计算机网络

* OSI七层协议模型
  + 应用层
    - 各种应用程序服务、协议，如HTTP、FTP、SMTP、POP3
    - FTP,WWW,Telnet,NFS,SMTP,Gateway,SNMP
  + 表示层
    - 信息的语法语义以及它们的关联，如加密解密、转换翻译、压缩解压缩
    - TIFF,GIF,JPEG,PICT,ASCII,EBCDIC,encryption,MPEG,MIDI,HTML
  + 会话层
    - 不同机器上的用户之间建立及管理会话
    - RPC,SQL,NFS,NetBIOS,names,AppleTalk,ASP,DECnet,SCP
  + 传输层
    - 接收会话层的数据，在必要时将这些数据分隔，并把数据交给网络层，且保证这些数据段有效地到达对端。TCP/UDP
    - TCP，UDP，SPX
  + 网络层
    - 控制子网的运行，比如逻辑编址、分组传输、路由传输，最复杂的IP
    - 路由器，利用IP地址来确定数据转发的地址
    - IP，IPX，AppleTalk DDP，【ARP,RARP】
  + 数据链路层
    - 物理寻址，同时将原比特流转变为逻辑传输路线
    - 网桥、交换机，交换机基于MAC地址识别，用来进行封转转发数据包的机器
    - Frame Relay, HDLC, PPP, IEEE 802.3/802.2, FDDI, ATM, IEEE 802.5/802.2
  + 物理层
    - 机械、电子、定时接口通信信道上的原始比特流传输
    - 网卡，网线、集线器、中继器、调制解调器，集线器采用广播形式传输信息
    - EIA/TIA-232, EIA/TIA-499, V.35, V.24, RJ45, Ethernet, 802.3, 802.5, FDDI, NRZI, NRZ, B8ZS
* TCP/IP参考模型（协议栈）
  + 应用层
    - Telnet,FTP,SMTP,SNMP.
  + 传输层
    - TCP ,UDP,UGP
  + 网络层
    - IP,ICMP,IGMP，【ARP,RARP】
  + 网络接口层
* 常用端口
* MAC和IP地址
* ARP协议
  + 地址解析，解决同一个局域网上的主机或路由器的IP地址和硬件地址之间的映射问题，将IP地址解析为MAC地址，RARP是逆地址解析，根据MAC找IP
* ICMP
  + 网络控制报文协议，IP层协议，用于在IP主机、路由器之间传递控制消息
  + 控制消息是指网络通不通，主机是否可达，路由器是否可用等网络本身的消息
* ping
  + 测试一台主机是否可达
  + 向指定的IP发送一定长度的ICMP协议的数据包，按照约定，如果指定IP地址存在的话就返回同样大小的数据包
  + 用类型码为0的ICMP发请求，收到请求的主机则用类型码为8的ICMP回应
  + 如果ping网站需要同DNS（应用层）将域名转换成IP地址
* traceroute
  + 用来侦测主机到目的主机之间所经路由情况的工具
  + 原理
    - 收到目的主机的IP后
    - 给目的主机发送一个TTL=1的UDP数据包
    - 经过第一个路由器收到这个数据包以后，TTL自动减一，同时产生一个主机可达的ICMP数据报给主机
    - 主机收到这个数据报之后再发送一个TTL=2的UDP数据报给目的主机，刺激第二个路由器给主机发ICMP数据报
    - 如此往复直到到达目的主机
    - 这样，traceroute就拿到了所有路由器的IP
* DNS
  + 域名系统， 将域名和IP地址 相互映射的一个分布式数据库
  + 运行在UDP协议上，使用端口号53
* TCP和UDP
  + 要求可靠传输
    - TCP
      * 可靠，面向连接，面向字节流
  + 要求实时传出
    - UDP
      * 不可靠，无连接，面向报文
  + 如何设计用UDP保证一定可靠性
  + 为什么TCP是可靠连接
* TCP（传输控制协议）
  + 应用
    - SMTP，电子邮件
    - TELNET，远程终端接入
    - HTTP，万维网
    - FTP，文件传输
  + TCP三次握手
    - 目的
      * 同步连接双方的序列号和确认号，并交换TCP窗口大小信息
    - 为什么需要三次握手
      * 防止已失效的连接请求报文突然又传送到了服务端，因而产生错误
    - 第一次握手
      * 客户端发送连接请求报文段，将SYN位置1，seq=x；客户端进入SYN\_SEND状态，等待服务器确认
    - 第二次握手
      * 服务器收到客户端的SYN报文段，需要对这个SYN报文段进行确认设置确认号ack=x+1；同时自己还要发送SYN同步请求，将SYN为置1，序列号seq=y；服务端将上述所有信息放到一个报文段（SYN+ACK），发送给客户端此时服务器进入SYN\_RECV状态
    - 第三次握手
      * 客户端收到SYN+ACK报文，设置序列号seq=x+1,确认号ack=y+1，向服务器发送ACK报文段发送完毕后，客户端和服务端都进入ESTABLISHED状态，三次握手完成
  + TCP四次挥手
    - 第一次挥手
      * 若A认为数据发送已完成，则向B发送连接释放请求，设置FIN=1，序列号seq=x，A进入FIN\_WAIT-1状态
    - 第二次挥手
      * B收到连接释放请求后，进入CLOSE\_WAIT状态，并向A发送连接释放的应答，设置ACK=1，seq=v，ack=u+1A收到应答后，进入FIN\_WAIT-2状态，等待B发送连接释放状态第二次挥手完成后，A到B方向上的连接已释放，B不会再接收数据，A不在发送数据，但B仍可以向A发送数据
    - 第三次握手
      * 当B向A发送完所有数据后，向A发送连接释放请求，设FIN=1，ACK=1，seq=w，ack=u+1，B进入LAST\_ACK状态
    - 第四次挥手
      * A收到释放请求后，向B发送确认应答，此时A进入TIME\_WAIT状态，该状态会持续2MSL时间，若该状态内没有收到B的重发请求，就进入CLOSED状态。B收到应答后，也进入CLOSED状态
    - 为什么需要四次挥手
      * TCP是全双工通信，前两次断开一个方向的连接，后两次断开一个方向的连接
    - 为什么需要2MSL
      * 为了保证A发送的最后一个确认报文能够到达B
      * 为了防止已失效的报文段出现在下一次的连接中
  + 拥塞控制
  + 流量控制
* UDP（用户数据报协议）
  + 应用
    - DNS，域名转换
    - TFTP，文件传输
    - SNMP，网络管理
    - NFS，远程文件服务器
  + 问题
    - 用udp下载的文件缺失了怎么判断，怎么补救
* HTTP
  + 超文本传输协议，客户端和服务器之间数据传输的格式规范
  + 状态码类别
    - 1XX
      * 信息，服务器收到请求，需要请求者继续执行操作
    - 2XX
      * 成功，操作被成功接收并处理
    - 3XX
      * 重定向，需要进一步操作以完成请求
    - 4XX
      * 客户端错误，请求包含语法错误或无法完成请求
    - 5XX
      * 服务器错误，服务器在处理请求时发生了错误
  + header
* 浏览器键入URL后的访问流程
  + 解析判断地址是否合法
  + 查看缓存：浏览器缓存，系统缓存、路由器缓存，有就直接显示界面
  + 发送HTTP请求前，通过DNS域名解析，将网址解析成IP地址
  + 浏览器向服务器发起TCP三次握手
  + 连接成功后，TCP分割报文段发送HTTP请求数据包
  + 服务器收到请求，返回数据，TCP重组报文段到浏览器
* 基础知识
  + OSI
    - Open Systems Interconnection，开放系统互连
  + ISO
    - International Standards Organization，国际标准化组织
  + 协议
    - 一整套标准化的允许两个电子设备互相连接和交换信息的规则