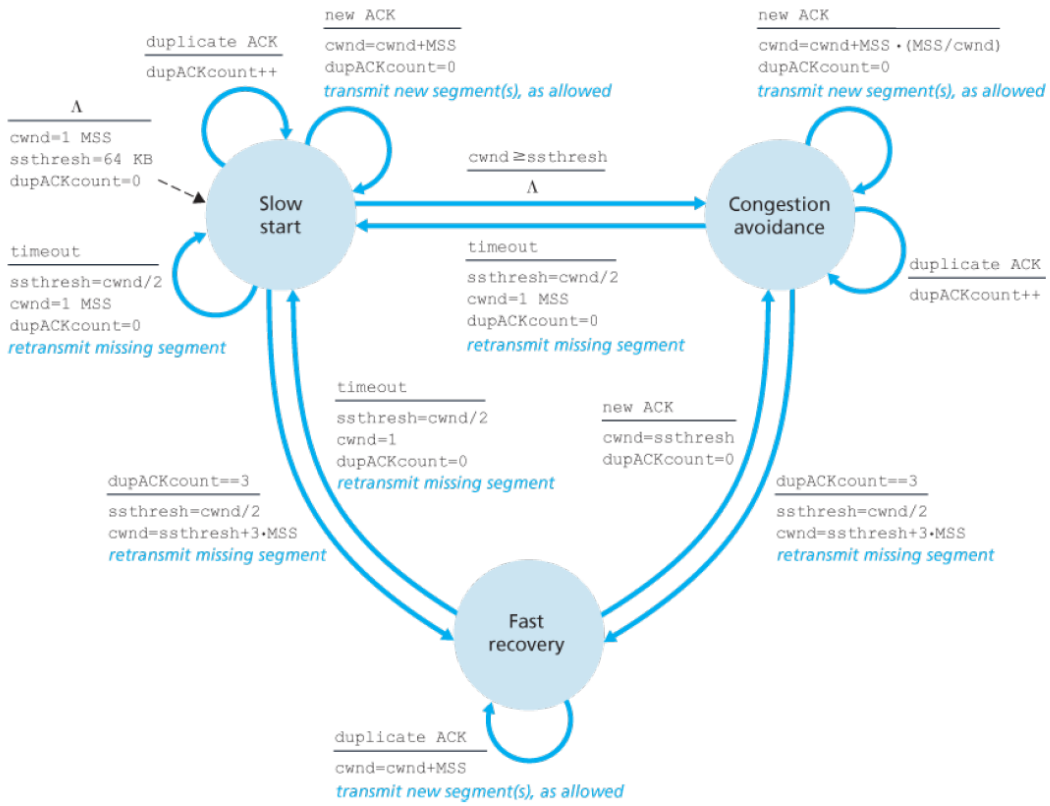
关于 TCP 拥塞控制机制的说明与理解

参考 RFC 5681 TCP congestion control

https://tools.ietf.org/html/rfc5681

TCP 的拥塞控制主要包括 3 个状态，RFC 文档中的调整策略是按照每收到一个 ACK 应答(Acknowledgment)来描述操作过程的，因此在理解的时候最好结合的是教材(第 7 版)中给出。黄色标记部分是关于“快速恢复”状态的理解。其余是关于其他部分的描述，供参考。



【关于状态转移图，圆圈表示状态；虚线箭头表示初始进入状态，以及对应的初始条件；绿色箭头实线表示状态转移方向；黑色实线的上部分表示状态转移的触发条件，下部分表示状态转移时对应的动作。】

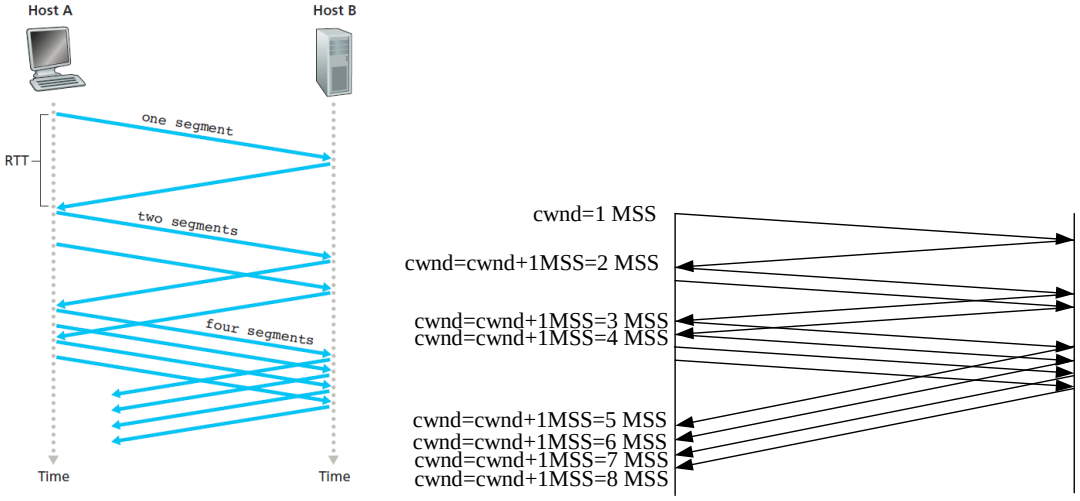
cwnd	拥塞控制窗口
MSS	(协商的)1 个最大报文段的长度，the maximum segment size
ssthresh	慢启动阈值
dupACKcount	重复 ACK 计数标记
timeout	计数器超时

TCP 中的状态主要包括 3 个，分别是：慢启动(slow start)、拥塞避免(congestion avoidance)、快速恢复(fast recovery)。根据状态转移图，启动时首先是慢启动状态（参考虚线箭头）：

- 在慢启动状态：初始时，cwnd 为 1 个报文段(MSS)，ssthresh(慢启动阈值)为 64KB，dupACKcount(重复 ACK 计数标记为 0)；
  - 慢启动状态下 cwnd 的增加规则为：每收到一个新的 ACK，cwnd=cwnd+1MSS，即拥塞窗口增加 1 个最大报文段长度；

    【这个规则的另一种描述为每 RTT 时间，cwnd 指数倍变化。例如第 1 个 RTT，cwnd

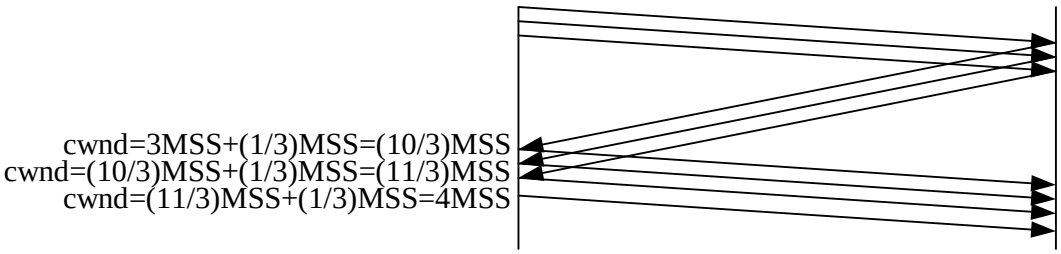
为 1 个 MSS，假设接收方立即返回 1 个 ACK，则第 2 个 RTT 开始之前，cwnd 变为 2 个 MSS，以此内推；第 2 个 RTT，发送方发出 2 个 MSS，接收方收到并返回 2 个 ACK，则第 3 个 RTT 开始之间，cwnd 变为 4 个 MSS（说明，收到第 1 个 ACK，cwnd 为 3，收到第 2 个 ACK，cwnd 为 4）；这是书上的原文解释】



- 当 cwnd 超过慢启动阈值(ssthresh)时，状态就从“慢启动”转为“拥塞避免”状态；

- 在拥塞避免状态，
  - 在拥塞避免状态，cwnd 的增加规则为：每收到一个新的 ACK，
    cwnd = cwnd + MSS*MSS/cwnd；注意这里不是增加 1 个最大报文段长度，而是增加（MSS/cwnd）个 MSS；

【这个规则的理解就是每个 RTT，cwnd 才总共增加 1 个 MSS，图示】



- 在拥塞避免状态（或慢启动状态），当收到 3 个重复 ACK，进入快速恢复状态，在状态转移的时候，执行的操作是将慢启动阈值设置为当前拥塞控制窗口的一半，即 ssthresh = cwnd/2；将新的拥塞控制窗口设置为新的慢启动阈值加 3 个 MSS，cwnd = ssthresh + 3MSS；重新传输丢失的报文段；

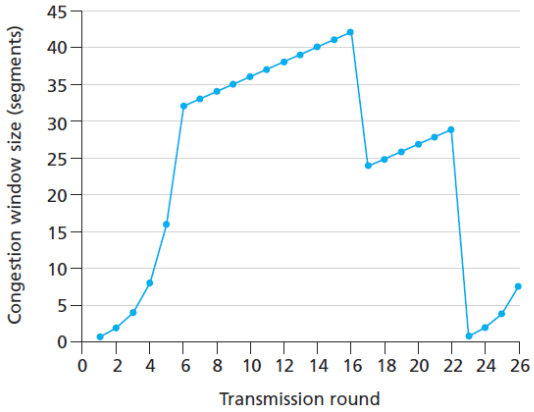
【举例说明：假设收到 3 个重复 ACK 时的 cwnd 为 10 个 MSS，则新的 ssthresh=5MSS，新的 cwnd=5MSS+3MSS=8MSS】

- **当处于快速恢复状态时比较复杂，有两种不同的操作，**
  - 每收到 1 个重复的 ACK，拥塞控制窗口的变化：cwnd = cwnd + MSS（注意这里是每收到一个重复 ACK，就增加 1 个 MSS）；然后传输允许的新的报文段【感觉不是很合理，但是 RFC 原文是这样写的】；
  - 当收到 1 个新的 ACK 时，从快速恢复状态返回拥塞避免状态，cwnd = ssthresh，这里的慢启动阈值是进入快速恢复状态时的计算值，沿用上面的例子，返回“拥塞避免”状态后，cwnd=5MSS；

因此，在出题的时候，只描述 TCP 进入快速恢复状态，而不说明后面到底发生的是哪种情况，学生容易分不清后续的 cwnd 窗口的变化。

例题分析：

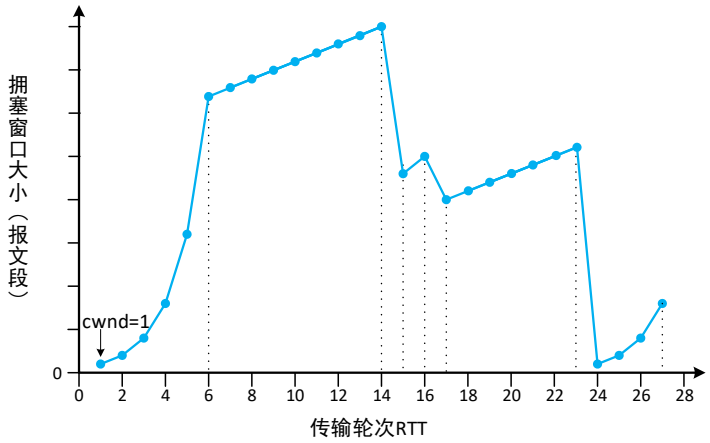
书上的课后题：



在这个题目中，[6,16]是拥塞避免状态，在第 16 个 RTT，cwnd=42 MSS，这个时候收到 3 个重复 ACK，因此第 17 个 RTT 时 ssthresh=21MSS，cwnd=(21+3)MSS，

如果按照书上的答案[17, 22]也处于拥塞避免状态，说明在进入快速恢复恢复状态时，很快就收到了新的 ACK 确认，这个时候 **cwnd=ssthresh=21MSS，然后再根据收到的新的 ACK 情况修改 cwnd 的值，因为是“拥塞回避”状态，所以这个 cwnd 增加最多就约 1 个 MSS，那么第 18 个 RTT 开始时的，cwnd 窗口到底是多少呢？22？这个就与书上的画法不同。准确的说应该是从 cwnd=21 开始线性增长。**

我看了一下 2018 年的中期考试这道题应该是正确的哈。可以参考这道题的出题方法来做。



(1) 轮次 6 开始时，拥塞窗口 cwnd=（ 32 ），

(2) 轮次 14 开始时，拥塞窗口 cwnd=（ 40 ），

(3) 轮次 15 开始时，拥塞窗口 cwnd=(23)，ssthresh=(20)

(4) 在轮次 15 到轮次 16 期间，发送方又收到了 2 个冗余的 ack，则轮次 16 开始时，拥塞窗口 cwnd=（ 25 ）

(5) 在轮次 16 快要结束时，由于收到了（ 新的确认(或 new ack) ），

导致 TCP 拥塞控制状态，在轮次 17 开始时，由快速恢复阶段迁移到拥塞避免阶段，则在轮次 17 开始时，拥塞窗口 cwnd=（ 20 ），ssthresh=（ 20 ）

(6) 轮次 23 开始时，拥塞窗口 cwnd=(26),ssthresh=(20)

(7) 轮次 24 开始时，拥塞窗口 cwnd=(1),ssthresh=(13)

假定在第 27 个传输轮次后，通过收到 3 个冗余的 ACK 检测到有分组丢失，则此后拥塞窗口 cwnd=（ 7 ），ssthresh=（ 4 ）