网络协议栈结构

TCP/IP是一个协议族的统称，并不是一个协议

TCP/IP四层模型

1. 应用层(SMTP/HTTPS/DNS/HTTP/Telnet/POP3/SNMP/FTP/NFS)
2. 传输层(TCP/UDP)
3. 网络层(IP/ICMP/ARP)
4. 网际接口层(PPP/Ethernet)

常见协议对应端口号

- SSH 22
- FTP 20 & 21
- Telnet 23
- SMTP 25
- TFTP 69
- HTTP 80
- HTTPs 443
- SNMP 161
- Ping 使用ICMP，无具体的端口号
- DNS 53

控制帧的传输 - 网际接口层

- 差错控制
    - 反馈重发
    - 计时器
    - 序号
- 流量控制

最大传输单元(MTU) - 网际接口层

- 为了提供足够快的网络响应时间，对数据帧长度进行了限制，为1500字节（有个协议是1492字节）
- 分类
    - 接口MTU
    - 路径MTU

IP协议 - 网络层

- 最核心的协议
- 提供的是**不可靠/无连接**的数据报传送服务

ARP(Address Resolution Protocol)地址解析协议 - 网络层

- 记录IP - MAC 对应关系
- ARP代理：发现自己可以通向主机的路径时会将自己代理应答，随后转发
- ARP欺骗：伪造ARP应答数据帧欺骗请求者，从而截获数据

RARP(Reverse Address Resolution Protocol) 逆向地址解析协议

- 极少使用

ICMP(Internet Control Message Protoco)控制报文协议 - 网络层

- 通信通信过程中发生各种问题时，ICMP 将问题反馈，通过这些信息，管理者可以对所发生的问题作出诊断，然后采取适当的措施去解决它。
- Ping和tracerounte是常见的基于ICMP协议的工具

UDP协议 - 传输层

- 从传输层来看，是发送方主机中的一个进程与接收方主机中的一个进程在交换数据，因此严格地讲，通信双方不是主机，而是主机中的进程。
- 特点
    - UDP 是无连接的，发送数据之前不需要建立连接(而 TCP 需要)，减少了开销和时延。
    - UDP尽最大努力交付，不保证交付可靠性。
    - UDP 是面向报文的，对于从网络层交付下来的 IP 数据报，只做很简单的封装(8 字节 UDP 报头)，首部开销小。
    - UDP 没有拥塞控制，出现网络拥塞时发送方也不会降低发送速率。这种特性对某些实时应用是很重要的，比如 IP 电话，视频会议等，它们允许拥塞时丢失一些数据，因为如果不抛弃这些数据，极可能造成时延的累积。
    - UDP 支持一对一、一对多、多对一和多对多的交互通信。

TCP协议 - 传输层

- 特点
    - TCP 提供 可靠的 数据传输服务，TCP 是 面向连接的 。应用程序在使用 TCP 通信之前，先要建立连接，这是一个类似“打电话”的过程，通信结束后还要“挂电话”。
    - TCP 连接是 点对点 的，一条 TCP 连接只能连接两个端点。
    - TCP 提供可靠传输，无差错、不丢失、不重复、按顺序。
    - TCP 提供 全双工 通信，允许通信双方任何时候都能发送数据，因为 TCP 连接的两端都设有发送缓存和接收缓存。
    - TCP 面向 字节流 。TCP 并不知道所传输的数据的含义，仅把数据看作一连串的字节序列，它也不保证接收方收到的数据块和发送方发出的数据块具有大小对应关系。
- 连接的建立与释放
    - 三次握手
        - 客户端发出请求连接报文段，其中报头控制位 SYN=1，初始序号 seq=x。客户端进入 SYN-SENT(同步已发送)状态。
        - 服务端收到请求报文段后，向客户端发送确认报文段。确认报文段的首部中 SYN=1，ACK=1，确认号是 ack=x+1，同时为自己选择一个初始序号 seq=y。服务端进入 SYN-RCVD(同步收到)状态。
        - 客户端收到服务端的确认报文段后，还要给服务端发送一个确认报文段。这个报文段中 ACK=1，确认号 ack=y+1，而自己的序号 seq=x+1。这个报文段已经可以携带数据，如果不携带数据则不消耗序号，则下一个报文段序号仍为 seq=x+1。
    - 四次挥手
        - 此时 TCP 连接两端都还处于 ESTABLISHED 状态，客户端停止发送数据，并发出一个 FIN 报文段。首部 FIN=1，序号 seq=u（u 等于客户端传输数据最后一字节的序号加 1）。客户端进入 FIN-WAIT-1(终止等待 1)状态。
        - 服务端回复确认报文段，确认号 ack=u+1，序号 seq=v（v 等于服务端传输数据最后一字节的序号加 1），服务端进入 CLOSE-WAIT(关闭等待)状态。现在 TCP 连接处于半开半闭状态，服务端如果继续发送数据，客户端依然接收。

        - 客户端收到确认报文，进入 FIN-WAIT-2 状态，服务端发送完数据后，发出 FIN 报文段，FIN=1，确认号 ack=u+1，然后进入 LAST-ACK(最后确认)状态。
        - 客户端回复确认报文段，ACK=1，确认号 ack=w+1（w 为半开半闭状态时，收到的最后一个字节数据的编号），序号 seq=u+1，然后进入 TIME-WAIT(时间等待)状态。
- 超时重传
    - TCP 规定，接收者收到数据报文段后，需回复一个确认报文段，以告知发送者数据已经收到。而发送者如果一段时间内(超时计时器)没有收到确认报文段，便重复发送。
- 连续的ARQ协议
    - 采用了流水线传输：发送方可以连续发送多个报文段(连续发送的数据长度叫做窗口)，而不必每发完一段就停下来等待确认。实际应用中，接收方也不必对收到的每个报文都做回复，而是采用累积确认方式：接收者收到多个连续的报文段后，只回复确认最后一个报文段，表示在这之前的数据都已收到。
- 流量控制和拥塞控制
    - 慢启动: 初始的窗口值很小，但是按指数规律渐渐增长，直到达到慢开始门限(ssthresh)
    - 加性增: 窗口值达到慢开始门限后，每发送一个报文段，窗口值增加一个单位量
    - 乘性减: 无论什么阶段，只要出现超时，则把窗口值减小一半

DNS(Domain Name Service域名服务)协议 - 应用层

- 该协议提供域名与IP对应的关系
- DNS服务器是一个分层次系统：
    - 根 DNS 服务器： 全世界共有 13 台根域名服务器，编号 A 到 M，其中大部分位于美国。
    - 顶级(TLD)DNS 服务器： 负责如 com 、org 、edu 等顶级域名和所有国家的顶级域名(如 cn 、uk 、jp)。
    - 权威 DNS 服务器： 大型组织、大学、企业的域名解析服务。
    - 本地 DNS 服务器： 通常与我们主机最近的 DNS 服务器。

FTP(File Transfer Protocol文件传输协议) - 应用层

- 它的主要功能是减少或消除在不同操作系统下处理文件的不兼容性，以达到便捷高效的文件传输效果。
    - FTP 只提供文件传输的基本服务，它采用 客户端—服务器 的方式，一个 FTP 服务器可同时为多个客户端提供服务。
    - 在进行文件传输时，FTP 的客户端和服务器之间会建立两个 TCP 连接：21 号端口建立控制连接，20 号端口建立数据连接。
    - FTP 的传输有两种方式：ASCII 传输模式和二进制数据传输模式。

HTTP(Hyper Text Transfer Protocol超文本传输协议) - 应用层

- 基于TCP，使用端口为80或8080
- 原理
    - 点击一个链接后，浏览器向服务器发起 TCP 连接；
    - 连接建立后浏览器发送 HTTP 请求报文，然后服务器回复响应报文；
    - 浏览器将收到的响应报文内容显示在网页上；
    - 报文收发结束，关闭 TCP 连接。
- 响应报文的状态码
    - 1xx：通知信息，如收到或正在处理。
    - 2xx：成功接收。
    - 3xx：重定向。
    - 4xx：客户的差错，如 404 表示网页未找到。
    - 5xx：服务器的差错，如常见的 502 Bad Gateway。

Telnet协议

- 基于TCP，使用端口23
- 远程登陆服务的过程：
  - 本地与远程主机建立连接。该过程实际上是建立一个 TCP 连接，用户必须知道远程主机的 IP 地址或域名;
  - 将本地终端上输入的用户名和口令及以后输入的任何命令或字符以 NVT (Net Virtual Terminal) 格式传送到远程主机。该过程实际上是从本地主机向远程主机发送一个 IP 数据包;
  - 将远程主机输出的 NVT 格式的数据转化为本地所接受的格式送回本地终端，包括输入命令回显和命令执行结果;
  - 最后，本地终端对远程主机进行撤消连接。该过程是撤销一个 TCP 连接。

SMTP(SImple Mail Transfer Protocol简单邮件传输协议)和POP3(Post Office Protoco Version3邮局协议版本3)

- 使用TCP协议，使用端口25(SMTP),110(POP3)
- SMTP的链接和发送过程
  - 建立 TCP 连接
  - 客户端向服务器发送 HELO 命令以标识发件人自己的身份，然后客户端发送 MAIL 命令
  - 服务器端以 OK 作为响应，表示准备接收
  - 客户端发送 RCPT 命令
  - 服务器端表示是否愿意为收件人接收邮件
  - 协商结束，发送邮件，用命令 DATA 发送输入内容
  - 结束此次发送，用QUIT命令退出
- POP3工作过程
  - 用户运行用户代理（如Foxmail, Outlook Express）
  - 用户代理（以下简称客户端）与邮件服务器（以下简称服务器端）的 110 端口建立 TCP 连接
  - 客户端向服务器端发出各种命令，来请求各种服务（如查询邮箱信息，下载某封邮件等）
  - 服务端解析用户的命令，做出相应动作并返回给客户端一个响应
  - 上述的两个步骤交替进行，直到接收完所有邮件转到下一步，或两者的连接被意外中断而直接退出
  - 用户代理解析从服务器端获得的邮件，以适当地形式（如可读）的形式呈现给用户
- STMP和POP3协同工作
  - 通过 smtp 协议连接到 smtp 服务器，然后发一封邮件给 sohu 的 smtp 服务器
  - 通过 smtp 协议将邮件转投给 sina 的 smtp 服务器（邮件发送服务器）
  - 将接收到的邮件存储到 gacl@sina.com 这个邮件账号分配的存储空间中
  - 通过 POP3 协议连接到 POP3 服务器收取邮件
  - 从 gacl@sina.com 账号的存储空间当中取出邮件
  - POP3 服务器将取出来的邮件回送给 gacl@sina.com 账户