- 第5章 传输层
- ..提供的服务
  - 功能(1通信 2用
  - **传输层' 寻址与端口**(2分类 1socket
  - **无连接服务**(UDP **面向..**
-
- **UDP协议**
  - 数据**报**(3无1小 4format
  - 检验(1首部 1检验:构补 划 和反
- TCP**协议**
  - ..特点(2面向 1传输 1通信
  - **报文段**(211含义:2号 1位 1wnd
  -
  - 连接**管理**(2过程: 三握link 四挥release 3reason:why 3/4? TIME_WAIT?
  - **可靠传输**3机制:序 确 重 2重传:超时,快速(冗余ACK
  -
  - 流量**控制**(2wnd: 滑动.. , rwnd..
  - **拥塞..**(4算法:慢 避 快*2 ; 2变化:cwnd ssthresh 1count:TCP拥塞(3+2
- 本章小结及疑难点

好的，同学们，我们来到了第五章《传输层》！ 这一章是整个协议栈的**中枢**，直接为应用层服务，重要性不言而喻。**TCP协议是本章的绝对核心**，也是整个计网课程的**灵魂**所在。它的连接管理（三次握手、四次挥手）、可靠传输、流量控制和拥塞控制机制，是每年考研大题的**常客和主角**。
这一章必须做到滴水不漏。跟上我的思路，我们用最精炼的方式，彻底征服TCP/UDP！

第5章 传输层

本章定位：如果说网络层是“跨国物流”，负责把包裹从**主机A**送到**主机B**的大门口，那传输层就是"**前台/收发室**"，负责把包裹从主机B的大门口，精准地送到**某个具体的部门/人（进程）**手中。它实现了**端到端（进程到进程）**的逻辑通信。

..提供的服务

功能(1通信 2用

- **核心考点**:
  - 提供**进程到进程 (Process-to-Process)** 的逻辑通信。
  - **复用 (Multiplexing)** 与 **分用 (Demultiplexing)**。
  - 对报文进行差错检测。

- **形象记忆**:
  - **进程到进程通信**: 网络层把信送到你家（主机），传输层负责把信交到**本人**(进程)手上，而不是你爸妈手上。
  - **复用/分用 (收发快递)**:
    - **复用 (发送时)**: 你电脑上同时开着微信、浏览器、游戏（多个进程），它们的数据都打包交给传输层这**一个**快递员发出去。
    - **分用 (接收时)**: 传输层这个快递员收到一堆包裹，根据包裹上的“收件人部门”（端口号），准确地分发给微信、浏览器、游戏。

传输层' 寻址与端口(2分类 1socket

- **核心考点**:
  - 端口号的作用。
  - 套接字(Socket)的定义。
- **形象记忆**:
  - **IP地址 vs. 端口号**:
    - `IP地址`: 一栋大楼的**街道地址**（定位到主机）。
    - `端口号`: 大楼里每个房间的**门牌号**（定位到进程）。
  - **端口号分类**:
    - **熟知**端口 (0-1023): 饭店、银行等**公共服务机构**的门牌号，众所周知 (如HTTP-80, FTP-21)。
    - **客户端**端口 (短暂端口): 你去办事时，银行给你的**临时排队号**，办完就失效了。

- **套接字 (Socket)**:
  - **定义**: `Socket = (IP地址 : 端口号)`
  - **作用**: 全网 唯一 的通信标识符，就像完整的收件地址：“**某市某街道100号大楼的808房间**”。

无连接服务(UDP 面向..

- **核心考点**: UDP和TCP的核心区别与适用场景。
- **形象记忆**:
  - **UDP (用户数据报协议) - 无连接服务**:
    - **比喻**: 寄**平信**。
    - **特点**: 简单、快、开销小。不用先打电话确认对方在不在家。直接把信扔邮筒里就完事了。信可能会丢，也可能先发的后到，它都不管。**不可靠，尽力而为**。
    - **适用**: **实时**性要求高、能容忍少量丢包的场景（视频直播、语音通话、DNS查询）。

  - **TCP (传输控制协议) - 面向连接服务**:
    - **比喻**: 打**电话**。
    - **特点**: 必须先“喂，听得到吗？”（**建立连接**），然后才能开始有序、可靠地对话，最后还要“再见”（**释放连接**）。开销大，但**可靠**。
    - **适用**: 可靠性要求高的场景（文件传输FTP、网页浏览HTTP）。

UDP协议

UDP:User Datagram Protocol,用户数据报协议

数据报(3无1小 4format

- **核心考点**: UDP的特点、首部格式。
- **形象记忆**:

- **特点**: “四无产品” -> **无连接、无拥塞控制、无连接状态、首部开销小**。
- **首部格式 (8字节)**:
  - `[源端口(2B)] [目的端口(2B)] [长度(2B)] [检验和(2B)]`
  - **记忆**: 一个极简的信封，只写了“**你的门牌号**”、“**我的门牌号**”，标了一下**信**有多**重**，再盖个“**检验邮戳**”。非常高效。

检验(1首部 1检验:构补 划 和反

- **核心考点**: UDP检验和的计算范围包含一个“**伪首部**”。
- **形象记忆**:
  - **伪首部**: 包含**源IP、目的IP、协议号**等信息。
  - **作用**: 就像邮递员在盖“检验邮戳”时，不仅要确认信封没破（UDP报文），还要再次核对信封上的 街道地址（**IP**地址）是不是真的送对了楼，防止IP层送错地方而传输层不知道。

- **【计算题模板】: UDP检验和计算**
  i. **构造**: 在UDP报文前添加一个12字节的**伪首部**。
  ii. **补齐**: 如果UDP报文（包含伪首部）总字节数是奇数，在末尾**补一个全0字节**（仅为计算，不发送）。

---

  iii. **划段**: 将整个数据（伪首部+UDP报文+可能有的填充字节）按16位（2字节）划分为多个段。

---

  iv. **求和**: 对所有16位段进行**反码算术运算求和**。（高位溢出加到低位）
  v. **取反**: 将最终的和**按位取反**，得到检验和。

TCP协议

TCP:Transport Control Protocol,传输控制协议

这是 **【本章的绝对核心，每年必考的“巨无霸”】**。

..特点(2面向 1传输 1通信

- **核心考点**: `面向连接` 、 `可靠传输` 、 `全双工通信` 、 `面向字节流` 。
- **形象记忆**:
  - **面向字节流**: TCP像一条**水管**，应用层把数据（水）倒进去，TCP不关心你分几次倒、每次倒多少，它只保证水管另一头流出来的水和你倒进去的**总量、顺序**完全一样。

报文段(211含义:2号 1位 1wnd

- **核心考点**: TCP首部各字段含义，特别是**序号、确认号、6个控制位、窗口大小**。
- **形象记忆 (打电话)**:
  - **序号 (seq)**: “我现在说的这句话，是我整段话里的第 `seq` 个字。”
  - **确认号 (ack)**: “你刚才说的话，我已经完整听到第 `ack-1` 个字了，你下次请从第 `ack` 个字开始说。”

  - **6个控制位 (标志位)**: 电话中的“暗语”。
    - `SYN` : “喂，我想跟你**建立连接**。”
    - `ACK` : “我**收到**了/同意。”
    - `FIN` : “我说完了，想**挂电话**了。”

    - `RST` : “出大问题了，**强制挂断**！”
    - `PSH` : “别攒着了，赶快把数据**推送**给应用程序！”
    - `URG` : “十万火急，这份是**紧急**数据！”

- **窗口大小 (rwnd)**: “我的耳朵（接收缓存）还能听 `rwnd` 个字，你看着说。”

🔢 连接管理(2过程: 三握link 四挥release 3reason:why 3/4? TIME_WAIT?

这是 **【面试、笔试的明星考点，必须倒背如流】**。

- **【模板】三次握手 (建立连接)**
  i. **客户端 -> 服务器**: `SYN=1，seq=x`
    - **人话**: “服务器你好，我想和你通话，我的开场白编号是x。”
  ii. **.. <- ..**: `SYN=1，ACK=1，seq=y，ack=x+1`
    - **人话**: “客户端你好，同意通话。我的开场白编号是y，我确认收到了你的x号，期待你的x+1号。”
  iii. **.. -> ..**: `ACK=1，seq=x+1，ack=y+1`
    - **人话**: “服务器我收到。这是我的x+1号，我确认收到了你的y号，期待你的y+1号。”

- **为何要三次?**: 主要**防止**已失效的连接请求突然传到服务器，导致服务器错误开辟资源。

- **【模板】四次挥手 (释放..)**
  i. **A -> B**: `FIN=1，seq=u`
    - **人话**: “B，我这边说完了，请求关闭（我到你）的单向通道。” (A进入FIN_WAIT_1)
  ii. **.. <- ..**: `ACK=1，ack=u+1`
    - **人话**: “好的，收到你的关闭请求。但我可能还有话没说完，你稍等。” (B进入CLOSE_WAIT，A进入FIN_WAIT_2，此时为**半关闭**状态)

---

  iii. **.. <- ..**: `FIN=1，seq=w`

- **人话**: “A，我这边也说完了，请求关闭（我到你）的单向通道。” (B进入LAST_ACK)

iv. **.. -> ..**: `ACK=1，ack=w+1`

- **人话**: “收到，再见。” (A进入**TIME_WAIT**，等待2MSL后才彻底关闭)

- **为何要四次?**: 因为TCP是**全双工**，一方关闭发送后，另一方可能还有数据要发，所以FIN和ACK需要分开发送。
- **..TIME_WAIT?**: 确保 最后一次**ACK**能成功被对方接收。如果ACK丢失，对方会重传FIN，自己还能响应。

可靠传输3机制:序 确 重 2重传:超时,快速(冗余ACK

- **核心考点**: 可靠传输的实现机制：`序号`、`确认`、`重传`。
- **重传机制**:
  - **超时重传**: 发送一个包后启动计时器，超时未收到确认则重传。
  - **快速..**: 发送方连续收到**3个**对于同一个包的**冗余ACK**（重复确认），不等计时器超时，立即重传该包。

流量控制(2wnd: 滑动.. , rwnd..

- # 核心考点: 利用滑动窗口和接收方通告的 接收窗口(rwnd) 进行流量控制。

- **形象记忆**:
  - **场景**: 你在给朋友灌输知识。
  - **rwnd**: 朋友（接收方）通过表情告诉你“我的大脑（接收缓存）还能接收 `rwnd` 个知识点”。
  - **滑动窗口**: 你（发送方）根据朋友的表情（`rwnd`值）调整自己灌输知识的速度。如果朋友眉头紧锁(`rwnd`变小)，你就说慢点；如果朋友豁然开朗(`rwnd`变大)，你就可以说快点。
  - **目的**: 防止把朋友**说懵**（防止撑爆接收方缓存）。

🧮 拥塞..(4算法:慢 避 快*2 ; 2变化:cwnd ssthresh 1count:TCP拥塞(3+2

- **核心考点**: `慢开始`、`拥塞避免`、`快重传`、`快恢复`四个算法，以及**拥塞窗口(cwnd)** 和 **慢开始门限(ssthresh)的变化。【计算分析天王山】**
- **形象记忆 (开车上高速)**:
  - `cwnd`: 你当前的车速。
  - `ssthresh`: 你认为前方路况能承受的安全巡航速度。
  - **发送窗口**:
    `实际车速 = min(你当前想开多快, 朋友让你开多快)` 即
    `min(cwnd, rwnd)`。

- **【计算/分析模板】: TCP拥塞控制算法**
  i. **初始阶段 (上匝道)**:
    - `cwnd = 1 MSS`, `ssthresh` 设为较大值。
    - 进入 **慢开始** 状态。
  ii. **慢开始** (Slow Start, cwnd < ssthresh):
    - **规则**: 每经过一个RTT， `cwnd` **翻倍** (指数增长)。
    - **行为**: 在匝道上**猛踩油门**，快速提速。
  iii. **拥塞避免** (Congestion Avoidance, cwnd >= ssthresh):
    - **规则**: 每经过一个RTT， `cwnd` **加 1** (线性增长)。
    - **行为**: 上了高速，达到安全速度后，**平稳地、慢慢地**加速。

- **遇到拥塞事件**:
  - 事件1: 超时 (Timeout) - **严重**堵车
    - **动作**:
      - `ssthresh = cwnd / 2` (降低心理预期)
      - cwnd = 1 MSS (**一脚刹车踩到底**)
      - **返回慢开始阶段** (重新从匝道上高速)。
  - 事件2: 收到3个冗余ACK - **轻微**拥堵 (触发**快重传/快恢复**)

    - **动作**:
        - `ssthresh = cwnd / 2` (降低心理预期)
        - cwnd = ssthresh (**松油门，减速到安全速度**)
        - **直接进入拥塞避免阶段** (继续在高速上平稳加速)。

本章小结及疑难点

- **TCP vs GBN/SR**: TCP是GBN和SR的混合体。它像GBN一样使用**累积确认**，但又像SR一样可以**缓存失序分组**并发起**快速重传**，而不是重传所有后续分组。
- **MSS (最大报文段长度)**: 是**数据**部分的最大长度，是为了避免在IP层被分片。
- **可靠性并非多余**: 即使链路不出错，路由器也可能因拥塞丢弃分组，分组也可能失序到达，所以TCP的可靠交付功能必不可少。

第五章的内容，TCP是重中之重。**三次握手、四次挥手**的图一定要会画，`seq`和`ack`的变化要了然于心。**拥塞控制**的四个算法以及`cwnd`和`ssthresh`的变化曲线图是分析大题的关键，必须熟练掌握。把这些核心拿下，传输层就不在话下！