分层的概念

问题 1：OSI 有哪几层，会画出来，知道主要几层的各自作用

1. 应用层（数据）：确定进程之间通信的性质以满足用户需要以及提供网络与用户应用
2. 表示层（数据）：主要解决拥护信息的语法表示问题，如加密解密
3. 会话层（数据）：提供包括访问验证和会话管理在内的建立和维护应用之间通信的机制，如服务器验证用户登录便是由会话层完成的
4. 传输层（段）：实现网络不同主机上用户进程之间的数据通信，可靠与不可靠的传输，传输层的错误检测，流量控制等
5. 网络层（包）：提供逻辑地址（IP）、选路，数据从源端到目的端的传输
6. 数据链路层（帧）：将上层数据封装成帧，用 MAC 地址访问媒介，错误检测与修正
7. 物理层（比特流）：设备之间比特流的传输，物理接口，电气特性等

问题 2：知道各个层使用的是哪个数据交换设备。（交换机、路由器、网关）

1. 网关：应用层、传输层（网关在传输层上以实现网络互连，是最复杂的网络互连设备，仅用于两个高层协议不同的网络互连。网关的结构也和路由器类似，不同的是互连层。网关既可以用于广域网互连，也可以用于局域网互连）
2. 路由器：网络层（路由选择、存储转发）
3. 交换机：数据链路层、网络层（识别数据包中的 MAC 地址信息，根据 MAC 地址进行转发，并将这些 MAC 地址与对应的端口记录在自己内部的一个地址表中）
4. 网桥：数据链路层（将两个 LAN 连起来，根据 MAC 地址来转发帧）
5. 集线器（Hub）：物理层（纯硬件设备，主要用来连接计算机等网络终端）
6. 中继器：物理层（在比特级别对网络信号进行再生和重定时，从而使得它们能够在网络上传输更长的距离）

数据链路层

ARP 协议

问题 1：ARP 的作用?

ARP 为 IP 地址到对应的硬件地址提供动态映射。

问题 2：点对点链路使用 ARP 吗?

不使用

问题 3：ARP 高效运行的关键是什么?

关键是每个主机上都有一个 ARP 的高速缓存。

问题 4：ARP 报文的各个字段以及含义？

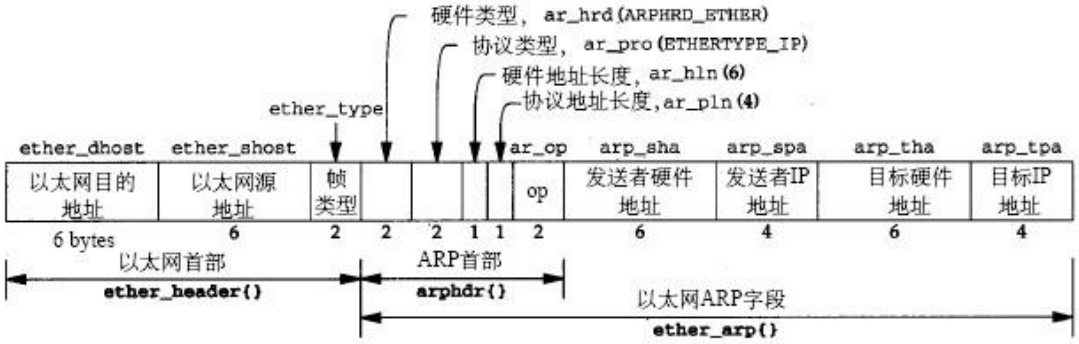


图21-7　在以太网上使用时ARP请求或回答的格式

帧类型：ARP：0x0806　(2)
ARP 首部：
硬件类型：硬件地址的类型，1 表示以太网地址。(2)
协议类型：协议地址的类型，0x0800 表示 IP 地址。(2)
硬件地址长度：字节为单位 6　(1)
协议地址长度：字节为单位 4　(1)
操作类型：2 个字节。 ARP 请求 1，ARP 回复 2，RARP 请求 3，RARP 应答 4。(2)
发送者硬件地址：6 个字节（6）
发送者 IP 地址：4 个字节（4）
目标硬件地址：6 个字节（6）
目标 IP 地址：4 个字节（4）
CRC 校验：4 个字节　(4)

总结：
arp 总共 28 个字节。
记忆方法：　以太网先目地后源，ARP 先发送端后目地端。先硬件后协议。

问题 5：ARP 协议有什么弱点？

1）缓存：主机的地址映射是基于高速缓存的，动态更新的。地址刷新是有时间限制的。 可以通过下次更新之前修改计算机上的地址缓存，造成拒绝服务攻击或者 ARP 欺骗。
2）广播：攻击者可以伪装 ARP 应答。
3）ARP 应答没有认证，都是合法的。可以在不接受到请求的时候就发出应答包。

问题 6：ARP 代理的概念和应用场景

若 ARP 请求是从一个网络的主机发送给另一个网络上的主机，那么连接这两个网络的路由器就可以回答该请求，这个过程叫做 ARP 代理。ARP 代理路由器响应 ARP 请求的 MAC 地址为路由器的 MAC 地址而非 ARP 请求的主机的 MAC 地址。
ARP 代理的应用环境：
两个物理网络之间的路由是使用相同的网络号，两个路由器设置成 ARP 代理，实现相互隐瞒物理网络

问题 7：免费 ARP

指主机发送 ARP 查找自己的 IP 地址，即数据链路层 SIP=DIP

作用有两个：

1）一个主机使用免费 ARP 确定是有存在有其他主机设置了相同的 IP 地址

2）如果发送免费 ARP 的主机改变了 MAC 地址，可以通过发送免费 ARP 的方式告知其他主机端更新 ARP 表

问题 8：数据链路层 MTU 的最大值和最小值是多少?

1）数据链路层的**最小 MTU 为 64 字节**。对于 IEEE802.3，两个站点的最远距离不超过 2500m，由 4 个中继器连接而成，其冲突窗口为 51.2us(2 倍电缆传播延迟加上 4 个中继器的双向延迟).对于 10Mbps 的 IEEE802.3 来说，这个时间等于发送 64 字节，即 512 位的时间，64 字节就是由此而来的。如果一个站点已经传输了 512bit，就认为它已经占用了这个信道。

2）数据链路层的**最大 MTU 为 1500 字节**，即数据字段的最大长度

网络层

IP 协议

问题 1：如何理解 IP 的不可靠和无连接。

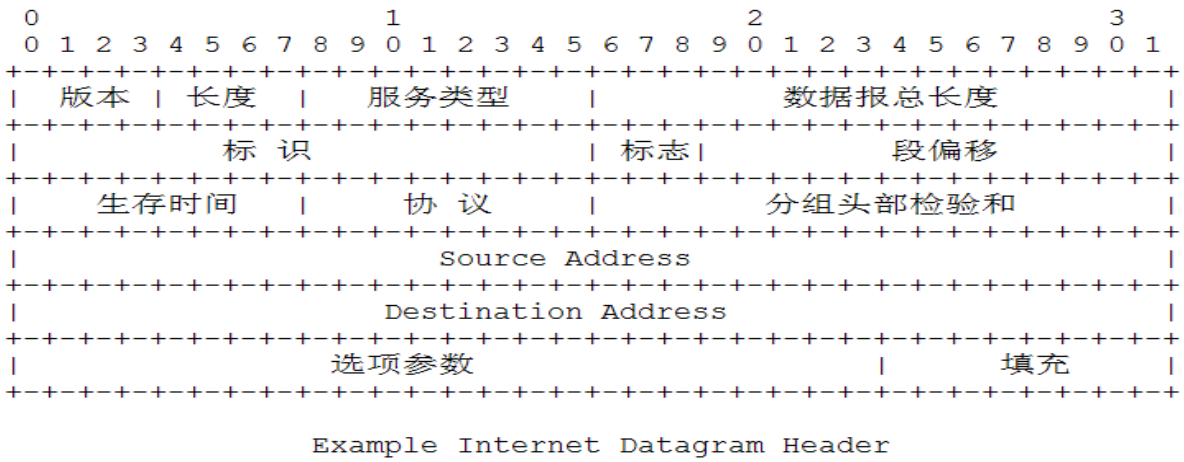
不可靠：指的是不能保证数据报能成功地到达目的地。

发生错误时候，丢弃该数据包，发送 ICMP 消息给信源端。 可靠性由上层提供。

无连接：IP 不维护关于后续数据报的状态信息。

体现在，IP 数据可以不按顺序发送和接收。A 发送连续的数据报，到达 B 不一定是连续的，来回路由选择可能不一样，路线也不一样，到达先后顺序也不一样。

问题 2：IP 报文的格式和各个字段的含义。



Example Internet Datagram Header

版本号： IPV4 就是 4，IPV6 就是 6 （4）

首部长度：4 个字节为单位。最小为 5，最大为 15。所以最小长度 20 个字节，最大为 60 个字节。(4)

服务类型： Qos 用，目前不怎么使用。(8)
总长度：字节为单位。 最多可以传送 65535 字节的 IP 数据包。(16)
标识字段（8）
标志 （3）
段偏移（5）与分片有关。
生存时间 TTL：经过一个路由器减一。 字段为 0 时，数据报被丢弃，并且发送 ICMP 报文通知源主机。目的是防止数据报在选路时无休止地在网络中流动。(8)
协议：区分上层协议 （8）
首部校验和：仅对首部进行校验。(16)【对比： ICMP，IGMP，TCP，UDP：对首部和数据进行校验】
源地址：(32)
目的地址：(32)

问题 3：为什么 IP 首部中要有总长度字段?
因为一些数据链路（以太网）需要填充一些数据以达到最小长度。因为以太网帧的最小长度是 46 个字节，但是 IP 长度可能更短，所以需要总长度来确定 IP 数据部分的内容。

问题 4：IP 首部校验和怎么计算的，与 ICMP，IGMP，TCP，UDP 的首部校验和有什么区别与共同点?
（1） 先把校验和字段置 0。
（2） 对首部中每个 16 位比特进行二进制反码求和。
（3） 结果存在检验和字段中。
（4） 收到一份 IP 数据包后，同样对首部中每个 16bit 二进制反码求和。
（5） 最后结果全为 1，表示正确，否则表示错误。
（6） 如果是错误的，IP 就丢弃该数据报，但是不生成差错报文，由上层去处理。

共同点：用到的算法都是一样的。
区别：IP 计算的时候没有将数据包括在内。
ICMP，IGMP，TCP，UDP 同时覆盖首部和数据检验码。

问题 5：主机和路由器本质区别是?
主机从不把数据报从一个接口转发到另一个接口，而路由器则要转发数据报。

问题 6：IP 路由选择的过程是怎么样的?
根据最长匹配原则，找到条目，发送到指定的路由器。如果不能找到，返回一个“主机不可达”或“网络不可达”的错误。

问题 7：IP 路由选择的特性有什么?
（1） IP 路由选择是逐跳进行的。
IP 并不知道到达任何目的的完整路径，只提供下一跳地址。
（2） 为一个网络指定一个路由器，而不是为每个主机指定一个路由器。
这样可以缩小路由表规模。

问题 8：IP 搜索路由表的步骤

搜索匹配的主机地址 ----》搜索匹配的网络地址 ----》搜索默认选项

IP 层进行的选路实际上是一种选路机制，它搜索路由表并决定向哪个网络接口发送分组。

问题 9：如果路由表中没有默认项，而又没有找到匹配项，这时如何处理?

结果取决于该 IP 数据报是由主机产生的还是被转发的。

如果数据报是由本机产生的，那么就给发送该数据报的应用程序返回一个差错，或者是“主机不可达差错”或者是“网络不可达差错”。

如果是被转发的数据报，就给原始发送一份 ICMP 主机不可达的差错报文。

问题 10：IP 地址的分类，如何划分的，及会计算各类地址支持的主机数

1. A 类地址：首位为 0，1.0.0.1~~126.255.255.254；主机号 24 位
2. B 类地址：首位为 10，128.0.0.1~~191.255.255.254；主机号 16 位
3. C 类地址：首位为 110，192.0.0.1~~223.255.255.254；主机号 8 位
4. D 类地址（多播地址，也叫做组播地址）：首位为 1110，224.0.0.1~~239.255.255.254
5. E 类地址：此类地址是保留地址，首位为 11110，240.0.0.1~~254.255.255.254

ICMP 协议

问题 1：ICMP 的层次和作用。

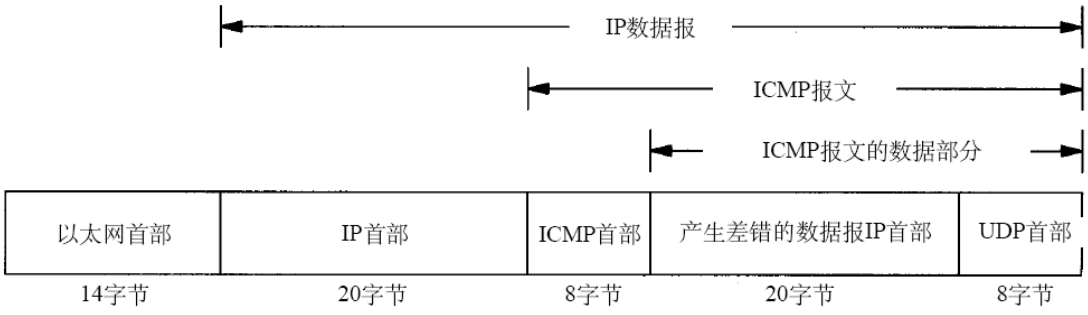


图6-9 “UDP端口不可达”例子中返回的ICMP报文

ICMP 一般认为是在三层的。主要传递一些差错报文和其他需要注意的信息。

问题 2：ICMP 报文的分类?

ICMP 分为两类，一类是 ICMP 查询报文，另一类是 ICMP 差错报文。

ICMP 报文	类型的值	ICMP 报文的类型
差错报告报文	3	终点不可达
	4	源点抑制
	11	时间超过
	12	参数问题
	5	改变路由
询问报文	8 或0	回送请求或回答
	13 或14	时间戳请求或回答

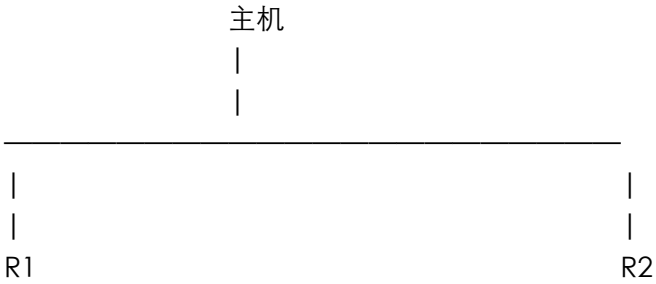
问题 3：ICMP 的主机不可达报文是在什么情况下发出的?

三层设备（路由器）给该主机寻路时，没有找到相应路径，向源 IP 发回 ICMP 主机不可达

问题 4：什么情况不会导致产生 ICMP 差错报文?

1） ICMP 差错报文。
2） 目的地址是广播地址或者多播地址的 IP 数据报。
3） 链路层广播的数据报
4） 不是 IP 分片的第一片
5） 源地址不是单个主机的数据包。

问题 5：ICMP 重定向差错报文是怎么来的，在何种场合出现?



1） 主机发送 IP 数据报给 R1，因为主机的默认路由指向的下一跳是 R1。
2） R1 收到数据报并且检查它的路由表，发现 R2 是发送该数据报的下一跳。当他将数据报发送给 R2 的时候，发现发送的接口与接受的端口是一样的，因此同时发送一个 ICMP 重定向报文给主机。
3） R1 接受到 ICMP 重定向报文后，接下来的数据报就发送给 R2，而不再发送给 R1。

问题 6：重定向报文有什么规则?

重定向报文只能有路由器生成。

重定向报文是为主机而不是为路由器使用的。

问题 7：Ping 命令的具体过程是怎么样的?

参考文章：《对于 Ping 的过程，你真的了解吗？》
https://mp.weixin.qq.com/s/DfQT3Vw2xaq60Ylil-7Yxw

传输层

UDP 协议

问题 1：UDP 和 TCP 的简单介绍。

UDP 是一个简单的面向数据报的运输层协议：进程的每个输出操作都正好产生一个 UDP 数据报，并组装成一份待发送的 IP 数据报。

TCP 是面向流字符，应用程序产生的全体数据与真正发送的单个 IP 数据报可能没什么联系。

问题 2：UDP 报头字段和含义?

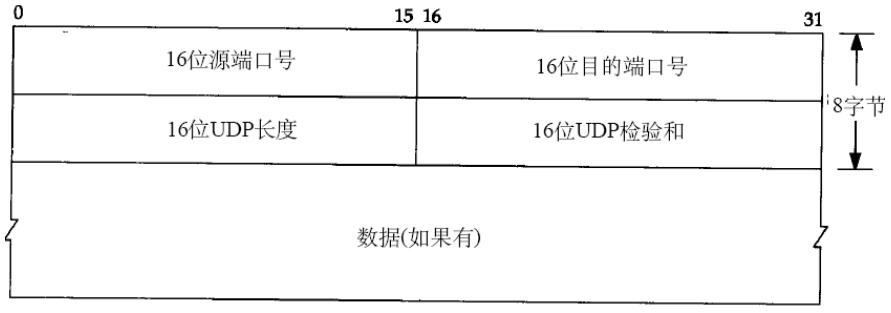


图11-2 UDP首部

源端口号（2）

目地端口号（2）

UDP 长度：是 UDP 的报文总长度，是多于的。 IP 总长度减去首部长度就是此值。(2)

UDP 校验和：注意点：校验和是可选的。 （TCP 是必选的）校验和覆盖 UDP 首部和数据（TCP 也一样覆盖首部和数据，但是 IP 指覆盖首部）(2)

问题 3： UDP 的校验和是怎么计算的?

UDP 的校验和要计算首部和数据部分。首部还包括伪首部。

图11-3 UDP检验和计算过程中使用的各个字段

多了 12 个字节的伪首部。

注意点：UDP 长度计算两次。

如果检验和有错，则 UDP 数据报被悄悄丢弃，不产生任何差错报文。

问题 4：为什么要加有伪首部?

目的是让 UDP 两次检查数据是否已经正确到达目的地。

IP 接受正确的目的地址，传送到正确的上层程序。

所有伪首部包括：源 IP 地址，目的 IP 地址，0，协议号，UDP 长度。

TCP 协议

问题 1：TCP 通过哪些方式来保证可靠性?

1） 应用数据被分割成 TCP 认为最适合发送的数据块。

2） 确认机制，发送报文后，等待确认。

3） 重发机制，没有收到确认，将重发数据段。

4） 保持它首部和数据的校验和。确认数据的准确性。

5） 排序，丢弃重复的，流量控制。

问题 2：TCP 与 UDP 的概念相互的区别及优劣

1. TCP 面向连接，UDP 面向无链接
2. TCP 面向报文，UDP 面向字节流
3. TCP 提供可靠传输服务（数据顺序、正确性），UDP 传输不可靠
4. TCP 协议传输速度慢，UDP 协议传输速度快
5. TCP 协议对系统资源要求多（头部开销大），UDP 协议要求少

问题 3：TCP、UDP 为什么存在伪包头?

UDP(TCP)检验和：是根据 UDP(TCP)数据报和伪报头计算得到的差错检测值。
伪报头包含源和目的 IP 地址，以及来自 IP 数据报报头的协议值。IP 数据报在网络中传送时包含 UDP 数据报。
伪报头并不会在网络中传送，校验和中所包含的伪报头内容可以避免目的端错误地接收错误路由的数据报。校验和值的计算方法和 IP 报头检验和的计算方法类似

问题 4：为什么要 3 次握手，4 次挥手

1. 3 次握手：防止已过期的连接请求报文突然又传送到服务器，因而产生错误
2. 4 次挥手：确保数据能够完成传输，但关闭连接时，当收到对方的 FIN 报文通知时，它仅仅表示对方没有数据发送给你了；但未必你所有的数据都全部发送给对方了，所以你可以未必会马上会关闭 SOCKET,也即你可能还需要发送一些数据给对方之后，再发送 FIN 报文给对方来表示你同意现在可以关闭连接了，所以它这里的 ACK 报文和 FIN 报文多数情况下都是分开发送的

问题 5：TCP 的流量控制机制

主要是下面的四种机制：

慢启动（慢开始）：

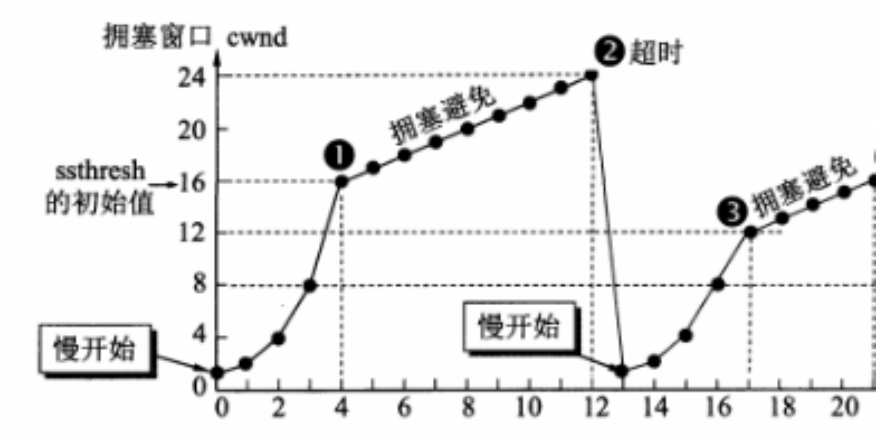
1. 慢开始不是指 cwnd 的增长速度慢（指数增长），而是指 TCP 开始发送设置 cwnd=1。
2. 思路：不要一开始就发送大量的数据，先探测一下网络的拥塞程度，也就是说由小到大逐渐增加拥塞窗口的大小。这里用报文段的个数的拥塞窗口大小举例说明慢开始算法，实时拥塞窗口大小是以字节为单位的。
3. 为了防止 cwnd 增长过大引起网络拥塞，设置一个慢开始门限（ssthresh 状态变量）

当 cnwd < ssthresh，使用慢开始算法
当 cnwd=ssthresh，既可使用慢开始算法，也可以使用拥塞避免算法
当 cnwd > ssthresh，使用拥塞避免算法

拥塞避免：

1. 拥塞避免并非完全能够避免拥塞，是说在拥塞避免阶段将拥塞窗口控制为按线性规律增长，使网络比较不容易出现拥塞。
2. 思路：让拥塞窗口 cwnd 缓慢地增大，即每经过一个往返时间 RTT 就把发送方的拥塞控制窗口加一。

无论是在慢开始阶段还是在拥塞避免阶段，只要发送方判断网络出现拥塞（其根据就是没有收到确认，虽然没有收到确认可能是其他原因的分组丢失，但是因为无法判定，所以都当做拥塞来处理），就把慢开始门限设置为出现拥塞时的发送窗口大小的一半。然后把拥塞窗口设置为 1，执行慢开始算法。 如图所示：

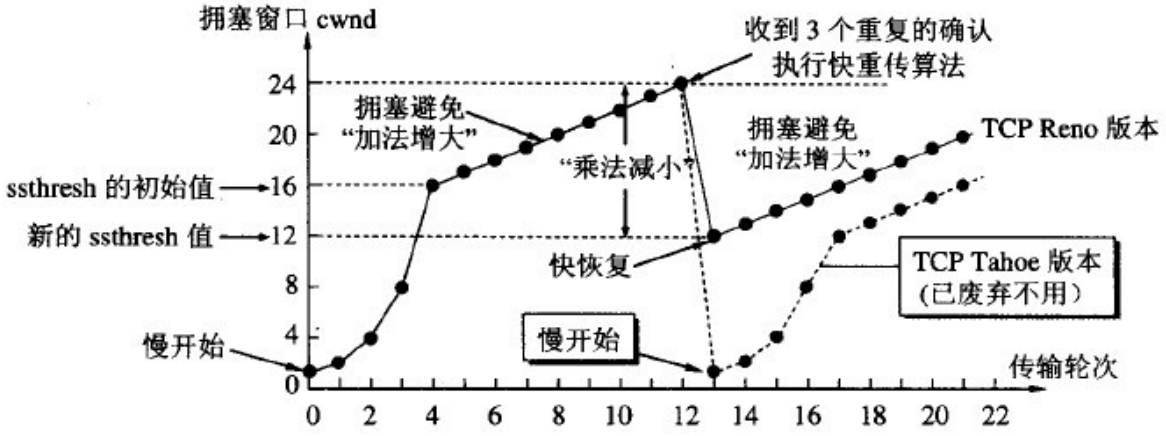


快速重传：

1. 快重传要求接收方在收到一个失序的报文段后就立即发出重复确认（为的是使发送方及早知道有报文段没有到达对方）而不要等到自己发送数据时捎带确认。快重传算法规定，发送方只要一连收到三个重复确认就应当立即重传对方尚未收到的报文段，而不必继续等待设置的重传计时器时间到期。
2. 由于不需要等待设置的重传计时器到期，能尽早重传未被确认的报文段，能提高整个网络的吞吐量。

快速恢复：

1. 当发送方连续收到三个重复确认时，就执行“乘法减小”算法，把 ssthresh 门限减半。但是接下去并不执行慢开始算法。
2. 考虑到如果网络出现拥塞的话就不会收到好几个重复的确认，所以发送方现在认为网络可能没有出现拥塞。所以此时不执行慢开始算法，而是将 cwnd 设置为 ssthresh 的大小，然后执行拥塞避免算法。



应用层

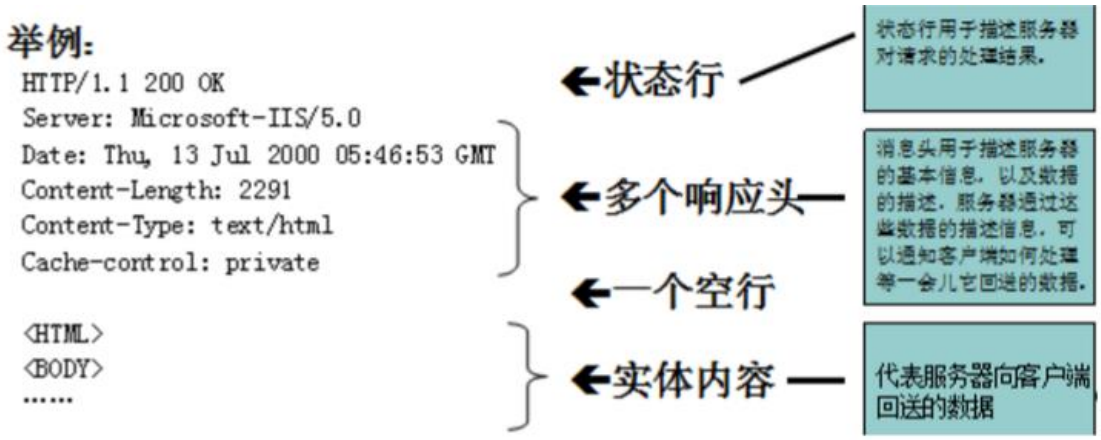
问题 0：DNS 的概念，用途，DNS 查询的实现算法

- 概念
  - 域名解析，www.xxx.com 转换成 ip，能够使用户更方便的访问互联网，而不用去记住能够被机器直接读取的 ip 地址
  - DNS 协议运行在 UDP 协议之上，使用端口号 53
- 主机解析域名的顺序
  - 浏览器缓存
  - 找本机的 hosts 文件
  - 路由缓存
  - 找 DNS 服务器(本地域名、顶级域名、根域名)
    - 迭代查询、递归查询

问题 1：http 基本格式

http 请求：

http 响应：



问题 2：GET、POST 区别

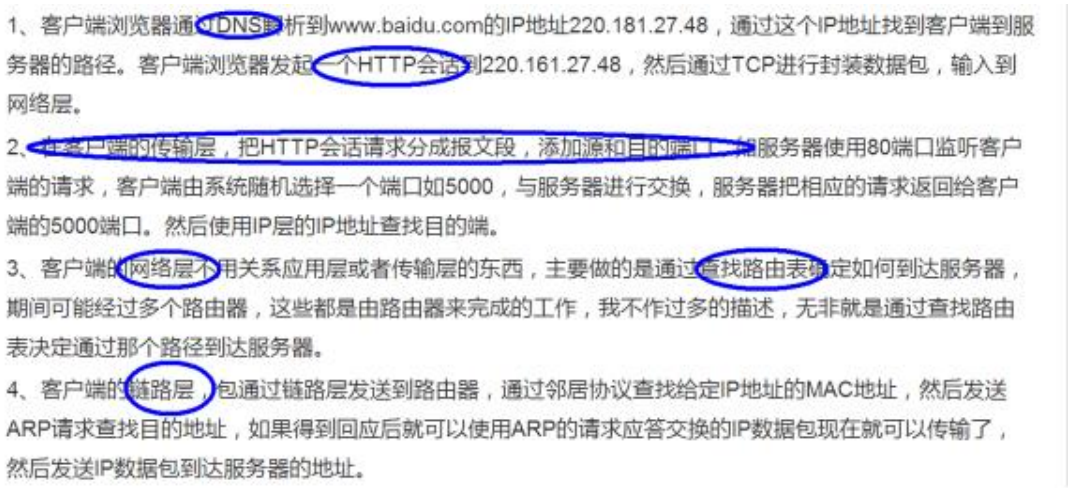
操作方式	数据位置	明文密文	数据安全	长度限制	应用场景
GET	HTTP包头	明文	不安全	长度较小	查询数据
POST	HTTP正文	可明可密	安全	支持较大数据传输	修改数据

问题 3：Cookies 和 Session 的区别

1. cookie 是一种发送到客户浏览器的文本串句柄，并保存在客户机硬盘上，可以用来在某个 WEB 站点会话间持久的保持数据
2. session 其实指的就是访问者从到达某个特定主页到离开为止的那段时间。 Session 其实是利用 Cookie 进行信息处理的，当用户首先进行了请求后，服务端就在用户浏览器上创建了一个 Cookie，当这个 Session 结束时，其实就是意味着这个 Cookie 就过期了
3. cookie 数据保存在客户端，session 数据保存在服务器端

问题 4：一次完整的 HTTP 请求所经历的步骤

比如：在浏览器中输入 www.baidu.com 后执行的全部过程



也可以参考文章：《抓包实战 | 浏览器里的 HTTP 请求到底是如何完成的？》
https://mp.weixin.qq.com/s/_fB7r53BGZRvpG9YAPRQ8A

问题 5：http2.0 和 http1.1 的区别