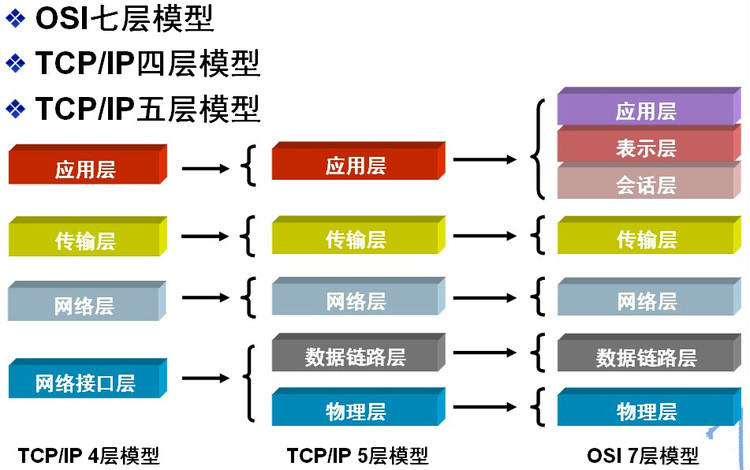
* 计算机网络定义：一个将分散的具有独立功能的计算机系统，通过通信设备与线路连接起来，有功能完善的软件实现资源共享和信息传递的系统，简而言之是一些互联的自治的计算机系统的集合
* 计算机网络组成
  + 组成部分：硬件+软件+协议
  + 工作方式：边缘+核心
  + 功能组成：通信子网+资源子网
* 计算机网络功能
  + 数据通信
  + 资源共享
  + 分布式处理
  + 提高可靠性
  + 负载均衡
* 计算机网络分类
  + 按范围：广域网、城域网、局域网、个人区域网
  + 按传播技术：广播式、点对点
  + 按拓扑结构（主要指通信子网）：星型、总线型、环形、网状
  + 按使用者：公用、专用
  + 按交换技术：电路交换、报文交换、分组交换
  + 按传输介质：有线、无线
* 计算机网络性能指标：带宽、时延、时延带宽积、往返时延、吞吐量、速率



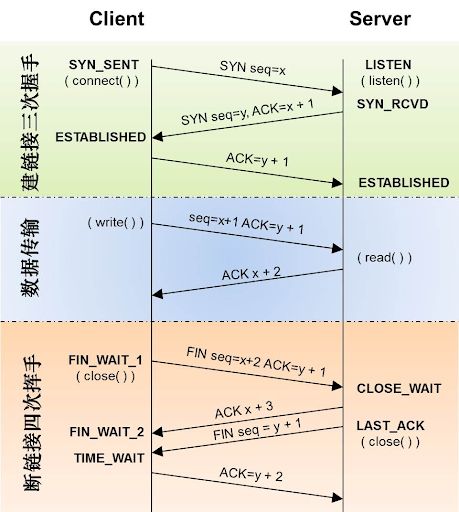
* OSI（开放式系统互连）参考模型七层体系结构与功能：OSI是一个开放性的通信系统互连参考模型，OSI 七层模型通过七个层次化的结构模型使不同的系统不同的网络之间实现可靠的通讯，每个层屏蔽下层的差异并向上层提供服务

|  |  |  |
| --- | --- | --- |
| 协议 | 层 | 对应功能 |
| HTTP,FTP,SMTP,DNS,POP3 | 应用层 | 为用户提供应用服务 |
| JPEG,ASCII | 表示层 | 为数据提供表示、加密、压缩 |
| SQL,NFS,NetBIOS | 会话层 | 确定数据是否要进行网络传输 |
| TCP,UDP | 传输层 | 对报文进行分组、传输协议、端口封装、差错校验 |
| IP,ICMP | 网络层 | IP地址编译、路由选择 |
| CSMA/CD，HDLC | 数据链路层 | MAC地址编译 |
| RS-232-C | 物理层 | 数据实际传输、电气特性定义 |

* TCP/IP四层协议体系结构：应用层、传输层、网络层、网络接口层
* 五层协议体系结构：应用层、传输层、网络层、数据链路层、物理层
  + 应用层：通过应用进程间的交互来完成特定网络应用，协议有域名系统DNS、支持电子邮件的SMTP，交互的数据单元称为报文
  + 传输层：负责向两台主机进程之间的通信提供通用的数据传输服务，主要使用以下两种协议：传输控制协议 TCP（Transmisson Control Protocol）--提供面向连接的，可靠的数据传输服务。用户数据协议 UDP（User Datagram Protocol）--提供无连接的，尽最大努力的数据传输服务
  + 网络层：实现两个主机系统之间的数据透明传送，具体功能包括寻址和路由选择、连接的建立、保持和终止等
  + 链路层：物理地址寻址、数据的成帧、流量控制、数据的检错、重发等
  + 物理层：确保原始的数据可在各种物理媒体上传输
* TCP和UDP比较

|  |  |
| --- | --- |
| TCP | UDP |
| 面向连接，需要建立连接，结束后释放 | 无连接 |
| 提供可靠交付的服务 | 尽最大努力交付，不保证可靠 |
| 面向字节流 | 面向报文 |
| 有拥塞控制 | 无拥塞控制 |
| 每条链接1：1，全双工 | 支持M：N交互通信 |
| 首部开销大 | 首部开销小 |
| 适用于文件传输 | 适用于即时通信 |

* TCP三次握手和四次挥手



* + 三次握手：目的是建立可靠的通信信道，确认自己与对方的收发正常，缺一不可
    - 第一次握手：C什么都不能确认，S确认对方发送正常
    - 第二次握手：C确认自己发送、接收正常，对方发送、接收正常；S确认自己接收正常，对方发送正常
    - 第三次握手：双方确认自己和对方发送、接收正常
    - 三次握手传回 SYN和ACK的原因：为了建立起可靠的TCP链接。传回SYN证明发送方到接收方的通道，发送ACK来验证接收方到发送方的通道
  + 四次挥手：目的是使数据传输完成并正确关闭连接
    - 第一次挥手：C发送一个FIN，用来关闭客户端到服务器的数据传输
    - 第二次挥手：S收到FIN，发回一个ACK，确认序号为收到的信号+1.
    - 第三次挥手：S关闭与C的连接，发送一个FIN给客户端
    - 第四次挥手：C发回ACK确认，并将确认信号序号设为收到序号+1
* TCP为什么是可靠的：确认和重传机制；数据排序；流量控制；拥塞控制
  + 确认和重传机制
  + 数据排序
  + TCP利用滑动窗口实现流量控制
    - 流量控制：为了控制发送方发送速率，保证接收方来得及接收
    - 滑动窗口：窗口大小决定接收方还有多大缓冲区用于接收数据，发送方可以通过滑动窗口的大小来确定发送多少字节数据。当滑动窗口大小为0时，发送方一般不能再发数据。例外如下
      * 发送紧急数据
      * 发送1B数据报来通知接收方重新声明它希望接收的下一个字节及发送方的滑动窗口大小
  + TCP有拥塞控制机制
    - 拥塞控制：**防止过多数据注入到网络中导致路由器或链路过载**，为此发送方要维护一个拥塞窗口cwnd的状态变量
    - 算法
      * 慢开始：由小到大逐渐增大发送窗口，cwnd初始为1，每次加倍
      * 拥塞避免：每经过一个往返时间RTT就把发送放的cwnd加1
      * 快重传与快恢复：如果接收机接收到一个不按顺序的数据段，它会立即给发送机发送一个重复确认。如果发送机接收到三个重复确认，它会假定确认件指出的数据段丢失了，并立即重传这些丢失的数据段
* 五类IP地址
  + A类地址：
    - 第1字节为网络地址，其它3个字节为主机地址
    - 地址范围：1.0.0.1—126.155.255.254
    - 10.X.X.X是私有地址，127.X.X.X是保留地址
  + B类地址
    - 第1字节和第2字节为网络地址，其它2个字节为主机地址
    - 地址范围：128.0.0.1—191.255.255.254
    - 172.16.0.0—172.31.255.255是私有地址，169.254.X.X是保留地址
  + C类地址
    - 第1字节、第2字节和第3个字节为网络地址，第4个个字节为主机地址
    - 地址范围：192.0.0.1—223.255.255.254
    - 192.168.X.X是私有地址
  + D类地址
    - 不分网络地址和主机地址，它的第1个字节的前四位固定为1110
    - 地址范围：224.0.0.1—239.255.255.254
  + E类地址
    - 不分网络地址和主机地址，第1个字节的前五位固定为11110
    - 地址范围：240.0.0.1—255.255.255.254
* 在浏览器中输入url地址后显示主页的过程
  + DNS域名解析
  + 建立TCP连接
  + 发送HTTP请求
  + 服务器处理请求
  + 返回HTTP报文
  + 关闭TCP连接
  + 浏览器解析HTML
  + 浏览器布局渲染
* HTTP请求
  + 请求行：请求方法\space URL \space 协议版本
    - 请求方法
      * GET：通常用于请求服务器发送某个资源，而且应该是**安全的和幂等**的
      * POST：向服务器提交数据，比如完成表单数据的提交，将数据提交给服务器处理
      * PUT：让服务器用请求的主体部分来创建一个由所请求的URL命名的新文档；如果那个文档存在的话，就用这个主体来代替它
      * DELETE：请求服务器删除指定URL所对应的资源。但是，客户端无法保证删除操作一定会被执行，因为HTTP规范允许服务器在不通知客户端的情况下撤销请求
      * HEAD方法与GET方法的行为很类似，但服务器在响应中只返回实体的主体部分。这就允许客户端在未获取实际资源的情况下，对资源的首部进行检查
      * TRACE方法会在目的服务器端发起一个“回环”诊断
      * OPTIONS方法用于获取当前URL所支持的方法。若请求成功，则它会在HTTP头中包含一个名为“Allow”的头，值是所支持的方法，如“GET, POST”
    - URL：统一资源定位符，<协议>://<主机>:<端口>/<路径>，端口号可省略，默认为80
    - 协议版本：HTTP/主版本号.次版本号，常用的有HTTP/1.0和HTTP/1.1
  + Header请求头：{头部字段名：值}\*



Referer

Content-Length

Content-Type

* + 请求体：请求方法是post时才有，get则没有请求体，直接跟在？后面，用&隔开
* 常见HTTP状态码
  + 1xx：表示通知信息，例如表示收到或正在处理
  + 2xx：表示成功，例如表示接受或知道了
  + 3xx：表示重定向
  + 4xx：表示客户端错误
  + 5xx：表示服务器错误
* HTTPS是如何保证安全的：在HTTP跟TCP中间加多了一层加密层TLS/SSL。HTTPS使用非对称加密，加密数据用的密钥（公钥），跟解密数据用的密钥（私钥）是不一样的
* 如何理解HTTP协议是无状态的：**协议对事物没有记忆能力，服务器不知道客户端是什么状态**
* ping命令基于哪一层协议的原理是什么：基于网络层的命令，是基于ICMP协议工作的
* 常见协议与对应端口号
  + FTP：21
  + DNS：53
  + HTTP：80
  + HTTPS：443
  + MySQL：3306