传输层-Tcp和Udp

# 共同点

Tcp和Udp都属于传输层，都是面向端口的通讯

# 2.Tcp详解（Transmission Control Protocol）

### 2.1数据的可靠传输(可靠交付)

重发机制：TCP可靠传输的实现正是基于这样的例子，**对于发送方发送的数据，接收方在接受到数据之后必须要给予确认，确认它收到了数据。如果在规定时间内，没有给予确认则意味着接收方没有接受到数据**，然后发送方对数据进行**重发**。

**TCP会将用户需要传输的数据进行分组，即将数据进行切割，分成多个数据段(data segment)，并给每个数据段编号**。

**TCP报文段首部中通过长度为32bit的字段来表示这个TCP报文段的序号，另外通过长度为32bit的字段来表示确认报文段要确认的报文段的序号。**

总结：发送前 对**数据进行切割，分成多个报文段(data segment)，并给每个数据段编号。 发送后 ，等待对方确认，如果在规定时间内没有接收到确认数据，就重发。**

流量控制：当数据到达主机之后，TCP会将该数据放入相应的队列(又称为缓冲区)(如果让你自己基于UDP实现一个TCP模块供自己的应用程序使用，你也会采用这种方式)，等待监听该端口的应用程序从队列中获取数据，应用程序一次所能处理的数据有限，因此不可能一次性取出队列中的所有数据，当队列已经满了，则无法再存放新的数据，只能将接受到的数据丢弃，因此**TCP协议需要提供流量控制的能力，控制发送方每次发送数据的大小**。

总结：Tcp会将数据放入到队列（缓冲区）中，然后等待应用程序从缓冲区中取出数据处理，当队列（缓存区）满了之后，就无法存放新的数据，因此Tcp还具有流量控制的能力（控制发送方每次发送数据的大小）

拥塞控制

## 2.2 Tcp的连接控制

**连接是可靠传输的前提，而不是可靠传输的保证**。**连接的作用就是让通讯双方知道并准备好通讯**。

TCP属于全双工通信，连接建立之后，任何一方都可以随时发送数据，而不需要理会另外一方是否也在发送数据。

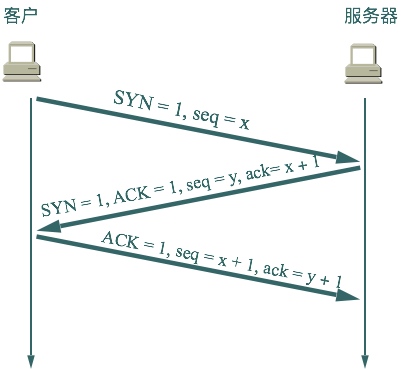
TCP连接的建立采用的是C/S(Client/Server,客户端/服务器)模型。由客户端发出建立连接的请求

### 2.2.1 Tcp的三次握手

* Client首先向Server发出建立连接的TCP报文段(SYN=1,seq=x(表示随机产生一个值,我们假设为100))，并等待Server的确认(确认收到该建立连接的请求)

1.客户端发出连接请求的报文，并等待

* Server收到该建立连接的请求之后，如果同意建立连接，则发送TCP报文段(SYN=1,seq=y(也是随机产生一个值，我们假设为21),ACK=1,ack=x+1(这里表示发送方的x号数据段已经收到,这里为101))给Client表示确认接收到该TCP报文，并等待Client确认该TCP报文已经收到
* Client收到Server的确认报文段之后，再次向Server发送TCP报文段(seq=x+1(这里为101),ACK=1,ack=y(这里为22))，确认Server的确认报文已经收到。

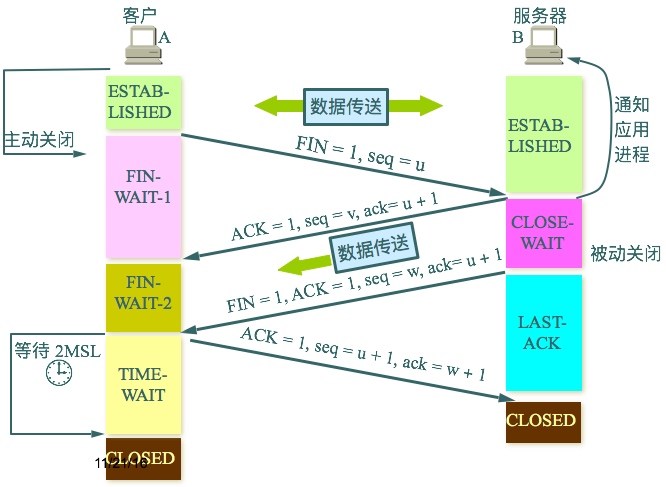


我们可以看到，控制字段中的SYN是用来建立一个TCP连接的，只在前两次"握手"中置为1，第三次"握手"置为0."我想和你谈话(SYN),我愿意和你谈话(SYN,ACK),那我们开始吧(ACK)".

### 2.3连接的断开

通过TCP三次握手建立的通信属于全双工通信，因此**每个方向都必须单独地进行关闭**。关闭的原则就是当其中一方A完成数据的传输并且不再传输数据时，发送一个控制位**FIN=1**的TCP数据段来告知另一方B"我的数据已经传输完毕，并且不再传输数据"，另一方B收到数据之后依旧发送确认TCP数据段，来表示A=>B方向的一条连接已经断开，此时TCP连接处于**半关闭状态**。需要注意的是，**这里的不再传输数据是指A不再传输用户数据，对于B传输的数据,A仍然要接收并给予确认**。

当B发送完数据之后并准备断开连接时，发送一个控制位**FIN=1,ACK=1(ack=A的FIN报文段的序号)**的TCP报文段，并等待A的确认，A收到B的FIN报文段后给予确认，至此整个TCP连接关闭。

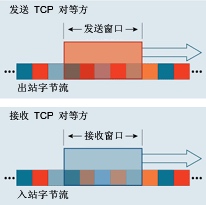


为了防止因为数据传输延时造成B的FIN报文段比B的有效数据传输报文段(用户数据)提前到达，因A会等待2MSL之后才真正关闭TCP连接。

**滑动窗口机制**

虽然上层应用和TCP的交互是一次一个数据快(大小不等)，但是TCP把上层应用程序交付下来的数据看成仅仅是一串连续的**无结构字节流**。

* 发送窗口:在为收到对方的ACK确认的情况下，只有发送窗口内的数据才能连续地发送出去。凡事已经发送过的数据，在未收到ACK确认之间都必须暂时保留在发送窗口内，以便超时重传使用。
* 接收窗口:缓冲区，用来接收发送方的TCP数据段。



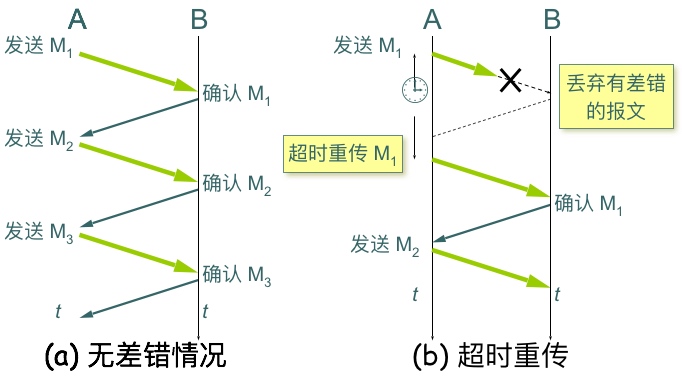
**停止-等待协议**

发送方和接收方都采用窗口大小为1的滑动窗口,即发送窗口和接受窗口都为1个最大TCP数据段的大小。  
停止等待协议的规则是:

* 发送方发完1个分组并收到接收方ACK确认之后才能发送下一个分组;
* 如果接收方收到一个错误的分组，则给发送方发送一个否认分组NAK,发送方收到NAK分组后重发，并继续等待发送方的ACK确认
* 如果发送方在规定的时限内(发送完一个分组，就开启一个定时器)没有收到接收方的ACK确认分组，则重新发送该分组。

总结：发送方：1.发送方只有在发送之后，等待接收方确认。如果在规定时间内没有确认则超时重发

接收方：收到错误分组后，发送否认分组的NAk，然后等待正确的NAk分组重发，然后再等待发送方的ACK确认：无差错传输



详情看以下链接

转载至：https://www.cnblogs.com/kakawater/p/6088948.html