### 概述：

TCP实现了可靠传输。

那些自身不包含可靠传输数据机制的协议，如IP、UDP，它们仅使用校验和或CRC这样的数学函数来检测差错，而没有实现差错纠正。

ARQ（automatic repeat request）自动重传：简单的进尝试重新发送，直到信息最终被接收，它构成了TCP在内的许多通信协议的基础。

### ARQ：

TCP使用ACK（acknowledgment）机制保证分组被正确的接收。发送方发完分组后，等待ACK消息，接收方接收到数据后发送ACK消息到发送方，发送方接收到ACK后继续发送下一个分组。

ACK机制会出现一些问题：1.发送等待ACK的时间。2.ACK丢失怎么处理。3.分组接收到后发现有错误怎么处理。

1. 见《TCP超时重传》。
2. ACK丢失，发送方无法区分是否为分组丢失，所以简单的重新发送分组。
3. 接收分组有错误时，就不发送ACK，直到发送方超时后重新发送分组。

上述情况都会有重新发送分组的操作，而接收方对于重复的分组需要丢弃。通过将每个分组都附带上一个唯一的序列号，接收方通过对比序列号即可判断是否是重复分组。

### 连续ARQ：

ARQ实现了可靠的传输，但等待ACK的机制导致效率低下，没有充分利用网络。

为了提升效率，需要允许下一个分组在未收到ACK的情况下就提前发送，提升网络的吞吐量。

发送方维护一个分组窗口，作为已发送但未接收到ACK的分组的集合。若需要重传则直接从分组窗口中取得对应分组重新发送。接收方维护分组窗口，用于维护到达分组的顺序。

### 可变窗口（流量控制、拥塞控制）：

为了处理当接收方处理速度跟不上发送方发送速度的情况。

发送方将窗口大小设置为可变的，接收方将窗口大小的信息放入ACK中，发送方根据接收到的ACK动态调整窗口大小，以此实现流量控制。

当发送方的速率超过了中间网络的处理能力，如某个路由器处理能力跟不上，导致丢包。此时涉及到拥塞控制的技术，详见《TCP拥塞控制》。

# 关键：超时时间选择，流量控制，拥塞控制。

# TCP：

TCP是面向连接的、可靠的字节流服务。

面向连接：在交换数据之前，需要先建立连接。

面向字节流：TCP不会解读字节流中的内容，只管传输。

TCP中分组携带的序列号，是每个分组第一个子节在整个字节流中的子节偏移。

TCP将应用程序的数据打散成TCP认为的最佳大小进行来发送，一般使得每个报文段不会被分片的单个IP数据报的大小来划分。

TCP维持一个强制的校验和，校验涉及到头部、任何相关的应用程序数据及IP数据报头部，若无效，则TCP会丢弃该分组。

TCP有自适应的超时重传策略。

TCP的ACK是累积的，即一个ACK可以代表直到N的所有分组已经成功接收了。若其中有ACK丢失，后续的ACK就足以确认前面的分组，从而减少不必要的重传。（批量确认）

TCP提供的是双工服务，即数据可以在两个方向上流动。一个方向的分组同时还包含了反方向的ACK，以及对应的窗口大小通知。