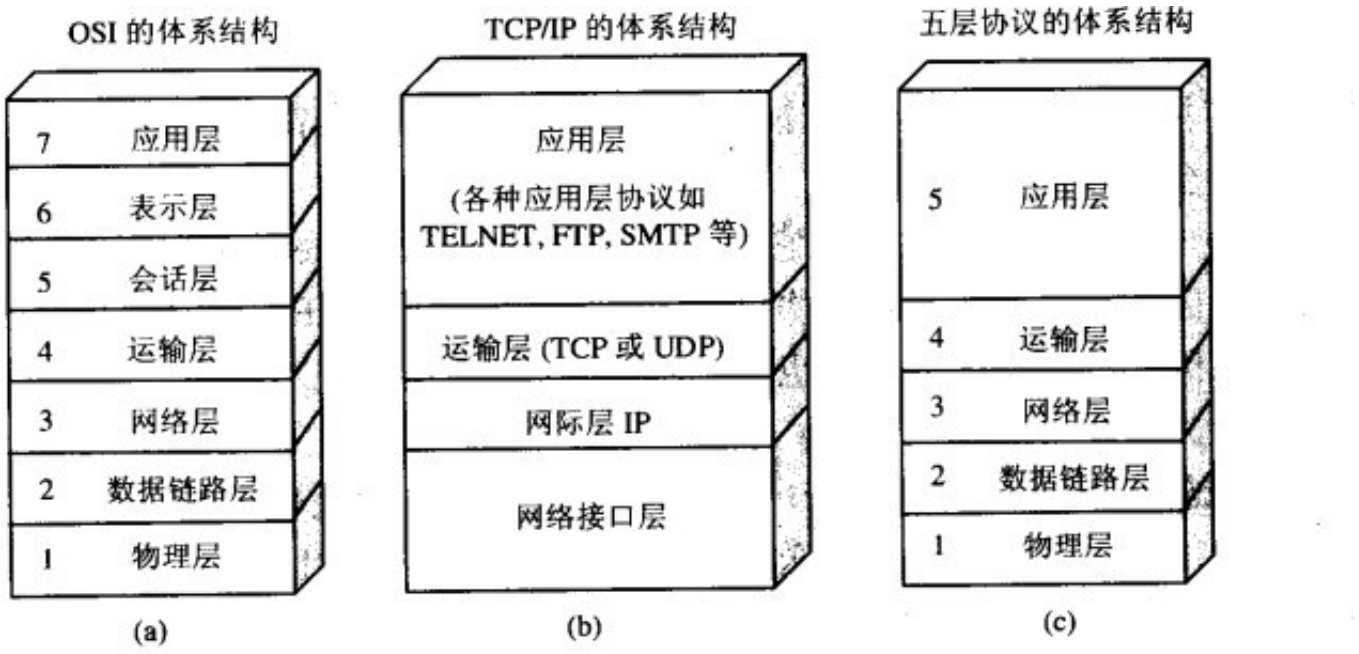
计算机网络总结

[搞定计算机网络面试，its enough]

1.OSI七层模型与TCP\IP四层模型



计算机网络体系结构：(a) OSI 的七层协议；(b) TCP/IP 的四层协议；(c) 五层协议

具体7层	数格式	功能与连接方式	相关协议	典型设备
应用层 application	数据 ATPU报文	是最靠近用户的OSI层。这一层为用户的应用程序（例如电子邮件、文件传输和终端仿真）提供网络服务。	FTP、DNS、HTTP、SMTP、www、Telnet、NFS	终端设备（pc、手机、平板）
表示层 presentation	数据 PTPU	数据表示、安全和压缩。可确保一个系统的应用层所发送的信息可以被另一个系统的应用层读取。例如，PC程序与另一台计算机进行通信，其中一台计算机使用扩展二一十进制交换码（EBCDIC），而另一台则使用美国信息交换标准码（ASCII）来表示相同的字符。如有	JPEG、MPEG、ASII	终端设备（pc、手机、

		必要，表示层会通过使用一种通格式来实现多种数据格式之间的转换。		平板）
会话层 session	数据 DTPU	通过运输层（端口号：传输端口与接收端口）建立数据传输的通路。主要在你的系统之间发起会话或者接受会话请求（设备之间需要互相认识可以是IP也可以是MAC或者是主机名）	NFS、 SQL、 netbios、 RPC	终端设备 （pc、 手机、 平板）
传输层 transport	数据组织成数据段 segment	定义了一些传输数据的协议和端口号（WWW端口80等），如：TCP（transmission control protocol –传输控制协议，传输效率低，可靠性强，用于传输可靠性要求高，数据量大的数据）。UDP（user datagram protocol–用户数据报协议，与TCP特性恰恰相反，用于传输可靠性要求不高，数据量小的数据，如QQ聊天数据就是通过这种方式传输的）。 主要是将从下层接收的数据进行分段和传输，到达目的地址后再进行重组。常常把这一层数据叫做段。	TCP/UDP、 SPX	终端设备 （pc、 手机、 平板）
网络层 network	分割和重新组合成数据报 packet	基于网络层IP地址在位于不同地理位置的网络中的两个主机系统之间提供连接和路径选择。Internet的发展使得从世界各站点访问信息的用户数大大增加，而网络层正是管理这种连接的层。	IP、 ICMP、 ARP、 RARP、 OSPF、 IPX、 RIP、 IGRP	网关、 路由器
数据链路层 data link	将比特信息封装成数据帧 frame	定义了如何让格式化数据以进行传输，以及如何让控制对物理介质的访问。这一层通常还提供错误检测和纠正，以确保数据的可靠传输。	PPP、FR、 HDLC、 VLAN、 MAC	
物理层 physical	传输比特 （bit） 流	主要定义物理设备标准，如网线的接口类型、光纤的接口类型、各种传输介质的传输速率等。它的主要作用是传输比特流（就是由1、0转化为电流强弱来进行传输,到达目的地后在转化为1、0，也就是我们常说的数模转换与模数转换）。这一层的数据叫做比特。		同轴电缆、光纤

TCP/IP

第7层　应用层

各种应用程序协议，如HTTP、FTP、SMTP、POP3。

常见使用TCP协议的应用层服务

HTTP 超文本传输协议	FTP 文件传输协议	SMTP 简单邮件传输协议	TELNET TCP/IP终端仿真协议
POP3 邮局协议第3版	Finger 用户信息协议	NNTP 网络新闻传输协议	IMAP4 因特网信息访问协议第四版

UNIX网络服务

LPR UNIX远程打印协议	Rwho UNIX远程Who协议	Rexec UNIX远程执行协议
Login UNIX远程登陆协议		RSH UNIX远程Shell协议

常见使用UDP协议的应用层服务

- BOOTP 引导协议
- DHCP 动态主机配置协议
- NTP 网络时间协议
- TFTP 简单文件传输协议

HP网络服务

NTF HP 网络文件传输协议	RDA HP 远程数据库访问协议	VT 虚拟终端仿真协议	RFA HP 远程文件访问协议	RPC Remote Process Comm.

同时使用TCP和UDP协议的应用层服务

SOCKS 安全套接字协议	FANP 流属性通知协议
SLP 服务定位协议	MSN 微软网络服务
Radius 远程用户拨号认证服务协议	DNS 域名系统

- S-HTTP 安全超文本传输协议
- GDP 网关发现协议
- X-Window X-Window
- CMOT 基于TCP/IP的CMIP协议

SUN网络服务

NFS 网络文件系统协议	R-STAT SUN远程状态协议	PMAP SUN端口映射协议
NIS SUN网络信息系统协议	NSM SUN网络状态监测协议	Mount

- SNMP 简单网络管理协议

第6层　表示层

信息的语法语义以及它们的关联，如加密解密、转换翻译、压缩解压缩。

- DECnet NSP
- LPP 轻量级表示协议
- NBSSN NetBIOS会话服务协议
- XDP 外部数据表示协议
- IPX

第5层　会话层

不同机器上的用户之间建立及管理会话。

安全协议

SSL 安全套接字层协议	TLS 传输层安全协议

目录访问协议

DAP 目录访问协议	LDAP 轻量级目录访问协议

- RPC 远程过程调用协议

VFRP	NeTBIOS	IPX
VINES NETRPC		

第4层　传输层

接受上一层的数据，在必要的时候把数据进行分割，并将这些数据交给网络层，且保证这些数据段有效到达对端。

DSI	IP NeTBIOS	SMB
NetBIOS	ISO-TP SSP	MSRPC

- NetBIOS
- XOT 基于TCP之上的X.25协议
- Van Jacobson 压缩TCP协议
- ISO-DE ISO开发环境
- TALI 传输适配层接口协议
- RUDP 可靠的用户数据报协议
- Mobile IP 移动IP协议

TCP 传输控制协议

UDP 用户数据报协议

第3层　网络层

控制子网的运行，如逻辑编址、分组传输、路由选择。

安全协议

AH 认证头协议	ESP 安全封装有效载荷协议

- IP/IPv6 互联网协议/互联网协议第6版
- SLIP 串行线路IP协议

路由协议

EGP 外部网关协议	NHRP 下一跳解析协议	GGP 网关到网关协议	RSVP 资源预留协议	RIP2 路由信息协议第2版
OSPF 开放最短路径优先协议	IE-IRGP 增强内部网关路由选择协议	VRRP 虚拟路由备冗余协议	PIM-DM 密集模式独立组播协议	PIM-SM 稀疏模式独立组播协议
IGRP 内部网关路由协议	RIPng for IPv6 IPv6路由信息协议	PGM 实际通用组播协议	DVMRP 距离矢量组播路由协议	MOSPF 组播开放最短路径优先协议

- X.25
- NetWare

ICMPv6 互联网控制信息协议第6版
ICMP 互联网控制信息协议
IGMP 互联网组管理协议

第2层　数据链路层

物理寻址，同时将原始比特流转变为逻辑传输线路。

隧道协议

MPLS 多协议标签交换协议	XTP 压缩传输协议	DCAP 数据转接客户访问协议
SLE 串行连接封装协议	IPinIP IP套IP封装协议	

PPTP 点对点隧道协议	L2TP 第二层隧道协议
L2F 第二层转发协议	ATMP 接入隧道管理协议

Cisco协议

CDP 思科发现协议
CGMP 思科组管理协议

地址解析协议

ARP 地址解析协议
RARP 逆向地址解析协议

第1层　物理层

机械、电子、定时接口通信信道上的原始比特流传输。

- IEEE 802.2
- Ethernet v.2
- Internetwork

1.1 OSI七层模型的每一层协议有哪些？SSL工作在那一层？

答：协议见👆表格，SSL工作在表示层。

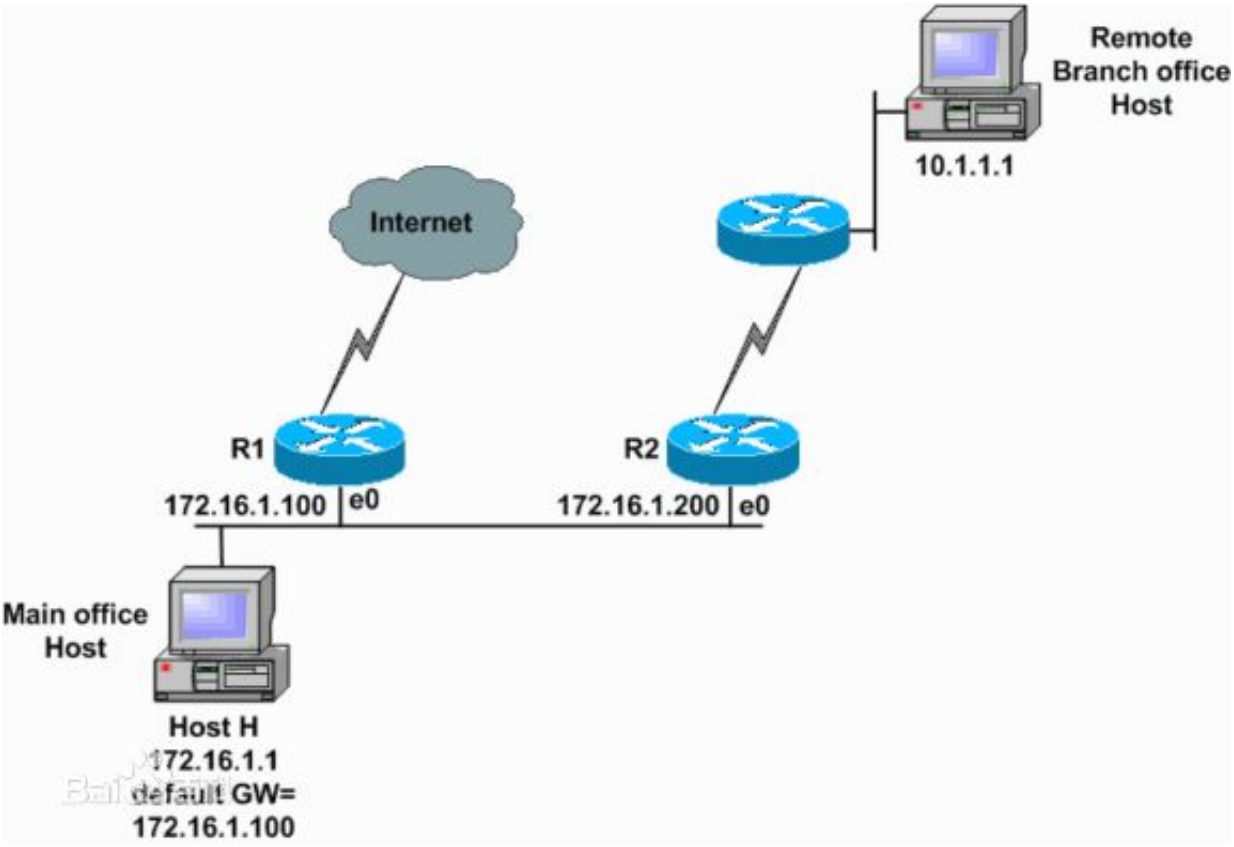
2. 说一下TTL

time to live。数据包在传输过程中，每经过一个路由器，TTL就减少1，直到TTL=0时，数据包被丢弃，并发送ICMP time exceeded报文通知主机防止重复发送。
（百度百科）TTL 是由发送主机设置的，以防止数据包不断在IP互联网络上永不终止地循环。转发IP数据包时，要求路由器至少将 TTL 减小 1。

生存时间，就是一条域名解析记录在DNS服务器中的存留时间。当各地的DNS服务器接受到解析请求时，就会向域名指定的NS服务器(权威域名服务器）发出解析请求从而获得解析记录；在获得这个记录之后，记录会在DNS服务器(各地的缓存服务器，也叫递归域名服务器）中保存一段时间，这段时间内如果再接到这个域名的解析请求，DNS服务器将不再向NS服务器发出请求，而是直接返回刚才获得的记录；而这个记录在DNS服务器上保留的时间，就是TTL值。

3. ping和tracert命令用什么协议?

ICMP协议（网络层）。
ICMP提供一致易懂的出错报告信息。发送的出错报文返回到发送 图1 ICMP原理 图1 ICMP原理 原数据的设备，因为只有发送设备才是出错报文的逻辑接受者。发送设备随后可根据ICMP报文确定发生错误的类型，并确定如何才能更好地重发失败的数据包。但是ICMP唯一的功能是报告问题而不是纠正错误，纠正错误的任务由发送方完成。



4. MTU是什么?

答：最大传输单元。
最大传输单元（Maximum Transmission Unit，MTU）是指一种通信协议的某一层上面所能通过的最大数据报大小（以字节为单位）。最大传输单元这个参数通常与通信接口有关（网络接口卡、串口等）。

因特网协议允许IP分片，这样就可以将数据报分成足够小的片段以通过那些最大传输单元小于该数据报原始大小的链路了。这一分片过程发生在IP层（OSI模型的第三层，即网络层），它使用的是将分组发送到链路上的网络接口的最大传输单元的值。原始分组的分片都被加上了标记，这样目的主机的IP层就能将分组重组成原始的数据报了。

在因特网协议中，一条因特网传输路径的“路径最大传输单元”被定义为从源地址到目的地址所经过“路径”上的所有IP跳的最大传输单元的最小值。或者从另外一个角度来看，就是无需进一步分片就能穿过这条“路径”的最大传输单元的最大值。

RFC 1191描述了“路径最大传输单元发现方法”，这是一种确定两个IP主机之间路径最大传输单元的技术，其目的是为了避免IP分片。在这项技术中，源地址将数据报的DF（Don't Fragment，不要分片）位置位，再逐渐增大发送的数据报的大小——路径上任何需要将分组进行分片的设备都会将这种数据报丢弃并返回一个“数据报过大”的ICMP响应到源地址——这样，源主机就“学习”到了不用进行分片就能通过这条路径的最大的最大传输单元了。

通常来说MTU越小，你所发送包的频率越快，在一些游戏和网络软件中可以通过更改mtu获得更好的效果。mtu代表的是封装包的大小，封闭包的大小决定你发送包的发送频率。

5. 三次握手（建连）

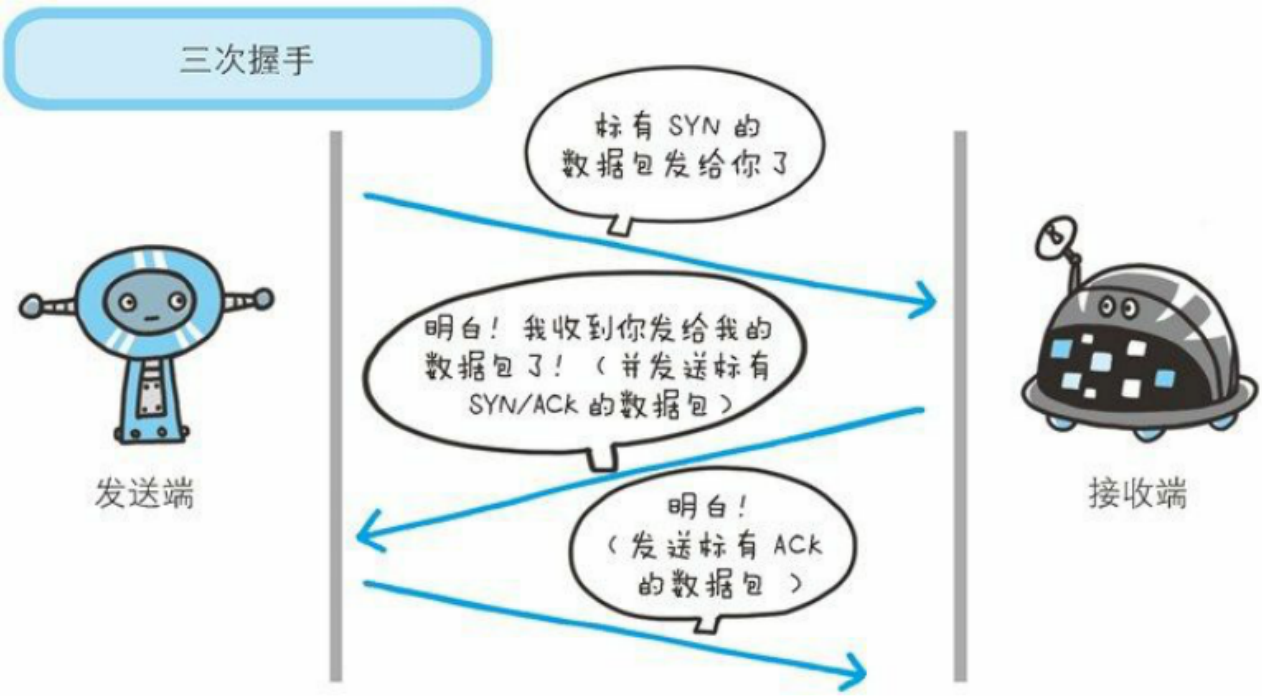
5.1 部分报头信息声明

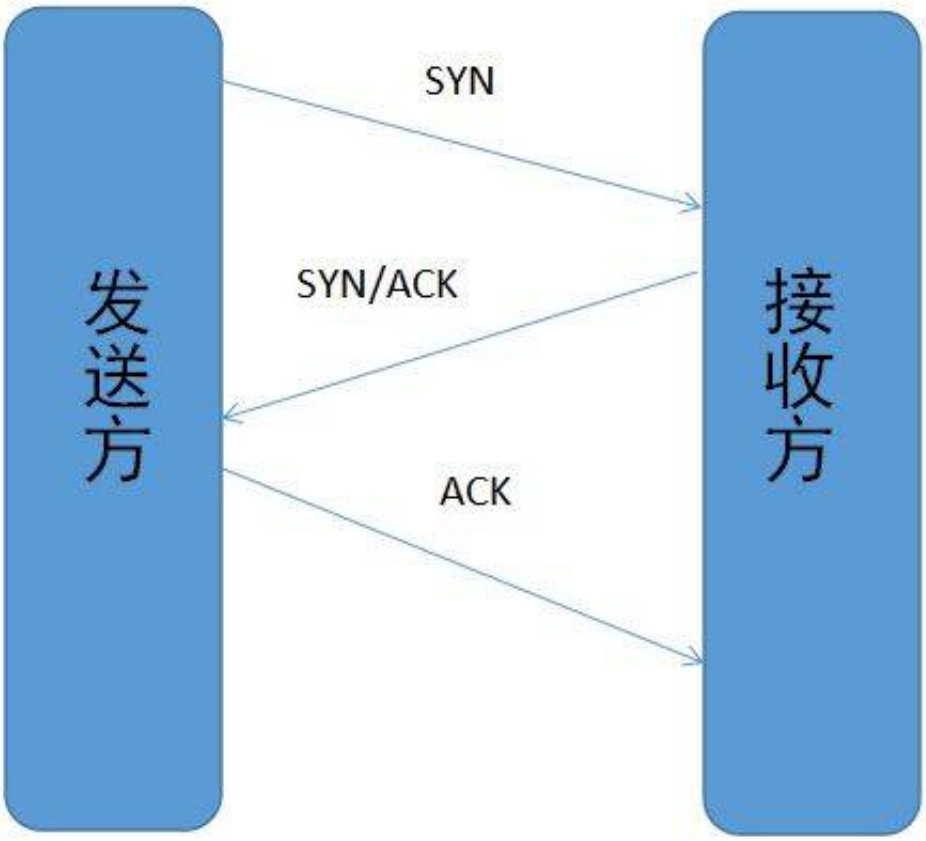
seq： 序列号，表示数据第一个字节的序号
ack： 确认序列号，表示期望收到的第一个字节的序号。

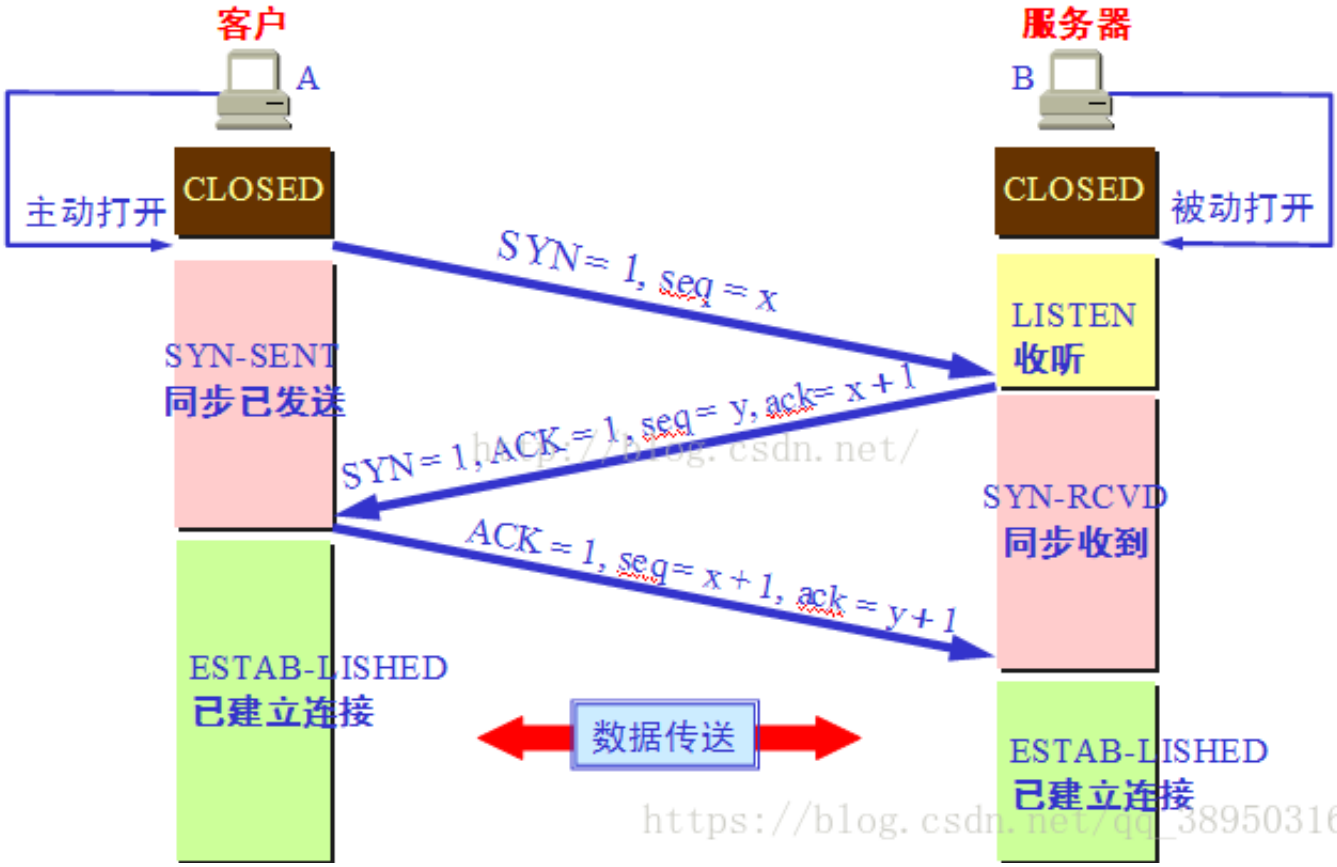
5.2 flag位

常用的有SYN、ACK、FIN。
SYN：用作连接建立时的同步信号。
ACK：用于对收到的数据进行确认。
FIN：表示后面没有数据需要发送，连接需要关闭。

5.3 握手过程







第一次握手：建立连接时，客户端发送syn包（syn=j）到服务器，并进入**SYN_SENT**状态，等待服务器确认；SYN：同步序列编号（Synchronize Sequence Numbers）。

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=j+1），同时自己也发送一个SYN包（syn=k），即SYN+ACK包，此时服务器进入**SYN_RCVD**状态；

第三次握手：客户端收到服务器的SYN+ACK包，向服务器发送确认包ACK(ack=k+1），此包发送完毕，客户端和服务器进入**ESTABLISHED**（TCP连接成功）状态，完成三次握手。

6. 为什么要三次握手？两次握手会怎么样?

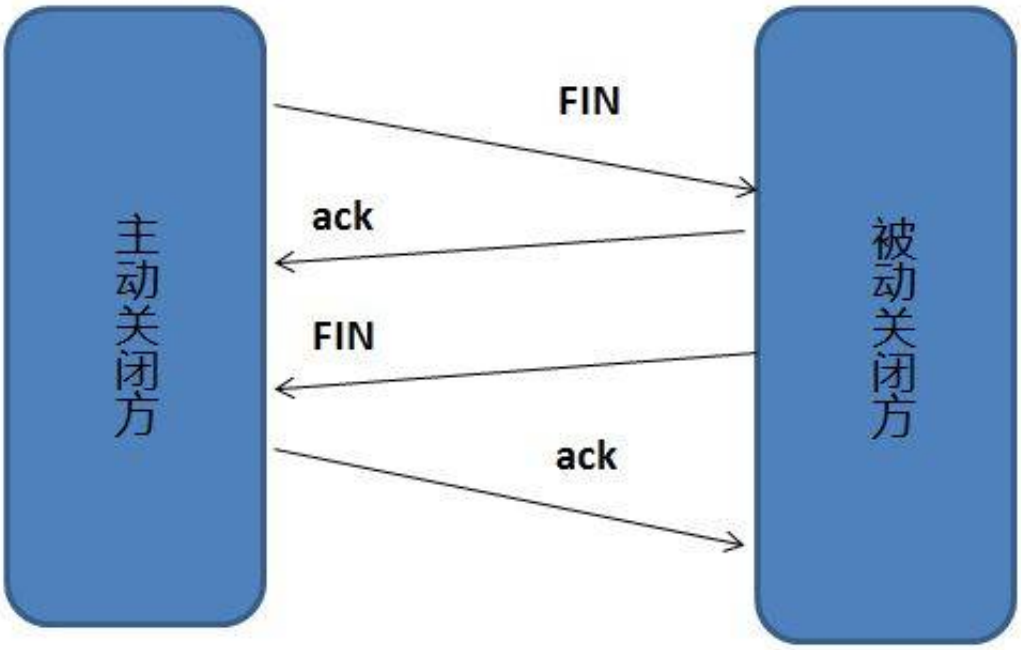
答： 三次我握手是为了 1. 保持信息对等；2.防止请求超时导致脏连接。

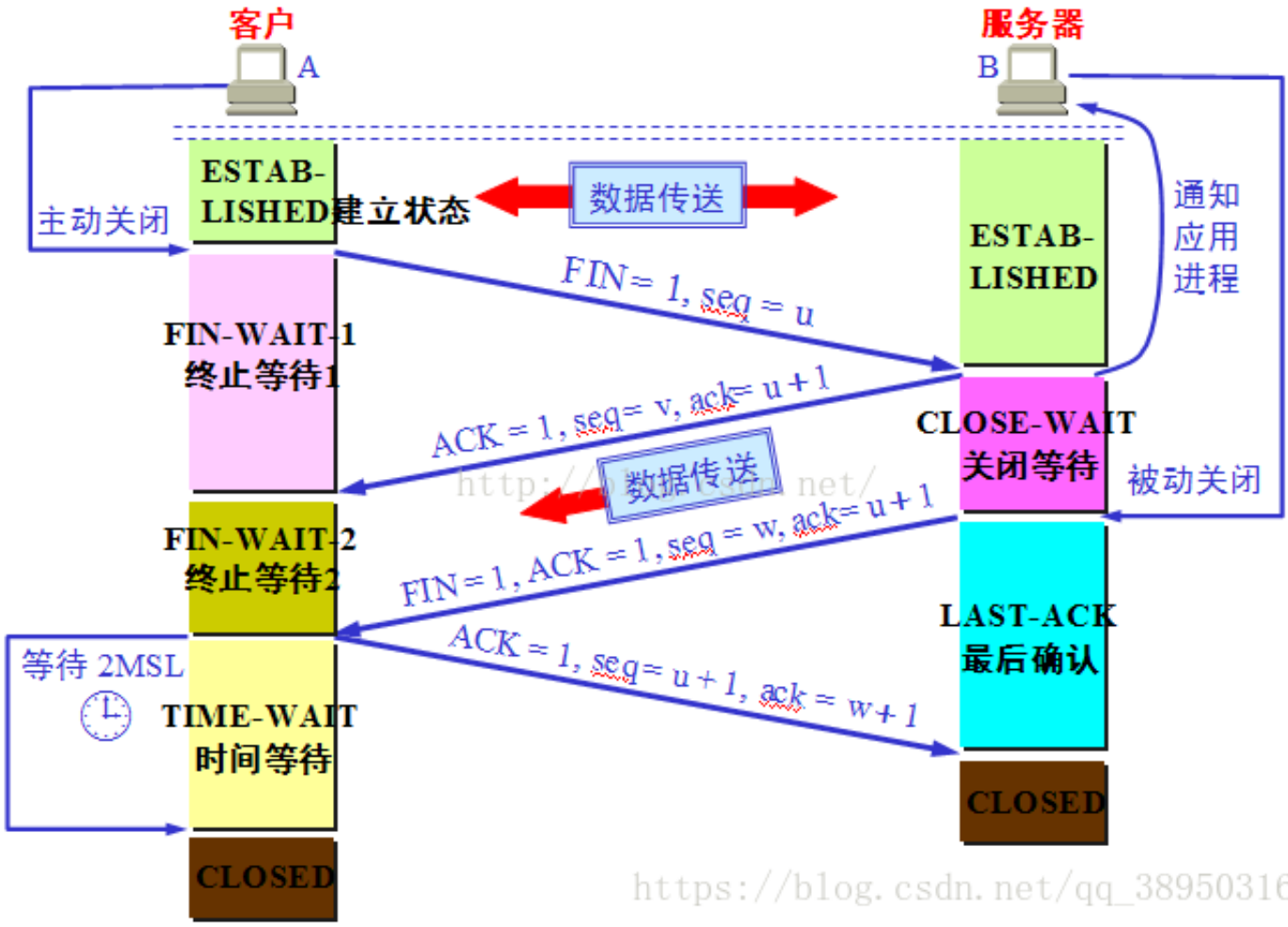
```
3次握手完成两个重要的功能，既要双方做好发送数据的准备工作(双方都知道彼此已准备好)，

也要允许双方就初始序列号进行协商，这个序列号在握手过程中被发送和确认。
三次握手的目的是建立可靠的通信信道，说到通讯，简单来说就是数据的发送与接收，
而三次握手最主要的目的就是双方确认自己与对方的发送与接收是正常的。
第一次握手：Client 什么都不能确认；Server 确认了对方发送正常
第二次握手：Client 确认了：自己发送、接收正常，对方发送、接收正常；
Server 确认了：自己接收正常，对方发送正常
第三次握手：Client 确认了：自己发送、接收正常，对方发送、接收正常；
Server 确认了：自己发送、接收正常，对方发送接收正常
所以三次握手就能确认双发收发功能都正常，缺一不可。

现在把三次握手改成仅需要两次握手，死锁是可能发生的。
作为例子，考虑计算机s和c之间的通信，假定c给s发送一个连接请求分组，s收到了这个分组，
并发送了确认应答分组。
按照两次握手的协定，s认为连接已经成功地建立了，可以开始发送数据分组。
可是，c在s的应答分组在传输中被丢失的情况下，将不知道s 是否已准备好，
不知道s建立什么样的序列号，c甚至怀疑s是否收到自己的连接请求分组。
在这种情况下，c认为连接还未建立成功，将忽略s发来的任何数据分组，只等待连接确认应答分组。
而s在发出的分组超时后，重复发送同样的分组。这样就形成了死锁。
```

7. 四次挥手（断连）





1）客户端进程发出连接释放报文，并且停止发送数据。释放数据报文首部，FIN=1，其序列号为seq=u（等于前面已经传送过来的数据的最后一个字节的序号加1），此时，客户端进入FIN-WAIT-1（终止等待1）状态。 TCP规定，FIN报文段即使不携带数据，也要消耗一个序号。

2）服务器收到连接释放报文，发出确认报文，ACK=1，ack=u+1，并且带上自己的序列号seq=v，此时，服务端就进入了CLOSE-WAIT（关闭等待）状态。TCP服务器通知高层的应用进程，客户端向服务器的方向就

释放了，这时候处于半关闭状态，即客户端已经没有数据要发送了，但是服务器若发送数据，客户端依然要接受。这个状态还要持续一段时间，也就是整个CLOSE-WAIT状态持续的时间。
3）客户端收到服务器的确认请求后，此时，客户端就进入FIN-WAIT-2（终止等待2）状态，等待服务器发送连接释放报文（在这之前还需要接受服务器发送的最后的数据）。
4）服务器将最后的数据发送完毕后，就向客户端发送连接释放报文，FIN=1，ack=u+1，由于在半关闭状态，服务器很可能又发送了一些数据，假定此时的序列号为seq=w，此时，服务器就进入了LAST-ACK（最后确认）状态，等待客户端的确认。
5）客户端收到服务器的连接释放报文后，必须发出确认，ACK=1，ack=w+1，而自己的序列号是seq=u+1，此时，客户端就进入了TIME-WAIT（时间等待）状态。注意此时TCP连接还没有释放，必须经过2**MSL（最长报文段寿命）的时间后，当客户端撤销相应的TCB后，才进入CLOSED状态。
6）服务器只要收到了客户端发出的确认，立即进入CLOSED状态。同样，撤销TCB后，就结束了这次的TCP连接。可以看到，服务器结束TCP连接的时间要比客户端早一些。

常见面试题

【问题1】为什么连接的时候是三次握手，关闭的时候却是四次握手?

答：因为当Server端收到Client端的SYN连接请求报文后，可以直接发送SYN+ACK报文。
其中ACK报文是用来应答的，SYN报文是用来同步的。
但是关闭连接时，当Server端收到FIN报文时，很可能并不会立即关闭SOCKET，
所以只能先回复一个ACK报文，告诉Client端，"你发的FIN报文我收到了"。
只有等到我Server端所有的报文都发送完了，我才能发送FIN报文，因此不能一起发送。
故需要四步握手。

【问题2】为什么TIME_WAIT状态需要经过2MSL(最大报文段生存时间)才能返回到CLOSE状态?

答：虽然按道理，四个报文都发送完毕，我们可以直接进入CLOSE状态了，
但是我们必须假象网络是不可靠的，有可能最后一个ACK丢失。
所以TIME_WAIT状态就是用来重发可能丢失的ACK报文。在Client发送出最后的ACK回复，但该ACK可能丢失。Server如果没有收到ACK，将不断重复发送FIN片段。所以Client不能立即关闭，它必须确认Server接收到了该ACK。
Client会在发送出ACK之后进入到TIME_WAIT状态。Client会设置一个计时器，等待2MSL的时间。
如果在该时间内再次收到FIN，那么Client会重发ACK并再次等待2MSL。
所谓的2MSL是两倍的MSL(Maximum Segment Lifetime)。
MSL指一个片段在网络中最大的存活时间，2MSL就是一个发送和一个回复所需的最大时间。
如果直到2MSL，Client都没有再次收到FIN，那么Client推断ACK已经被成功接收，则结束TCP连接。

【问题3】为什么不能用两次握手进行连接?

答：3次握手完成两个重要的功能，既要双方做好发送数据的准备工作(双方都知道彼此已准备好)，
也要允许双方就初始序列号进行协商，这个序列号在握手过程中被发送和确认。
现在把三次握手改成仅需要两次握手，死锁是可能发生的。
作为例子，考虑计算机S和C之间的通信，假定C给S发送一个连接请求分组，S收到了这个分组，并发 送了确认应答分组。
按照两次握手的协定，S认为连接已经成功地建立了，可以开始发送数据分组。
可是，C在S的应答分组在传输中被丢失的情况下，将不知道S 是否已准备好，
不知道S建立什么样的序列号，C甚至怀疑S是否收到自己的连接请求分组。
在这种情况下，C认为连接还未建立成功，将忽略S发来的任何数据分组，只等待连接确认应答分组。
而S在发出的分组超时后，重复发送同样的分组。这样就形成了死锁。

【问题4】如果已经建立了连接，但是客户端突然出现故障了怎么办?

TCP还设有一个保活计时器，显然，客户端如果出现故障，服务器不能一直等下去，白白浪费资源。
服务器每收到一次客户端的请求后都会重新复位这个计时器，时间通常是设置为2小时，
若两小时还没有收到客户端的任何数据，服务器就会发送一个探测报文段，
以后每隔75秒钟发送一次。若一连发送10个探测报文仍然没反应，服务器就认为客户端出了故障，接着就关闭连接。

8. TCP和UDP的区别？HTTP与HTTPS的区别?

TCP和UDP的区别:
TCP面向连接，UDP面向非连接

TCP提供可靠的服务（数据传输无差错、不丢失、不重复、按序到达），UDP不可靠

TCP面向字节流，UDP面向报文

TCP数据传输慢，UDP数据传输快

TCP首部开销20字节，UDP8字节

HTTPS和HTTP的区别主要如下：

1、https协议需要到ca申请证书，一般免费证书较少，因而需要一定费用。

2、http是超文本传输协议，信息是明文传输，https则是具有安全性的ssl加密传输协议。

3、http和https使用的是完全不同的连接方式，用的端口也不一样，前者是80，后者是443。

4、http的连接很简单，是无状态的；HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，比http协议安全。

8.1 TCP如何保证可靠传输

1.应用数据被分割成 TCP 认为最适合发送的数据块。

2.TCP 给发送的每一个包进行编号，接收方对数据包进行排序，把有序数据传送给应用层。

3.**校验和：** TCP 将保持它首部和数据的检验和。这是一个端到端的检验和，目的是检测数据在传输过程中的任何变化。如果收到段的检验和有差错，TCP 将丢弃这个报文段和不确认收到此报文段。

4.TCP 的接收端会丢弃重复的数据。

5.**流量控制：** TCP 连接的每一方都有固定大小的缓冲空间，TCP的接收端只允许发送端发送接收端缓冲区能接纳的数据。当接收方来不及处理发送方的数据，能提示发送方降低发送的速率，防止包丢失。TCP 使用的流量控制协议是可变大小的滑动窗口协议。 （TCP 利用滑动窗口实现流量控制）。

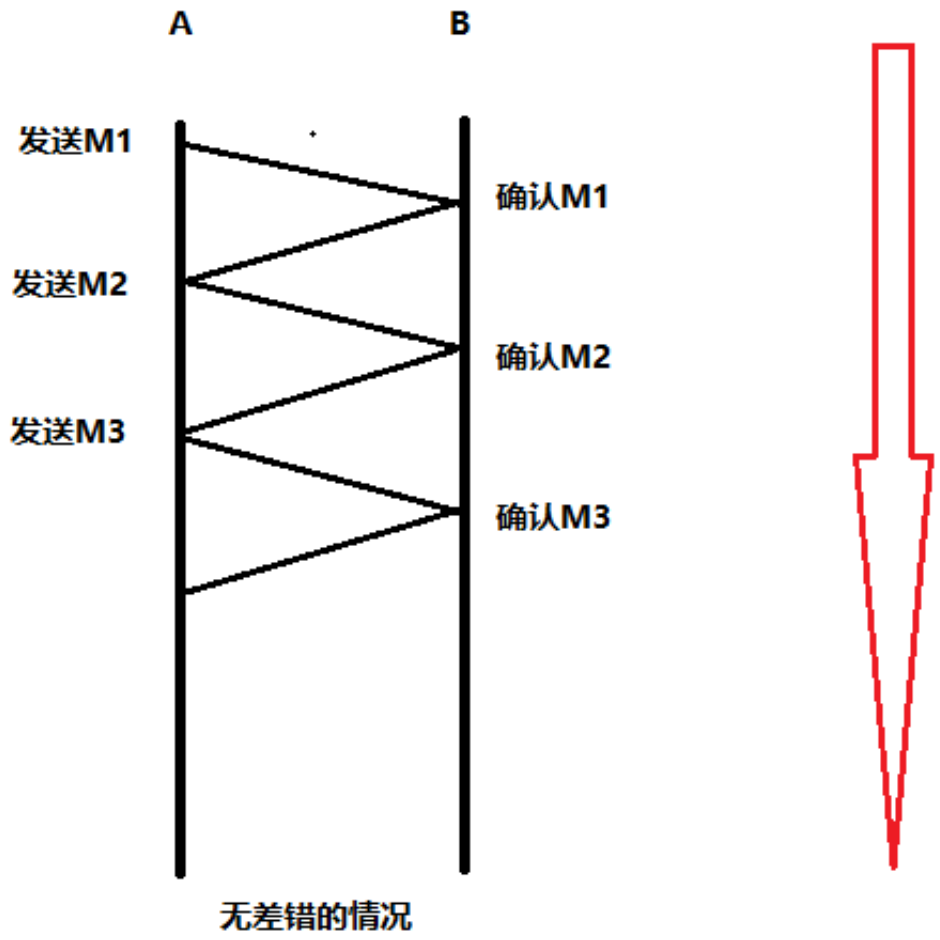
6.**拥塞控制：** 当网络拥塞时，减少数据的发送。

7.**停止等待协议** 也是为了实现可靠传输的，它的基本原理就是每发完一个分组就停止发送，等待对方确认。在收到确认后再发下一个分组。 超时重传： 当 TCP 发出一个段后，它启动一个定时器，等待目的端确认收到这个报文段。如果不能及时收到一个确认，将重发这个报文段。

停止等待协议

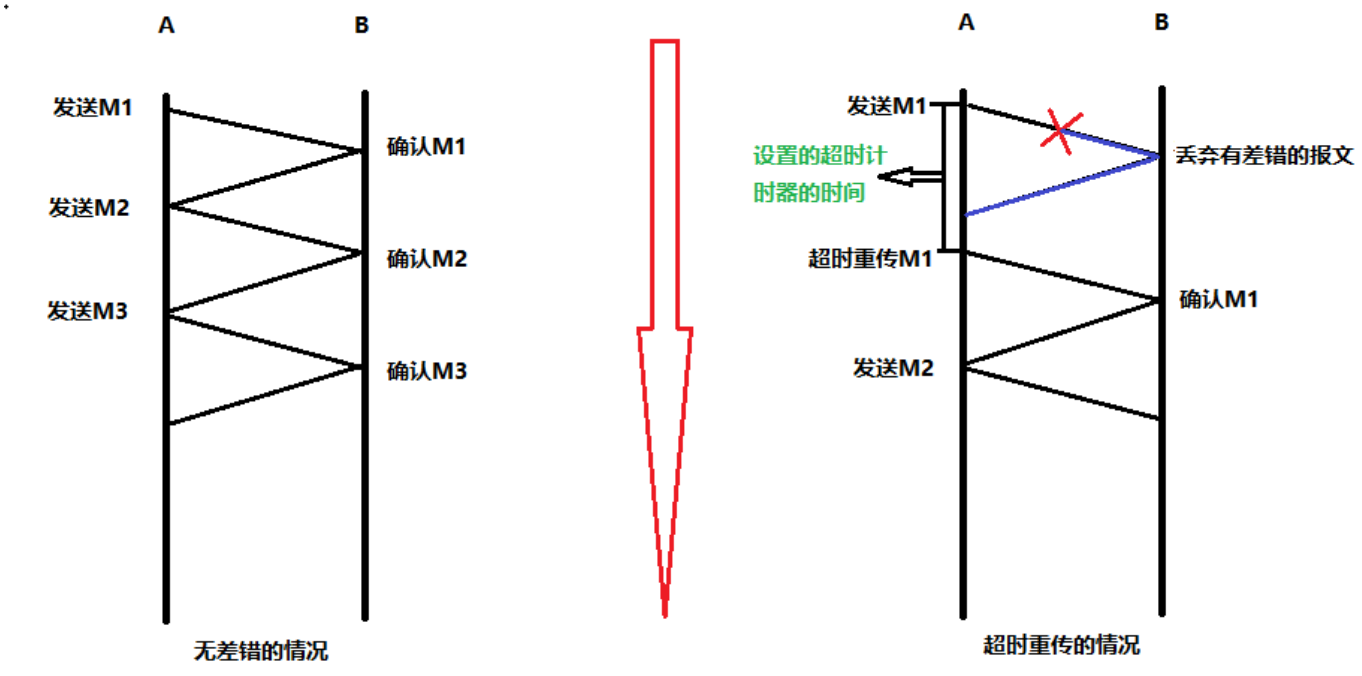
停止等待协议是为了实现可靠传输的，它的基本原理就是每发完一个分组就停止发送，等待对方确认。在收到确认后再发下一个分组； 在停止等待协议中，若接收方收到重复分组，就丢弃该分组，但同时还要发送确认；

1) 无差错情况:



发送方发送分组,接收方在规定时间内收到,并且回复确认.发送方再次发送。

2) 出现差错情况（超时重传）：



停止等待协议中超时重传是指只要超过一段时间仍然没有收到确认，就重传前面发送过的分组（认为刚才发

送过的分组丢失了）。因此每发送完一个分组需要设置一个超时计时器，其重转时间应比数据在分组传输的平均往返时间更长一些。这种自动重传方式常称为 自动重传请求 ARQ 。另外在停止等待协议中若收到重复分组，就丢弃该分组，但同时还要发送确认。连续 ARQ 协议 可提高信道利用率。发送维持一个发送窗口，凡位于发送窗口内的分组可连续发送出去，而不需要等待对方确认。接收方一般采用累积确认，对按序到达的最后一个分组发送确认，表明到这个分组位置的所有分组都已经正确收到了。

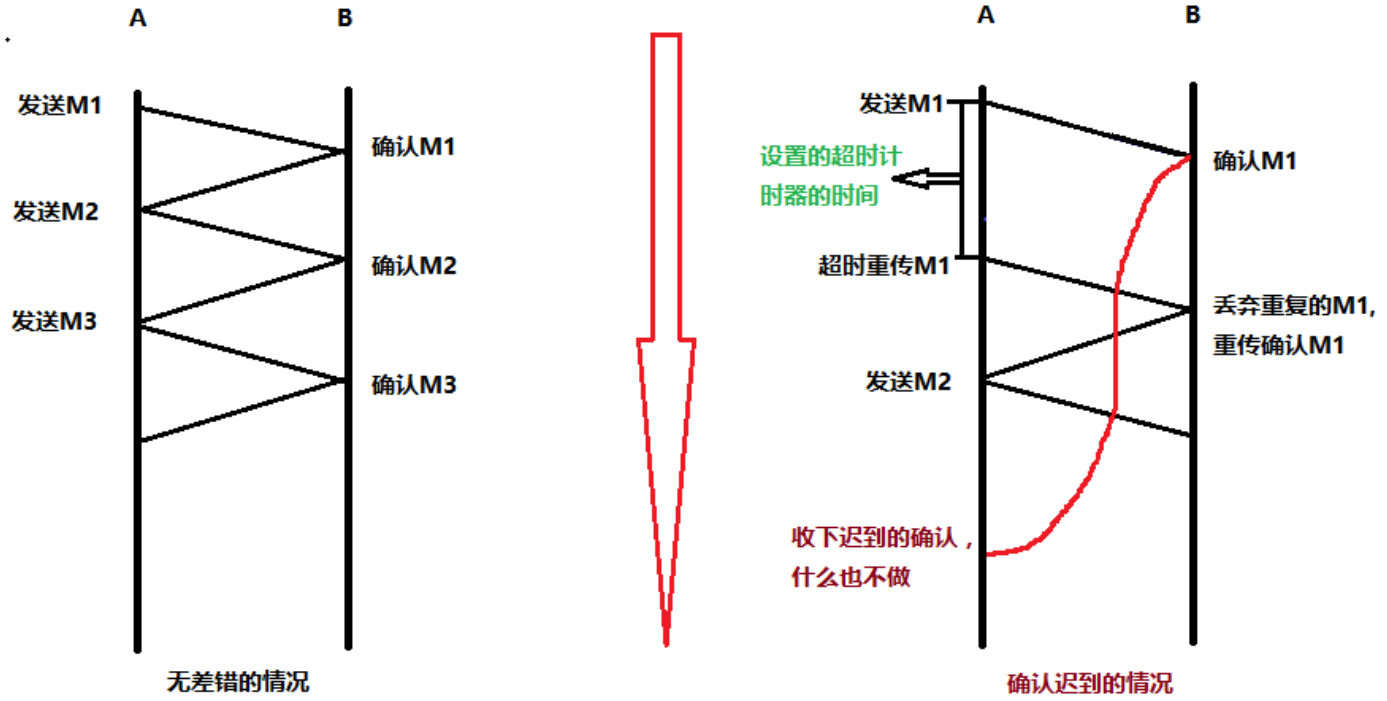
3) 确认丢失和确认迟到

确认丢失：确认消息在传输过程丢失

当A发送M1消息，B收到后，B向A发送了一个M1确认消息，但却在传输过程中丢失。而A并不知道，在超时计时过后，A重传M1消息，B再次收到该消息后采取以下两点措施：

```
1.丢弃这个重复的M1消息，不向上层交付。
2.向A发送确认消息。（不会认为已经发送过了，就不再发送。A能重传，就证明B的确认消息丢失）。
```

**确认迟到 **:
确认消息在传输过程中迟到



A发送M1消息，B收到并发送确认。在超时时间内没有收到确认消息，A重传M1消息，B仍然收到并继续发送确认消息（B收到了2份M1）。此时A收到了B第二次发送的确认消息。接着发送其他数据。过了一会，A收到了B第一次发送的对M1的确认消息（A也收到了2份确认消息）。处理如下：

```
1.A收到重复的确认后，直接丢弃。
2.B收到重复的M1后，也直接丢弃重复的M1。
```

自动重传请求 ARQ 协议

停止等待协议中超时重传是指只要超过一段时间仍然没有收到确认，就重传前面发送过的分组（认为刚才发

送过的分组丢失了）。因此每发送完一个分组需要设置一个超时计时器，其重转时间应比数据在分组传输的平均往返时间更长一些。这种自动重传方式常称为自动重传请求ARQ。
优点： 简单
缺点： 信道利用率低

连续ARQ协议

连续 ARQ 协议可提高信道利用率。发送方维持一个发送窗口，凡位于发送窗口内的分组可以连续发送出去，而不需要等待对方确认。接收方一般采用累计确认，对按序到达的最后一个分组发送确认，表明到这个分组为止的所有分组都已经正确收到了。
优点： 信道利用率高，容易实现，即使确认丢失，也不必重传。
缺点： 不能向发送方反映出接收方已经正确收到的所有分组的信息。 比如：发送方发送了 5条 消息，中间第三条丢失（3号），这时接收方只能对前两个发送确认。发送方无法知道后三个分组的下落，而只好把后三个全部重传一次。这也叫 Go-Back-N（回退 N），表示需要退回来重传已经发送过的 N 个消息。

滑动窗口

- TCP 利用滑动窗口实现流量控制的机制。
- 滑动窗口（Sliding window）是一种流量控制技术。早期的网络通信中，通信双方不会考虑网络的拥挤情况直接发送数据。由于大家不知道网络拥塞状况，同时发送数据，导致中间节点阻塞掉包，谁也发不了数据，所以就有了滑动窗口机制来解决此问题。
- TCP 中采用滑动窗口来进行传输控制，滑动窗口的大小意味着接收方还有多大的缓冲区可以用于接收数据。发送方可以通过滑动窗口的大小来确定应该发送多少字节的数据。当滑动窗口为 0 时，发送方一般不能再发送数据报，但有两种情况除外，一种情况是可以发送紧急数据，例如，允许用户终止在远端机上的运行进程。另一种情况是发送方可以发送一个 1 字节的数据报来通知接收方重新声明它希望接收的下一字节及发送方的滑动窗口大小。

流量控制

- TCP 利用滑动窗口实现流量控制。
- 流量控制是为了控制发送方发送速率，保证接收方来得及接收。
- 接收方发送的确认报文中的窗口字段可以用来控制发送方窗口大小，从而影响发送方的发送速率。将窗口字段设置为 0，则发送方不能发送数据。

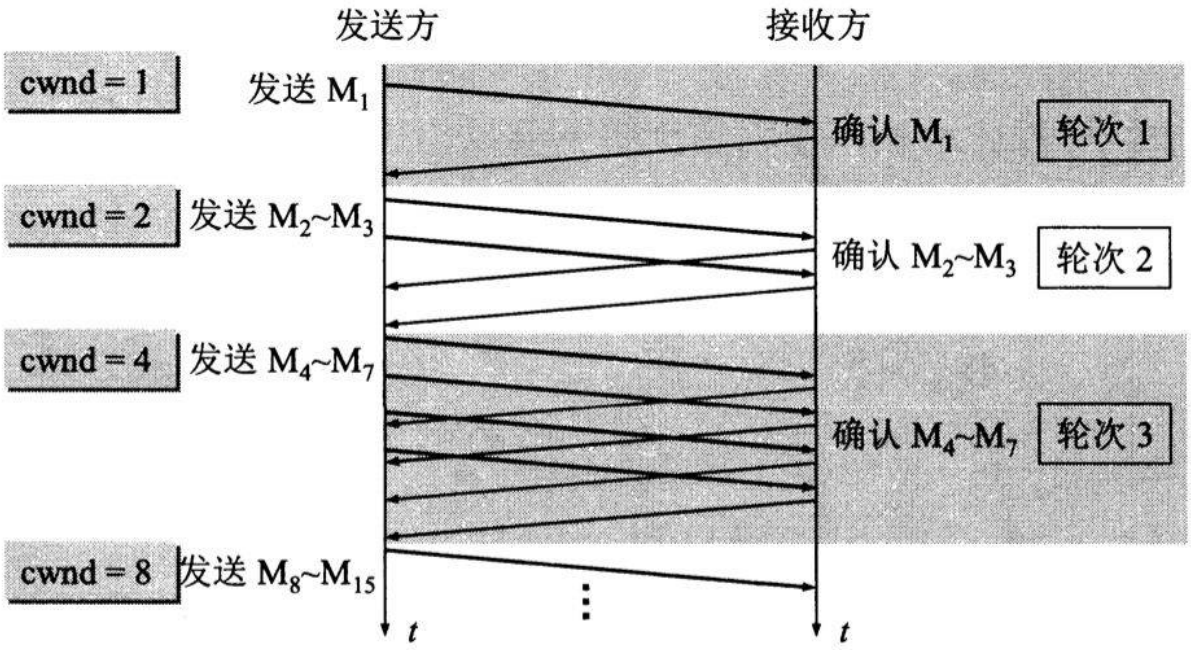
拥塞控制

在某段时间，若对网络中某一资源的需求超过了该资源所能提供的可用部分，网络的性能就要变坏。这种情况就叫拥塞。拥塞控制就是为了防止过多的数据注入到网络中，这样就可以使网络中的路由器或链路不致过载。拥塞控制所要做的都有一个前提，就是网络能够承受现有的网络负荷。拥塞控制是一个全局性的过程，涉及到所有的主机，所有的路由器，以及与降低网络传输性能有关的所有因素。相反，流量控制往往是点对点通信量的控制，是个端到端的问题。流量控制所要做到的就是抑制发送端发送数据的速率，以便使接收端来得及接收。

为了进行拥塞控制，TCP 发送方要维持一个 拥塞窗口(cwnd) 的状态变量。拥塞控制窗口的大小取决于网络

的拥塞程度，并且动态变化。发送方让自己的发送窗口取为拥塞窗口和接收方的接受窗口中较小的一个。TCP的拥塞控制采用了四种算法，即 慢开始 、 拥塞避免 、快重传 和 快恢复。在网络层也可以使路由器采用适当的分组丢弃策略（如主动队列管理 AQM），以减少网络拥塞的发生。

- **慢开始：**慢开始算法的思路是当主机开始发送数据时，如果立即把大量数据字节注入到网络，那么可能会引起网络阻塞，因为现在还不知道网络的符合情况。经验表明，较好的方法是先探测一下，即由小到大逐渐增大发送窗口，也就是由小到大逐渐增大拥塞窗口数值。cwnd初始值为1，每经过一个传播轮次，cwnd加倍。



- **拥塞避免**： 拥塞避免算法的思路是让拥塞窗口cwnd缓慢增大，即每经过一个往返时间RTT就把发送放的cwnd加1.

- **快重传与快恢复**： 在 TCP/IP 中，快速重传和恢复（fast retransmit and recovery，FRR）是一种拥塞控制算法，它能快速恢复丢失的数据包。没有 FRR，如果数据包丢失了，TCP 将会使用定时器来要求传输暂停。在暂停的这段时间内，没有新的或复制的数据包被发送。有了 FRR，如果接收机接收到一个不按顺序的数据段，它会立即给发送机发送一个重复确认。如果发送机接收到三个重复确认，它会假定确认件指出的数据段丢失了，并立即重传这些丢失的数据段。有了 FRR，就不会因为重传时要求的暂停被耽误。 当有单独的数据包丢失时，快速重传和恢复（FRR）能最有效地工作。当有多个数据信息包在某一段很短的时间内丢失时，它则不能很有效地工作。

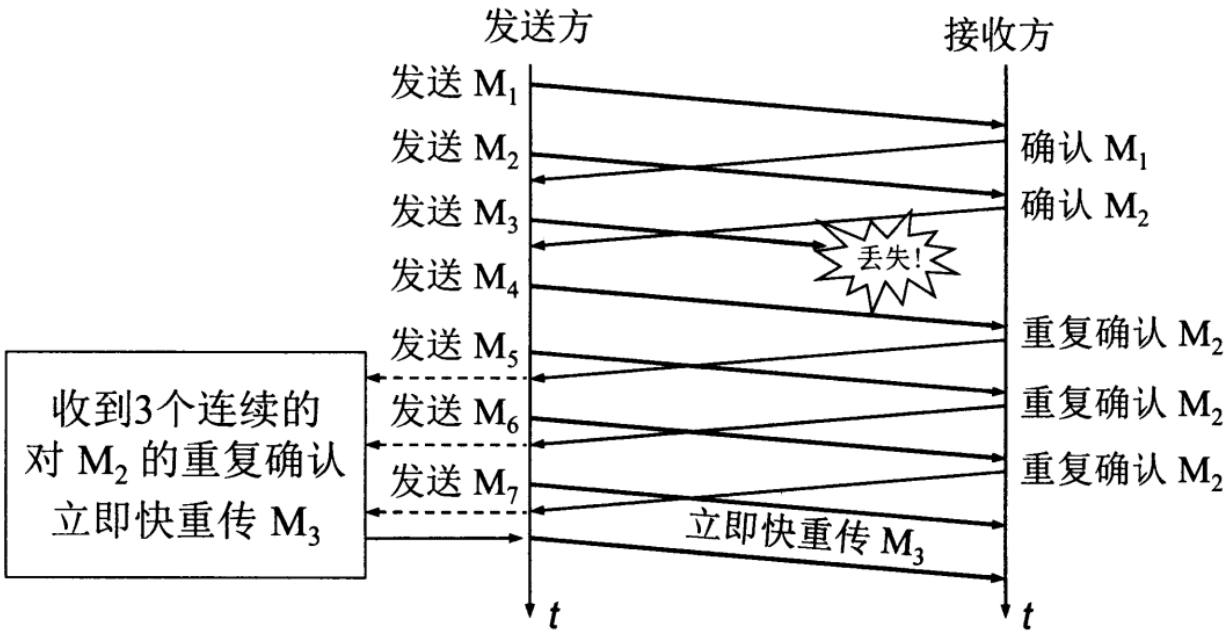


图 5-26　快重传的示意图

9. 状态码

	类别	原因短语
1XX	Informational（信息性状态码）	接收的请求正在处理
2XX	Success（成功状态码）	请求正常处理完毕
3XX	Redirection（重定向状态码）	需要进行附加操作以完成请求
4XX	Client Error（客户端错误状态码）	服务器无法处理请求
5XX	Server Error（服务器错误状态码）	服务器处理请求出错

9.1 常用端口以及对应服务

常见服务	端口
HTTP	80
FTP	21
DNS	53
POP3	110
SMTP	25
SSH	22
nginx	80
MEMCACHED	11211
MYSQL	3306
TOMCAT	8080
NFS	2049
TLENET	23
HTTPS	443
SAMBA	UDP139 TCP139
POSTFIX	25
IMAP	143
ZABBIX	10051
DHCP	56

10. 在浏览器中输入网址之后执行会发生什么？（面试常客）

DNS解析，找到对应ip地址

客户端发起http/https请求,然后交给传输层

传输层将请求分成报文段，添加目标源和端口，并随机用一个本地接口封装进报头，然后交给网络层。

网络层加上双方的ip地址信息，并负责路由分发。

链路层中，包通过链路层发送到路由器，通过邻居协议查找给定IP地址的MAC地址，然后发送ARP请求查找目的地址，如果得到回应后就可以使用ARP的请求应答交换的IP数据包进行传输了，然后发送IP数据包到达

服务器的地址。

1.确定访问的目标地址

浏览器中输入网址后，首先浏览器需要知道访问的目标地址（也就是IP地址），如果，输入的是IP地址（如：172.217.24.14），那就跳过这一步了。
如果是字符串域名，先去DNS域名服务器查询对应域名的IP地址，DNS服务器返回该域名的IP地址，域名服务器的默认访问端口是53。

2.准备HTTP通信包，分为请求头，请求行，请求体，详细的细节这里不在叙述，网上有很多资料。

浏览器（应用层）根据获得的目标IP组装好HTTP请求包。假如是4096个字节，通过建立传输层TCP连接进

行通信。

3.准备TCP通信包

首先TCP通信包分为包头（head）和包体（data），上一步中的http通信包放在TCP通信包的包体（data）部分。TCP数据包需要设置端口。

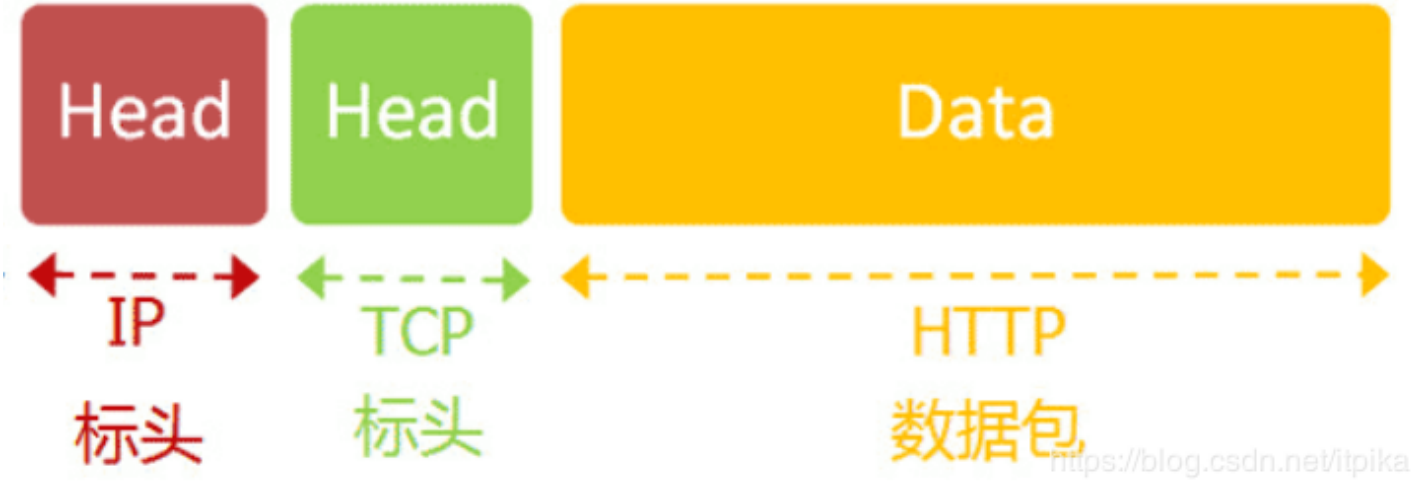
第一步可以拿到接收方的端口，发送方的端口是一个随机端口，范围为0~65535，0~0124端口默认系统占用，一般使用大于1024的端口

TCP数据包包头长度为20字节，加上HTTP协议包4096个字节，总共是 4116字节。

4.准备网络层通信包

IP协议包分为包头和包体，将TCP协议包再嵌入IP协议包的包体（data）部分，IP数据包需要设置访问方和接收方的IP地址，到这一步，IP地址已经是已知的了。

IP协议包的包头为20个字节，加上TCP协议包，总共是4136字节。



5.准备以太网协议通信包

以太网协议包分为包头和包体，将TCP协议包嵌入以太网协议包的包体（data），以太网协议包需要设置双方的MAC地址（物理网卡地址），

发送方为本机的MAC地址，接收方的MAC地址分两种情况：

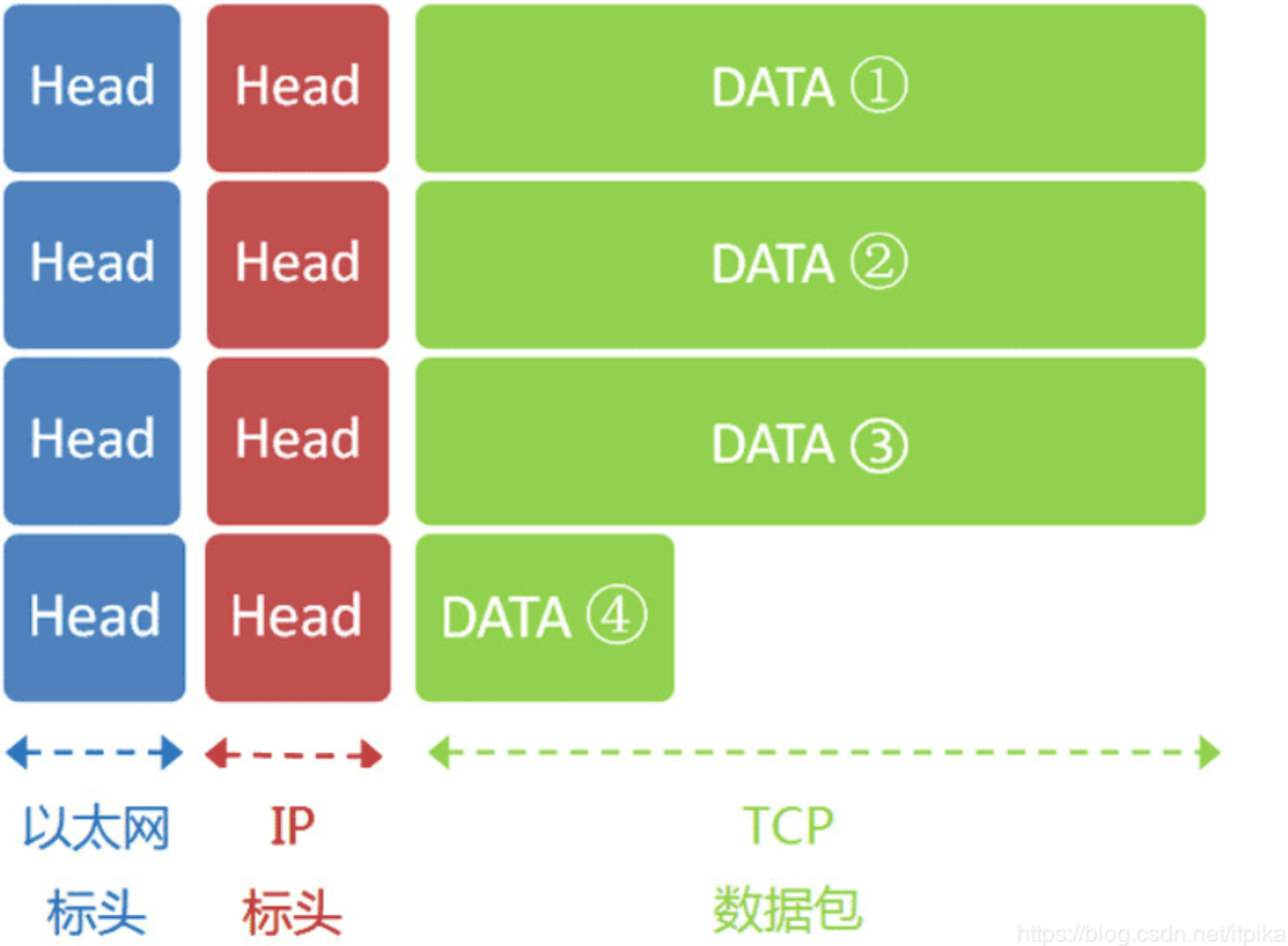
用双方的IP地址分别与访问方的子网掩码做二进制的与运算，如果结果一致，则双方处于统一子网下，

如果双方处于同一子网下，可以通过ARP协议拿到对方的MAC地址。反之不相等，则不处于同一子网下，比如我的电脑和谷歌的服务器肯定处于不同子网下，

那接收方的MAC地址就是当前子网的网关的MAC地址，以太网协议数据包的数据部分，最大长度为1500字

节，所以IP数据包需要拆分成三个包，

每个包都有自己包头，所以三个IP协议包长度为1500，1500，1196



备注：上图多画了一个数据包，应该是三个不是四个

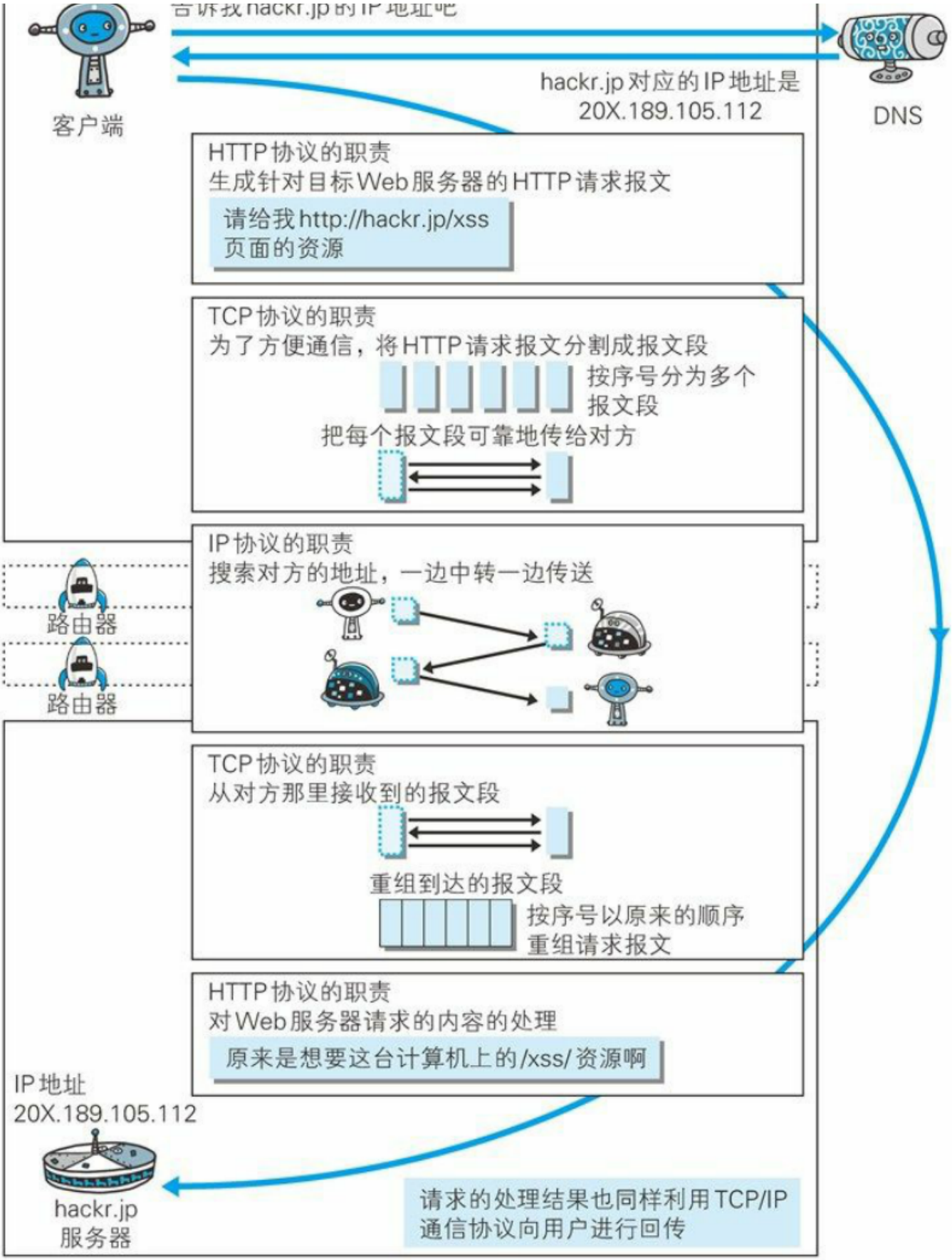
6.发送数据包

以太网数据包通过网卡发送到多重网关发送到目标服务器，目标服务器根据IP表头的序号，将三个包拼接起来，取出完成的TCP数据包，读出里面的HTTP请求包，

做出响应后，再通过上面的顺便发送响应包到接收方的电脑上，呈现在浏览器中，至此，一次http通信就完成了。

11. 各种协议与HTTP协议之间的关系

一般面试官会通过这样的问题来考察对计算机网络知识体系的理解


客户端
DNS
hackr.jp对应的IP地址是
20X.189.105.112
HTTP协议的职责
生成针对目标Web服务器的HTTP请求报文
请给我http://hackr.jp/xss
页面的资源
TCP协议的职责
为了方便通信，将HTTP请求报文分割成报文段
按序号分为多个
报文段
把每个报文段可靠地传给对方
IP协议的职责
搜索对方的地址，一边中转一边传送
路由器
路由器
TCP协议的职责
从对方那里接收到的报文段
重组到达的报文段
按序号以原来的顺序
重组请求报文
HTTP协议的职责
对Web服务器请求的内容的处理
原来是想要这台计算机上的/xss/资源啊
IP地址
20X.189.105.112
hackr.jp
服务器
请求的处理结果也同样利用TCP/IP
通信协议向用户进行回传

11.1 ip地址分类?

A类地址：以0开头，第一个字节范围：0~127（1.0.0.0 - 126.255.255.255)；

B类地址：以10开头，第一个字节范围：128~191（128.0.0.0 - 191.255.255.255);

C类地址：以110开头，第一个字节范围：192~223（192.0.0.0 - 223.255.255.255);

内部地址：10.0.0.0—10.255.255.255， 172.16.0.0—172.31.255.255， 192.168.0.0—192.168.255.255。

12. APR是什么协议？工作原理是什么？

> 地址解析协议。

1.每个主机都会在自己的ARP缓冲区中建立一个ARP列表，以表示IP地址和MAC地址之间的对应关系。

2.当源主机要发送数据时，首先检查ARP列表中是否有对应IP地址的目的主机的MAC地址，如果有，则直接发送数据，如果没有，就向本网段的所有主机发送ARP数据包，该数据包包括的内容有：源主机 IP地址，源主机MAC地址，目的主机的IP地址。

3.当本网络的所有主机收到该ARP数据包时，首先检查数据包中的IP地址是否是自己的IP地址，如果不是，则忽略该数据包，如果是，则首先从数据包中取出源主机的IP和MAC地址写入到ARP列表中，如果已经存在，则覆盖，然后将自己的MAC地址写入ARP响应包中，告诉源主机自己是它想要找的MAC地址。

4.源主机收到ARP响应包后。将目的主机的IP和MAC地址写入ARP列表，并利用此信息发送数据。如果源主机一直没有收到ARP响应数据包，表示ARP查询失败。

广播发送ARP请求，单播发送ARP响应。

13. DHCP协议的工作原理与作用？

工作原理：DHCP服务的工作过程是这样的：

1. 发现阶段，即DHCP客户机寻找DHCP服务器的阶段。DHCP客户机以广播方式（因为DHCP服务器的IP地址对于客户机来说是未知的）发送DHCP discover发现信息来寻找DHCP服务器，即向地址255.255.255.255发送特定的广播信息。网络上每一台安装了TCP/IP协议的主机都会接收到这种广播信息，但只有DHCP服务器才会做出响应。

2. 提供阶段，即DHCP服务器提供IP地址的阶段。在网络中接收到DHCP discover发现信息的DHCP服务器都会做出响应，它从尚未出租的IP地址中挑选一个分配给DHCP客户机，向DHCP客户机发送一个包含出租的IP地址和其他设置的DHCP offer提供信息。

3. 选择阶段，即DHCP客户机选择某台DHCP服务器提供的IP地址的阶段。如果有多台DHCP服务器向DHCP客户机发来的DHCP offer提供信息，则DHCP客户机只接受第一个收到的DHCP offer提供信息，然后它就以广播方式回答一个DHCP request请求信息，该信息中包含向它所选定的DHCP服务器请求IP地址的内容。之所以要以广播方式回答，是为了通知所有的DHCP服务器，他将选择某台DHCP服务器所提供的IP地址。

4. 确认阶段，即DHCP服务器确认所提供的IP地址的阶段。当DHCP服务器收到DHCP客户机回答的DHCP request请求信息之后，它便向DHCP客户机发送一个包含它所提供的IP地址和其他设置的DHCP ack确认信息，告诉DHCP客户机可以使用它所提供的IP地址。然后DHCP客户机便将其TCP/IP协议与网卡绑定，另外，除DHCP客户机选中的服务器外，其他的DHCP服务器都将收回曾提供的IP地址。

5. 重新登录。以后DHCP客户机每次重新登录网络时，就不需要再发送DHCP discover发现信息了，而是直接发送包含前一次所分配的IP地址的DHCP request请求信息。当DHCP服务器收到这一信息后，它会尝试让DHCP客户机继续使用原来的IP地址，并回答一个DHCP ack确认信息。如果此IP地址已无法再分配给原来的DHCP客户机使用时（比如此IP地址已分配给其它DHCP客户机使用），则DHCP服务器给DHCP客户机回答一个DHCP nack否认信息。当原来的DHCP客户机收到此DHCP nack否认信息后，它就必须重新发送DHCP discover发现信息来请求新的IP地址。

6. 更新租约。DHCP服务器向DHCP客户机出租的IP地址一般都有一个租借期限，期满后DHCP服务器便会收回出租的IP地址。如果DHCP客户机要延长其IP租约，则必须更新其IP租约。DHCP客户机启动时和IP租约期限过一半时，DHCP客户机都会自动向DHCP服务器发送更新其IP租约的信息。

为了便于理解，我们把DHCP客户机比做餐馆里的客人，DHCP服务器比做服务员（一个餐馆里也可以有多个服务员），IP地址比做客户需要的食物。那么可以这样描述整个过程：客人走进餐馆，问：“有没有服务员啊？”（DHCP discover），多个服务员同时回答：“有，我这有鸡翅”“有，我这有汉堡”（DHCP offer）。客人说：“好吧，我要一份汉堡”（DHCP request，这个客人比较死板，总是选择第一次听到的食物），端着汉堡的服务员回应了一声：“来啦”（DHCP ack），并把食物端到客人面前，供其享用（将网卡和IP地址绑定）。客人下次来的时候，就直接找上次那个服务员点自己喜欢的汉堡了（DHCP request），如果还有汉堡，服务员会再次确认并上菜（DHCP ack），而如果已经卖完了，服务员则会告诉客人：“不好意思，已经卖完了”（DHCP nack）。当然，服务员隔一段时间会来收拾一次桌子，除非客人特别说明这菜还要继续吃的，服务员会将剩菜端走。
作用：一个局域网的网络协议，使用UDP协议工作，用途：给内部网络或网络服务供应商自动分配IP地址，给用户或者内部网络管理员作为对所有计算机作中央管理的手段。

14. HTTP长连接、短连接

在HTTP/1.0中默认使用短连接。也就是说，客户端和服务器每进行一次HTTP操作，就建立一次连接，任务结束就中断连接。当客户端浏览器访问的某个HTML或其他类型的Web页中包含有其他的Web资源（如JavaScript文件、图像文件、CSS文件等），每遇到这样一个Web资源，浏览器就会重新建立一个HTTP会话。

而从HTTP/1.1起，默认使用长连接，用以保持连接特性。使用长连接的HTTP协议，会在响应头加入这行代码：

```
Connection:keep-alive
```

在使用长连接的情况下，当一个网页打开完成后，客户端和服务器之间用于传输HTTP数据的TCP连接不会关闭，客户端再次访问这个服务器时，会继续使用这一条已经建立的连接。Keep-Alive不会永久保持连接，它有一个保持时间，可以在不同的服务器软件（如Apache）中设定这个时间。实现长连接需要客户端和服务端

都支持长连接。

15. 子网掩码的作用?

子网掩码只有一个作用，就是将某个IP地址划分成网络地址和主机地址两部分。

通过IP和子网掩码计算网络号(笔试题)
计算出IP二进制和子网掩码的二进制，然后取与

通过IP和子网掩码计算主机号(笔试题)
将子网掩码的二进制取反，然后与IP的二进制取与。