

计算机网络-实践课

Computer Networks

王春阳

2025. 10



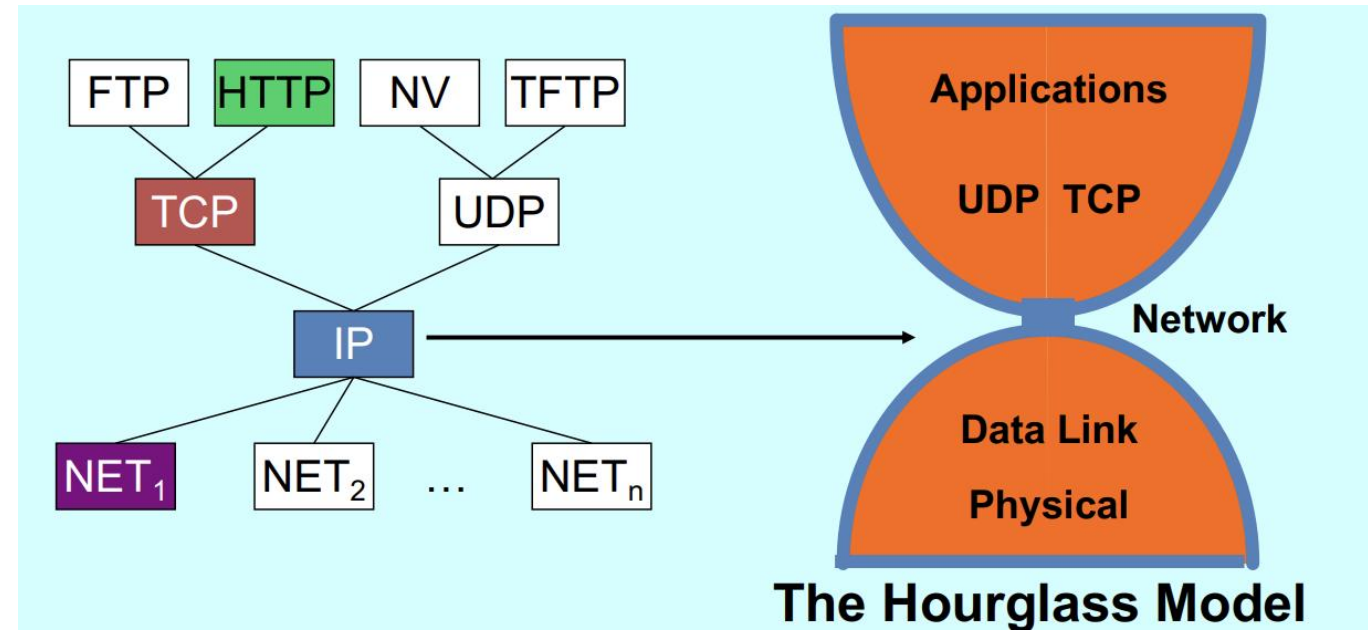
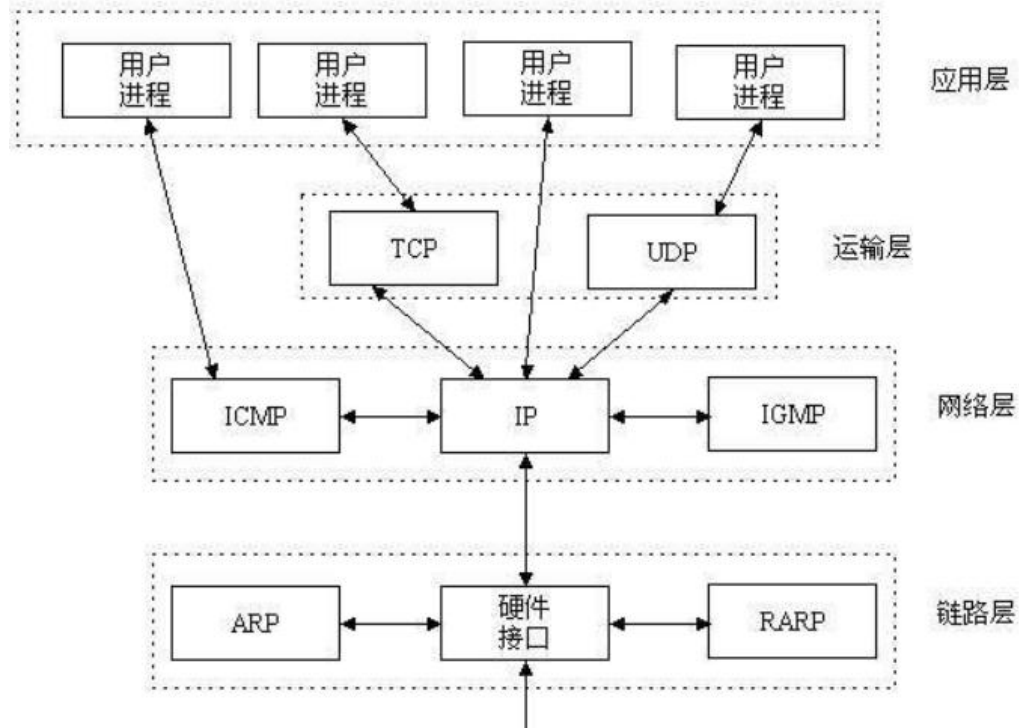
实践课教务

- 主讲老师: 王春阳
- 办公室: 地理馆315
- Email: cywang[@dase.ecnu.edu.cn](mailto:cywang@dase.ecnu.edu.cn)
- 助教TA: 刘蔚美 (51285903074@stu.ecnu.edu.cn)
杨嘉莉 (10234804407@stu.ecnu.edu.cn)

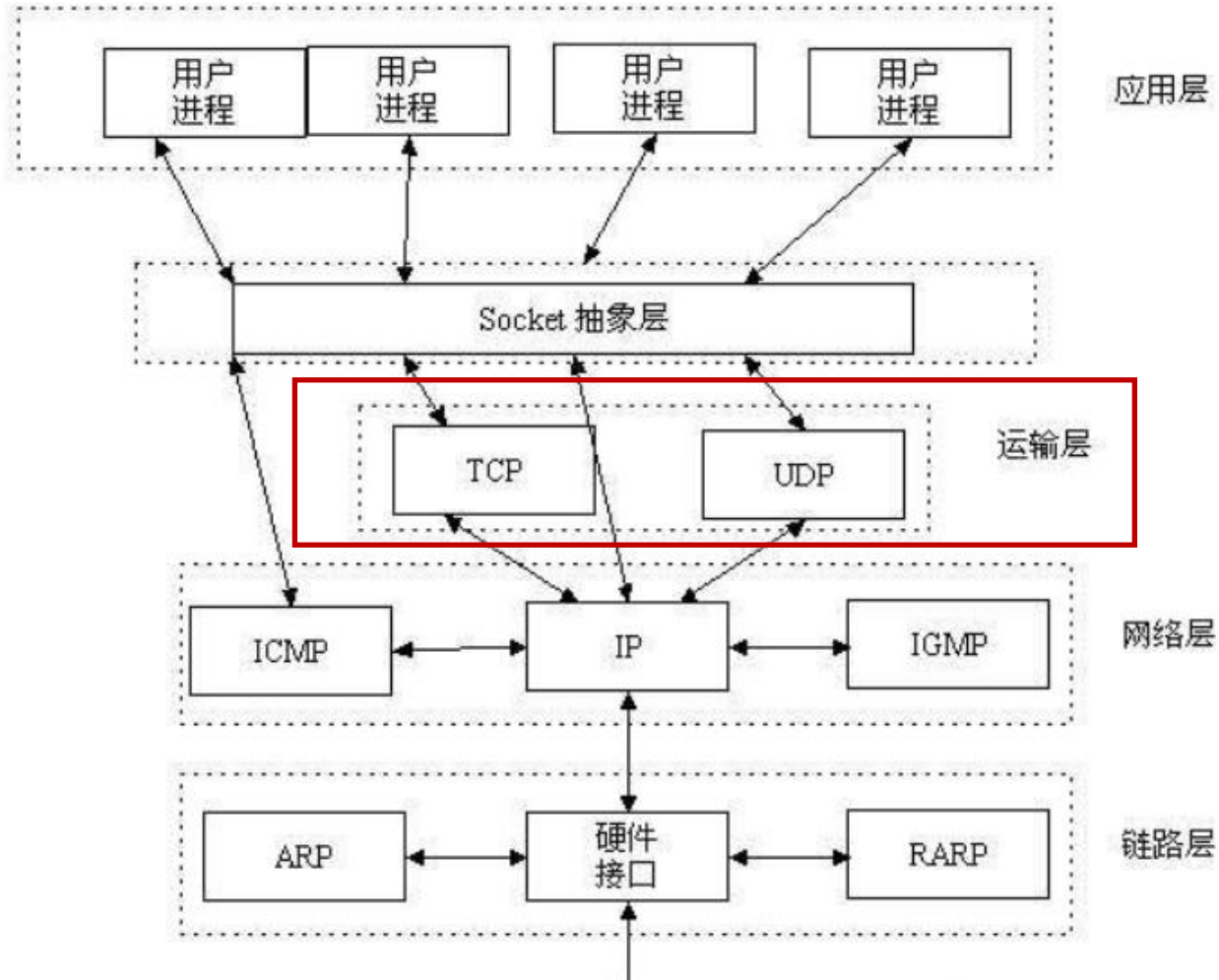
Assignment 2.

TCP Congestion Control and Bufferbloat

五层网络模型 - TCP/IP协议族



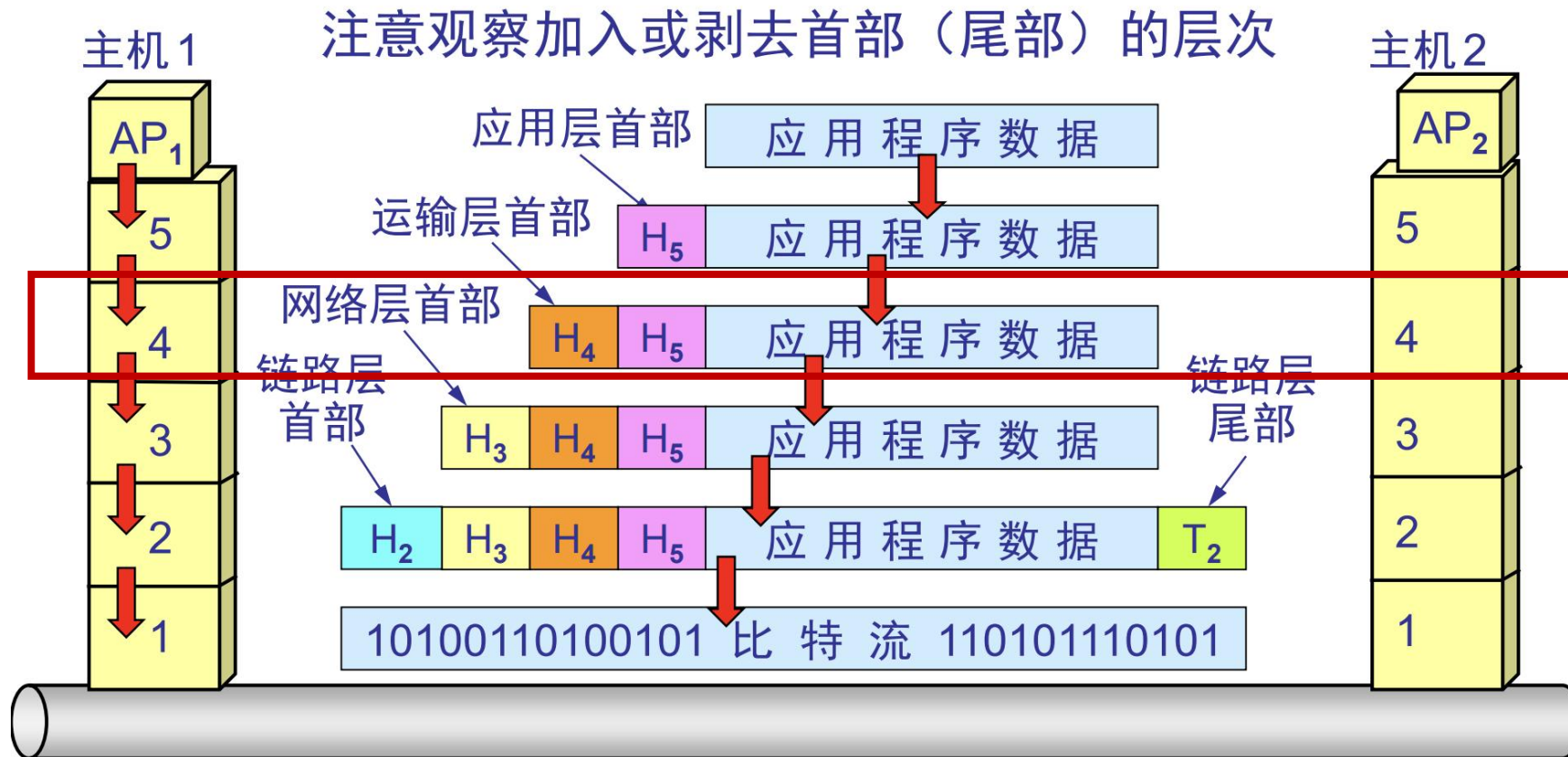
五层网络模型 - TCP/IP协议族



运输层协议

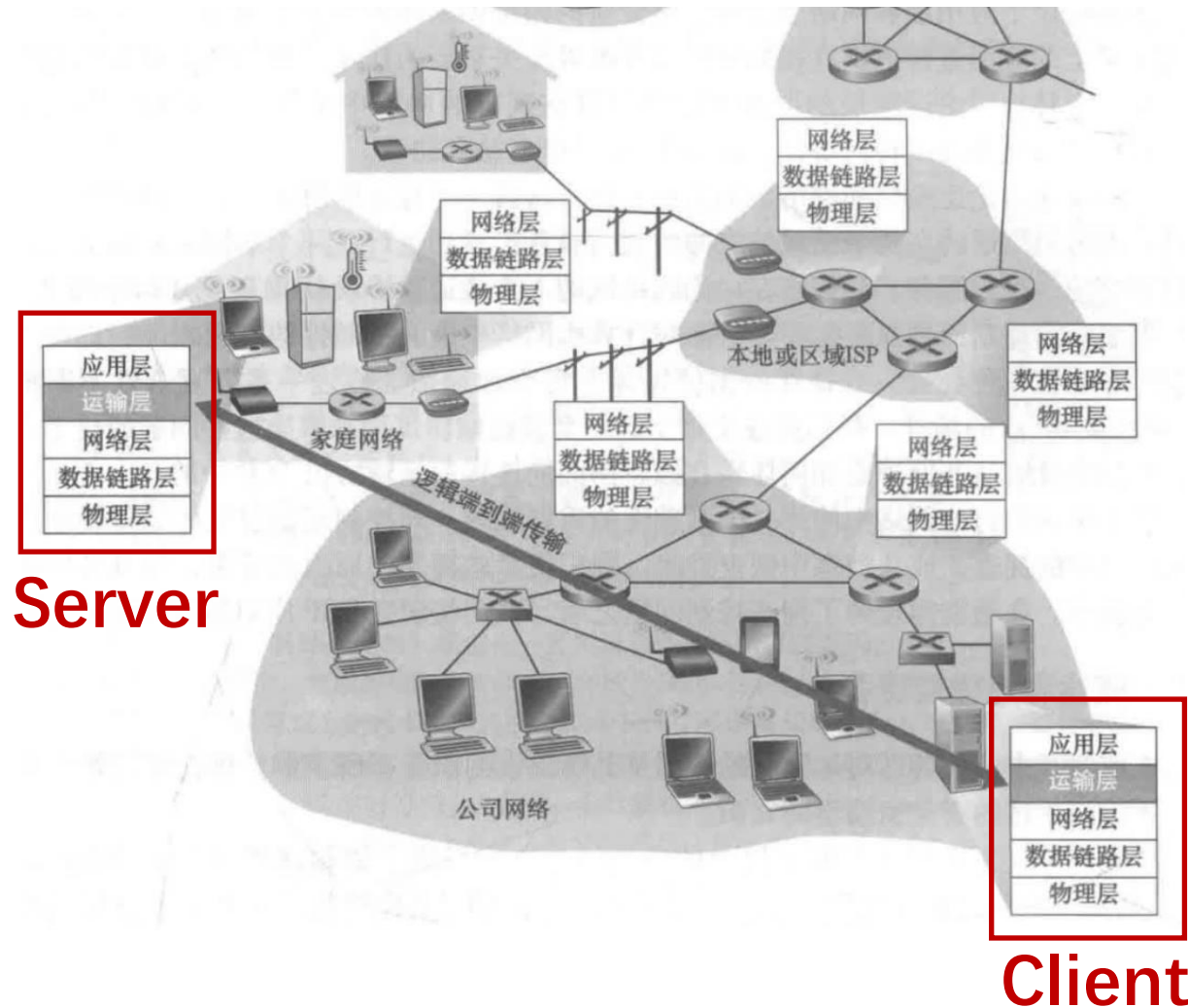
五层网络模型中的数据流实例

主机 1 向主机 2 发送数据



运输层协议 (Transport Layer)

- 概述: 为运行在不同主机的应用进程提供了**逻辑通信** (logic Communication)
- 仅存在于端系统, 而不在路由器中。因此, 路由器不检查运输层报文段。
- 借助运输层提供的逻辑通信功能, 不同主机(Server & Client)上的应用进程彼此发送报文, 而无需考虑底层的物理基础设置细节



运输层协议 TCP & UDP

- TCP (Transmission Control Protocol: “传输控制协议”) :
 - ✓[进程间数据交付] 数据可靠传输：面向连接的的报文段(segment)发送
 - ✓[差错检查]
 - ✓[拥塞控制 (congestion control)]: 调节TCP连接的发送端速率，避免拥塞
- UDP (User Datagram Protocol: “用户数据报协议”):
 - [进程间数据交付] 不可靠传输：无连接的报文段(segment)发送
 - [差错检查]
 - 流量不可调节，可根据需要以任意速率发送数据

TCP vs UDP

指标	TCP	UDP
是否连接	面向连接	无连接
传输可靠性	可靠	不可靠
速度	较慢	较快
传输质量	较高	较差

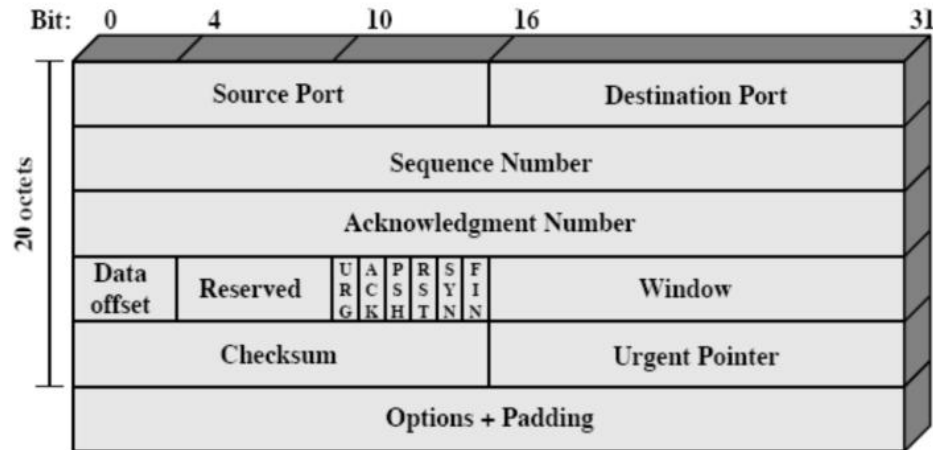
TCP vs UDP

应用	应用层协议	下面的运输协议
电子邮件	SMTP	TCP
远程终端访问	Telnet	TCP
Web	HTTP	TCP
文件传输	FTP	TCP
远程文件服务器	NFS	通常 UDP
流式多媒体	通常专用	UDP 或 TCP
因特网电话	通常专用	UDP 或 TCP
网络管理	SNMP	通常 UDP
名字转换	DNS	通常 UDP

图 3-6 流行的因特网应用及其下面的运输协议

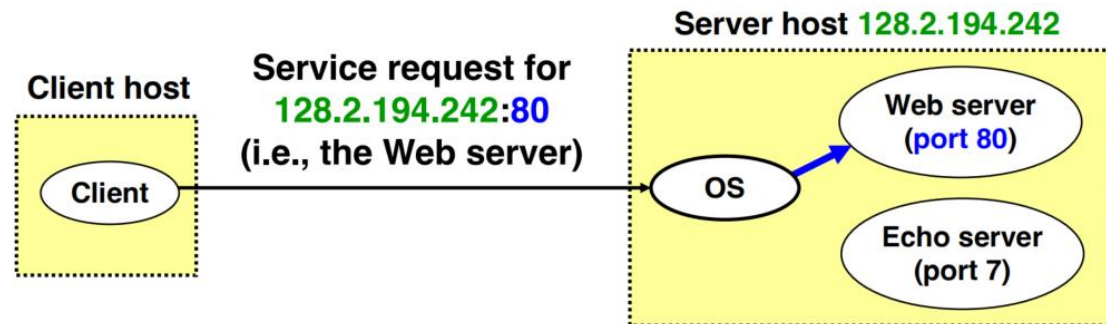
TCP (Transmission Control Protocol)

- 多路复用 (multiplexing) : socket → 运输层报文段
 ➤ 需要封装TCP首部信息



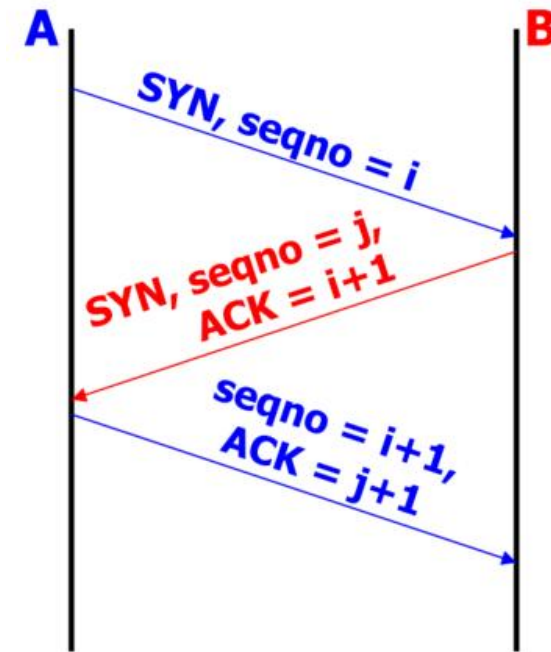
传输层头部

- 多路分解 (Demultiplexing) : 运输层报文段 → socket



TCP (Transmission Control Protocol)

- 可靠数据传输 (Reliable data transfer)
 - 确保接收到无损坏、无间隙、非冗余、按序的数据流，与输出完全相同
 - TCP是可靠数据传输协议，但下层IP网络层协议是不可靠数据传输协议。
- TCP 基本功能：
 - 面向连接：三次握手（发送预备报文段）



TCP (Transmission Control Protocol)

- 可靠数据传输 (Reliable data transfer)
 - 确保接收到无损坏、无间隙、非冗余、按序的数据流，与输出完全相同
 - TCP是可靠数据传输协议，但下层IP网络层协议是不可靠数据传输协议。
- TCP 基本功能：
 - 面向连接：三次握手（发送预备报文段）
 - 可靠数据传输：
 - 基于序号、确认号字段
 - 超时重传

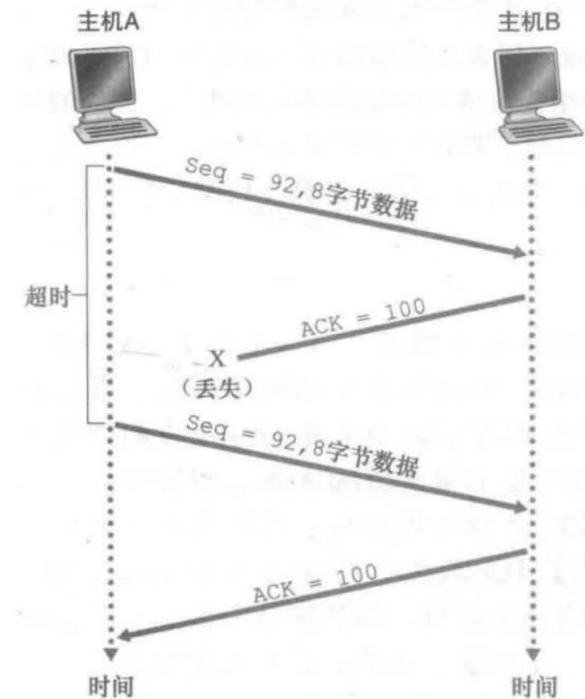


图 3-34 由于确认丢失而重传

TCP (Transmission Control Protocol)

- 可靠数据传输 (Reliable data transfer)
 - 确保接收到无损坏、无间隙、非冗余、按序的数据流，与输出完全相同
 - TCP是可靠数据传输协议，但下层IP网络层协议是不可靠数据传输协议。
- TCP 基本功能：
 - 面向连接：三次握手（发送预备报文段）
 - 可靠数据传输：
 - 基于序号、确认号字段
 - 超时重传
 - 流量控制：匹配接收方速度，避免接受缓存溢出
 - 接受窗口：剩余缓存空间

$$rwnd = RcvBuffer - [LastByteRcvd - LastByteRead]$$

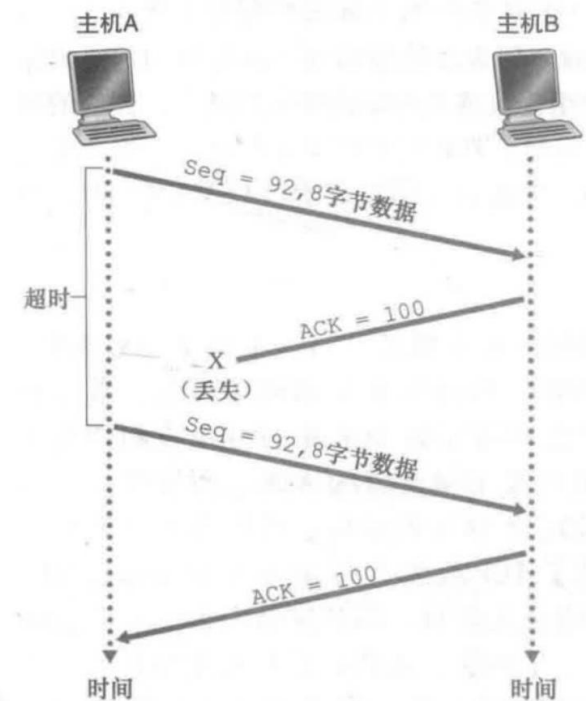


图 3-34 由于确认丢失而重传

拥塞控制 (congestion control)

- 网络拥塞 (congestion):
 - 太多数据源希望以高速率发送数据
 - 造成丢包、超时、过高的排队时延
 - 分组重传机制可以应对丢包，但无法缓解拥塞
- 拥塞控制 (congestion control):
 - ✓ 端到端拥塞控制：端系统自行观测网络行为进行阻塞监测（分组丢失、时延等）
 - ✓ 网络辅助的拥塞控制：路由器向发送方提供网络拥塞状态的显示反馈

TCP拥塞控制

- Basic Idea:
 - ✓ 每一个发送方根据感知网络拥塞程度 (How?)
 - ✓ 限制发送数据的速率 (How?)
- 发生丢包事件： 超时 或 三个冗余ACK
- 拥塞窗口 (congestion window) : cwnd 【动态变化， 取决于拥塞程度】

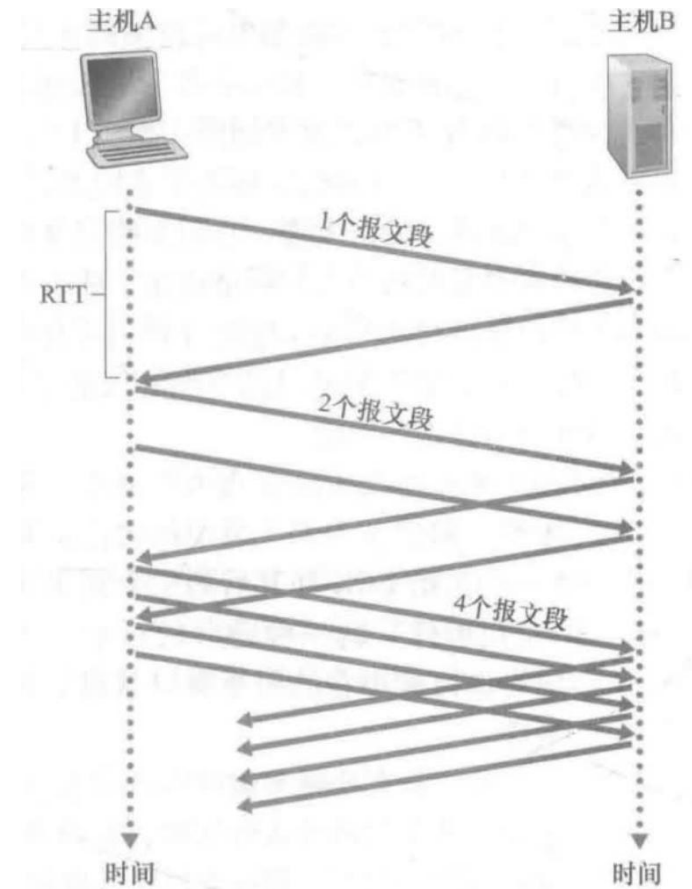
$$\text{LastByteSent} - \text{LastByteAcked} \leq \min \{ \text{cwnd}, \text{rwnd} \}$$

TCP拥塞控制

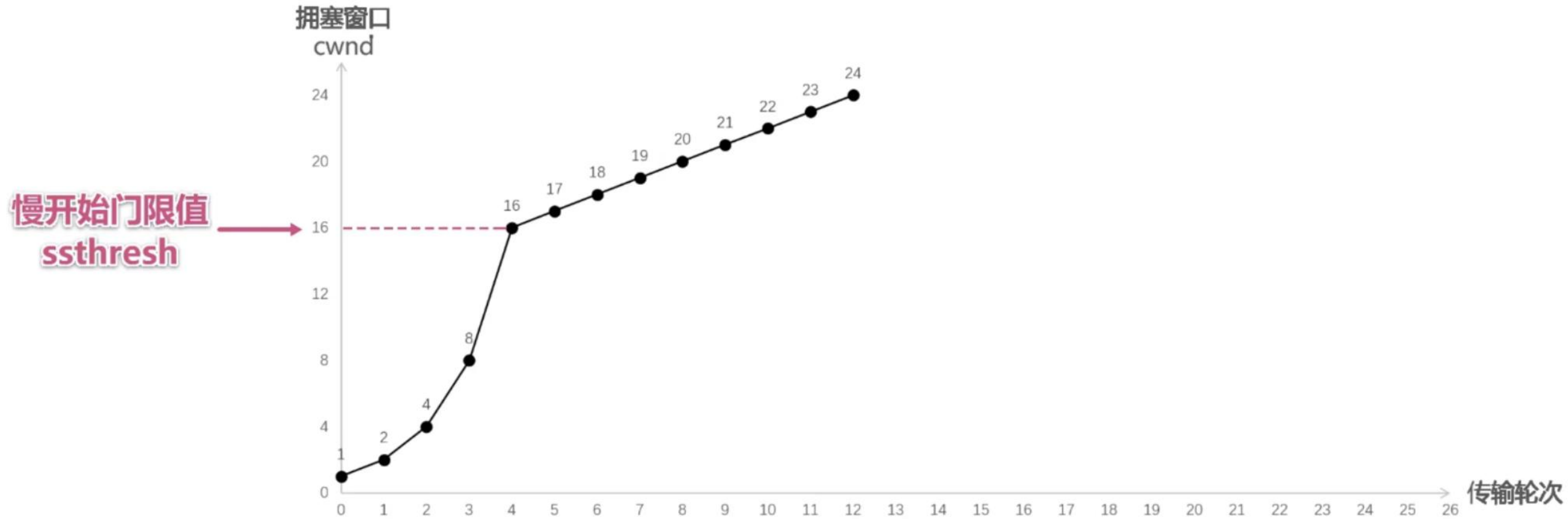
- 原则：
 - ✓ 拥塞(丢包事件)： 缩小拥塞窗口 $cwnd$
 - ✓ 无拥塞： 逐渐增大拥塞窗口 $cwnd$

TCP拥塞控制算法：

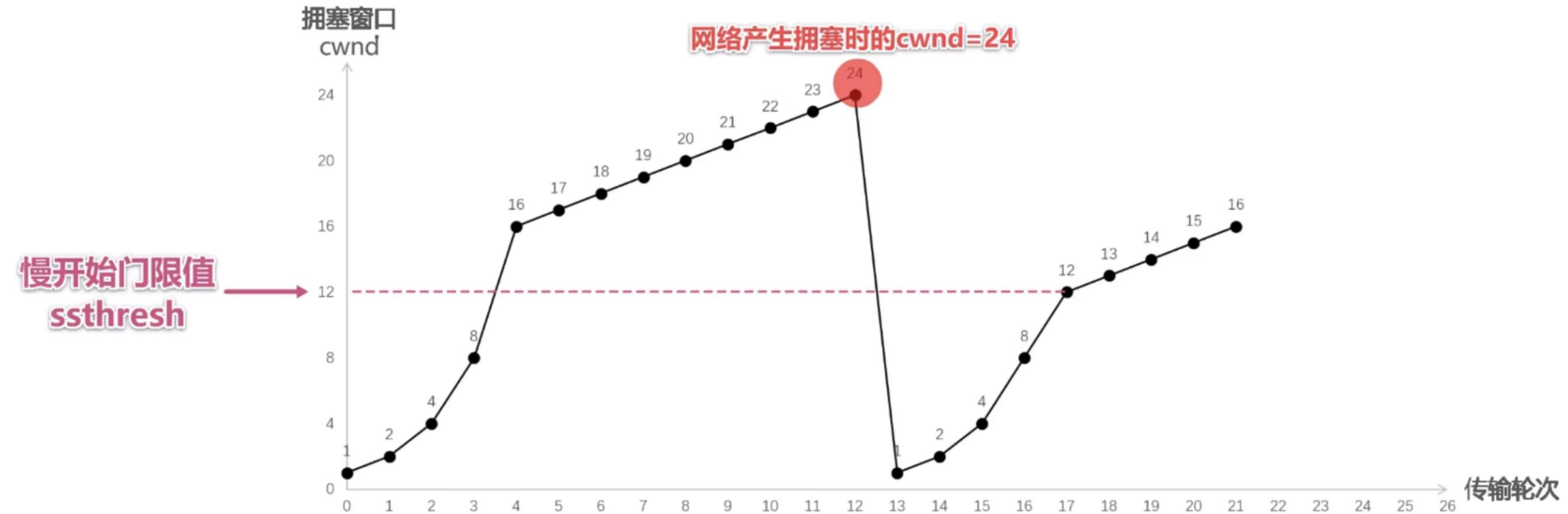
- ✓ 慢启动(Slow-start)
- ✓ 拥塞避免
- ✓ 快恢复



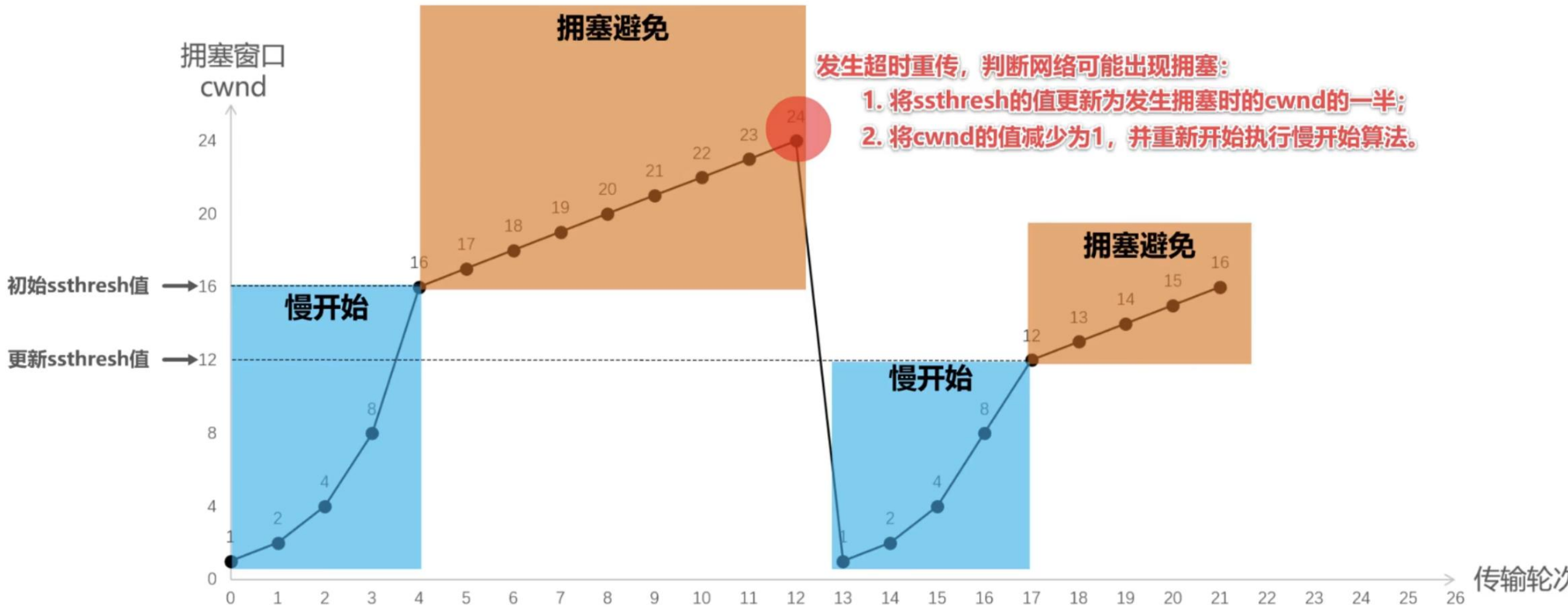
TCP拥塞控制-案例



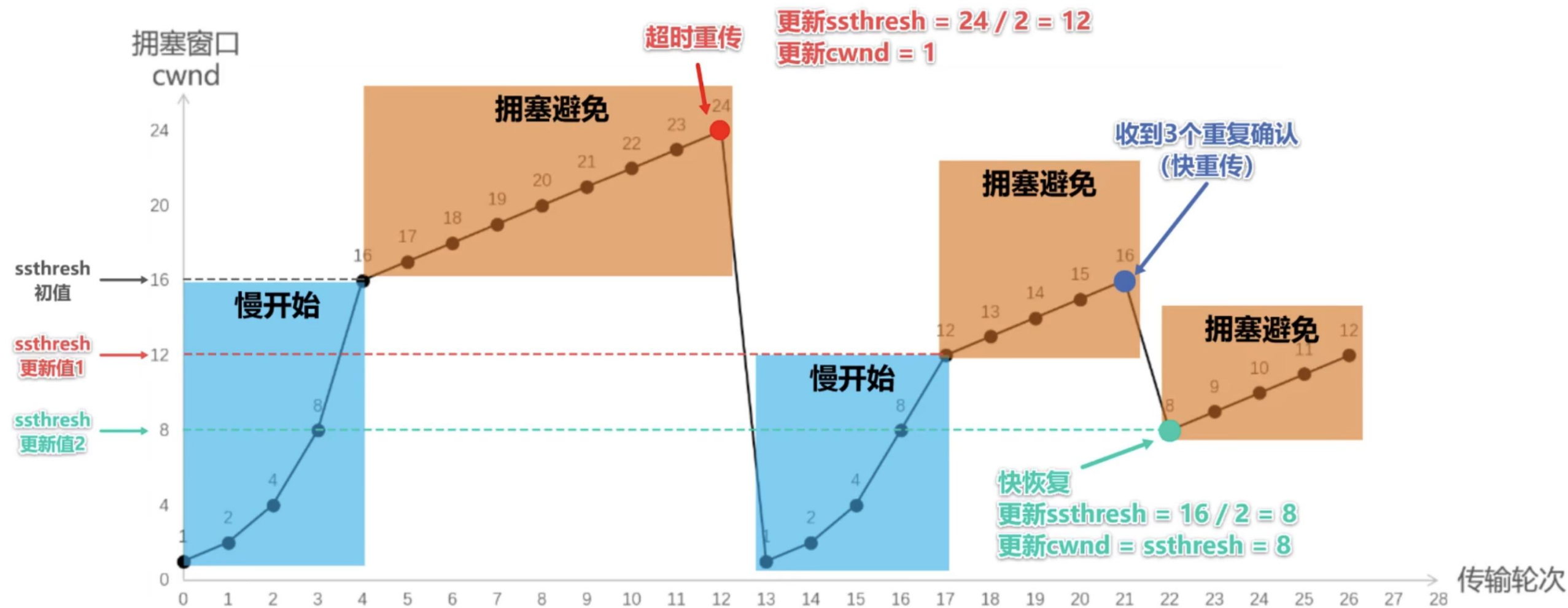
TCP拥塞控制-案例



TCP拥塞控制-案例

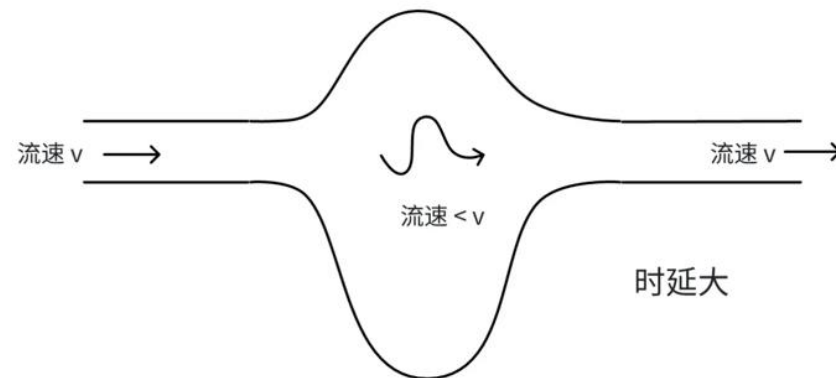


TCP拥塞控制 - 快恢复



缓冲膨胀 (BufferBloat)

- 概述: **缓冲膨胀**: 当交换设备配置为使用过大的缓冲区(Buffer)时, 这反过来会导致高延迟和数据包 延迟变化 (抖动)
- 【拥塞控制】 TCP 发送方以越来越快的速度发送, 直到看到丢失的数据包为止
- 但如果缓冲区很大, 发送者就无法看到丢失的数据包, 直到缓冲区被填满为止, 发送速率已经远远超过了网络的容量。



作业验收及提交

- 实践课现场验收（11.6），报告提交（11.9）
- 发送TA邮箱：
 - 点名册1-10：杨嘉莉 (10234804407@stu.ecnu.edu.cn)
 - 点名册11-23：刘蔚美 (51285903074@stu.ecnu.edu.cn)
- 邮件主题：CN-ASS2-学号-姓名
- 邮件附件：CN-ASS2-学号-姓名.zip
- 提交内容：
 - ✓.ipynb文件
 - ✓实验报告：需要包括关键代码解析、实验结果截图与分析

严禁抄袭!!!