

本篇文章来学习下著名的OSI七层模型和TCP/IP四层模型，以及深入理解为什么要分层，将计算机网络体系结构一网打尽！

## 一、计算机网络体系结构

通过学习因特网的历史我们知道，上个世纪由于没有统一的标准，各个厂商生产出来的计算机无论是硬件还是软件都互不相同，也无法正常通信，更别提世界范围内的计算机互连了。

为了使不同体系结构的计算机网络都能互连，国际标准组织于1977年成立了专门机构研究该问题，不久提出了一个试图使各种计算机在世界范围内互连成网的标准框架，也就是著名的“开放系统互连参考模型”，简称OSI，该模型是一个七层协议的体系结构。



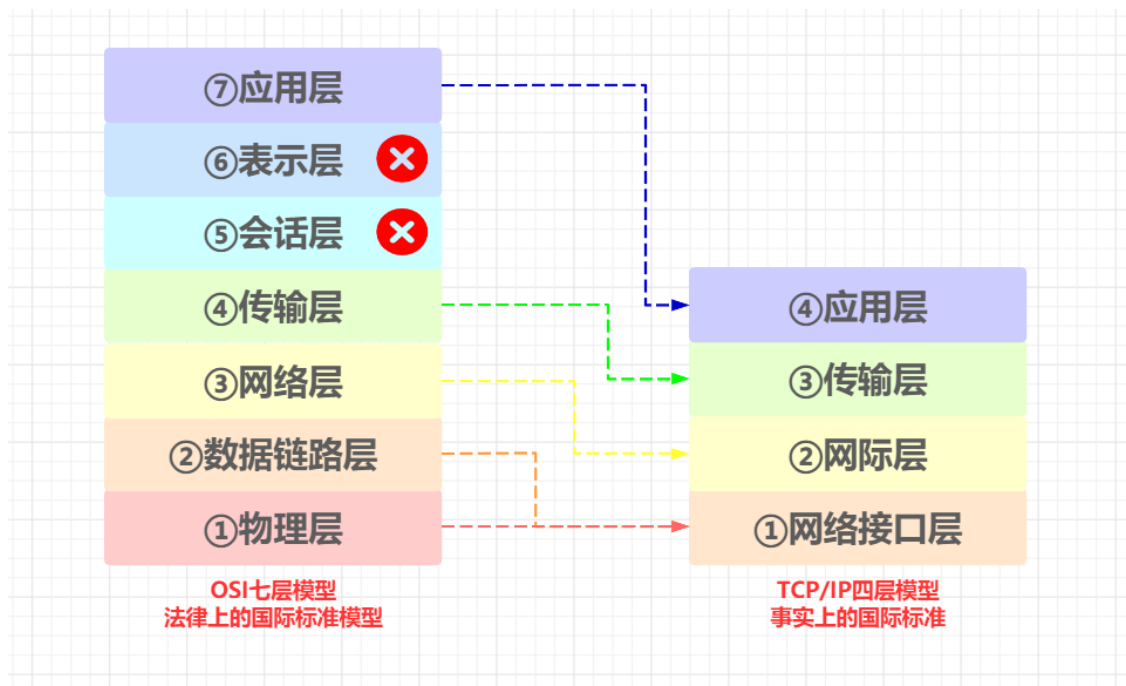
法律上的国际标准模型

OSI七层标准模型是法律上的国际标准模型。为什么强调这一点呢？因为到了20世纪90年代初期，虽然整套的OSI国际标准都已经制订出来了，但是由于因特网已经抢先在全世界覆盖了相当大的范围，因特网从1983年开始使用TCP/IP协议族，并逐步演变出TCP/IP四层模型。

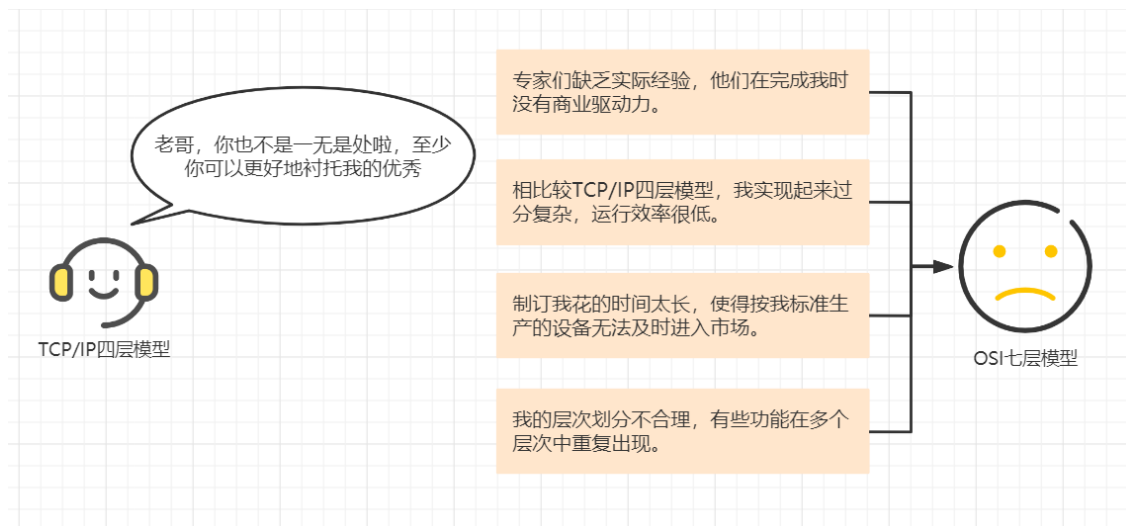


TCP/IP四层模型是事实上的国际标准，从这种意义上来说，能占领市场、被人们广泛认可的才会成为真正的标准。

我们对比两个模型可以看出，实际上TCP/IP四层模型就是对OSI七层模型的一个合并和简化。（由于IP全称叫做Internet Protocol，我们一般称为[网际互连协议](#)，因此在TCP/IP协议中称为[网际层](#)）



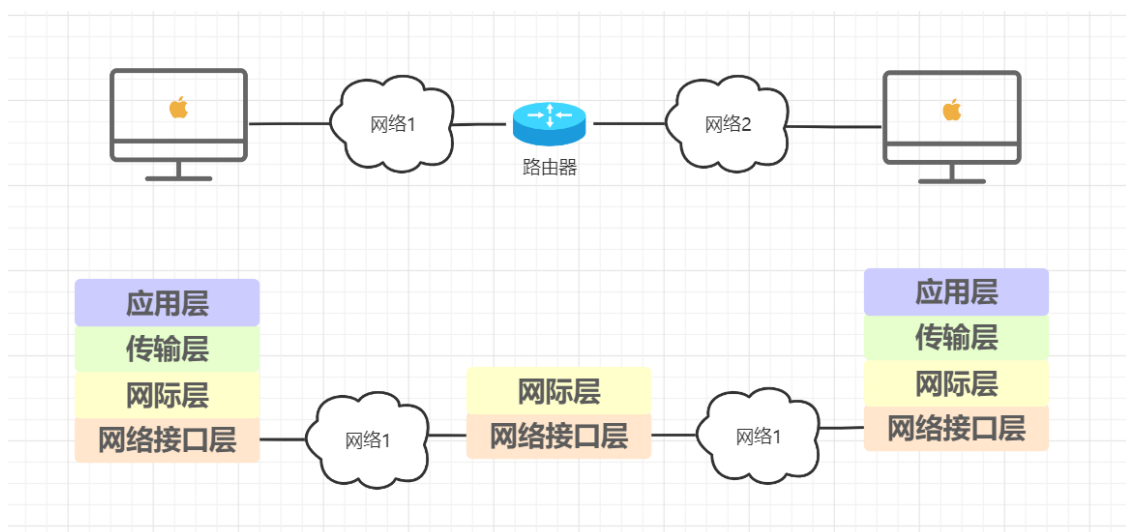
TCP/IP四层模型之所以能占领市场，OSI七层模型也得找找自身原因，OSI标准失败的原因可归纳为如下一些原因。



大多数用户每天都有接入因特网的需求，这就**要求用户的主机必须使用TCP/IP协议**。

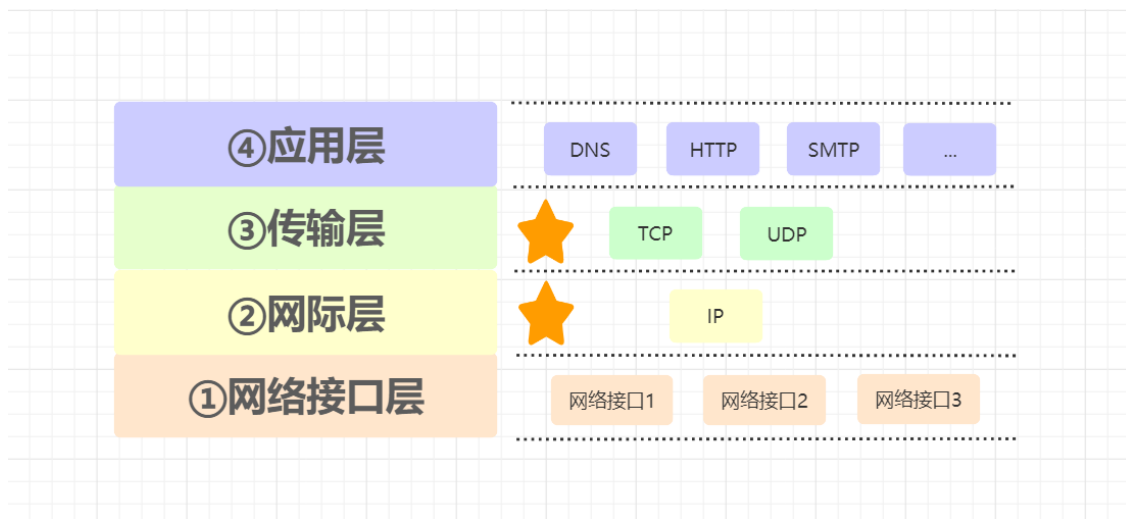
在**用户主机的操作系统中**，通常都带有符合TCP/IP体系结构标准的TCP/IP协议族。

用于网络互连的**路由器中**，也带有符合TCP/IP体系结构标准的TCP/IP协议族，只不过**路由器一般只包含网络接口层和网际层**。



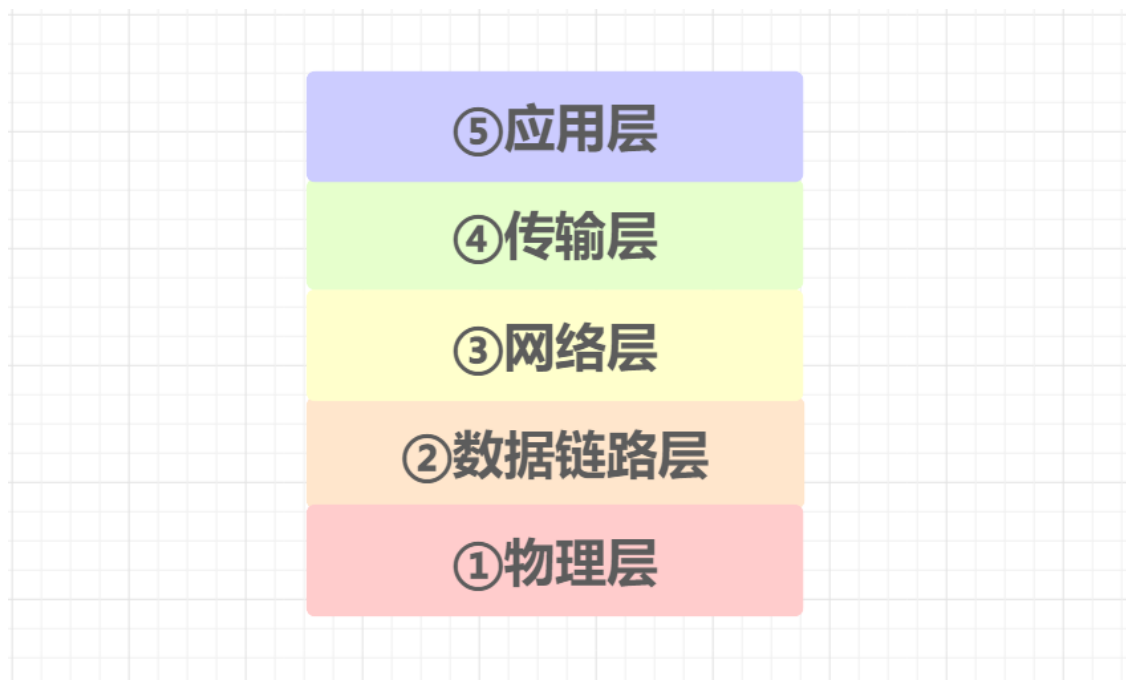
IP协议可以将不同网络进行互连，并向其上的TCP协议和UDP协议提供网络互连服务，TCP协议在享受IP协议提供的网络互连服务的基础上，可向应用层提供可靠传输的服务。

IP协议和TCP协议是其中非常重要的两个协议，因此用TCP和IP两个协议来表示整个协议大家族，常称为TCP/IP协议族。



TCP/IP体系结构的网络接口层并没有规定什么具体内容，这样做的目的是可以互连全世界各种不同的网络接口，例如有线的以太网接口，无线的WIFI接口。

然而，这对于我们学习计算机网络的完整体系而言，就会缺少一部分内容，因此在学习计算机网络原理时，会综合OSI和TCP/IP模型的优点，学习五层模型，这样更有利于我们对计算机网络原理的学习。



好了，请不要忘记这张图，这张图就是我们学习计算机网络的主体脉络，也是未来打交道最多的一张图。

## 二、计算机网络体系结构分层的必要性

我们先来考虑最简单的情况，两台计算机要互相通信，如何做呢？



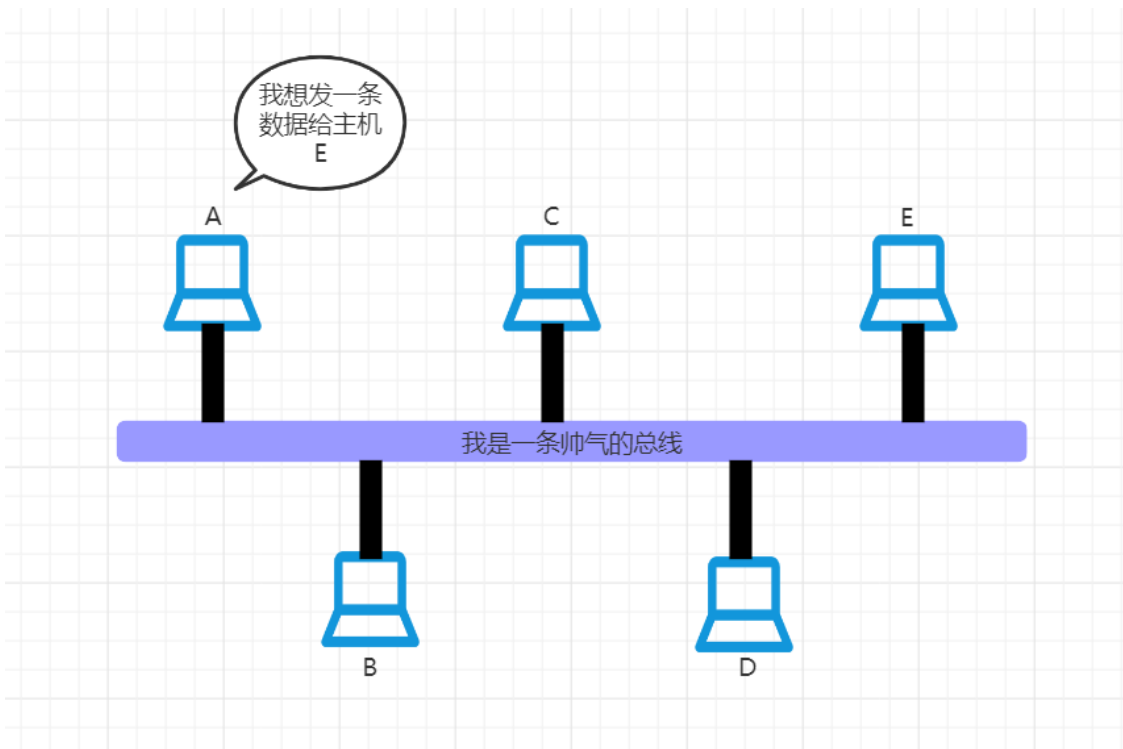
我们需要考虑以下几个问题：

- **采用什么样的传输介质**？是用我们平时用的双绞线？还是光纤呢？
- **采用什么样的物理接口**？比如平时咱们用的RJ45接口？
- **采用怎样的信号来表示比特0和1呢**？用方波信号？高电平表示1？低电平表示0？

以上问题，实际上是物理层所要关注的事情。

严格来说，传输介质不属于物理层，它并没有被包含在体系结构之中；计算机网络中传输的信号并不是简单的方波信号。

假设我们已经解决了上面的问题，两台主机之间可以互相发送比特0和1了，我们继续思考，实际上计算机网络是由多台计算机构成，比如下面图示这种，利用总线型拓扑结构构建的一个网络：



假设主机A想发送一条消息给主机E，我们知道，在总线型网络拓扑结构中，表示消息的信号都是经过这条总线传输的，所有其他主机都可以收到这个信号，那么自然而然就会有一个问题：**主机E如何知道这条消息是发送给我的呢？其他主机是如何知道这个信息不是发送给我而是要丢弃的呢？**

如何解决这个问题呢？**这就引出如何标识网络中各个主机的问题了，大家可能听说过网卡的MAC地址，实际上我们就是用这个MAC地址来标识唯一一个网卡。**（切记切记，**MAC地址是唯一标识网卡的**，MAC地址不等同于主机地址，因为一台主机可能有多个网卡）

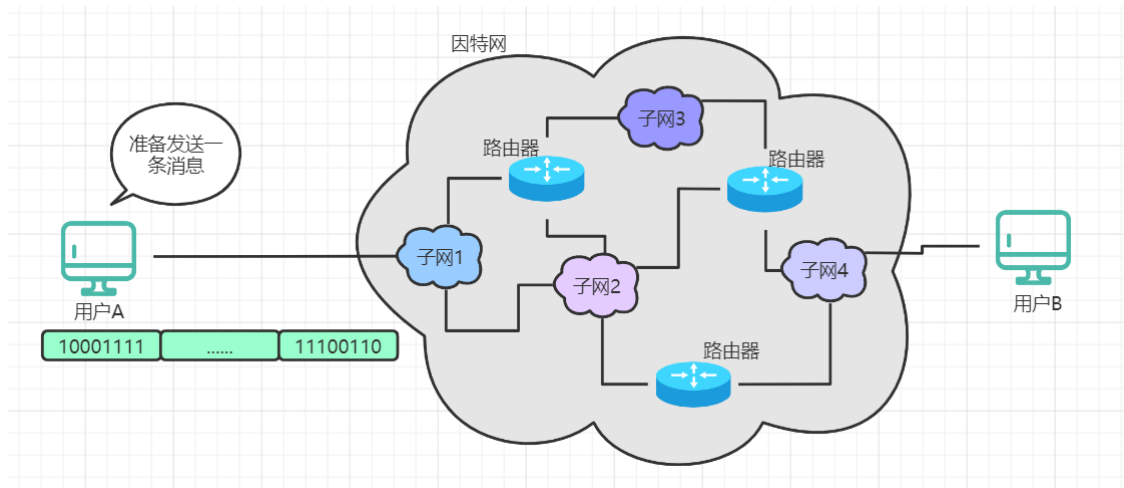
此外，**既然有了地址信息，那么就得区分地址信息和实际数据，如何从信号中区分出地址和数据也是个问题。**

再此外，**这种总线型网络拓扑结构存在消息碰撞问题，这个该如何解决呢？**后续文章将展开说明，这里点到为止。另外需要注意的是，**总线型网络拓扑结构早已淘汰，目前主流是用交换机将多台主机互连形成交换式以太网，那么以太网交换机又是如何实现的呢？**

**我们可将这些问题归结到数据链路层。**

好了，假设我们也已经解决了前面所有的问题，**那么我们会神奇地发现，在一个网络中，主机之间可以进行通信了！**

但是，我们还要继续，仅仅依靠上面两层，我们还无法使用因特网，因为我们每天使用的因特网是由多个网络组合而成，这些网络由路由器互连：

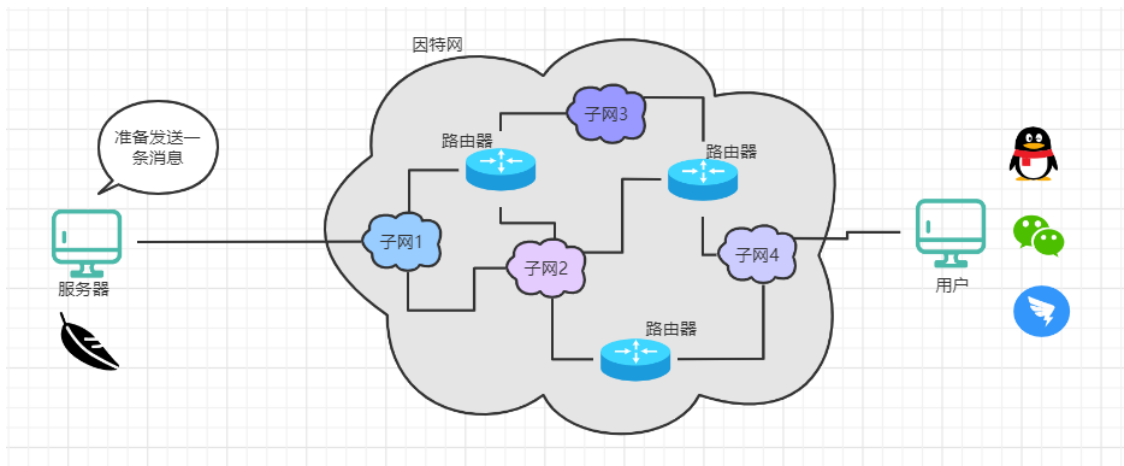


面对这个结构，我们就需要思考如下问题：

- **如何标识各网络以及网络中各主机的问题**，即网络和主机共同编址问题，例如IP地址
- **如何进行路由选择的问题**（我们可以看到分组报文可以有佷多路径可走，尽可能少走弯路或者走宽路是一个比较重要的问题）

**这个就是网络层所解决的事情。**

好了，现在一条消息可以经过多个网络后传送到另外一台主机了，但是还有问题要解决。



比如服务器返回了一个信息回来，客户端有多个应用进程在监听，那么收到的分组到底是交给哪个进程处理呢？

因此我们就需要思考：

- 如何标识与网络通信相关的应用进程，进而解决进程之间基于网络通信的问题
- 如果某个分组在传输过程中出现了误码，或者由于路由器繁忙丢弃分组，这种错误场景下如何处理呢

我们可以将这些问题划分到传输层。

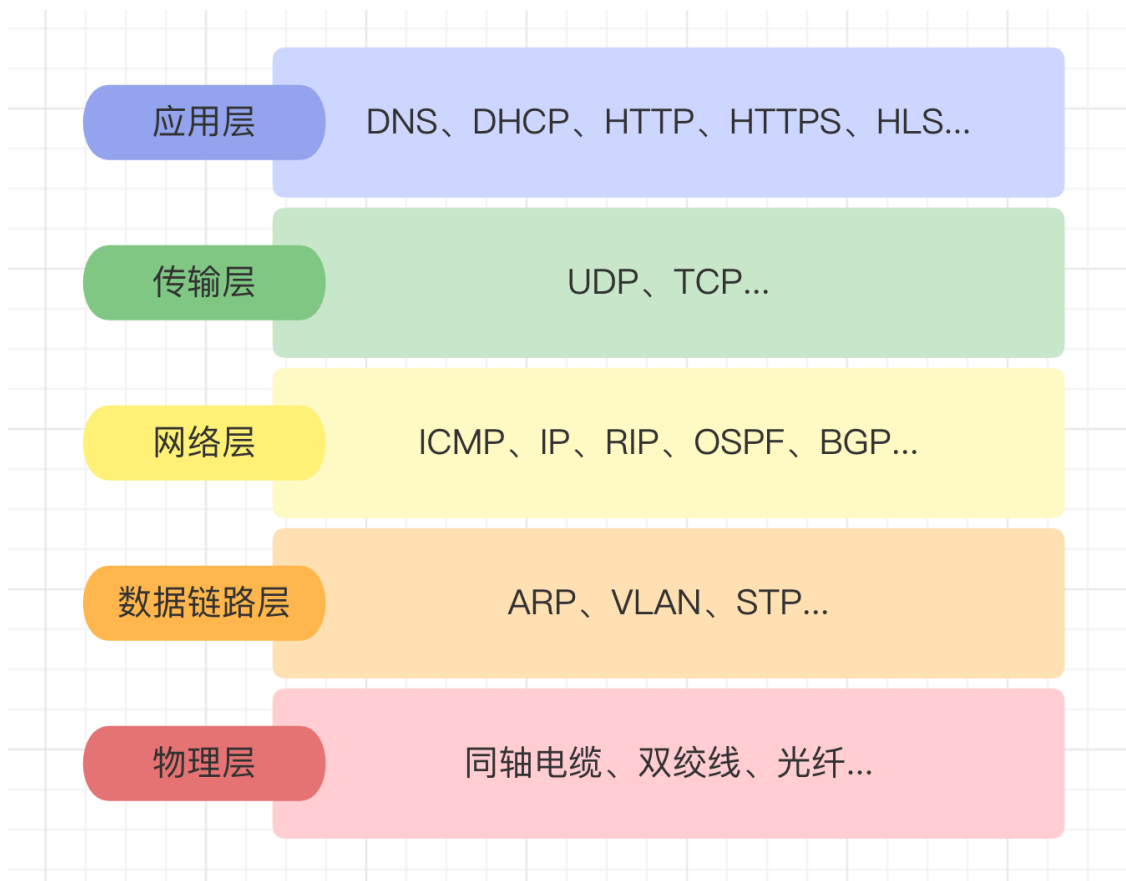
如果以上问题都已解决，那么我们就可以实现进程基于网络的通信，在此基础上，只需制订各种应用层协议，并按协议标准编写相应的应用程序，即可通过应用进程间的交互来完成特定的网络应用，比如支持万维网应用的HTTP协议，支持电子邮件的SMTP协议，支持文件传送的FTP协议等等。

我们将这些问题划分到应用层。

下面简单总结各层的基本作用：



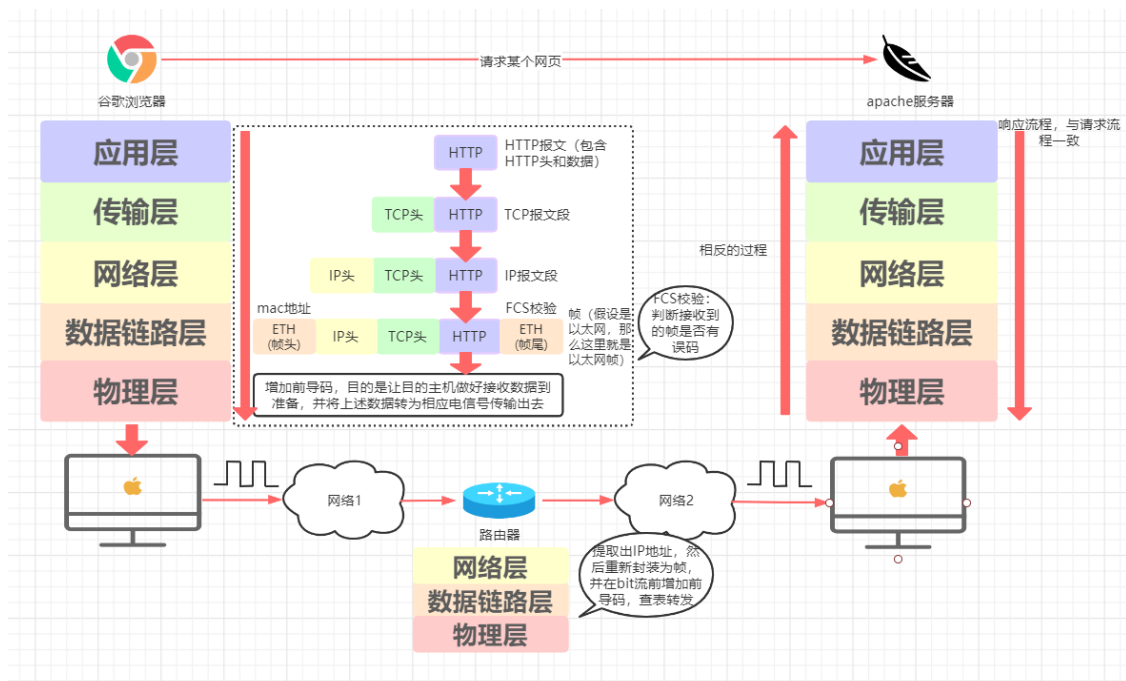
每一层都有相应的网络协议来专门干一件事儿，这里将后面即将逐个接触到的协议列举出来：



试想，如果是你，是把这些问题的放在一个模块中解决，还是应该分模块来解决呢？我想，分层的必要性已经深入人心了。

了解了五层模型后，我们以浏览器向服务器请求一个网页为例，看下整个五层模型的大致工作情况。





- 1、应用层按HTTP协议的规定，**构建一个HTTP请求报文**。
- 2、应用层将HTTP报文交给传输层处理，**传输层给HTTP请求报文增加一个TCP首部，使之成为TCP报文段**。该首部的作用是区分应用进程，以及实现可靠传输。
- 3、传输层将TCP报文段交付给网络层处理，**网络层给TCP报文段增加一个IP首部，使之成为IP数据报**。该首部的作用是标识了IP地址，可以让数据报被路由器转发。
- 4、网络层将IP数据报交付给数据链路层处理，**数据链路层给IP数据报添加一个首部和一个尾部，使之成为帧**。帧首部作用是标识了MAC地址，使之可以在一个网络内被目的主机接收。帧尾部是诸如FCS校验，作用是让目的主机判断所接收到的帧是否有误码。
- 5、**数据链路层将数据交付给物理层，物理层将帧看作是比特流**，我们这里用以太网为例，比特流前面要增加前导码，目的是让目的主机做好接收帧的准备，并**将上述比特流转为相应电/光信号通过传输介质传输出去**。
- 6、**信号通过传输介质到达路由器**，路由器的物理层将信号变换为比特流，然后去掉前导码后将其交付给数据链路层，数据链路层即可拿到帧，数据链路层再将帧首部和尾部去掉后交付给网络层，此时就是IP数据报，网络层拿到IP数据报后解析首部，从中提取出目的网络地址，然后查找自身路由表，确定转发端口，以便进行转发。OK，确定好如何转发后，这个数据就需要再次发送出去，那么同样地，网络层交付给数据链路层，数据链路层交付给物理层，物理层同样将比特流加上前导码，转为相应电/光信号通过传输介质传输出去。
- 7、信号终于来到web服务器，**物理层将收到的信号转为比特流**，去掉前导码交付给数据链路层，**数据链路层拿到帧后**，去掉帧的首部和尾部，交付给网络层，**此时网络层拿到了IP数据报**，网络层将IP数据报首部去掉后，交付给传输层，**此时传输层拿到了TCP报文段**，传输层将TCP报文段的首部去掉后，交付给应用层，**此时应用层即可拿到HTTP请求报文了**。随后应用层就会对HTTP请求报文进行解析，然后**给客户端发回HTTP响应报文**。

- 8、HTTP响应报文与之前的过程类似，**也需要经过层层封装**，经过物理层变换为相应信号，最后通过传输介质传输，客户端收到信号后，将其转换为比特流，再**通过层层解封**，最终取出HTTP响应报文，客户端即可根据响应报文进行页面渲染，我们终于看到了一个网页。

## 三、分层模型的两个重要原则

### 3.1、每一层都是相对独立的

**每一层都有其自己的使命，每一层的变革都不需要更改其他层**，例如网络层，目前用的是IPV4（32位），但是由于地址数量已经不够使用，需要逐步演化到IPV6（128位）。

这个时候只需要更改第三层即网络层协议，其他层可以保持不变。

此外，**下层协议服务于上层协议，并且对于上层协议是透明的**，即上层只需要享受下层服务即可，不必了解实现细节。

### 3.2、每一层只能和相邻的层通信

我们知道，当我们浏览如 [www.google.com](http://www.google.com) 的时候，实际上是穿越了整个TCP/IP模型的：应用层-->传输层-->网络层-->数据链路层-->物理层。

谷歌服务器在处理的时候反之：物理层-->数据链路层-->网络层-->传输层-->应用层。

**得亏这个原则，保证了通信过程中各层都会被使用，也就保证每一层的作用都能得到实现。**