计算机网络知识点

**1.OSI 7层参考模型：**应用层， 表示层，会话层，**传输层，网络层**，数据链路层，物理层；

**2.传输层功能：**

复用：在发送端，多个应用公用一个传输层；

分用： 在接受端，传输层会根据端口将数据分派给不同的进程；

**3.传输层和网络层区别：**

网络层为不同主机提供通信服务，传输层为不同应用提供通信服务；

网络层只对报文头部进行差错检测，传输层对整个报文进行差错检测。

**4.TCP与UDP**

UDP：

udp只在ip数据报服务上增加少量功能：复用与分用，对整个报文进行差错校验；

udp是无连接的;

udp是不可靠的，不保证每一个数据报送达以及有序，带宽问题会导致缓冲区溢出丢失数据，通信双方不知道彼此状态；

udp面向报文，不会对数据进行拆分与拼接操作；

没有拥塞控制， 可能会引起报文丢失，但对实时性要求较高的场景适用；

支持一对一，多对多，一对多通信；

首部开销小，8字节（16位（源端口号，目的端口号，长度，校验和））

TCP:

Tcp是面向连接的，通信时需要建立连接以及结束后释放连接；

提供可靠的交付服务，发送的数据无重复，有序，无错，不丢失；

面向字节流，tcp以字节为单位，将传输的数据划分为一个个数据报，接收端接受的数据会和发送端数据一模一样；

提供全双工通信，每个端都可以作为发送端/接收端；

tcp只能提供一对一通信(点到点通信)；

首部开销20(20-60)字节；

**5.保护消息边界和流**

保护消息边界：传输协议把数据当作一条独立的消息在网络上传输，接收端只能接收独立的消息;

面向流：无保护消息边界，发送端连续发送数据，接收端在接收过程中，会接受多个数据报；

**6.粘包**

发送方发送的若干数据包数据到接收方接收时变成一包，从接收缓冲区看，后一包数据头紧接着前一包数据尾；

粘包可能引起接收方对数据的解析问题；

出现原因：发送方需要等待缓冲区满才发送出去，造成沾包；

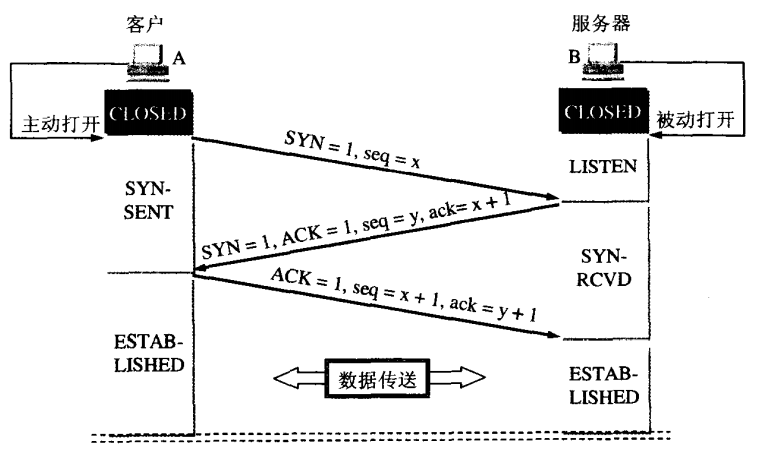
接收方不及时接收缓冲区的包，造成多个包接收；

解决：

1.用户在编程时进行设置，强制将每次要发送的数据直接发送，而不必等待发送缓冲区满；

2. 对于接收方引起的沾包，通过优化程序设计，提高接收进程优先级，及时接收数据，从而避免引起沾包现象；

**7.TCP三次握手，四次挥手**



**第一次握手：**

客户端向服务器发送连接请求报文段，报文头部有SYN= 1(同步位) , ACK=0,seq=x（字节流初始序号），客户端进程进入SYN-SENT（同步已发送）状态;

**第二次握手：**

服务器向客户端发送确认SYN = 1,ACK= 1(同意连接的应答报文),seq =y(服务器作为发送端，发送字节流初始序号),ack = x+1(期望的下一个数据报序号);

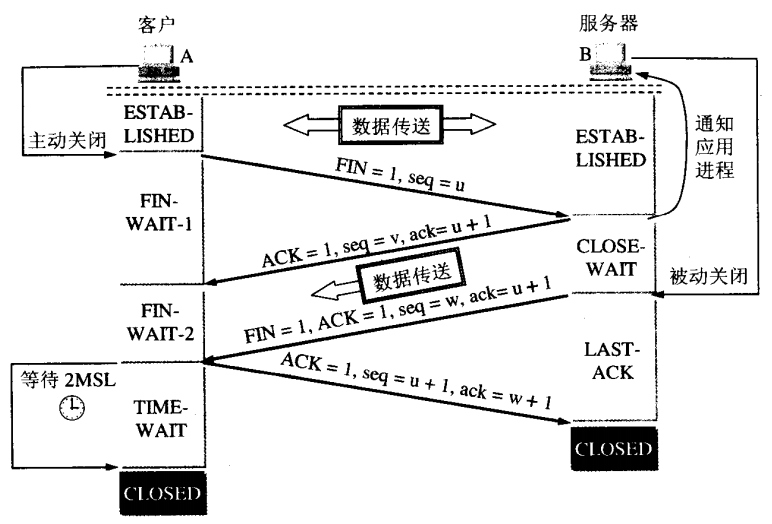
**第三次握手：**

客户端收到同意应答后，向服务器发送确认报文段，ACK =1, seq = x+1, ack = y+1;

**为什么需要三次握手：**

为了防止已经失效的链接请求报文段又传输到服务端而产生错误。

**4次挥手：**

 **第一次挥手**：客户端认为数据发送完成，向服务器发送连接释放请求。该请求只有报文头，FIN =1(连接释放请求信号), seq =u(客户端向服务器发送的最后一个字节的序号);此时，客户端进入FIN-WAIT-1状态；

**第二次挥手**：服务器收到连接释放请求后，通知相应程序客户端已经将自己端的连接释放了，服务器进入CLOSE-WAIT状态，并向客户端发送连接释放应答ACK =1(tcp通信中所有数据报的应答都是1,表示应答), seq = v(标识服务端向客户端发送的最后一个字节序号的下一位是v),ack =u+1（表示希望收到字节为u+1的数据端，前u个字节已经成功接收);

客户端收到消息后进入FIN-WAIT2阶段，等待服务端释放连接，客户端向服务端发送数据的连接已经释放；

**第三次挥手**：当服务端向客户端发送完所有数据后，向客户端发送连接释放请求，FIN = 1 ACK = 1，seq =w, ack = u+1;

**第四次**：客户端收到服务端连接释放请求后，向服务端发送确认应答，并进入TIME-WAIT状态并持续2MSL时间。在此时间内没有服务端重发请求的话，进入CLOSED状态，撤销TCB(进程控制块)，服务端接收到客户端应答后，也进入CLOSED状态，并撤销TCB。

**8.ARQ(Automatic Repeat-reQuest)协议与滑动窗口协议**

滑动窗口协议：TCP使用的流量控制的一种方法。该协议允许发送方在停止并等待确认前可以连续发送多个分组，以加速数据的传输。

ARQ：OSI模型中数据链路层纠错协议之一。它通过确认和超时机制，在不可靠的服务基础上实现可靠的信息传输。包括停止等待ARQ协议，回退ARQ协议和连续ARQ协议，错误检测，正面确认，超时重传和负面确认和重传等机制。

**分类：**

停止等待ARQ协议：发送一个报文后等待确认返回消息，再继续发送下一个报文;

回退n帧ARQ协议：发送窗口>1,接收窗口=1,允许发送方连续发送n个信息帧，一旦某一帧发生错误，必须重新发送该帧以及其后的n帧；

选择性重发ARQ协议：接收方将收到的非预期的帧缓存起来，并告知发送方自己期待的帧，发送方收到来自消息后，发送接收方缺失的帧而不是发送从该帧后所有数据。

连续ARQ协议(滑动窗口协议):发送者拥有一个发送窗口，发送者在没有得到应答的情况下可以连续发生窗口中的分组；接受者也有一个接收窗口，接受者不需要每收到一个分组就返回应答，可以连续收到分组之后统一返回一个应答；同一时刻，发送窗口不一定要和接收窗口一样大。接收端会缓存未按序到达的字节，以节约带宽。

**9.流量控制**

含义：为了避免发送端与接收端速度不匹配，导致接收端处理不了发送端发来的数据而丢弃分组，控制发送端的发送速度，使得接受者来得及接收发送端的数据。

目的：防止分组丢失;

实现：滑动窗口协议，保证了分组无差错，有序接收以及流量控制；

死锁：接受者缓冲区满后会向发送者发送一个窗口为0的应答，使发送者停止发送数据，接受者处理完数据后，会向发送者发送继续接收消息的请求，若该请求传输过程中丢失，则发送者/接收者一直等待对方，形成死锁；

持续计时器：解决死锁，当发送者收到一个0窗口应答后启动，时间一到便主动发送报文询问接受者窗口大小。若接受者仍返回0窗口应答，则重置计数器继续等待，若不为0，则表示接受者发送的报文丢失。发送者重置发送窗口后开始继续发送数据；

**10.拥塞控制与流量控制**

拥塞控制作用于网络，防止过多的数据注入网络，导致网络负载过大；(缓解网路压力/保证分组按时到达)

流量控制作用于接收者，控制发送者发送速度来保证接受者能够及时处理；

**11.慢开始算法和拥塞避免算法**

发送方维护一个发送窗口，发送窗口的大小取决于网络的拥塞情况和接收窗口的大小，动态变化;

发送方维护一个慢开始阈值：

发送窗口<阈值：使用慢开始算法；

发送窗口>阈值：使用拥塞避免算法；

发送窗口=阈值：两种算法选一个；

具体过程：

**慢开始：**

1.通信开始时，发送方的发送窗口设为1,并发送第一个分组M1；

2.发送方收到确认应答后，将发送窗口扩大为2倍；

**拥塞避免算法：**

3.当发送窗口>阈值时，每次收到确认应答后将发送窗口大小+1;

4.若出现了超时重传，表明可能发生网络拥塞，此时

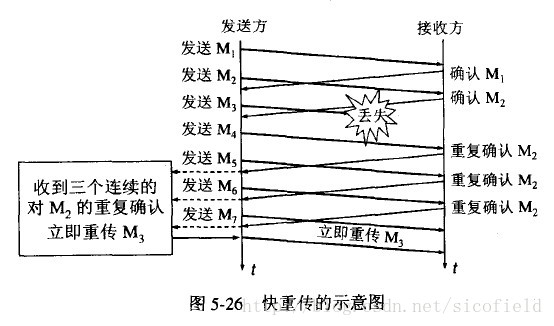
1).将阈值设为当前发送窗口的一半；

2).发送窗口设为1；

3).重新进行慢开始算法；

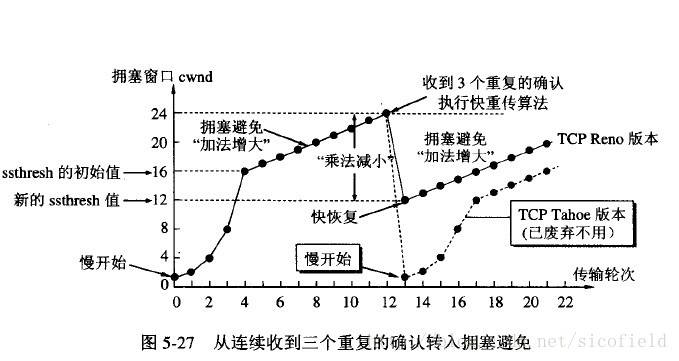
**12.快速重传算法与快恢复算法：**

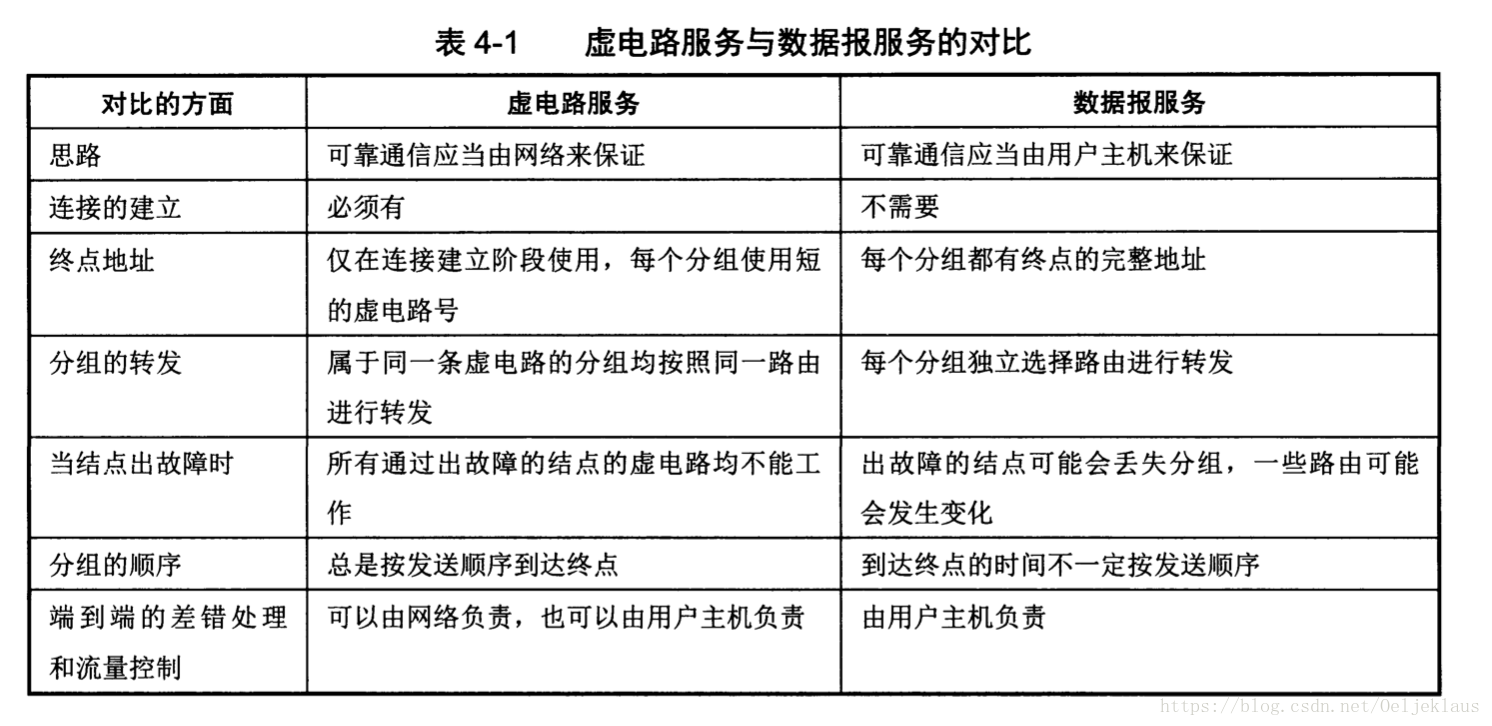
**快重传算法**：要求接收方在收到一个失序报文时立即发送重复确认而不要等待自己发送数据时捎带确认。



发送方只要收到三个重复确认后就启动快重传，而不需要等待超时；

**快恢复算法**：发送方在收到三个重复确认时，将发送窗口大小减半=新阈值，启动拥塞避免算法；



**13.TCP可靠性保证**

1.校验和：计算时加上12byte(字节)伪首部，对报文段划分为多个16位的段，将所有的段相加，对结果取反作为校验和．接收方计算结果与发送方不一致，则数据存在错误

2.序列号：TCP将每一个字节的数据都进行了编码．能够：保证数据的可靠性，保证数据按序到达，提高效率，去除重复数据

3.确认应答机制：接收方会对按序到达的数据发送确认应答告诉发送方该数据序号之前的所有数据已经按序到达．发送方收到应答报文后可以继续传输后续数据

4.超时重传机制：报文发送超过一定时间未收到来自接收方的应答，发送方会进行重传．接收方收到重复数据时，会将其丢弃，重新发送确认报文．

5.连接管理机制：建立连接的三次握手与断开连接的四次挥手

6.流量控制：发送方会根据接收方数据处理速度来改变发送速率

7.拥塞控制：发送方会使用慢启动，拥塞避免，快重传等方法来调整发送速率，以减少网络拥塞

**14. 点到点和端到端的区别**

**端到端通信**:针对传输层而言。它是一个网络连接，指在数据传输之前，在发送端和接收端之间为数据传输建立一条链路。链路建立以后，发送端可以发送数据，并且知道数据发送成功与接收端接收成功。即：在数据传输之前，现为数据传输建立一条通道，然后进行传输。

**优点**：通信链路建立以后，发送端知道接收端一定能够收到数据，并且经过中间交换设备时不需要进行存储转发，传输延时小。

**缺点**：

知道接收端收到数据为止，发送端设备一直要参与传输，如果整个传输的延迟很长，会对发送端设备造成影响。

如果接收设备关机或者故障，那么端到端传输不可能实现。

**点对点通信**：针对链路层或者网络层而言。基于IP地址或者是MAC地址。指一个设备发送数据给能够直连的其他设备，其他设备对接受到的数据进行转发，通过一台台设备连接转发将数据传输给接收方。

**优点：**

发送端发出数据后任务已经完成，不需要参与整个传输过程，这样不会造成发送端资源浪费。

接收端设备或者故障，点对点传输也可以采用存储转发技术进行缓冲

**15.使用TCP和UDP的协议**

使用TCP协议的有：HTTP，HTTPS,FTP，SSH ，TELNET， POP3， SMtP, IMAP等

使用UDP协议的有：SNMP(简单网络管理协议), NTP，DNS.RIP(路由选择协议)

**16. 私有IP**

私有IP有三个范围，处于该范围的都是私有IP

10.0.0.0-10.255.255.255

172.16.0.0-172.31.255.255

192.168.0.0-192.168.255.255

**17.ISO每一层包含哪些协议**

物理层：IEEE802.3，IEEE802.4 FDDI，ISDN

链路层： ARP ，RARP, PPP

网络层： IP，ICMP

传输层：TCP,UDP

应用层：http ，SMTP，POP3，FTP，DNS