

运输层

概要

- 功能
 - 提供进程之间的逻辑通信
 - 实现复用与分用
 - 差错检测
- 术语
 - 运输层服务访问点 ——— TSAP
 - 网络层服务访问点 ——— NSAP
- 端口Port
 - 复用和分用的基础
 - 16位，仅有本地意义
 - 分类
 - 熟知端口 0-1023
 - 登记端口 1024-49151
 - 客户使用端口 49152-65535
 - 通过报文队列实现

TPDU : Transport Protocol Data Unit

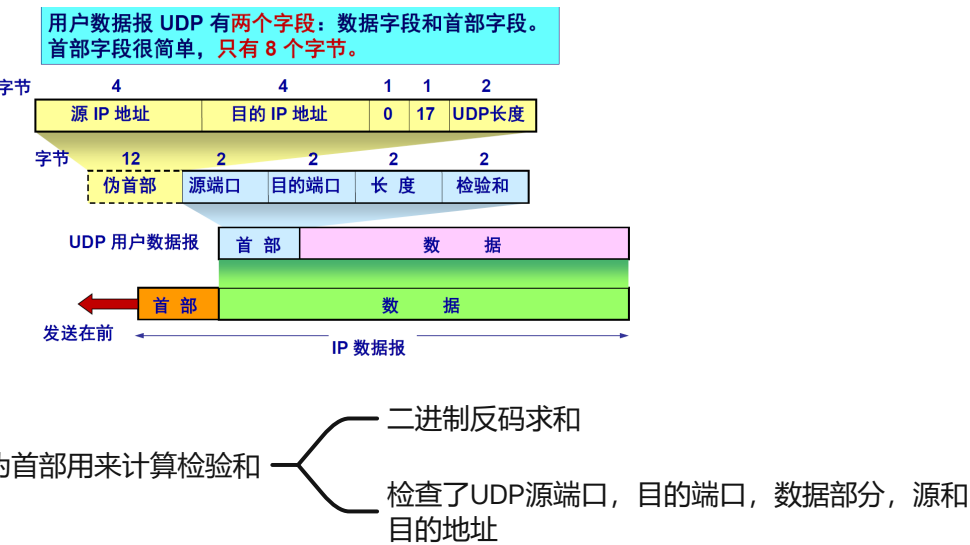
- TCP segment
- UDP datagram

UDP (User Datagram Protocol)

特点

- 无连接协议
- 不可靠交付
- 面向报文 ——— 不拆分
- 不需要接收方给出确认
- 有差错检测
- 首部开销小 8 Bytes

报文格式



协议

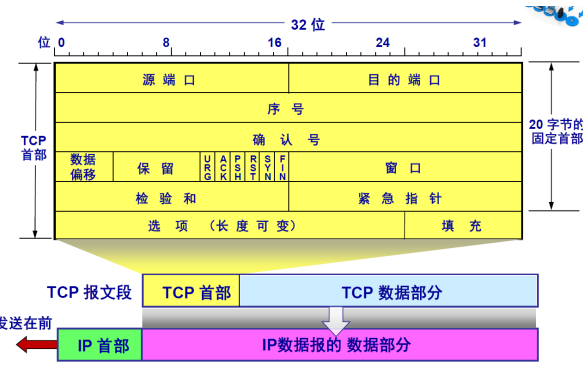
特点

- 面向连接 ——— 虚连接
- 可靠交付
- 面向字节流
- 需要确认，流量控制，计时器，连接管理等技术
- 只能点对点
- 全双工通信
- 端点为套接字 ——— 端口号+IP地址

可靠传输原理

- Stop and Wait
- ARQ (Automatic Repeat reQuest)
 - GO BACK N
 - Selective Repeat
- 滑动窗口
 - 以字节为单位
 - TCP接收方必须用累计确认机制

TCP报文格式



TCP的连接管理

- 三次握手
- 四次挥手 ——— 等待2MSL时间

超时重传的时间选择

- 自适应算法
 - RTTs
 - RTTd
 - RTO: Retransmission Time-Out
 - Karn 算法 ——— 重传一次九八RTO增大2倍

选择ACK

流量控制

- 实现 ——— 控制发送方的流量
- 具体
 - 利用滑动窗口机制
 - 持续计时器
 - 零窗口通知
 - 零窗口探测报文段 (携带 1 Byte数据)
 - 发送方糊涂 ——— Nagle算法
 - 接收方糊涂 ——— 等待到接收窗口足够大

拥塞控制

- 网络的拥塞 ——— 防止过多的数据注入到网络
- 思路
 - 开环控制
 - 闭环控制
 - 基于窗口的方法
 - 发送方维持一个拥塞窗口cwnd
 - Min (拥塞窗口，接收窗口)
 - 拥塞的判断
 - 重传定时器超时
 - 收到三个重复ACK
- 拥塞控制方法
 - 慢开始 ——— 确定网络的负载能力
 - 由小到大逐渐增大拥塞窗口数值
 - cwnd设置
 - ssthresh设置
 - 每收到一个确认cwnd就+1
 - 传输轮次产生指数增加 ——— 轮次时间为RTT
 - 拥塞避免
 - cwnd>ssthresh ——— 变为线性增加
 - 出现定时器超时
 - ssthresh=max(cwnd/2, 2)
 - cwnd=1
 - 执行慢开始算法
 - 快重传 ——— 重复收到三个ACK ——— 执行快恢复算法
 - 快恢复
 - ssthresh=cwnd/2
 - cwnd=ssthresh
 - 执行拥塞避免算法