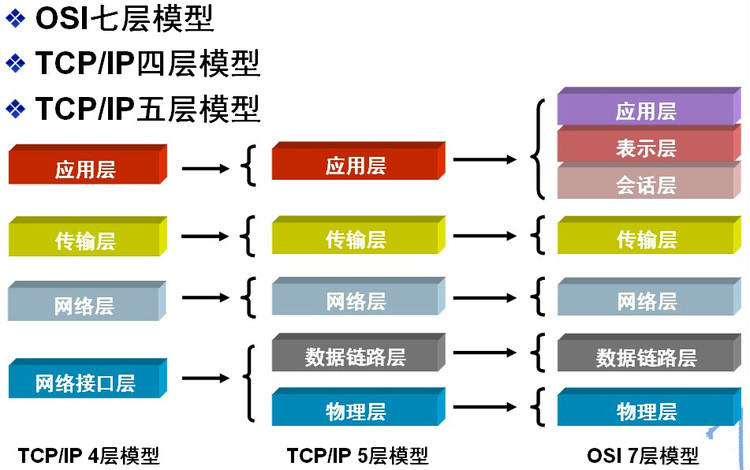
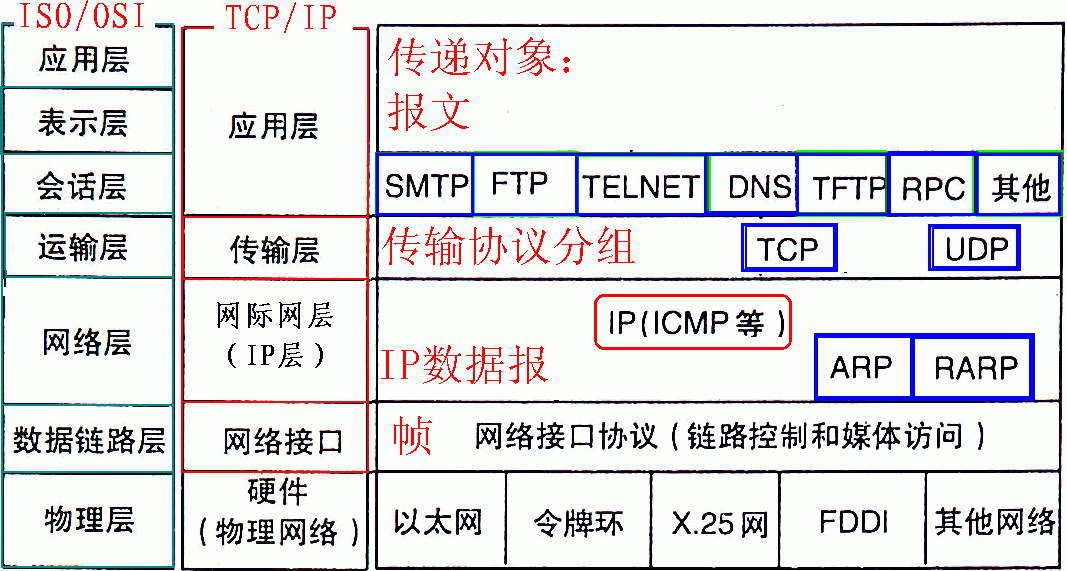
参考网站：<https://blog.csdn.net/freeygh/article/details/90957106>

**计算机网络学习的核心内容就是网络协议的学习：**



各层的协议：



**物理层：**

**该层为上层协议提供了一个传输数据的可靠的物理媒体。简单的说，物理层确保原始的数据可在各种物理媒体上传输。**物理层记住两个重要的设备名称，**中继器（Repeater，也叫放大器）和集线器**

**数据链路层：**

**该层的作用包括：物理地址寻址、数据的成帧、流量控制、数据的检错、重发等。**

**有关数据链路层的重要知识点：**

**1> 数据链路层为网络层提供可靠的数据传输；**

**2> 基本数据单位为帧；**

**3> 主要的协议：以太网协议；**

**4> 两个重要设备名称：**网桥和交换机**。**

****网络层：****

**1> 网络层负责对子网间的数据包进行路由选择。此外，网络层还可以实现拥塞控制、网际互连等功能；**

**2> 基本数据单位为**IP数据报**；**

**3> 包含的主要协议：**

**IP协议（Internet Protocol，因特网互联协议）;**

**ICMP协议（Internet Control Message Protocol，因特网控制报文协议）;**

**ARP协议（Address Resolution Protocol，地址解析协议）;**

**RARP协议（Reverse Address Resolution Protocol，逆地址解析协议）。**

1. **重要的设备：**路由器****

****传输层：****

**1> 传输层负责将上层数据分段并提供**端到端**的、可靠的或不可靠的传输以及端到端的**差错控制和流量控制**问题；**

**2> 包含的主要协议：TCP协议（Transmission Control Protocol，传输控制协议）、UDP协议（User Datagram Protocol，用户数据报协议）；**

**3> 重要设备：网关**

****会话层：****

**会话层管理主机之间的会话进程，即负责建立、管理、终止进程之间的会话。会话层还利用在数据中插入校验点来实现数据的同步。**

****表示层：****

**表示层对上层数据或信息进行变换以保证一个主机应用层信息可以被另一个主机的应用程序理解。**表示层的数据转换包括数据的加密、压缩、格式转换**等。**

****应用层：****

**为操作系统或网络应用程序提供访问网络服务的接口。**

**会话层、表示层和应用层重点：**

**1> 数据传输基本单位为报文；**

**2> 包含的主要协议：FTP（文件传送协议）、Telnet（远程登录协议）、DNS（域名解析协议）、SMTP（邮件传送协议），POP3协议（邮局协议），HTTP协议（Hyper Text Transfer Protocol）。**

**地址解析协议：**

**即ARP（Address Resolution Protocol），是根据IP地址获取物理地址的一个TCP/IP协议。**主机发送信息时将包含目标IP地址的ARP请求广播到网络上的所有主机，并接收返回消息，以此确定目标的物理地址；收到返回消息后将该IP地址和物理地址存入本机ARP缓存中并保留一定时间，下次请求时直接查询ARP缓存以节约资源。

**逆地址解析协议：**

与地址解析协议相反，只不过请求的目标是RARP服务器。

**常见的路由选择协议**有：RIP协议、OSPF协议。

RIP协议 ：底层是**贝尔曼福特算法**，它选择路由的度量标准（metric)是跳数，最大跳数是15跳，如果大于15跳，它就会丢弃数据包。

OSPF协议 ：Open Shortest Path First开放式最短路径优先，底层是**迪杰斯特拉算法**，T是链路状态路由选择协议，它选择路由的度量标准是带宽，延迟

**TCP/IP协议：**

TCP/IP协议是Internet最基本的协议、Internet国际互联网络的基础，由**网络层的IP协议和传输层的TCP协议组成**。通俗而言：TCP负责发现传输的问题，一有问题就发出信号，要求重新传输，直到所有数据安全正确地传输到目的地。而IP是给因特网的每一台联网设备规定一个地址。

IP数据包是不可靠的

TCP是面向连接的通信协议，通过三次握手建立连接，通讯完成时要拆除连接，由于TCP是面向连接的所以只能用于端到端的通讯。TCP提供的是一种可靠的数据流服务，采用“**带重传的肯定确认**”技术来实现传输的**可靠性**。TCP还采用一种称为“**滑动窗口**”的方式进行**流量控制**，所谓窗口实际表示接收能力，用以限制发送方的发送速度**。**

**[注：tcp协议如何保证传输的可靠性](https://blog.csdn.net/jhh_move_on/article/details/45770087" \t "https://www.cnblogs.com/inception6-lxc/p/_blank)**

流量控制主要针对的是**端到端传输中控制流量大小并保证传输可靠性**（未收到ack就不滑动）。流量控制往往是指点对点通信量的控制，**所要做的是抑制发送端发送数据的速率**。

**拥塞控制主要是一个全局性过程，涉及到所有主机，路由器，以及与降低网络传输性能有关的所有因素**。**防止过多的数据注入到网络中**。如果有发生丢包则通过拥塞控制减小窗口，确定出合适(慢启动 拥塞避免 快重传 快恢复)的拥塞窗口（增性加乘性减）

**三次握手和四次挥手：**



注：seq:"sequance"序列号；ack:"acknowledge"确认号；SYN:"synchronize"请求同步标志；；ACK:"acknowledge"确认标志"；FIN："Finally"结束标志。

TCP连接建立过程：首先Client端发送连接请求报文，Server段接受连接后回复ACK报文，并为这次连接分配资源。Client端接收到ACK报文后也向Server段发生ACK报文，并分配资源，这样TCP连接就建立了。

TCP连接断开过程：假设Client端发起中断连接请求，也就是发送FIN报文。Server端接到FIN报文后，意思是说"我Client端没有数据要发给你了"，但是如果你还有数据没有发送完成，则不必急着关闭Socket，可以继续发送数据。所以你先发送ACK，"告诉Client端，你的请求我收到了，但是我还没准备好，请继续你等我的消息"。这个时候Client端就进入FIN\_WAIT状态，继续等待Server端的FIN报文。当Server端确定数据已发送完成，则向Client端发送FIN报文，"告诉Client端，好了，我这边数据发完了，准备好关闭连接了"。Client端收到FIN报文后，"就知道可以关闭连接了，但是他还是不相信网络，怕Server端不知道要关闭，所以发送ACK后进入TIME\_WAIT状态，如果Server端没有收到ACK则可以重传。“，Server端收到ACK后，"就知道可以断开连接了"。Client端等待了2MSL后依然没有收到回复，则证明Server端已正常关闭，那好，我Client端也可以关闭连接了。Ok，TCP连接就这样关闭了！

**为什么要三次握手和四次挥手，自己回答。**

**DNS**是域名系统(DomainNameSystem)的缩写，该系统用于命名组织到域层次结构中的计算机和网络服务，**可以简单地理解为将URL转换为IP地址。**

****超文本传输协议**（HTTP，HyperText Transfer Protocol)是互联网上应用最为广泛的一种网络协议。所有的WWW文件都必须遵守这个标准。**

****HTTP 协议包括哪些请求？****

**GET：请求读取由URL所标志的信息。**

**POST：给服务器添加信息（如注释）。**

**PUT：在给定的URL下存储一个文档。**

**DELETE：删除给定的URL所标志的资源。**

****HTTP 中， POST 与 GET 的区别****

**1）Get是从服务器上获取数据，Post是向服务器传送数据。**

**2）Get是把参数数据队列加到提交表单的Action属性所指向的URL中，值和表单内各个字段一一对应，在URL中可以看到。**

**3）**Get传送的数据量小，不能大于2KB**；**Post传送的数据量较大，一般被默认为不受限制。****

**4）根据HTTP规范，GET用于信息获取，而且应该是**安全的和幂等的**。**

**I. 所谓**安全的**意味着该操作用于获取信息而非修改信息。换句话说，GET请求一般不应产生副作用。就是说，它仅仅是获取资源信息，就像数据库查询一样，不会修改，增加数据，不会影响资源的状态。**

**II.**幂等的**意味着对同一URL的多个请求应该返回同样的结果。**

****http消息结构：**典型的请求头有：User-Agent：产生请求的浏览器类型。Accept：客户端可识别的内容类型列表。**

****

****HTTP响应消息：****

**在接收和解释请求消息后，服务器返回一个HTTP响应消息。**

**HTTP响应也是由三个部分组成，分别是：状态行、消息报头、响应正文**

****HTTP-Version Status-Code Reason-Phrase CRLF****

**其中，HTTP-Version表示服务器HTTP协议的版本；Status-Code表示服务器发回的响应状态代码；Reason-Phrase表示状态代码的文本描述。**

1xx：指示信息--表示请求已接收，继续处理  
2xx：成功--表示请求已被成功接收、理解、接受  
3xx：重定向--要完成请求必须进行更进一步的操作  
4xx：客户端错误--请求有语法错误或请求无法实现  
5xx：服务器端错误--服务器未能实现合法的请求

**http1.0和http1.1:**

在HTTP/1.0中默认使用短连接。也就是说，客户端和服务器每进行一次HTTP操作，就建立一次连接，任务结束就中断连接

http1.1在响应头加入这行代码:Connection:keep-alive

在使用长连接的情况下，当一个网页打开完成后，客户端和服务器之间用于传输HTTP数据的TCP连接不会关闭，客户端再次访问这个服务器时，会**继续使用这一条已经建立的连接。**

**Http和Https的区别:**

Http协议运行在TCP之上，明文传输，客户端与服务器端都无法验证对方的身份；Https是身披SSL(Secure Socket Layer)外壳的Http，运行于SSL上，SSL运行于TCP之上，是添加了加密和认证机制的HTTP。二者之间存在如下不同：

1)端口不同：Http与Https使用不同的连接方式，用的端口也不一样，前者是80，后者是443；

2)资源消耗：和HTTP通信相比，Https通信会由于加减密处理消耗更多的CPU和内存资源；

3)开销：Https通信需要证书，而证书一般需要向认证机构购买；(要发送证书验证)

Https的加密机制是一种共享密钥加密和公开密钥加密并用的混合加密机制

**对称加密与非对称加密：**

　　对称密钥加密是指加密和解密使用同一个密钥的方式，这种方式存在的最大问题就是密钥发送问题，即如何安全地将密钥发给对方；而非对称加密是指使用一对非对称密钥，即公钥和私钥，公钥可以随意发布，但私钥只有自己知道。发送密文的一方使用对方的公钥进行加密处理，对方接收到加密信息后，使用自己的私钥进行解密。

　　由于非对称加密的方式不需要发送用来解密的私钥，所以可以保证安全性；但是和对称加密比起来，它非常的慢，所以我们还是要用对称加密来传送消息，**但对称加密所使用的密钥我们可以通过非对称加密的方式发送出去**

**客户端不断进行请求链接会怎样？DDos(Distributed Denial of Service)攻击？**

服务器端会为每个请求创建一个链接，并向其发送确认报文，然后等待客户端进行确认

**DDos 攻击：**

客户端向服务端发送请求链接数据包

服务端向客户端发送确认数据包

客户端不向服务端发送确认数据包，服务器一直等待来自客户端的确认

**DDos 预防** ( 没有彻底根治的办法，除非不使用TCP )

限制同时打开SYN半链接的数目

缩短SYN半链接的Time out 时间

关闭不必要的服务

**交换机与路由器有什么区别？**

1）工作所处的OSI层次不一样，交换机工作在OSI第二层数据链路层，路由器工作在OSI第三层网络层

2）寻址方式不同：交换机根据MAC地址寻址，路由器根据IP地址寻址

3）转发速不同：交换机的转发速度快，路由器转发速度相对较慢。

**ICMP协议？**

ICMP是InternetControl Message Protocol，因特网控制报文协议。它是TCP/IP协议族的一个子协议，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。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。ICMP报文有两种：差错报告报文和询问报文。

**DHCP协议？**

动态主机配置协议，是一种让系统得以连接到网络上，并获取所需要的配置参数手段。通常被应用在大型的局域网络环境中，主要作用是**集中的管理、分配IP地址，**使网络环境中的主机动态的获得IP地址、Gateway地址、DNS服务器地址等信息，并能够提升地址的使用率

**Session、Cookie**：

Cookie和Session都是客户端与服务器之间保持状态的解决方案

参考：（查漏补缺，但是重点不突出）

**DNS服务器：**

https://mp.weixin.qq.com/s?\_\_biz=MzAwNDA2OTM1Ng==&mid=2453143291&idx=1&sn=d22c5bc68b8503998511c317874f4eac&chksm=8cf2dc78bb85556ef7e3e461b6d6a0d1422a312f3e84c184457eab6df07b014451bbef49b092&mpshare=1&scene=1&srcid=&sharer\_sharetime=1591750875536&sharer\_shareid=605b52fecbd4f990f7fcc587be229eee&key=1e758dfedc4d5b90255f96c400713656cd1de223ed5945b904f636b7cefbbc2282c2c66d2ac3bbeaa3b9c235b33c1e5258335428bfcb6ffa37bfc500632228839725a9bebbfc0eeb6dacc0c724fe4de3&ascene=1&uin=Mjk2NzIzNzgyNA%3D%3D&devicetype=Windows+10+x64&version=62090070&lang=zh\_CN&exportkey=Ay%2Fft9ezHxbdgPeqoXmGx3E%3D&pass\_ticket=7O6g4E3e7rUiRrKFxMRayIIoCG2YyVwLvNrtZoeuGhUcMdtoWZJDqJsF6x18fHu%2F