1、 TCP 和 UDP 的区别？

（1）TCP 提供面向连接的传输，通信前要建立连接，即三次握手机制；UDP 提供无连接的传输，通信前不需要建立连接。（2）TCP 提供可靠的传输（有序，无差错，不丢失，不重复）；UDP 提供不可靠的传输。（3）TCP 面向字节流的传输，能够将信息分割成组，并在接收端将其重组；UDP 是面向数据报的传输，没有分组开销。（4）TCP 提供拥塞控制和流量控制机制，UDP 不提供这两种机制。

2、 TCP 三次握手和四次挥手？

（1）第一次握手：建立连接时，客户端 A 发送 SYN(SYN=j)包到服务器 B，并且进入 SYN_SEND 状态，等待服务器的确认。（2）第二次握手：服务器 B 收到 SYN 包，必须确认客户端 A 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 B 进入 SYN_RECV 状态。（3）第三次握手：客户端 A 收到服务器 B 的 SYN+ACK 包，向服务器 B 发送确认包 ACK(ACK=k+1)，这个包发送完毕，完成三次握手，然后客户端与服务器开始传送数据。

客户端 TCP 状态迁移：CLOSED---SYN_SEND---ESTABLISHED---FIN_WAIT_1---FIN_WAIT_2---TIME_WAIT---CLOSED。服务器端 TCP 状态迁移：CLOSED---LISTEN---SYN_RECV---ESTABLISHED---CLOSE_WAIT---LAST_ACK---CLOSED

TCP 三次握手各个状态意义：LISTEN，侦听来自远方 TCP 端口的连接请求。SYN_SEND，发送连接后等待匹配的连接请求。SYN_RECV，在收到和发送一个连接请求后等待确认。ESTABLISHED，代表一个打开的连接，可以传输数据。FIN_WAIT_1，等待远程 TCP 的连接中断请求。FIN_WAIT_2，从远程 TCP 等待连接中断请求。CLOSE_WAIT，等待本地用户发来的连接中断请求。CLOSING，等待远程 TCP 对中断的确认。LAST_ACK，等待中断请求的确认。TIME_WAIT，等待足够的时间确保远程 TCP 接受到连接中断请求确认。CLOSED，没有任何连接的状态。

CLOSE_WAIT，发起 TCP 连接的称为客户端，被动关闭的称为服务端，被动关闭的服务器端收到 FIN 后还没有发出 ACK 确认的状态为 CLOSE_WAIT。一般都是由于服务器端的代码的问题。

TIME_WAIT，根据 TCP 定义的 3 次握手断开连接的规定，发起主动关闭 socket 的一方，socket 将进入 TIME_WAIT 状态。TIME_WAIT 状态下的 socket 不能被回收使用。对于一个处理大量连接的服务器，如果服务器主动关闭客户端的连接，将导致服务器端大量的 TIME_WAIT 状态的 socket，严重影响服务器的性能，甚至用完所有可用的 socket，停止服务。

由于 TCP 连接时全双工的，一个 TCP 连接存在双向的读写通道。因此每个方向都必须单独进行关闭。这个原则是当一方完成了它的数据发送任务后就能发送一个 FIN 来终止这个方向上的连接。收到一个 FIN 只是意味着这一方向上没有数据流动，一个 TCP 连接在收

到一个 FIN 后仍然能够发送数据。首先进行关闭的一方将执行主动关闭，而被动关闭的一方将执行被动关闭。

（1）客户端 A 发送一个 FIN，用来关闭客户端 A 到服务器 B 的数据传送。（2）服务器 B 收到这个 FIN，发回一个 ACK，确认序号为收到的序号加 1。（3）服务器 B 关闭与客户端 A 的连接，发送一个 FIN 给客户端 A。（4）客户端 A 发回 ACK 报文确认，并将确认序号设为收到序号加 1。

在 TCP 连接中，服务器端的 SYN 和 ACK 向客户端发送是一次性发送的，而在断开连接的过程中，B 端向 A 端发送数据的 ACK 和 FIN 是分两次发送的。因为在 B 端接收到 A 端的 FIN 后，B 端可能还有数据要传输，所以先发送 ACK，等 B 端处理完之后再发送 FIN 断开连接。

TCP 的读写通道是先关闭读，再关闭写。一共需要四个阶段，以客户端发起关闭连接：服务器读通道关闭---客户端写通道关闭---客户端读通道关闭---服务器写通道关闭。

关闭行为是在发起方数据发送完毕之后，给对方发出一个 FIN 数据段，直到接收到对方发送的 FIN，且对方收到了接收确认的 ACK 之后，双方的数据通信完全结束，过程中每次都需要返回确认数据段 ACK。

TCP/IP 协议中，中继器和集线器在物理层，网桥和交换机在数据链路层，路由器在网络层，网关在应用层。

3、 ARP 地址解析协议及原理？

（1）首先，每个主机都会在自己的 ARP 缓冲区中建立一个 ARP 列表，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，则向本网段的所有主机发送 ARP 数据包。数据包的内容有源主机 IP 地址，源主机 MAC 地址和目的主机 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，告诉源主机自己是目的 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，并发送数据，源主机一直没有收到 ARP 响应的数据包，则表示 ARP 查询失败。广播发送 ARP 请求，单播发送 ARP 响应。

4、 计算机网络协议？

（1）ICMP 协议：因特网控制报文协议，属于 TCP/IP 协议的子协议，用于在 IP 主机和路由器之间传递控制消息。（2）TFTP 协议：是 TCP/IP 协议族中的一个用来在客户机与服务器之间进行简单文件传输协议，提供不复杂，开销不大的文件传输服务。（3）HTTP 协议：超文本传输协议，是一个属于应用层的面向对象的协议，用于分布式超媒体信息系统。（4）DHCP 协议：动态主机配置协议，是一种让系统连接到网络上，并获取所需要的配置参数手

段。使用 UDP 协议工作，用于内部网络或者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。（5）NAT 协议：网络地址转换属接入广域网技术，是一种将私有地址转为合法 IP 地址的转换技术。

5、 TCP 为什么是三次握手？

采用三次握手是为了防止失效的连接请求报文段突然又传送到主机 B 上产生错误。失效的连接请求报文段是指：主机 A 发出的连接请求没有收到主机 B 的确认，过了一段时间，主机 A 又重新向主机 B 发送连接请求，且连接成功，顺序完成数据传输。主机 A 第一次发送的连接请求并没有丢失，而是因为网络节点导致延迟到达主机 B，主机 B 以为是主机 A 又发起了新的连接，主机 B 同意连接，并向主机 A 发回确认，主机 A 不理，主机 B 一直等待主机 A 发动数据，导致主机 B 的资源浪费。

6、 交换机、路由器、网关？

（1）交换机工作在数据链路层。交换机内部的 CPU 会在每个端口成功连接时，通过 ARP 协议学习 MAC 地址，保存一张 ARP 表。在后边的通讯中，发往该 MAC 地址的数据包将仅发到对应的端口，而不是所有的端口。因此，交换机可用于划分数据链路层广播，即冲突域，但不能划分网络层广播，即广播域。交换机有二层交换机、三层交换机、四层交换机、七层交换机。（2）路由器是一种计算机网络设备，提供了路由与转送两种重要机制，可以决定数据包从来源端到目的端所经过的路由路径，这个过程称为路由。将路由器输入端的数据包移送到合适的路由器输出端，称为转送。路由器工作在网络层，路由器的一个作用是连通不同的网络，另一个作用是选择信息传送的线路。（3）网关是连接两个网络的设备，网关能在不同协议间移动数据，路由器在不同网络间移动数据。

7、 TCP 对应的协议和 UDP 对应的协议？

TCP 对应的协议：（1）FTP 定义了文件传输协议，使用 21 端口。（2）Telnet 是一种用于远程登录的端口，使用 23 端口，用户可以以自己的身份登录连接到计算机上，可提供基于 DOS 模式下的通信服务。（3）SMTP 是邮件传送协议，用于发送邮件，开放 25 号端口。（4）POP3 用于接受邮件，使用 110 端口。（5）HTTP 是从 Web 服务器传输超文本到本地浏览器的传送协议。

UDP 对应的协议：（1）DNS 用于域名解析服务，将域名地址转为 IP 地址，使用 53 号端口。（2）SNMP 是简单网络管理协议，使用 161 端口，用来管理网络设备。（3）TFTP 是简单文件传输协议，在 69 端口。

8、 TCP 流量控制？

TCP 使用滑动窗口机制进行流量控制。建立连接时，各端分配一个缓冲区用来存储接收的数据，并将缓冲区的尺寸发送给另一端。接收方发送的确认消息中包含了自己剩余的缓冲区尺寸，剩余缓冲区空间的数量叫窗口。

9、 TCP 拥塞控制？

拥塞控制：防止过多的数据注入到网络中，可以使网络中的路由器或者链路不至于阻塞。拥塞控制是一个全局性的过程，流量控制是一个点对点的过程。

拥塞控制的方法：慢开始和拥塞避免算法。发送端维持一个叫做拥塞窗口的状态变量。拥塞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等于拥塞窗口，另外考虑到接收放的接收能力，发送窗口可能小于拥塞窗口。慢启动算法是不要一开始就发送大量的数据，先测试一些网络的拥塞程度，从小到大增加拥塞窗口的大小。慢启动算法是乘法增长，拥塞窗口加倍，拥塞避免算法是加法增长。

为了防止拥塞窗口 cwnd 增长过大引起网络拥塞，还需要设置一个慢开始门限 ssthresh 状态变量。当 cwnd<ssthresh 时，采用慢开始算法，当 cwnd>ssthresh 时，采用拥塞避免算法。当 cwnd=ssthresh 时，任意选。

拥塞避免算法让拥塞窗口缓慢增长，每经过一个往返时间 RTT 就把发送方的拥塞窗口 cwnd 加 1，而不是加倍，拥塞窗口按照线性规律缓慢增长。无论是在慢开始阶段还是拥塞避免阶段，只要发送方判断网络出现了拥塞，根据就是没有收到确认，就把慢开始门限设置为拥塞时的发送窗口的一半，然后把拥塞窗口设置为 1，执行慢开始算法，实际拥塞窗口大小为字节。

10、 快速重传和快速恢复？

快速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，为的是使得发送方早点知道有报文段没有到达，而不是等到自己发送数据的时候顺便确认。快速重传规定，发送方连续收到三个重复确认就应立即重传对方没有收到的报文段。

快速恢复算法，当发送方连续收到是三个重复确认时，就执行乘法减小算法，将 ssthresh 门限减半，但是接下去并不执行慢开始算法。考虑到网络出现拥塞的话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。

停止-等待协议是 TCP 保证传输可靠的重要途径，停止-等待协议就是指发送完一个分组之后停止发送，等待对方的确认，只有对方确认，才发送下一个分组。

回退 N 帧协议，由于停止等待协议要为每个帧进行确认后才继续发送下一帧，降低了信道的利用率。回退 N 帧协议中，发送方在发送完一个数据帧后，不停下来应答帧，而是连续发送若干个数据帧。即使在连续发送的过程中接收到了对方发来的应答帧，也可以继续发送。发送方在每发送一个数据帧时都要设置超时定时器。只要在所设置的超时时间内仍没有收到确认，就要重发相应的数据帧。回退 N 帧协议因为连续发送数据帧提高了效率，但是在重传时将原来已经正确发送的数据帧重传，降低了效率。

选择重传协议，当接收方发现某一帧出错后，其后续发来的正确帧虽然不能立即递交给接收到，但是接受党仍可以接收下来存放在一个缓冲区，同时要求发送方重传出错的那一帧。