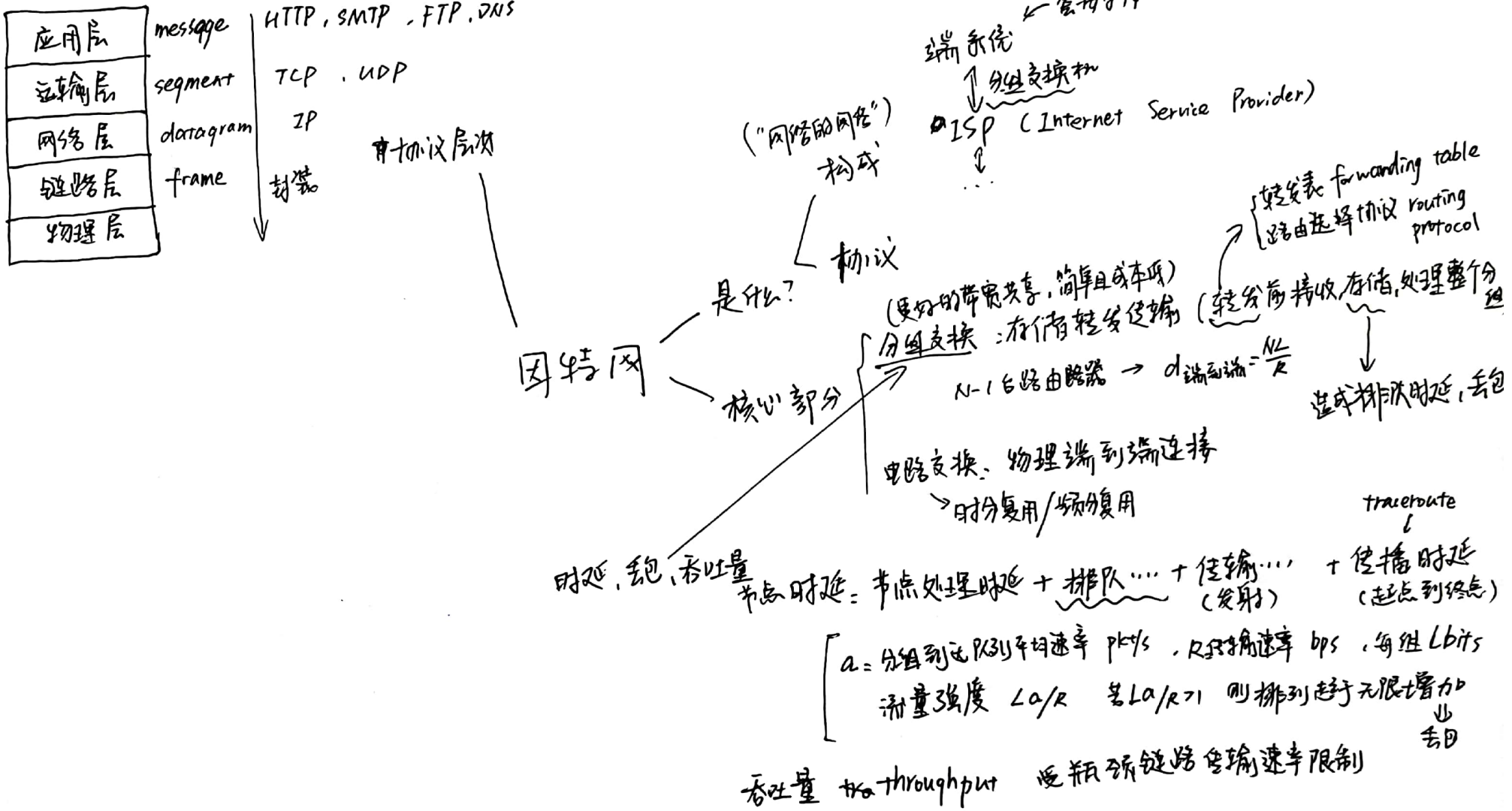
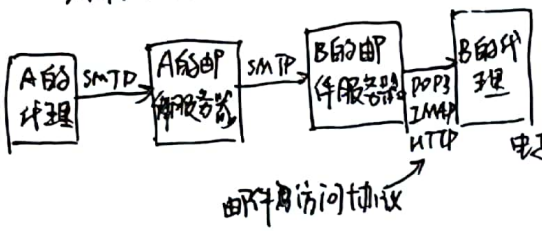


SMTP (Simple Mail Transfer Protocol), 基于TCP

报文中包含 HELO; MAIL FROM; RCPT TO; DATA; QUIT (P78)

SMTP发推协议 (而HTTP发拉协议)



电子邮件

定义不同端系统上的应用程序进程如何相互传递报文

↑

应用层协议原理

- 应用程序体系结构
  - 客户-服务器 (client-server architecture) → Web, FTP, Telnet, email
  - P2P → BitTorrent
- 进程间通信: 套接字 连接进程与计算机网络
  - IP + Port 寻址
- 应用层根据需求选择运输层协议: 可靠, 吞吐量, 定时, 安全
  - TCP: 连接, 可靠, 拥塞控制
  - UDP: 无连接, 不可靠

应用层

DNS (Domain Name System)

实现主机名到IP地址的转换

解析层次数据库: 根DNS / 顶级域DNS (TLD) / 权威DNS

类型, 报文 P89-90

基于UDP

Web 和 HTTP

(HyperText Transfer Protocol)

运行在TCP上, 非持续连接/持续连接

HTTP 请求报文: GET /somedir/page.html HTTP/1.1

```
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

响应:

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 18 Aug 2015 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 18 Aug 2015 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
(data.....)
```

报文包含 Cookie: 实现站点对用户行为的跟踪

Web 缓存 + 条件 GET (if modified)

TCP连接：三次握手：① SYN，初始化报文

② SYNACK，分配资源，初始化报文

③ SYN=0，即分配资源

TCP

提供进程间通信

(源)多路复用 ↓ ↑ (目的)多路分解 ⇐ Segment中包含源、目的端口

对比网络层，主机间通信

UDP：实现复用/分解，少量差错检测（校验和），不可靠，无连接，无拥塞控制。

（DNS → 避免TCP的连接创建时延）

运输层

可靠数据传输原理：

ARQ (Automatic Repeat reQuest)：

差错检测 + 接收方反馈（发送方修复） + 重传

问题：Ack/NAK可能受损 → 引入冗余分组 → 无法判断是新分组还是重传分组 → 添加序号字段 (0,1)...

丢包问题 → 引入倒计时器定时器，时间内未收到ACK则重传

信道号被信道利用率低 → 流水线技术 ⇐ 扩展

差错恢复

回退N步：滑动窗口 + 累积确认（接收不缓存）

选择重传：滑动窗口 + 累积确认 + 接收缓存（选择重传）

TCP报文段：首部，确认号（累积确认）

估计往返时间 EstimatedRTT = (1−α)·EstimatedRTT + α·SampleRTT

DevRTT = (1−β)·DevRTT + β·|SampleRTT − EstimatedRTT|

设置时间间隔 TimeoutInterval = EstimatedRTT + 4·DevRTT

超时间隔加倍 → 拥塞控制

3个冗余ACK ⇒ 快速重传

流量控制：使发送速率与接收速率匹配

⇒ 发送方维护一个接收窗口变量 rwnd，指明接收方缓存大小

拥塞控制：因IP网络拥塞而遏制发送速率

维护一个拥塞窗口变量 cwnd：LastByteSent − LastByteAcked ≤ min{cwnd, rwnd}

① 慢启动：每有1个ACK，cwnd就加1个MSS（2次方指数增加），直到超时 或 cwnd≥ssthresh 或 3个冗余ACK

cwnd=1, ssthresh=cwnd/2

② 拥塞控制：每一个RTT，cwnd加1个MSS（线性增加），直到超时 或 3个冗余ACK

③ 快速恢复：每个冗余ACK，cwnd加1个MSS，超时 或 新ACK

ssthresh=cwnd/2 cwnd=ssthread+3·MSS

网络层

控制平面：路由选择

控制平面：路由选择 ← 每路由器控制 / 逻辑集中式控制、SDN

路由选择算法

- 集中式：Dijkstra 最短路径
- 分散式：DV算法，Bellman-Ford方程

路由器分组组织成自治系统 AS (Autonomous System)

- AS内路由选择：OSPF 开放最短路优先
- 跨AS路由选择：BGP (Broder Gateway Protocol) P356
  - iBGP，eBGP对话 → 传播可达性信息
  - 热土豆协议 → 尽快送出某AS

SDN

网络控制应用程序接口
网络范围状态管理层
通信层

ICMP

ICMP：用于主机、路由器彼此沟通网络层信息

差错报告 → Traceroute程序原理

SNMP

SNMP (Simple Network Management Protocol)

用于在管理服务器和代表服务管理服务器之间传递网络管理控制和信息报文。

数据平面：转发

路由器

输入端口 ⇒ 交换结构 ⇒ 输出端口

- 输入端口：(协议处理，查找，转发，排队)
  - IP最长前缀匹配 forward table
- 交换结构：{内存 / 总线 / 互联网络}
- 输出端口：(排队，协议处理)
  - 分组调度：{先进先出 / 优先权排队 / 循环排队}

IPv4数据报 (P314)

IPv4编址：IP地址对应一个接口；4字节 → 点分十进制记法

地址分配策略 CIDR Classless InterDomain Routing：IP地址 { a.b.c.d/x：子网掩码 + 32-x 比特：区分子网内设备

设备自动获取IP地址：DHCP (Dynamic Host Configuration)

① DHCP服务器发现：客户向255.255.255.255广播，本机IP源为0.0.0.0

② DHCP服务器提供：服务器广播，提供IP地址，网络掩码，IP地址租用期等

③ DHCP请求：客户选择一个服务器提供，发送请求

④ DHCP ACK：响应，证实参数

问题：子网变大，地址不够 → 使用专用网络 + NAT (Network Address Translation)

整个子网视为1个主机，IP地址已变为NAT路由器对外接口的IP地址

NAT路由器实现WAN端和LAN端地址转换

IPv6

IPv6：① 地址容量扩大 32bit → 128bit ② 首部简化：③ 分片/重组，首部检验和，选项字段 删去

③ 流标签

通用转发

通用转发：流表 flow table 对协议栈的多个首部字段进行匹配

MAC地址：每个适配器(网络接口)的地址；固定6字节；十六进制表示法

IP与MAC转换 ⇒ 地址解析协议 ARP (Address Resolution Protocol)
(同一子网中)　　源结ARP分组 沟通 获得IP对应MAC转移入 ARP表中

以太网：无连接：不与发面端握手
不可靠：帧未通过CRC校验时丢弃但不报错

链路层交换机：过滤(是否转发)+转发
交换机表(switch table)（地址 接口 时间）
自学习的

交换局域网

链路层局域网

虚拟局域网

提供的可能服务：
- 成帧
- 链路接入　MAC (Medium Access Control) 规定了多路访问问题
- 可靠交付
- 差错检测和纠正
  - 奇偶校验
  - 检验和 (常用于运输层)
  - 循环冗余检验 (常用于链路层)

主体部分：网络适配器 (network adapter)

多路访问链路和协议：各个发送、接收节点，需解决碰撞问题。

① 信道划分协议
- 时分多路复用(TDM)：时间 → 时间帧 → N个时隙
- 频分多路复用(FDM)：将R bps信道划分N个频段
- 码分多址(CDMA)：对每个节点分配独立编码

② 随机接入协议：~~发生~~碰撞后每个节点独立地选择随机时延
- 时隙ALOHA
- ALOHA
- 载波侦听多路访问(CSMA)：载波侦听和碰撞检测

③ 轮流协议
- 轮询协议：主节点循环轮询每个节点
- 令牌传递协议：无主节点