS server；
    - local server 再询问 authoritative server 得到 abc.com 的IP地址；
    - local server 将IP地址返回给 user 的 DNS client。
- 递归查询：
    - user 的 DNS client 请求 local name server；
    - local server 向 root DNS server 请求 abc.com 的 IP地址；
    - root server 请求顶级域名 com 的 TLD server，询问 abc.com 的 IP地址；
    - TLD server 再请求 abc.com 的 authoritative server，得到 abc.com 的IP地址；
    - TLD server 将IP地址返回 root server；
    - root server 将IP地址返回 local server；
    - local server 将IP地址返回 user 的 DNS client。
- 迭代查询 local server 负载更大（多次询问 DNS server），递归查询整个DNS系统负载更大（DNS server 互相询问）。

传输层

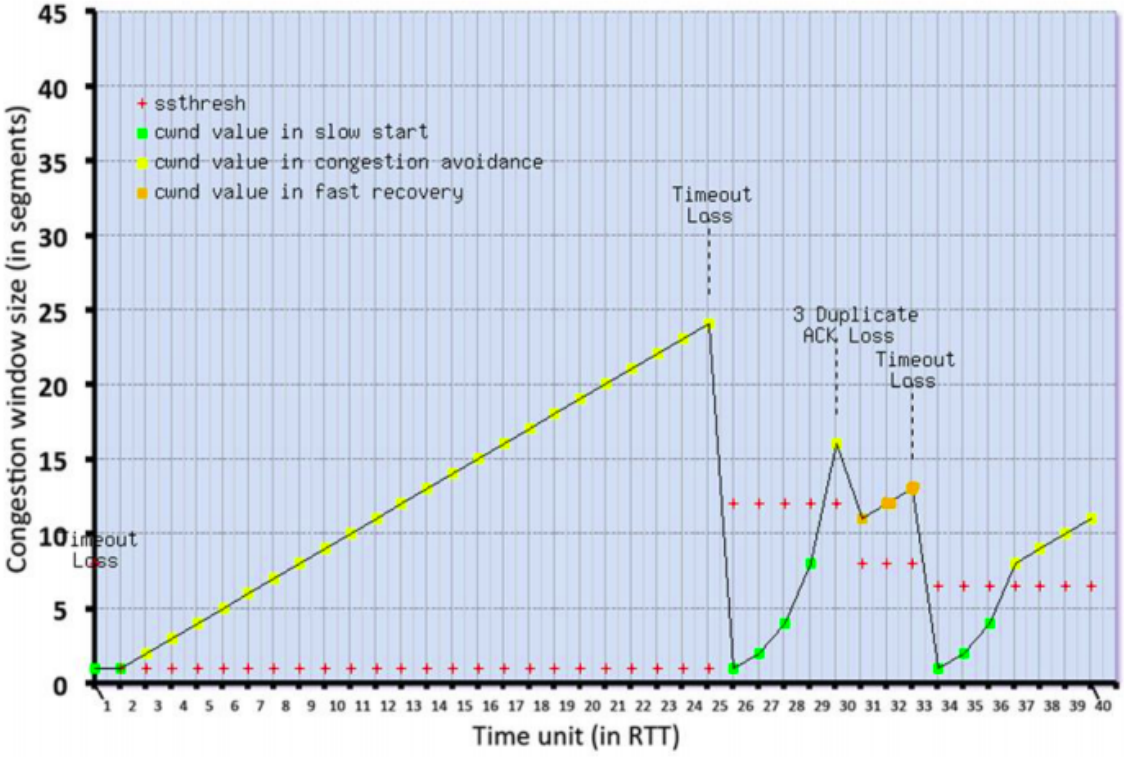
TCP/UDP 多路复用分解的区别。

- UDP：传输层数据段到达主机后，仅需要通过 目的IP地址+端口号 来定位数据交付的socket。
- TCP：需要通过 (源IP地址, 源端口号, 目的IP地址, 目的端口号) 四元组，来定位数据交付的socket。大概是因为，一个socket只能维护一个TCP连接，因此即使你们请求的都是我的熟知端口号，我也要分流一下……?
- 对于TCP：端口号 → 多个socket → 多个TCP连接 → 多个进程。UDP不能这样。

拥塞控制 & 流量控制的区别。

- 拥塞控制：控制sender发送包的速率，使数据进入网络的速度不会导致网络过载，或出现拥塞时减少进入网络的数据流量。TCP通过根据timeout、duplicate ACK等信息，控制发送窗口大小来实现。
- 流量控制：控制sender发包速率与receiver向上层交付的速率相匹配。通过TCP首部字段的 window size 来实现。

TCP拥塞控制，cwnd变化图。



计算UDP校验和。

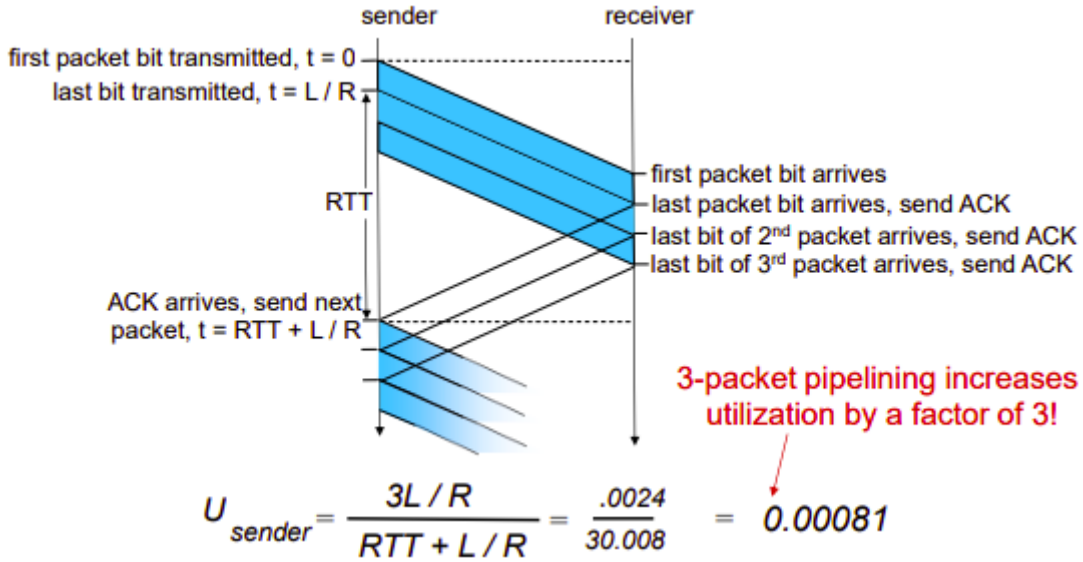
- 01111001 10111001 11101010 00001100
- 16位数相加，进位再相加：01100011 11000110
- 反码：10011100 00111001，即校验和。

简述rdt协议的推导过程，总结可靠性机制。

- rdt1.0：完全可靠的信道。
- rdt2.0：bit翻转的信道：引入ACK / NAK。sender 2个状态 等上层要求发包 → 发完包等ACK，receiver 1个状态。
- rdt2.1：考虑ACK/NAK受损：引入seq#。sender 发0 → 等0的ACK → 发1 → 等1的ACK。receiver 收0 → 收1。

- rdt2.2：使用上一个seq#的ACK代替NAK。
- rdt3.0：bit翻转+丢包的信道，alternating-bit protocol。引入timer，sender一段时间后等不到ACK就重传。
- GBN：
  - 滑动窗口机制：保证流水线中已发送未确认的包的数量≤N。
  - 仅对窗口最左端未确认的包，使用一个计时器。
  - 超时后，重传所有未确认的包。
  - receiver丢弃乱序到来的包，使用累计确认，没有NAK。
- SR：
  - 滑动窗口机制：保证流水线中已发送未确认的包的数量≤N，receiver的窗口大小需要≥N。
  - 对每个未确认的包，使用一个计时器。
  - 一个包超时后，单独重传。
  - receiver缓存乱序到来的包，发送单个确认（而非累计确认），按序交付给上层。
- 可靠性机制：校验和，序列号，停等，ACK，计时+超时重传，累计ACK，快速重传。
- TCP的应用：
  - 校验和，连接管理（三次握手+四次挥手），流量控制。
  - 拥塞控制：序列号，计时+超时重传，累计ACK，单个重传，快速重传。
  - 【TCP的seq#，是字节流编号！】
  - 【仅对最早未确认的报文段，使用一个重传定时器】
  - 【仅在超时后，重发最早未确认的报文段】
  - 【收到 3 duplicate ACK，在超时前快速重传该segment】

计算TCP可靠传输流水线的信道利用率。



【TCP & UDP 的 原理 & 应用场景】

- UDP：
  - 原理：尽力而为的、无连接的不可靠传输。
  - 特点：对数据的正确性、完整性和顺序要求不高，传输代价少、快速高效。
  - 应用场景：适合少量数据传输，适合流媒体传输，适合支持大量活跃客户。
- TCP：
  - 原理：面向连接的、提供流量控制 & 拥塞控制的可靠传输。
  - 特点：适用于对数据可靠性（正确性、完整性和顺序）要求高的场景
  - 应用场景：在线交易，HTTP，电子邮件，telnet。

网络层

IP地址划分子网，写子网路由表。

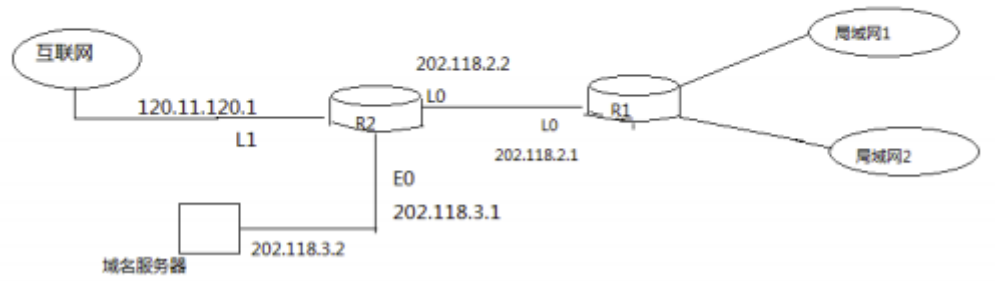
8. 某网络拓扑结构如下图所示，路由器 R1 通过接口 E1，E2 连接局域网 1 和 2，通过接口 L0 连接路由器 R2，并通过 R2 连接域名服务器和互联网。R1 的 L0 接口的 IP 地址是 202.118.2.1/24，R2 的 L0 接口的 IP 地址是 202.118.2.2/24，L1 接口的 IP 地址是 100.11.120.1/24，E0 接口的 IP 地址是 202.118.3.1/24，域名服务器的 IP 地址是 202.118.3.2/24。

路由器 R1 和 R2 的路由表结构为：

目的网络地址	子网掩码	下一跳 IP 地址	接口

试解决以下问题：（20 分）

1. 将 IP 地址空间 202.118.1.0/24 划分为两个子网，分别分配给局域网 1 和局域网 2，每个局域网 IP 地址数不少于 120 个，请给出子网划分结果，说明理由或给出必要的计算过程。
2. 请给出 R1 的路由表，使其明确包括至局域网 1 的路由、局域网 2 的路由、域名服务器的路由和互联网的路由。



- 子网划分：

子网	子网IP
局域网1	202.118.1.0 / 25
局域网2	202.118.1.128 / 25

- 路由表：

目的网络地址	子网掩码	下一跳IP地址	接口
202.118.1.0	255.255.255.128	直达	E1
202.118.1.128	255.255.255.128	直达	E2
202.118.3.2	255.255.255.0	202.118.2.2	L0
0.0.0.0	0.0.0.0	202.118.2.2	L0

- 再来一题：
- 6. 企业组建内部网，拟将 200.1.1.0/24 网段分配给 4 个部门组建子网。若已知 4 个部门拟接入企业网的主机数分别为 72、35、20、18 台，请给出各个子网的子网掩码，子网地址和 ip 地址范围。若支持这些主机接入因特网，企业必须配置那些设施。（15’）
- 可用IP范围，要预留全0（子网地址）全1（广播）。

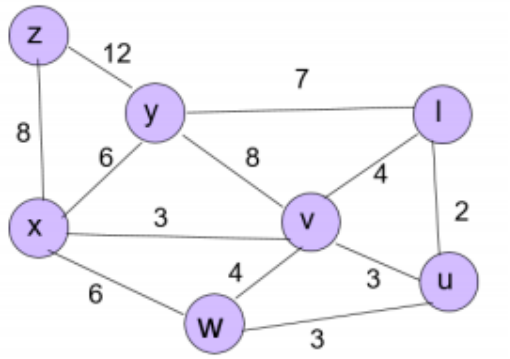
子网	子网地址	子网掩码	可用IP范围
72	200.1.1.0 / 25	255.255.255.128	1 - 126
35	200.1.1.128 / 26	255.255.255.192	129 - 190
20	200.1.1.192 / 27	255.255.255.224	193 - 221
18	200.1.1.224 / 27	255.255.255.224	225 - 254

IP报文分片+重组。

- 提醒：偏移量以 8 个字节为一个单位。
- 考虑向具有 700 字节 MTU 的一条链路发送一个（包括IP首部）2400 字节的数据报。假定初始数据报标有标识号 422。将会生成多少个分片？在生成相关分片的数据报中各个字段的值是多少？
- 分片个数 = (2400 - 20) / (700 - 20) = 3.5 → 4。
- 分片0：id=422，fragflag=1，offset=0，length=700；
- 分片1：id=422，fragflag=1，offset=85，length=700；
- 分片2：id=422，fragflag=1，offset=170，length=700；
- 分片3：id=422，fragflag=0，offset=255，length=360；

link state / distance vector 算节点间距离。

- dijk：



答：

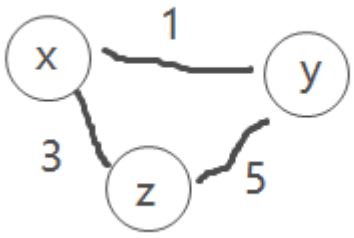
A:

步骤	N'	D(v),p(v)	D(w),p(w)	D(l),p(l)	D(u),p(u)	D(y),p(y)	D(z),p(z)
0	x	3,x	6,x	∞	∞	6,x	8,x
1	xv		6,x	7,v	6,v	6,x	8,x
2	xvw			7,v	6,v	6,x	8,x
3	xvwu			7,v		6,x	8,x
4	xvwuy			7,v			8,x
5	xvwuyl						8,x
6	xvwuylz						

B:

步骤	N'	D(v),p(v)	D(w),p(w)	D(l),p(l)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	3,u	3,u	2,u	∞	∞	∞
1	ul	3,u	3,u		∞	9,l	∞
2	ulv		3,u		6,v	9,l	∞
3	ulvw				6,v	9,l	∞
4	ulvwx					9,l	14,x
5	ulvwxy						14,x
6	ulvwxyz						

- bellman-ford：



3	u	v	w	x	y
u	0	3	7	10	12
v	3	0	4	7	9
w	∞	∞	∞	∞	∞
x	∞	∞	∞	∞	∞
y	∞	∞	∞	∞	∞

- 一开始只有邻接信息。拿别人传过来的信息更新自己：如果把你作为我的第一跳，接下来都按照你的路径走，会更近吗？

主机接入internet，需要配置什么？

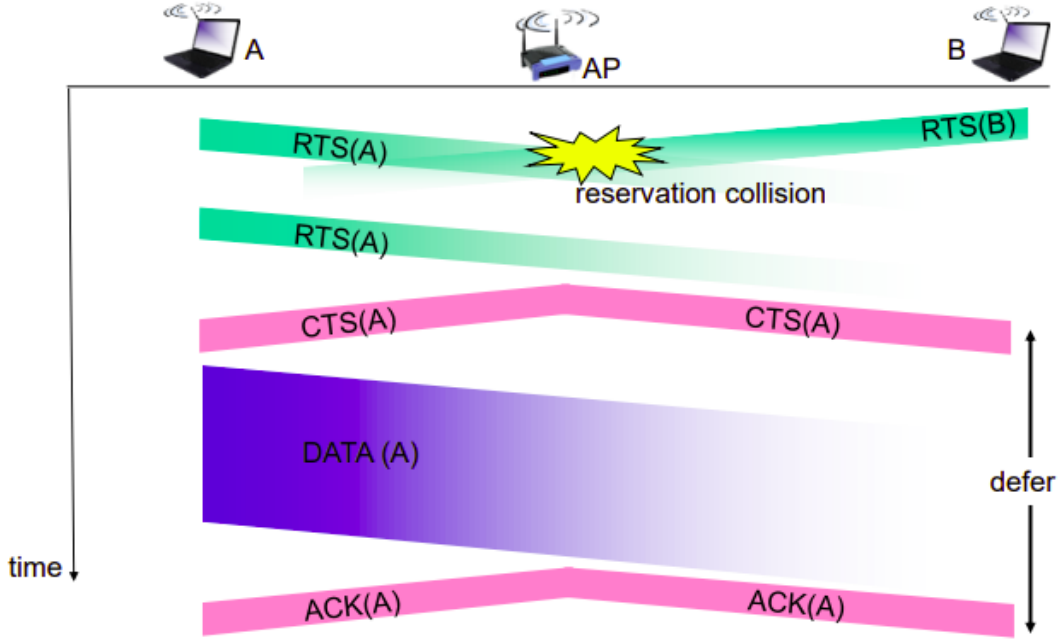
- IP地址，子网掩码（用来判断和别人是否同一子网），默认网关的IP地址（第一跳路由器），DNS服务器的IP地址。
- ARP协议（局域网内通信），IP协议（点到点传输），TCP/UDP（端到端传输），DNS协议（域名解析服务），如果动态配置IP地址 DHCP协议，HTTP/SMTP/...
- 组网：交换机，配备了 OSPF/RIP等 + BGP 的路由器，DNS服务器，ARP服务器，（DHCP服务器），物理链路。

【tracert实现原理】

- 发送不可达的UDP报文，利用ICMP回显信息，得到数据传输的路径，及各节点时延。
- 设置UDP报文的TTL字段：首先TTL=1，此时报文会到达路径的第一跳路由器，路由器将报文丢弃，并发送ICMP TTL过期报文，该报文包含路由器的IP地址，可以通过测量响应时间来得到节点时延（每一个节点测3遍）。然后TTL=2，可以得到第二跳路由器的IP大致和节点时延；然后TTL=3，...
- 当TTL=路径上节点数量，刚好可以到达目的地时，因为UDP目的端口号是30000以上的不常用端口，所以对方会发送ICMP port unreachable报文，我们即可结束tracert过程。

链路层

【简述 CSMA/CD 和 CA 的应用背景 & 工作原理】

- CSMA：载波侦听多路访问，carrier sense multiple access，指的是接入前先侦听信道是否忙碌，不忙碌再接入。
- CSMA/CD：在CSMA的基础上，如果在数据传输过程中检测到冲突，立刻停止传输。应用背景：有线传输，易于检测冲突。
- CSMA/CA：
  - 首先侦听信道，信道空闲一个DIFS后（对不同优先级设备不一样），发送RTS（request-to-send），如果RTS冲突就二进制回退。AS收到RTS，等一个SIFS后回复（广播）一个CTS（clear-to-send），此时发送者已经预约了信道。
  - 发送者收到CTS，等一个SIFS，发送数据。AP收到数据，等一个SIFS，发送ACK。
  - 

  - 应用背景：无线传输，不好检测冲突（没法CD）。预约的帧数据量小，冲突的代价小，但数据传输冲突的代价大，因此预约机制合算。

计算CRC校验码。

- G=10011（多项式），D=10001010，R?

- 先左移4位，然后进行异或的除法，结果是0011。
- 直接取余数，不要求反。

为什么链路层对帧的长度有限制。

- 不能过长：会导致其他设备长时间无法发送数据，会导致缓冲区溢出。
- 不能过短：如果发完整个帧才检测到冲突，数据损坏了，但设备以为发送成功了。因此，发送帧的过程不能过短，即帧长不能过小，要大于整个网络的最大时延位（最大时延时间内可以传输的数据位）。

简述交换机自学习的过程。

- 当帧来到交换机，交换机会记录 (帧的源MAC地址, 进入接口) 的映射。
- 交换机去查找，帧的目的MAC地址对应的接口，找不到就广播。

综合

【访问一个网页的全过程】

- 进行DHCP： discover 广播 → offer 单播或广播 → request 广播 → ack 单播，得知IP地址，子网掩码，默认网关IP地址，DNS服务器IP地址。
- DNS，询问网页的IP地址：DNS → UDP 53 → IP，成帧需要得知DNS服务器的MAC地址，于是
- ARP，得到DNS服务器的MAC地址：广播询问，单播回复。如果在子网外，则得到网关路由器MAC地址。
- DNS成帧，询问local DNS server：迭代 / 递归，root → TLD → authoritative，得到网页域名的IP地址。
- HTTP → TCP 80，需要先三次握手建立连接。
- 三次握手：SYN → SYNACK → ACK。
- HTTP GET → TCP 80 → IP → frame（MAC是网关），收到HTTP应答，browser渲染网页。
- 渲染成功，TCP四次挥手关闭连接：FIN → ACKFIN → ACK。

地址有哪些，作用是什么，如何互相映射，映射关系怎么样（一对一 / 一对多 / 多对多）。

- 地址：进程pid，端口号，IP地址+端口号，域名，IP地址，MAC地址。
- 端口号 pid os维护，域名 IP地址 DNS，IP地址 MAC地址 ARP。
- 一个端口号 → 多个socket → 多个pid。
- 一个域名 → 多个IP地址。
- 一个IP地址 → 多个MAC地址（内网地址）。一个MAC地址 → 多个IP地址（虚拟接口）。

定时器与缓存在哪里有应用？（应用层 传输层 网络层）

- 应用层：web缓存。DNS缓存，定时清理缓存。SMTP mail server 邮件发送失败会先缓存在 server 上，反复定时发送。
- 传输层：TCP可靠传输，为ACK定时。TCP流水线机制缓存包。TCP四次挥手，等待2MSL后关闭连接。
- 网络层：OSPF BGP 路由表缓存（？），OSPF 定时更新链路状态（？）
- 链路层：ARP缓存，定时清理缓存。switch缓存接口信息，定时清理缓存。无线通信，AP周期性（定时）发送信标帧，方便主机与其关联。

计算机网络的数据类型有哪些?

- data/message → TCP/UDP segment → IP datagram → link layer frame → bit。
- header，有效数据。