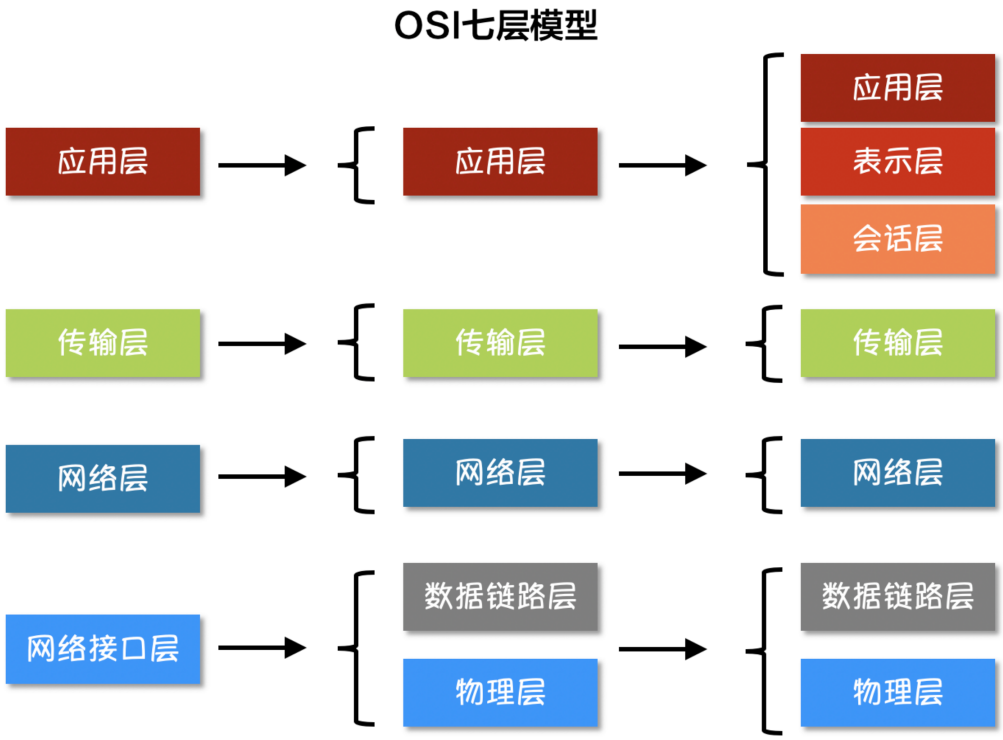
互联网协议

互联网的核心是一系列协议，总称为"互联网协议"（Internet Protocol Suite），正是这一些协议规定了电脑如何连接和组网。我们理解了这些协议，就理解了互联网的原理。由于这些协议太过庞大和复杂，没有办法在这里一概而全，只能介绍一下我们日常开发中接触较多的几个协议。

互联网分层模型

互联网的逻辑实现被分为好几层。每一层都有自己的功能，就像建筑物一样，每一层都靠下一层支持。用户接触到的只是最上面的那一层，根本不会感觉到下面的几层。要理解互联网就需要自下而上理解每一层的实现的功能。



如上图所示，互联网按照不同的模型划分会有不用的分层，但是不论按照什么模型去划分，越往上的层越靠近用户，越往下的层越靠近硬件。在软件开发中我们使用最多的是上图中将互联网划分为五个分层的模型。

接下来我们一层一层的**自底向上**介绍一下每一层。**物理层→数据链路层→网络层→传输层→应用层**

物理层

我们的电脑要与外界互联网通信，需要先把电脑连接网络，我们可以用双绞线、光纤、无线电波等方式。这就叫做"实物理层"，它就是把电脑连接起来的物理手段。它主要规定了网络的一些电气特性，作用是负责传送0和1的电信号。

物理层解决如何来连接各种计算机的传输媒体（电缆、光纤电缆等）上传输数据比特流，而不是指具体的传输媒体。

物理层主要任务：确定与传输媒体接口有关的一些特性 定义标准。

物理层四大特性：

机械特性： 定义物理连接的特性，规定物理连接时所采用的规格、接口形状、引线数目、引脚数量和排列情况等

电器特性：规定传输二进制位时，线路上信号的电压范围、阻抗匹配、传输速率和距离限制等。

功能特性：指明某条线路上出现的某一电平表示何种意义，接口不见的信号线的用途

规程特性（过程特性）：定义各物理线路的工作规程和时序关系。

数据链路层

单纯的0和1没有任何意义，所以我们使用者会为其赋予一些特定的含义，规定解读电信号的方式：例如：多少个电信号算一组？每个信号位有何意义？这就是"数据链接层"的功能，它在"物理层"的上方，确定了物理层传输的0和1的分组方式及代表的意义。早期的时候，每家公司都有自己的电信号分组方式。逐渐地，一种叫做"以太网"（Ethernet）的协议，占据了主导地位。

以太网规定，一组电信号构成一个数据包，叫做"帧"（Frame）。每一帧分成两个部分：标头（Head）和数据（Data）。其中"标头"包含数据包的一些说明项，比如发送者、接受者、数据类型等等；"数据"则是数据包的具体内容。"标头"的长度，固定为18字节。"数据"的长度，最短为46字节，最长为1500字节。因此，整个"帧"最短为64字节，最长为1518字节。如果数据很长，就必须分割成多个帧进行发送。

那么，发送者和接受者是如何标识呢？以太网规定，连入网络的所有设备都必须具有"网卡"接口。数据包必须是从一块网卡，传送到另一块网卡。网卡的地址，就是数据包的发送地址和接收地址，这叫做MAC地址。每块网卡出厂的时候，都有一个全世界独一无二的MAC地址，长度是48个二进制位，通常用12个十六进制数表示。前6个十六进制数是厂商编号，后6个是该厂商的网卡流水号。有了MAC地址，就可以定位网卡和数据包的路径了。

我们会通过ARP协议来获取接受方的MAC地址，有了MAC地址之后，如何把数据准确的发送给接收方呢？其实这里以太网采用了一种很"原始"的方式，它不是把数据包准确送到接收方，而是向本网络内所有计算机都发送，让每台计算机读取这个包的"标头"，找到接收方的MAC地址，然后与自身的MAC地址相比较，如果两者相同，就接受这个包，做进一步处理，否则就丢弃这个包。这种发送方式就叫做"广播"（broadcasting）。

数据链路层在物理层提供的服务基础上**向网络层提供服务**，其最基本的服务是将源自网络层来的数据**可靠地**传输到相邻结点的目标机网络层。其主要作用是**加强物理层传输原始比特流的功能**，将物理层提供的可能出错的物理连接改造为逻辑上无差错的数据链路，使之对网络层表现为一条无差错的链路。

网络层

按照以太网协议的规则我们可以依靠MAC地址来向外发送数据。理论上依靠MAC地址，你电脑的网卡就可以找到身在世界另一个角落的某台电脑的网卡了，但是这种做法有一个重大缺陷就是以太网采用广播方式发送数据包，所有成员人手一"包"，不仅效率低，而且发送的数据只能局限在发送者所在的子网络。也就是说如果两台计算机不在同一个子网络，广播是传不过去的。这种设计是合理且必要的，因为如果互联网上每一台计算机都会收到互联网上收发的所有数据包，那是不现实的。

因此，必须找到一种方法区分哪些MAC地址属于同一个子网络，哪些不是。如果是同一个子网络，就采用广播方式发送，否则就采用"路由"方式发送。这就导致了"网络层"的诞生。它的作用是引进一套新的地址，使得我们能够区分不同的计算机是否属于同一个子网络。这套地址就叫做"网络地址"，简称"网址"。

"网络层"出现以后，每台计算机有了两种地址，一种是MAC地址，另一种是网络地址。两种地址之间没有任何联系，MAC地址是绑定在网卡上的，网络地址则是网络管理员分配的。网络地址帮助我们确定计算机所在的子网络，MAC地址则将数据包送到该子网络中的目标网卡。因此，从逻辑上可以推断，必定是先处理网络地址，然后再处理MAC地址。

规定网络地址的协议，叫做IP协议。它所定义的地址，就被称为IP地址。目前，广泛采用的是IP协议第四版，简称IPv4。IPv4这个版本规定，网络地址由32个二进制位组成，我们通常习惯用分成四段的十进制数表示IP地址，从0.0.0.0一直到255.255.255.255。

根据IP协议发送的数据，就叫做IP数据包。IP数据包也分为"标头"和"数据"两个部分："标头"部分主要包括版本、长度、IP地址等信息，"数据"部分则是IP数据包的具体内容。IP数据包的"标头"部分的长度为20到60字节，整个数据包的总长度最大为65535字节。

传输层

有了MAC地址和IP地址，我们已经可以在互联网上任意两台主机上建立通信。但问题是同一台主机上会有许多程序都需要用网络收发数据，比如QQ和浏览器这两个程序都需要连接互联网并收发数据，我们如何区分某个数据包到底是归哪个程序的呢？也就是说，我们还需要一个参数，表示这个数据包到底供哪个程序（进程）使用。这个参数就叫做"端口"（port），它其实是每一个使用网卡的程序的编号。每个数据包都发到主机的特定端口，所以不同的程序就能取到自己所需要的数据。

"端口"是0到65535之间的一个整数，正好16个二进制位。0到1023的端口被系统占用，用户只能选用大于1023的端口。有了IP和端口我们就能实现唯一确定互联网上一个程序，进而实现网络间的程序通信。

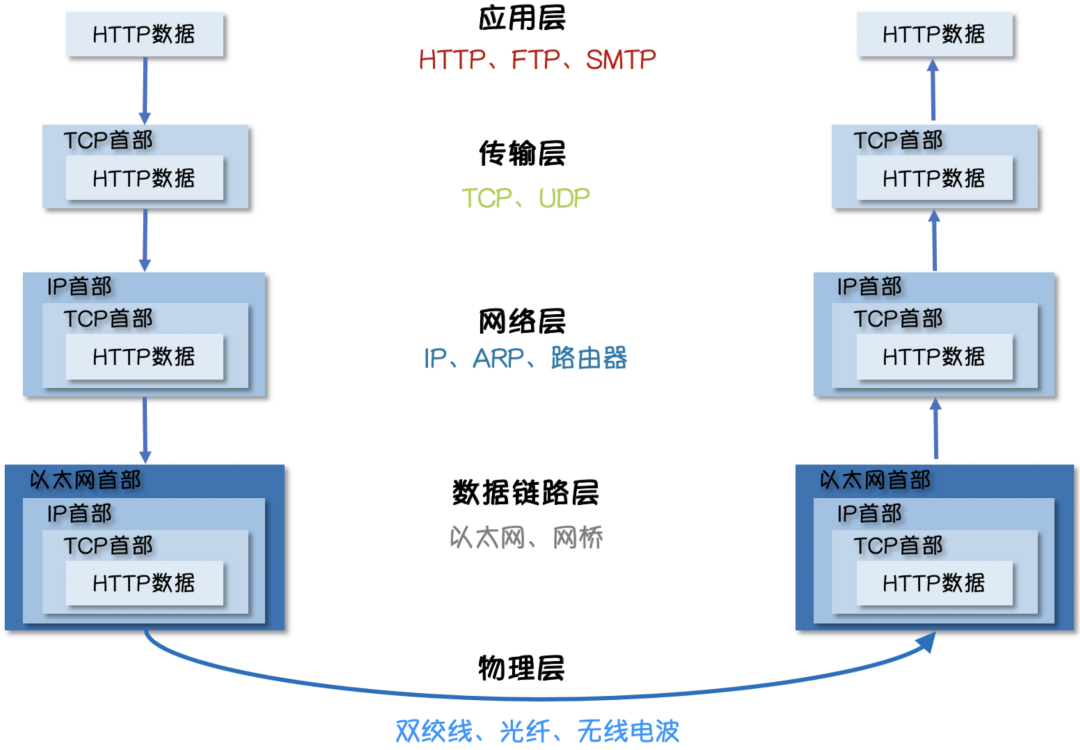
我们必须在数据包中加入端口信息，这就需要新的协议。最简单的实现叫做UDP协议，它的格式几乎就是在数据前面，加上端口号。UDP数据包，也是由"标头"和"数据"两部分组成："标头"部分主要定义了发出端口和接收端口，"数据"部分就是具体的内容。UDP数据包非常简单，"标头"部分一共只有8个字节，总长度不超过65,535字节，正好放进一个IP数据包。

UDP协议的优点是比较简单，容易实现，但是缺点是可靠性较差，一旦数据包发出，无法知道对方是否收到。为了解决这个问题，提高网络可靠性，TCP协议就诞生了。TCP协议能够确保数据不会遗失。它的缺点是过程复杂、实现困难、消耗较多的资源。TCP数据包没有长度限制，理论上可以无限长，但是为了保证网络的效率，通常TCP数据包的长度不会超过IP数据包的长度，以确保单个TCP数据包不必再分割。

应用层

应用程序收到"传输层"的数据，接下来就要对数据进行解包。由于互联网是开放架构，数据来源五花八门，必须事先规定好通信的数据格式，否则接收方根本无法获得真正发送的数据内容。"应用层"的作用就是规定应用程序使用的数据格式，例如我们TCP协议之上常见的Email、HTTP、FTP等协议，这些协议就组成了互联网协议的应用层。

如下图所示，发送方的HTTP数据经过互联网的传输过程中会依次添加各层协议的标头信息，接收方收到数据包之后再依次根据协议解包得到数据。


HTTP数据
TCP首部
HTTP数据
IP首部
TCP首部
HTTP数据
以太网首部
IP首部
TCP首部
HTTP数据
应用层
HTTP、FTP、SMTP
传输层
TCP、UDP
网络层
IP、ARP、路由器
数据链路层
以太网、网桥
物理层
双绞线、光纤、无线电波
HTTP数据
TCP首部
HTTP数据
IP首部
TCP首部
HTTP数据
以太网首部
IP首部
TCP首部
HTTP数据