**1、网络基础**

**Q：DHCP如何实现分配IP的？**

A： a、客户机请求IP（客户机发DHCP DISCOVER广播包）；

当DHCP客户端第一次登录网络的时候，计算机发现本机上没有任何IP地址设定，将以广播方式发送DHCP discover发现信息来寻找DHCP服务器，即向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这个广播信息，但只有DHCP服务器才会做出响应。

b、服务器响应（服务器发DHCP OFFER广播包）；

在网络中接收到DHCP discover发现信息的DHCP服务器就会做出响应，它从尚未分配的IP地址池中挑选一个分配给DHCP客户机，向DHCP客户机发送一个包含分配的IP地址和其他设置的DHCP offer提供信息。

c、客户机选择IP（客户机发DHCP REQUEST广播包）；

DHCP客户端接受到DHCP offer提供信息之后，选择第一个接收到的提供信息，然后以广播的方式回答一个DHCP request请求信息，该信息包含向它所选定的DHCP服务器请求IP地址的内容。

d、服务器确定租约（服务器发DHCP ACK）。

当DHCP服务器收到DHCP客户端回答的DHCP request请求信息之后，便向DHCP客户端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客户端可以使用它提供的IP地址。然后，DHCP客户机便将其TCP/IP协议与网卡绑定，另外，除了DHCP客户机选中的DHCP服务器外，其他的DHCP服务器将收回曾经提供的IP地址。

e、客户端重新登录（客户机发DHCP REQUEST、服务器发DHCP ack/DHCP nack）

以后DHCP客户端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，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。

f、客户端更新租约

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租约期限到达租约的50%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。

**Q：ARP/RARP**

A： ARP协议在主机发送帧前，根据目标IP地址获取目标主机的MAC地址。ARP协议只使用于局域网内部。

a、每个主机上保存有一个ARP高速缓冲区，先默认在缓冲区中查找目标IP对应的MAC地址。如果没有，进入第二步；

b、ARP协议应用在每一跳之间，发送主机和出口路由器之间、路由器和路由器之间、路由器和目的主机之间。

b、主机广播ARP请求包，其中包含自己的IP地址和MAC地址，目的主机的ARP层识别请求，单播发送自己的IP地址和MAC地址。

<https://www.cnblogs.com/luckyxiaoxuan/p/3395527.html>

**Q：ICMP协议**

A： ICMP协议是网络控制报文协议。用于传输查询报文和差错报文。

**Ping**

用于检测主机到目标主机之间是否可以通信，通过周期发送ICMP包，检测网络的丢包率、延迟等网络状况。

**Traceroute**

探测数据包从发送端到接收端的路由过程。

<https://blog.csdn.net/inject2006/article/details/2139149>

**Q：DNS协议域名查找过程？**

A： 1、查询浏览器缓存记录。

2、查询操作系统缓存数据，host文件；

3、向本地域名解析服务器查询，本地DNS服务器查询自己的高速缓存；

4、本地DNS服务器向根域名服务器查询，根域名返回顶级DNS服务器的地址；

5、本地域名服务器向顶级gTLD服务器发送请求

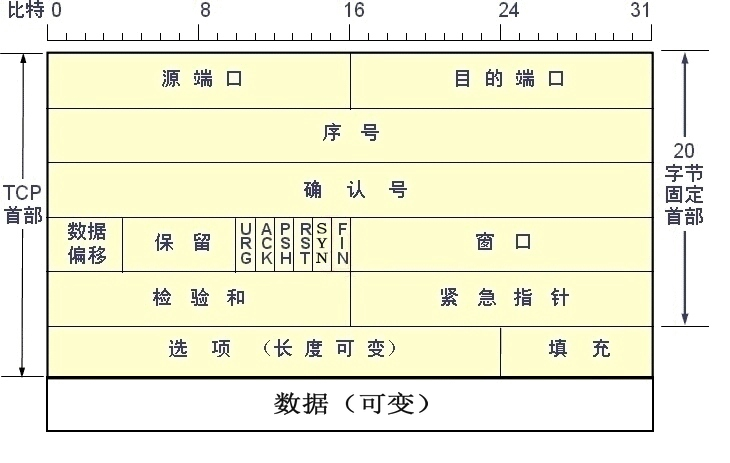
6、顶级服务器接收到本地DNS服务器的请求后，返回权限服务器的地址；

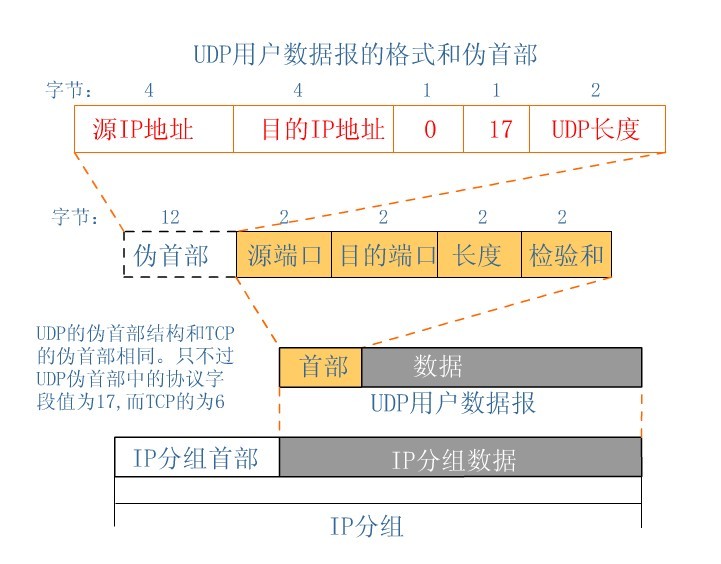
7、本地DNS服务器缓存域名和IP，并返回给客户；

8、客户本地缓存。

<http://blog.csdn.net/zhangyuan19880606/article/details/51141610>

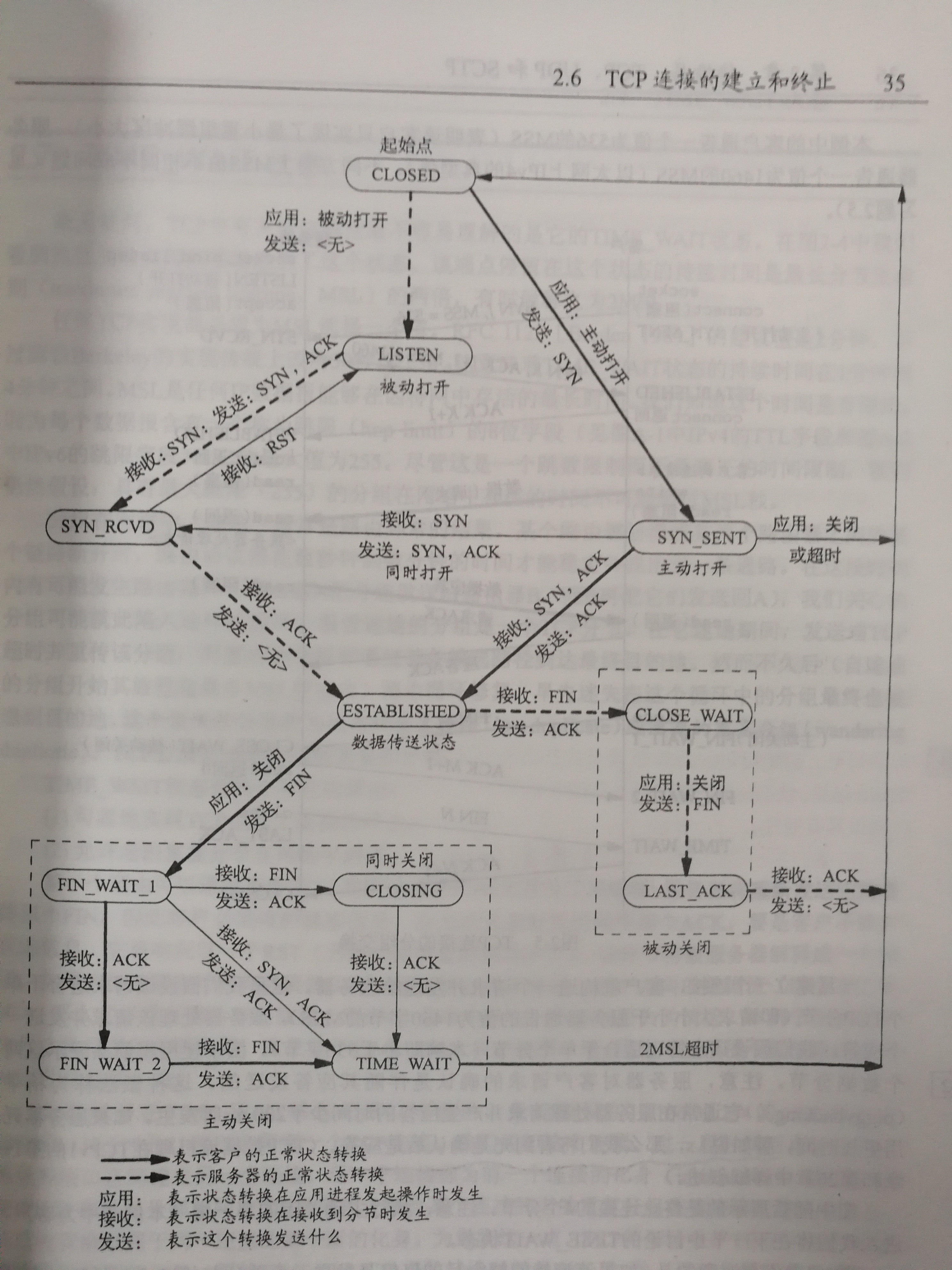
**2、TCP和UDP**

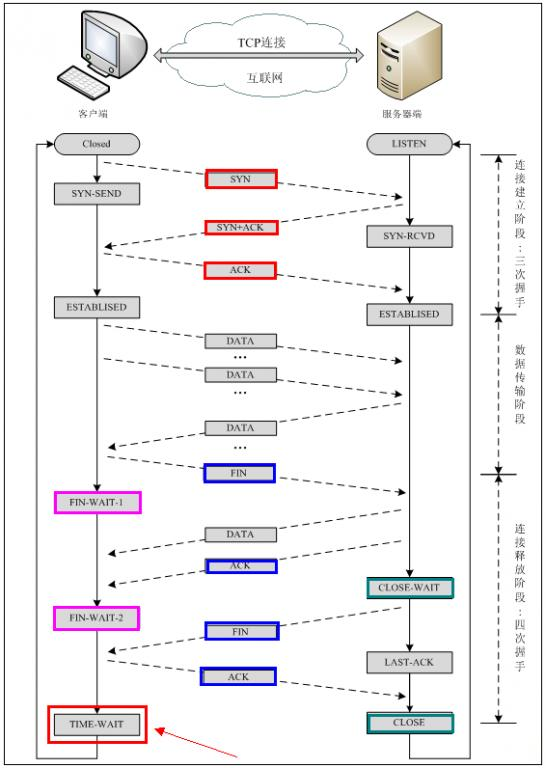




**Q：TCP三次握手、四次挥手（11种状态）**

A：





详见：程序员进阶-》计算机网络-》TCP协议中的三次握手和四次挥手

**三次握手**

a、客户端将标志位SYN置1，并产生一个随机值seq = J；发送数据包给服务器；客户端进入SYN\_SENT状态。

b、服务器端收到数据包，根据SYN位判断出是是客户端请求连接。将数据包的SYN和ACK标志位置1，设置确认号字段ack = J + 1；并产生一个随机数seq = K，设置序号字段为K，发送给客户端。

c、客户端收到数据包，检查ACK标志位是否为1，ack字段是否为J + 1。如果正确，回给服务器一个数据包，将ACK位置1，确认号字段为seq = K + 1。服务器收到数据包，检查ACK标志位是否为1，确认号字段是否为K + 1。至此，连接完成。

**四次挥手**

a、关闭端：FIN置1，seq = J；ESTABLISHED-> FIN\_WAIT1

b、对端：ACK置1，ack = J + 1；ESTABLISHED->CLOSE\_WAIT，FIN\_WAIT1->FIN\_WAIT2

c、对端：FIN置1，seq = K；CLOSE\_WAIT->LAST\_ACK

d、关闭端：ACK置1，ack = K + 1；FIN\_WAIT2->TIME\_WAIT

**CLOSING状态：**

当客户端和服务端同时关闭连接时，从FIN\_WAIT1切换到CLOSING状态。接收到对方的应答后，都进入TIME\_WAIT状态。

**Q：如TCP三次握手和四次挥手状态图所示，为什么需要TIME\_WAIT状态？**

A： 1、可靠地实现TCP的连接终止。

在终止TCP连接时有4个报文需要交换，其中最后一个ACK报文是由客户端发往服务器。假设这

个ACK报文在网络中被丢弃了，那么服务器端收不到这个确认ACK，服务器端会向客户端再次发送FIN。这就是为什么TIME\_WAIT状态持续2倍的最长报文段生存时间：1MSL时间留给最后的ACK确认报文段到达服务器端，1MSL时间留给服务器端再次发送的FIN。

2、确保老的重复的报文段在网络中过期失效，这样建立新的连接时将不再接受它们。

TCP协议采用的是出错重传，也就是会生成重复的报文，并且根据路由器的选择，这些重复的报文可能在连接终止后才到达，如果客户端/服务器端收到这个老的报文会把它误认为一个同一连接的新的报文，然后对这个报文进行处理，这样就会出现错误。从状态转换图我们可以看到从TIME\_WAIT到连接终止，中间有2MSL，这个时间足以让老的重复的报文段过期失效。

详见：程序员进阶-》计算机网络-》TCP协议中的三次握手和四次挥手

**Q：TIME\_WAIT状态的危害及解决办法**

A： 关闭连接的一方最后需要进入TIME\_WAIT状态。在这长达2MSL的时间里，该连接占用的套接口将一直被占用，影响连接的并发。

解决办法：

使用setsockopt函数，设置SO\_REUSEADDR选项为true。

Bool bReuseaddr=TRUE;

setsockopt(s,SOL\_SOCKET ,SO\_REUSEADDR,(const char\*)&bReuseaddr,sizeof(BOOL));

<http://blog.51cto.com/wpfbcr/1775701>

**Q：TCP协议三次握手和四次挥手过程中，传输报文除了请求连接信息还有哪些信息？**

A： a、MSS，最大分节大小，占用4byte。在SYN中，发送方通知接收方自己的MSS。后续发送方根据对方的MSS对TCP包进行分片。MSS过小，发送效率低；MSS过大，在IP层需要分片，当其中一片丢失时，丢弃整个包，影响效率。MSS受物理信道性能的影响，MSS随时进行动态调节。

b、滑动窗口规模，占用4byte，最大为65535（单位为byte）；

见：计算机网络-》TCP数据段格式+UDP数据段格式详解

**Q：TCP/UDP区别，分别适用于哪些场合？**

A： a、TCP面向连接；UDP是无连接的。

b、TCP通过滑动窗口、确认应答机制、分片等方式，提供可靠的服务；UDP尽最大努力交付，即不保证可靠传输。

c、TCP只支持点到点通信；UDP支持一对多的广播通信。

d、TCP首部开销20字节；UDP首部开销8个字节

e、TCP占用的系统资源多；TCP发消息过程相对复杂，效率比UDP低；

f、TCP是面向字节流的方式，可能发生粘包问题，UDP面向数据报，不存在粘包问题（UDP包头中有长度字段，但是TCP是字节流的，没有长度概念）

**应用场合：**

TCP：TCP适用于对效率要求相对低，但对准确性要求相对高的场景下。例如Telnet(远程登录)、FTP(文件传输协议)、SMTP(简单邮件传输协议)等。

UDP：UDP主要用于那些对高速传输和实时性有较高要求的通信或广播通信。例如：NFS(网络文件系统)、SNMP(简单网络管理系统)、DNS(主域名称系统)、TFTP(通用文件传输协议)、RIP（路由信息协议）、DHCP（动态主机配置协议）等

<http://blog.csdn.net/u013777351/article/details/49226101>

见：计算机网络-》TCP和UDP的比较

**Q：如何实现UDP模拟TCP？**

A： 传输层解决不了的问题，需要上层应用层解决。应用层增加包头，包头中增加以下内容：

a、序号：UDP对大数据包不分层，IP层对数据包分片后，如果任意一片丢失，整个数据包都丢弃，影响传输效率。在应用层对数据进行分片，保证传给UDP的数据包大小不超过1472。在包头中增加一个序号段，记录各分片。丢弃重复到达的数据包

b、确认号：在包头增加确认号，并采用确认应答机制，保证数据的传输。

c、窗口：UDP本身无拥塞控制，UDP接收窗口满了之后会直接覆盖之前消息。在包头增加窗口字段通知发送方自己的可用窗口大小。同时，在应用间发送消息之前，接收方需要先向发送方发送一条消息，告知自己滑动窗口大小。

d、校验和：为了保证包头数据传输过程中不被修改，同时由于UDP的校验和是选择性设置的。因此需要在应用层包头中增加对包头和数据的校验和。

e、另外，应用层需要测试数据包的平均往返时间，每次发送消息后，开启一个定时器，保证超时重传，防止出现死锁。

<http://blog.csdn.net/ls5718/article/details/52141571>

**Q：TCP/IP四层模型和OSI七层模型的对应关系，每层的主要作用和协议？**

A：OSI七层模型包括：物理层、数据链路层、网络层、传输层、会话层、表示层、应用层

a、**数据链路层**：分配信道，抓住信道，保证点到点之间的可靠传输。

协议：FDDI（分布式数据接口）、Ethernet、PPP（点对点协议）

b、**网络层**：路由选择、网络互连、拥塞控制；

协议：IP、ICMP（控制报文协议，传输出错报告控制信息）、ARP（根据IP地址解析物理地

址）、RARP（根据物理地址解析IP地址）

c、**传输层**：向上一层提供可靠的端到端（End-to-End）服务，确保“报文”无差错、有序、不丢

失、无重复地传输。它向高层屏蔽了下层数据通信的细节

协议：TCP（控制传输协议）、UDP（用户数据报协议）

d、**应用层**：为应用程序提供了网络服务

协议：HTTP、HTTPS、FTP、NFS、SMTP（简单邮件传输协议）、Telnet、SNMP（简单

网络管理协议）、DNS

补充：

**会话层**：负责建立、管理和终止表示层实体之间的会话连接（session）。

**表示层**：定义通信的数据格式及数据加密。

**Q：粘包问题**

A： TCP是面向字节流的，UDP是面向数据报的，所以只有TCP才会出现粘包问题，UDP没有（因为UDP不会使用块合并算法，每个包都有UDP的消息头）。TCP粘包是指发送方发送的若干包数据到接收方接收时粘成一包，从接收缓冲区看，后一包数据的头紧接着前一包数据的尾。

**粘包原因：**

a、发送方原因：发送方使用Nagle算法，会收集多个小分组，在一个确认到来时一起发送；此时接收方接收到的包就是首尾粘在一起包。

b、接收方原因：接收方接收到分组放到接收缓存，由应用程序去缓存中取数据。如果TCP接收分组的速度超过应用程序取分组的速度时，应用程序就会一次读到多个首尾相连的包。

**粘包解决办法：**

a、发送方关闭Nagle算法，TCP接收到消息就直接发送；

b、应用层保证发送的是定长包，浪费带宽；

c、应用层在每条消息前后加标志位或者在消息头部加包长度，每次读取时，先读长度，再读数据；

<https://www.cnblogs.com/kex1n/p/6502002.html>

<https://www.cnblogs.com/qiaoconglovelife/p/5733247.html>

**Q：TCP如何保证数据的可靠传输的**

A： a、消息确认、超时重传，保证数据成功传输；

b、数据校验和，保证数据无差错传输；

c、根据最大消息长度MSS，对大段数据进行分片和排序；

d、三次握手、四次挥手，保证连接和断开；

e、丢弃重复数据；

f、通过窗口公告，进行流量控制；

g、滑动窗口通过慢启动、拥塞避免、快重传、快恢复进行拥塞控制；

h、采用Nagle算法和TCP确认延迟机制，尽可能发送大块数据，避免网络中充斥许多小数据块。

<http://blog.csdn.net/cmm0401/article/details/77878998>

**Q：TCP拥塞控制方法**

A： a、慢启动：一开始不发送大量的数据，先探测一下网络的拥塞程度，然后逐渐翻倍增加拥塞窗口的大小，直到达到拥塞阈值。

b、拥塞避免：拥塞窗口达到拥塞阈值之后，线性增加拥塞窗口。直到出现拥塞，进入快重传和快恢复。

c、快重传：接收方收到失序报文后，立即发出重复确认。发送方连续接收到三个重复确认后，立即重传报文，而不必等待重传计时器到期。

d、快恢复 ：发送方连续收到三个重复确认时，执行“乘法减小”算法，阈值减小为出现拥塞时的拥塞阈值的一半，然后直接执行拥塞避免算法；

<http://blog.csdn.net/sicofield/article/details/9708383>

**Q：TCP滑动窗口算法**

A： a、TCP是全双工通信，所以发送方和接收方同时分别维护一段接收缓冲区和发送缓冲区。

b、发送窗口是发送缓冲区中的一部分，发送缓冲区中包括已发送但未收到确认的数据、允许发送但尚未发送的数据。

c、接收窗口是接收缓冲区的一部分，其包括已接收但未交付应用程序部分和可接收数据部分。

d、在三次握手建立连接阶段，接收方通知发送方自己的接收窗口大小，发送方设置发送窗口大小与接收窗口一致。

e、发送方通过慢启动给接收方发送数据，接收方接收数据，并在给发送方应答中捎带发送空闲窗口大小（单位为字节）。

f、当应用程序迟迟不从套接字接收缓冲区中取走数据时，接收方接收窗口会被塞满，此时给发送方发零窗口公告。

g、发送方根据接收方返回的窗口公告，动态调整发送窗口大小，减小或增大发送流量。

h、当接收方接收到数据并提交给应用程序时，接收窗口向后滑动；当发送方收到消息确认时，发送窗口向后滑动；

i、发送过程中，通过拥塞控制、快重传、快恢复保证发送效率。

j、发送方接收到接收方发送的零窗口公告后，停止发送消息。接收方有可用窗口时，给发送方新窗口公告，如果该报文丢失，可能出现发送方和接收方死锁的情况。为了防止该情况，发送方停止发送消息的同时启动一个计时器，时间到达时，发送探测报文。

<http://blog.chinaunix.net/uid-26275986-id-4109679.html>

**Q：syn攻击？**

A：利用TCP三次握手的缺陷，发送大量的SYN连接请求，服务器中将存在大量处在SYN\_RECV状态的进程。

**Q：黑客如何越过防火墙，对防火墙内部计算机进行攻击？**

A： 防火墙不会拦截已验证进程的数据传输，可以利用UDP的无连接特性，将计算机进程中的UDP通信套接字据为己用。比如说DNS。

**Q：计算机网络各层数据包的长度限制，为什么尽量使用长度有限的数据包？**

A： **长度限制：**

链路层：由以太网的物理特性决定了数据帧的长度为（46＋18）~（1500＋18），其中的18是数据帧 的头和尾，也就是说数据帧的内容最大为1500（不包括帧头和帧尾），即MTU（Maximum

Transmission Unit）为1500；

网络层：因为IP包的首部要占用20字节，所以这的MTU为1500－20＝1480；当IP包的大小超过1480时，需要进行分片；

传输层：由于IP数据包可以进行分片，所以原则上对TCP和UDP数据的大小没有限制。但是，由于UDP包的首部要占用8字节（TCP包首部占用20字节），当我们的UDP包中的数据多于MTU(1472)时，发送方的IP层需要分片进行传输，而在接收方IP层则需要进行数据报重组，由于UDP是不可靠的传输协议，如果分片丢失导致重组失败，将导致UDP数据包被丢弃，所以UDP数据最大为1472为好。

注意：IP数据包最大长度为1480是在以太网的限制下，当不受链路层限制时，IP数据包中用2个字节描述报文长度，其所能表达的最大数字就是65535。

**使用长度有限的数据包的目的：**

a、网络中每台计算机能为可能到来的最大数据包分配足够的缓冲空间

b、避免因为长消息不加分割地传递，引起的为等待通信通道空闲而出现的过度延迟；

**Q：平时用到了哪些协议、哪些端口**

A： HTTP协议、80端口

HTTPS：443

DHCP：客户端68，服务器端67；

FTP：20、21；

DNS：53

**3、算法**

**Q：加密算法有哪些？对称加密和非对称加密的区别？**

A： 对称加密和非对称加密，常用的对称加密算法有DES和AES，非对称加密算法有RSA、ECC等。

区别：

a、加密效率上来看，对称加密简单，加密解密效率快；

b、安全性上看，非对称加密消除了最终用户交换密钥的需要，保密性比较好；

由于非对称加密效率较低，非对称加密常用于少量数据传输。通信前，先用非对称加密传输对称加密的密钥，再用对称加密传输数据。

**Q：数字证书**

A： 在采用加密和对称加密过程中，数字证书用于验证发送消息的是本人。

a、A和B通信时，A为了证明自己的身份，需要向证书机构CA申请证书，证书机构采用非对称加密的方式，通过用自己的私钥对A的公钥等信息进行加密，并返回给A；

b、A向B发消息时，附上CA加密过的信息；B接收到消息后，尝试用CA的公钥解密，如果解密出的A的公钥和A发送给B的一致，说明对方是A；

**5、HTTP/HTTPS协议**

**Q：HTTP？**

A： a、超文本协议，通过因特网传送万维网文档的数据传送协议。

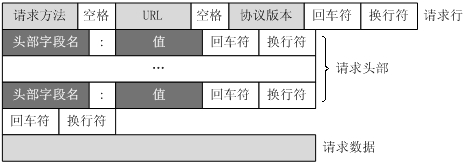
b、传输层使用TCP协议。

c、无状态协议

**超文本**：页面内包含普通图片、链接、音乐、视频等非文字元素的文本。

**Q：http报文格式**

A：http请求报文包括：请求行、请求头部、空行、请求数据。



a、请求行：由请求方法字段、URL字段和HTTP协议版本字段，组成，它们用空格分隔，例如：

GET /index.html  HTTP/1.1

b、请求头部：请求头部由key/value键值对组成，每行一对，key和value用冒号":"分隔，请求

头部通知服务器有关于client端的请求信息；

c、空行：发送回车符和换行符，通知服务器请求头结束

d、请求数据：请求数据不在GET方法中使用，而是在POST中使用

**Q：Get和Post的区别？**

A： a、提交方式：Get方式通过头部的URL提交数据，数据在URL中可以看到；POST方式，数据放置在HTTP请求消息中提交。

b、安全性：使用 Get 的时候，参数会显示在地址栏上，他人可以通过URL获取参数，而 Post 不会。

c、get只允许发ASCII字符，post没有限制。

d、数据长度：GET方式提交的数据最多只能有1024字节（因为URL有长度限制），而POST则没有此限制。

e、get可以被缓存并存入书签，参数会被保存在浏览历史中；post不可以

**Q：SESSION机制、cookie机制**

A： 由于HTTP协议是无状态的协议，所以服务端需要记录用户的状态时，就需要用某种机制来识别具体的用户，这个机制就是Session。

**session、cookie原理**

客户端首次通过http访问服务器时，服务器为客户创建一个session，每个session有一个唯一ID，服务器将该session ID返回给客户端。客户端将其保存在cookie中。客户端第二次访问服务器时，将session ID一起放在请求中发给服务器，服务器根据该ID可以找到用户对应的session。

除了存放session以外，cookie中还可以存放用户的登录名及密码等信息。

**cookie存在的问题：**

cookie中可能记录用户的账号信息，cookie被盗用之后，会被盗用身份；为了控制占用的空间，对cookie数量和长度有限制。

**客户端禁用cookie时如何解决：**

当客户端浏览器禁用了Cookie时，使用**URL重写技术**，即每次和HTTP交互，URL后都附上一个诸如sid = xxxxx的参数。

**cookie安全性问题解决办法：**

对cookie使用加密算法，并配合使用时间戳和IP戳。保证固定时间后cookie失效；并在登陆时验证客户端IP与cookie的IP是否对应。

<https://blog.csdn.net/attilax/article/details/7633025>

**Q：HTTP的长连接和短连接的区别，适用范围？**

A： **比较：**

a、在HTTP/1.0中，默认使用的是短连接。浏览器和服务器每进行一次HTTP操作，就建立一次连接，但任务结束就中断连接。

b、但从 HTTP/1.1起，默认使用长连接，用以保持连接特性。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，如果客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接。

**适用范围：**

长连接多用于操作频繁，点对点的通讯，而且连接数不能太多的情况；对于高并发量，但每个用户无需频繁操作情况下用短连好。

**Q：HTTP服务器可以主动给客户端发送数据吗？**

A： 可以采用轮询和基于HTTP长连接两种方式

**轮询**：客户端定时向服务器发送Ajax请求，服务器接到请求后马上返回响应信息，并关闭连接。

优点：后端程序编写比较容易。

缺点：请求中有大半是无用，浪费带宽和服务器资源。

实例：适于小型应用。

**长轮询**：客户端向服务器发送Ajax请求，服务器接到请求后保留连接，直到有新消息才返回响应信息并关闭连接，客户端处理完响应信息后再向服务器发送新的请求。

优点：在无消息的情况下不会频繁的请求。

缺点：服务器保留连接会消耗资源。

实例：网页版聊天软件：WebQQ、Hi网页版、Facebook IM。

**长连接**：在页面里嵌入一个隐蔵iframe，将这个隐蔵iframe的src属性设为对一个长连接的请求，服务器端就能源源不断地往客户端输入数据。

优点：消息即时到达，不发无用请求。

缺点：服务器维护一个长连接会增加开销。

实例：Gmail聊天

**Q：http2.0新特性1.1和1.0版本的区别？**

A： a、长连接和短连接：1.0短连接、1.1之后长连接

b、节约带宽：1.1之后支持传输资源的一部分。

c、host域：1.1在request消息头中增加了host域

d、多路复用：2.0中增加了多路复用技术

e、数据压缩：2.0增加了对header数据压缩

f、服务器推送：2.0增加了服务器给客户端推送消息的机制。

<http://blog.csdn.net/linsongbin1/article/details/54980801>

**Q：HTTPS协议和HTTP协议的区别**

A： HTTP 协议存在很多缺点：

a、通信使用明文（ 不加密） ， 内容可能会被窃听；

b、不验证通信方的身份， 因此有可能遭遇伪装；

c、无法证明报文的完整性， 所以有可能已遭篡改。

HTTPS是身披SSL（Secure Socket Layer，安全套接层）外壳的HTTP，将HTTP 通信接口部分用 SSL和TLS协议代替。发送数据时，HTTP将数据传给SSL，SSL先对数据进行加密处理，再传给TCP传输；接收数据时，TCP先将数据传给SSL，SSL通过证书对通信对方的身份进行验证，并验证数据的完整性，确认无误后再传给http层。

**HTTPS的缺点：**

和HTTP相比，HTTP发送请求时，除去和TCP连接外，还必须和SSL通信，因此整体处理信息量会不可避免的增加；另外，SSL需要对数据进行加密处理，服务器和客户端都需要进行加密/解密运算，会消耗更多的资源，导致负载增强。

**HTTP和HTTPS的选用：**

非敏感信息则使用 HTTP 通信，在包含个人信息等敏感数据时，利用 HTTPS 加密通信；

<https://www.cnblogs.com/wqhwe/p/5407468.html>

**Q：HTTP协议状态码**

A： 1xx：信息，服务器收到请求，需要请求者继续执行操作

2xx：成功--表示请求已被成功接收、理解、接受

3xx：重定向--要完成请求必须进行更进一步的操作

4xx：客户端错误--请求有语法错误或请求无法实现

5xx：服务器端错误--服务器未能实现合法的请求

常见状态码：

200 OK //客户端请求成功

301、302：重定向，301永久、302临时

400 Bad Request //客户端请求有语法错误，不能被服务器所理解

401 Unauthorized //请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用

403 Forbidden //服务器收到请求，但是拒绝提供服务

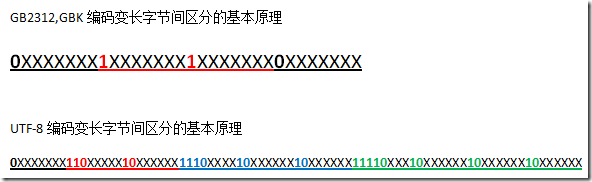
404 Not Found //请求资源不存在，eg：输入了错误的URL

500 Internal Server Error //服务器发生不可预期的错误

503 Server Unavailable //服务器当前不能处理客户端的请求，一段时间后可能恢复正常

**Q：变长编码的实现与传输问题？**

A： 以UTF-8为例，其支持1，2，3，4四种字节编码。其关键是，用高位保留位来做区分，四种长度保留位分别为：0、110、1110、11110。优点是兼容多种编码格式，包括ASCII，缺点是有效编码空间变少。



**变长编码的传输：**

**Q：打开网页到网页显示的过程**

A： a、在浏览器中输入网址；

b、发送至DNS服务器并获得域名对应的WEB服务器的IP地址；

c、与WEB服务器建立TCP连接；

d、浏览器向WEB服务器的发送相应的HTTP请求；

e、WEB服务器响应请求并返回指定URL的数据，或错误信息，如果设定重定向，则重定向到新的URL地址；

f、浏览器下载数据后解析HTML源文件，解析的过程中实现对页面的排版，解析完成后在浏览器中显示基础页面；

g、分析页面中的超链接并显示在当前页面，重复以上过程直至无超链接需要发送，完成全部显示；

h、如果采用的是短连接，通信完成关闭连接；如果是长连接，连接保持，等待发送新消息。

**Q：交换机和路由器的区别**

A： 交换机属于链路层设备，通过MAC地址寻址，通过站表选择路由；路由器属于网络层设备，根据IP地址寻址，通过路由表路由协议寻找路径。

**Q：路由算法**

A： 分为静态路由和动态路由，静态路由通过用户手动配置实现。

动态路由

a、**距离向量算法（RIP）**

每经过一个固定时间间隔，每个节点和其相邻节点交换路由信息；收敛较慢。

b、**链路状态算法（OSPF）**

与所在网络系统中的所有路由器交换消息；收敛快，但是网络中的广播消息会占用较多带宽。

**Q：IP地址的分类**

A： A类地址：1.0.0.0-126.0.0.0；

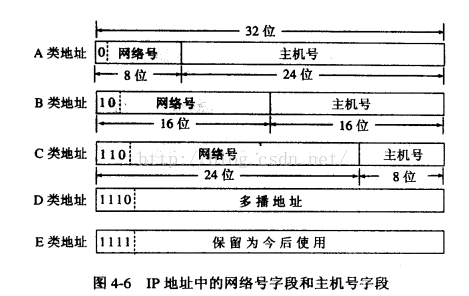
B类地址：128.1.0.0-191.255.0.0；

C类地址：192.0.1.0-223.255.255.0；

另外：

a、D类地址用于进行广播通信；

b、0.0.0.0和127.0.0.0都表示本机IP地址



//以上为整理过的面经