1.

TCP流量控制原理：所谓流量控制就是让发送速率不要过快，让接收方来得及接收。利用滑动窗口机制就可以实施流量控制。TCP连接双端都维护一个接受窗口的变量来提供流量控制，接收窗口大小是可变的，等于RcvBuffer-[LastByteRcvd-LastByteRead].

TCP可靠数据传输原理：首先，采用三次握手来建立TCP连接，四次握手来释放TCP连接，从而保证建立的传输信道是可靠的。其次，TCP采用了流水线传输，连续ARQ协议，累积确认，回退N，选择重传，快速重传，超时自动重传等来保证数据传输的正确性，使用滑动窗口协议来保证接方能够及时处理所接收到的数据，进行流量控制。最后，TCP使用慢开始、拥塞避免、快速恢复来进行拥塞控制，避免网络拥塞。

TCP拥塞控制的原理：拥塞控制就是防止过多的数据注入到网络中，这样可以使网络中的路由器或链路不致过载。TCP使用慢开始、拥塞避免、快速恢复来进行拥塞控制，避免网络拥塞。

2.

TCP：面向连接，不仅需要对端的IP地址和端口号PORT，还需要通过“三次握手”建立连接，“四次挥手”断开连接。可靠，TCP的报头中添加了很多字段来保证协议的可靠性，如：序号、确认序号、窗口大小，还具有确认应答，超时重传，流量控制和拥塞控制机制。面向字节流，在发送和接收数据时不必要一个报文一个报文的进行，可以发送/接收半个报文。具有发送缓冲区。所以TCP适用于效率要求相对低，但对准确性要求相对高的场景。因为传输中需要对数据确认、重发、排序等操作，相比之下效率没有UDP高。例如：文件传输、接受邮件、远程登录。

UDP：无连接，只需要提供对端的IP地址和端口号PORT就可以与对端进行通信。 不可靠，UDP的报头中只有校验和，但是校验和只能检查报头信息是否出错不足以保证协议的可靠性。面向数据报，在发送和接收数据时必须一个报文一个报文的进行，不能发送/接收半个报文。没有发送缓冲区；由于UDP是不可靠的，不需要将发送的数据保存至发送缓冲区再移交给内核，而是直接调用sendto发送给内核。所以UDP适用于效率要求相对高，对准确性要求相对低的场景。例如：在线视频、网络语音电话、广播通信。

3.

慢启动：主机开发发送数据报时，如果立即将大量的数据注入到网络中，可能会出现网络的拥塞。慢启动算法就是在主机刚开始发送数据报的时候先探测一下网络的状况，如果网络状况良好，发送方每发送一次文段都能正确的接受确认报文段。那么就从小到大的增加拥塞窗口的大小，即增加发送窗口的大小。

例子：开始发送方先设置cwnd=1,发送第一个报文段M1，接收方接收到M1后，发送方接收到接收方的确认后，把cwnd增加到2，接着发送方发送M2、M3，发送方接收到接收方发送的确认后cwnd增加到4，慢启动算法每经过一个传输轮次，拥塞窗口cwnd就加倍。

加法提速：当网络频发出现超时情况时，ssthresh就下降的很快，为了减少注入到网络中的分组数，而加法提速是指执行拥塞避免算法后，拥塞窗口缓慢的增大，以防止网络过早出现拥塞。

乘法降速：无论在慢启动阶段还是在拥塞控制阶段，只要网络出现超时，就是将cwnd置为1，ssthresh置为cwnd的一半，然后开始执行慢启动算法。

例子：TCP连接进行初始化的时候，cwnd=1,ssthresh=16。在慢启动算法开始时，cwnd的初始值是1，每次发送方收到一个ACK拥塞窗口就增加1，当ssthresh =cwnd时，就启动拥塞控制算法，拥塞窗口按照规律增长（加法提速）。当cwnd=24时，网络出现超时，发送方收不到确认ACK，此时设置ssthresh=12,设置cwnd=1（乘法降速）。然后开始慢启动算法，当cwnd=ssthresh=12,慢启动算法变为拥塞控制算法，cwnd按照线性的速度进行增长。

4.

假定通信双方如下，A发送4个TCP 报文给B，N-1成功到达

A方发送顺序 N-1，N，N+1，N+2

B方到达顺序

如果N没丢，但发生乱序了：

N-1，N，N+2，N+1

A收到1个ACK (N)

N-1，N+1，N，N+2

A收到2个ACK (N)

N-1，N+1，N+2，N

A收到3个ACK (N)

N-1，N+2，N，N+1

A收到2个ACK (N)

N-1，N+2，N+1，N

A收到3个ACK (N)

如果N丢了，没有到达B：

N-1，N+1，N+2

A收到3个ACK (N)

N-1，N+2，N+1

A收到3个ACK (N)

TCP报文乱序有40%的概率会造成A收到三次冗余ACK(N)，而如果N丢了，则会100%概率A会收到三次冗余ACK(N)。基于以上的统计，当A接收到三次冗余ACK(N)启动快速重传是合理的，即立马重传N，可以起到快速恢复的功效，快速修复一个丢包对TCP管道的恶劣影响。

5.

滑动窗口协议：滑动窗口协议是TCP使用的一种流量控制方法，协议中的发送和接受窗口大小从1到N，允许发送方发送多个分组而不需等待确认，但它受限于在流水线中为未确认的分组数不能超过某个最大允许数N。只有在接收窗口向前滑动时（同时接受方发送确认帧），发送窗口才有可能向前滑动。滑动窗口协议本质上是全双工的。

停止等待协议：发送窗口和接收窗口的大小都等于1，发送方发送单个分组并等待接收方的确认后才能发送下一个分组。停止等待协议本质上是半双工的。

所以滑动窗口协议效率比停止等待协议高很多。

6.

GBN：如果某个报文段没有被正确接收，则从这个报文段到后面的报文段都要重新发送。采用累计应答的方式。例如接收端返回ACK=3，则证明报文段3以及之前的报文段都被正确接收。在GBN机制下，在接收端的运输层一次只交付给上层一个分组，并且保证是按序交付的，因此如果分组k已接收，则所有序号小于k的分组也已经交付了。接收端不对失序到达的分组进行缓存。

SR：接收方设置缓存区，用于接收失序到达的分组。为每个报文段设置单独的计时器，单个分组计时器超时只重发这一个报文段。接收端返回ACK是当前接收成功报文段的序号，SR不采用累计应答的方式。

TCP：TCP使用累计应答的方式，这一点与GBN类似。TCP在接收端会设置缓存，来缓存正确接收但是失序的分组，这点与SR类似。TCP使用快速重传机制：如果收到对于一个特定报文段的3个冗余ACK，则在超时事件发生前就会对该报文段进行重传，这大大节约了时间。