

传输层

选择题

拥塞窗口

简答：

8. 停等协议主要使用何种机制解决 IP 层的丢包、乱序和重复？
9. 为什么需要进行流量控制？ TCP 采用何种机制进行流量控制？
为什么需要流量控制？
TCP 的流量控制机制：滑动窗口（Sliding Window）
10. TCP 已有流量控制，为何还需要拥塞控制？
流量控制与拥塞控制的本质区别
为何需要拥塞控制？
TCP 拥塞控制的核心机制

TCP格式

用三次握手以防止出现错误连接，当三次握手出现错误的连接，响应RST，ACK

选择题

拥塞窗口

1.TCP 采用慢启动进行拥塞控制，若 TCP 在某轮拥塞窗口为 8 时出现拥塞，经过 4 轮均成功收到应答，此时拥塞窗口为（ A）

A. 5 B. 6 C. 7 D. 8

- 慢启动（Slow Start）机制：
 - 出现拥塞时，阈值（sssthresh）设为当前拥塞窗口（CWND）的一半（本题中，原 CWND=8，sssthresh=4）。
 - 拥塞后，重新进入慢启动阶段，CWND 从 1 开始按指数增长，每轮成功收到应答则 CWND 翻倍，但不超过 sssthresh。
- 4 轮后的 CWND 变化：

- 第 1 轮：CWND=1（初始值）
- 第 2 轮：CWND=2（翻倍）
- 第 3 轮：CWND=4（翻倍，达到 ssthresh=4）
- 第 4 轮：进入拥塞避免阶段，CWND 按线性增长（+1），变为 5。

简答：

8. 停等协议主要使用何种机制解决 IP 层的丢包、乱序和重复？

停等协议（Stop-and-Wait Protocol）通过以下机制解决 IP 层的丢包、乱序和重复问题：

1. 超时重传（Timeout Retransmission）

- **解决丢包：**发送方发送数据后启动定时器，若未在超时时间内收到确认（ACK），则重新发送该数据。
- **原理：**利用定时器检测丢包（包括数据帧丢失或 ACK 丢失），通过重传恢复丢失的数据。

2. 序号（Sequence Number）

- **解决乱序和重复：**为每个数据帧分配唯一序号（0/1 交替）。为什么要01交替
 - **乱序：**接收方通过序号判断数据帧是否按顺序到达，丢弃序号错误的帧（如重复或提前到达的帧）。
 - **重复：**若接收方收到相同序号的帧，判定为重复数据，直接丢弃并重新发送 ACK。

总结：停等协议通过 **序号 + 超时重传** 的组合机制，确保数据在不可靠的 IP 层上实现可靠传输，解决丢包、乱序和重复问题。

9. 为什么需要进行流量控制？TCP 采用何种机制进行流量控制？

为什么需要流量控制？

- **核心目标：**协调发送方与接收方的速率，避免发送方发送数据过快导致接收方缓冲区溢出

（即“淹没”接收方）。

- **场景示例：**若发送方速率远高于接收方（如高速服务器向低速手机发送数据），接收方来不及处理数据，会导致缓冲区满、数据丢失及重传，降低整体效率。

TCP 的流量控制机制：滑动窗口（Sliding Window）

- **基本原理：**
 - 接收方在确认报文中通过 **窗口字段（Window Size）** 告知发送方自己当前的接收缓冲区可用大小（即“接收窗口”）。
 - 发送方根据接收窗口动态调整自己的发送窗口大小，确保发送的数据量不超过接收方的处理能力。
- **关键细节：**
 - a. **窗口字段：**占 16 位，单位为字节，最大值为 65535（可通过 **窗口扩大因子** 扩展）。
 - b. **动态调整过程：**
 - 接收方处理数据后，缓冲区空闲空间增加，会在 ACK 中增大窗口字段，允许发送方发送更多数据。
 - 若接收方缓冲区满，窗口字段设为 0，发送方停止发送（直到收到窗口更新通知）。
 - c. **零窗口通知（Zero Window）：**接收方缓冲区满时发送零窗口报文，发送方进入阻塞状态，等待接收方的 **窗口更新通知（Window Update）**。

总结：TCP 通过 **滑动窗口机制** 实现端到端的流量控制，基于接收方的实际处理能力动态调节发送速率，避免数据丢失和效率浪费。

10. TCP 已有流量控制，为何还需要拥塞控制？

流量控制与拥塞控制的本质区别

维度	流量控制	拥塞控制
目标	解决发送方与接收方的速率匹配问题	解决网络中所有主机、路由器等资源的拥塞问题
作用范围	端到端（发送方 ↔ 接收方）	全局（整个网络层面）

触发原因	接收方缓冲区不足	网络中链路带宽、路由器队列等资源不足
------	----------	--------------------

为何需要拥塞控制？

1. 流量控制的局限性：

- 流量控制仅关注发送方与接收方的局部速率匹配，无法感知网络整体状态。
- **示例：**若多个发送方同时向同一网络注入大量数据，即使每个接收方的流量控制正常，也可能导致网络拥塞（如路由器队列溢出、分组丢弃）。

2. 拥塞控制的必要性：

- 网络资源（带宽、缓冲区）是有限的，需全局协调所有发送方的速率，避免因过度占用资源导致网络瘫痪。
- **拥塞的后果：**分组延迟增加、丢弃率上升，甚至引发“拥塞崩溃（Congestion Collapse）”（网络吞吐量急剧下降）。

TCP 拥塞控制的核心机制

- **慢启动（Slow Start）：**初始阶段缓慢增加拥塞窗口（CWND），避免突发流量冲击网络。**理解其具体过程，可见上题**
- **拥塞避免（Congestion Avoidance）：**当 CWND 超过阈值（ssthresh）后，转为线性增长，降低拥塞概率。
- **快速重传（Fast Retransmit）：**通过重复 ACK 检测丢包，提前重传丢失的分组，减少等待超时的时间。
- **快速恢复（Fast Recovery）：**配合快速重传，在检测到拥塞后调整窗口大小，快速恢复传输效率。

总结：流量控制解决端到端的速率匹配，而拥塞控制解决全局网络资源竞争问题。两者缺一不可：

- 流量控制确保接收方不被“淹没”，
- 拥塞控制确保网络不被“压垮”，
- 共同保障 TCP 在复杂网络环境下的稳定性和效率。

