

第十六章七层协议

一 客户端服务端架构

二 互联网协议介绍

三 什么是网络

四 osi七层协议

1. 物理层

2. 数据链路层

3. 网络层

4. 传输层

5. 应用层

本文是Python通用编程系列教程，已全部更新完成，实现的目标是从零基础开始到精通Python编程语言。本教程不是对Python的内容进行泛泛而谈，而是精细化，深入化的讲解，共5个阶段，25章内容。所以，需要有耐心的学习，才能真正有所收获。虽不涉及任何框架的使用，但是会对操作系统和网络通信进行全局的讲解，甚至会对一些开源模块和服务端进行重写。学完之后，你所收获的不仅仅是精通一门Python编程语言，而且具备快速学习其他编程语言的能力，无障碍阅读所有Python源码的能力和对计算机与网络的全面认识。对于零基础的小白来说，是入门计算机领域并精通一门编程语言的绝佳教材。对于有一定Python基础的童鞋，相信这套教程会让你的水平更上一层楼。

一 客户端服务端架构

我们通过前面的学习已经掌握了如何写程序，客户端和服务端无非都是代码写成的，这个技能我们现在已经掌握了，如果我们写的程序不需要联网就能完成独立的运行，那么这个程序肯定商业价值不高，所以下下来我们就要来学习的是通信，即客户端与服务端是如何基于网络来进行通信的

C/S架构指的就是客户端服务端架构，C指的是client，S指的是server。从硬件上来，打印机和电脑就是一个C/S架构，从软件上来，C/S架构随处可见，比如你看的小电影网站是服务端，你的浏览器是客户端，准确的说这是B/S（browser <==> server）架构，但是B/S架构严格来讲也是C/S的一种。常见C/S架构很多，我们常用的qq，微信等等，我们常用的所有的手机软件几乎都是，也就是说我们使用的是客户端，而服务端肯定是在这个公司的服务器上运行着的。那么我们相互之间发送消息的过程其实就是先由我们客户端发送到的服务端，再由服务端发送到接受这个消息的另外一个客户端。

二 互联网协议介绍

假如现在把我打电话的过程比作上网（电话都是有线电话），首先要先连接好这个电话线，这个电话线四通八达，共同组成了像一张网一样的东西，现在我要给姚明打电话要将汉语，要给吉诺比利打电话要讲俄语，要给希腊怪兽字母哥打电话要讲希腊语，要给不同的人打电话就要用不同的语言，这显然不现实。所以我们需要一个统一的标准，所有人都会说，所有人都能听的懂，这门语言就是英语。

我们要想实现通信，前面我们使用打电话来举例，对于计算机来说上网首先需要插一根网线，再通过交换机实现多次连接，我们把计算机与计算机中间的这些设备统称为物理连接介质（通信载体），然后才是通信，通信需要有一个统一的标准，计算机按照这个标准组织数据，发送到另外一台计算机上，另外一台计算机再按照这个标准反解出来数据，这就拿到了真正想要的数据，这个标准就是互联网协议。所以，互联网协议就相当于计算机届的英语。

学习网络编程，就是要编写一个客户端和服务端软件，然后实现客户端与服务端基于网络通信

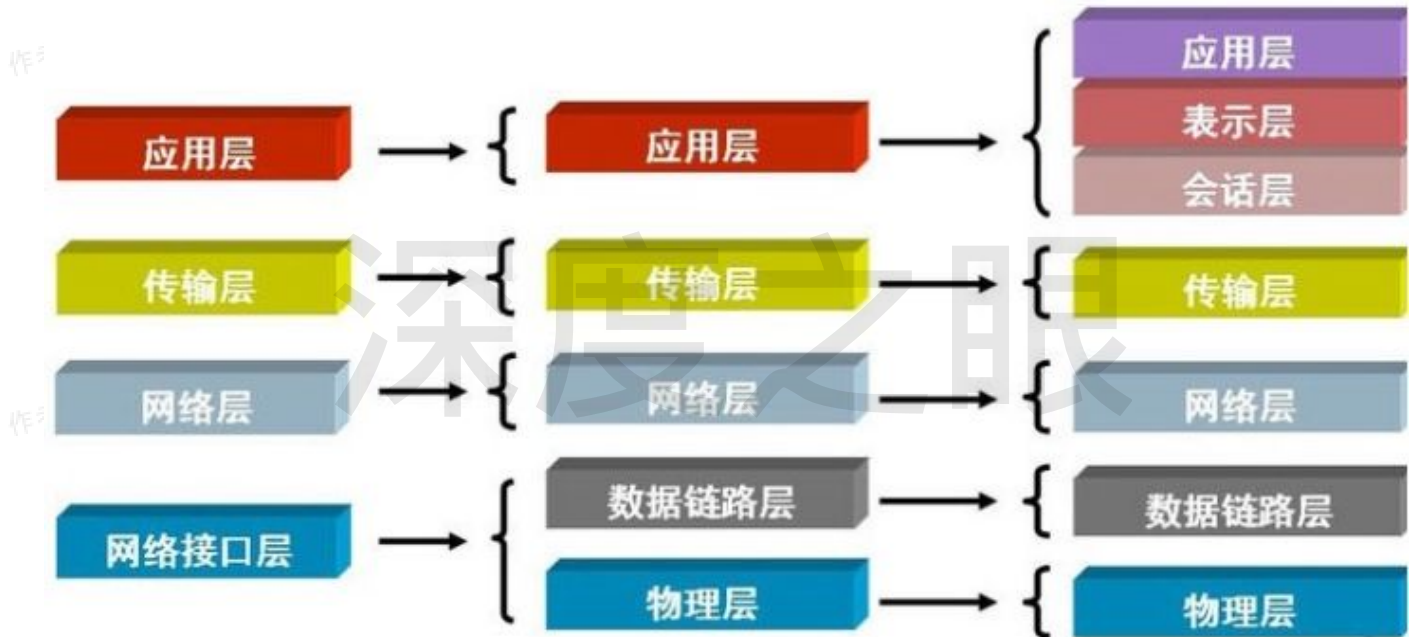
三 什么是网络

网络主要由两部分构成

1. 物理连接介质
2. 互联网协议

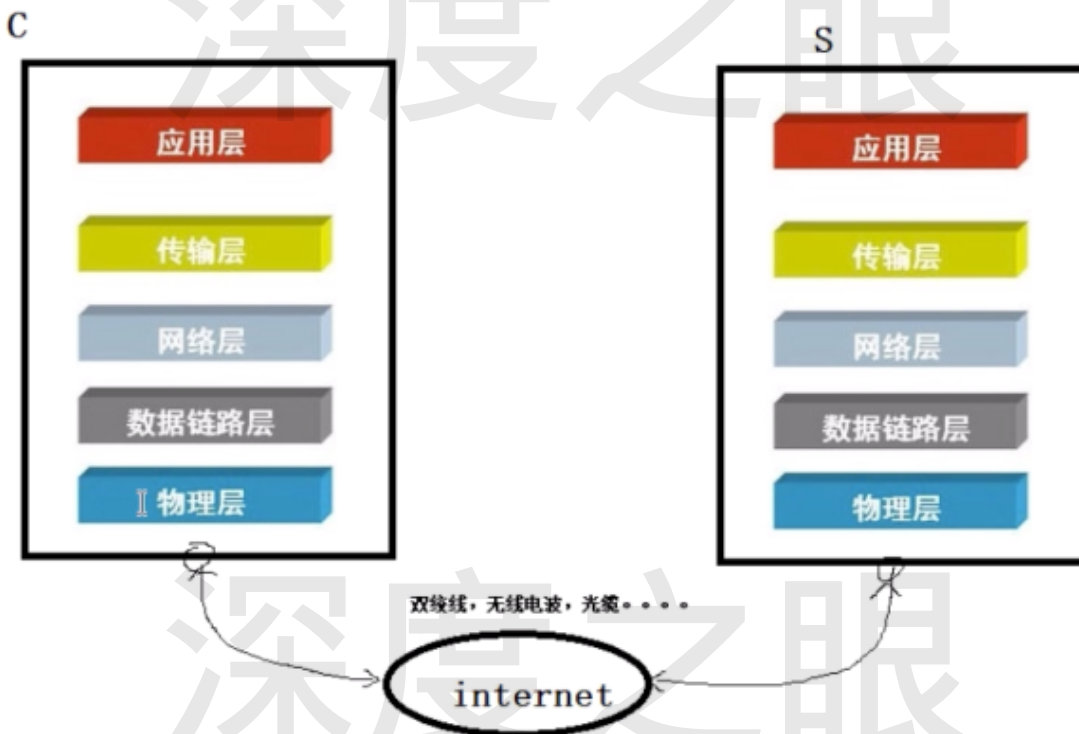
物理连接介质是由网络工程师来帮我们完成处理的，这个不需要我们操心，我们需要的是写程序，那就必须要清楚互联网协议，我们写的程序也要遵循这个协议。

四 osi七层协议



osi七层协议指的是应表会传网数物这七层，我们会把应用层，表示层和会话层统称为应用层，也可以把数据链路层和物理层统称为网络接口层。对于我们来说需要至少掌握五层协议：应用层，传输层，网络层，数据链路层和物理层。

既然互联网协议相当于是计算机届的英语，我们学习英语的时候都要先学字母，再学单词，再学句子，在学文章。那么计算机在学习互联网协议的时候也应该是一点一点循序渐进学习的，物理层==>数据链路层==>网络层==>传输层==>应用层。



1. 物理层

网线连接在客户端计算机上，其实是连接在了计算机的一个叫做网卡的设备上，网卡是专门负责与外界通信的。网线一般是双绞线或者光缆，也可以使用无线电波，中间经过交换机，路由器，防火墙等等一堆设备统称为物理连接介质，可以理解为经过互联网，再连接到服务端设备。首先工作的是物理层，发送电信号，电信号分为两种，高电平和低电平，高电平可以被定义成数字 1，低电平可以被定义成数字 0。假如我客户端发送一个 0010101100，服务端相应的就会收到这些数字。但是单纯的一段二进制数字这是没有意义的，一定要明确，从哪里开始到哪里结束这表示一段内容，从哪里开始到哪里结束这又表示另外一段内容。这也就是说，我们要给这些二进制数字进行分组，这不是物理层干的活，接下来就到了数据链路层。

2. 数据链路层

数据链路层的分组也应该按照一个统一的标准，早期的时候并没有形成统一的标准，每个公司有自己的标准，那么导致的结果就是天下大乱。后来形成了一个统一的标准，这个标准就叫做Ethernet（以太网协议）。以太网协议规定：

- 一组电信号构成一个数据报，叫做‘帧’
- 每一数据帧分成：报头head和数据data两部分
这一点和我们写信类似，有信封，有信的内容，信封上面会写明这封信的发送者接受者分别是谁，信里面的信纸上写的就是信的内容。
head包含：（固定18个字节）
 - 发送者 / 源地址，6个字节
 - 接收者 / 目标地址，6个字节
 - 数据类型，6个字节

data包含：（最短46字节，最长1500字节）

注意：头固定长度18个字节，也只有固定长度，接收者才知道按照什么标准来读取

- 数据报的具体内容

head长度 + data长度 = 最短64字节，最长1518字节，超过最大限制就分片发送

以太网规定head里面要有发送者的源地址和接受者的目标地址，源地址可以理解为是发送者的家，目标地址就是接收者的家。那么，在计算机中如何标识家在哪里？

使用mac地址，注意这个mac地址不是你用的苹果电脑那个mac，只是巧合同名了。MAC地址（Media Access Control Address），直译为媒体访问控制地址，也称为局域网地址（LAN Address），以太网地址（Ethernet Address）或物理地址（Physical Address），它是一个用来确认网上设备位置的地址。

mac地址是计算机上一个唯一的地址，是在计算机的网卡上的，每块网卡出厂时都被烧制上一个世界唯一的mac地址，长度为48位2进制，通常由12位16进制数表示（前六位是厂商编号，后六位是流水线

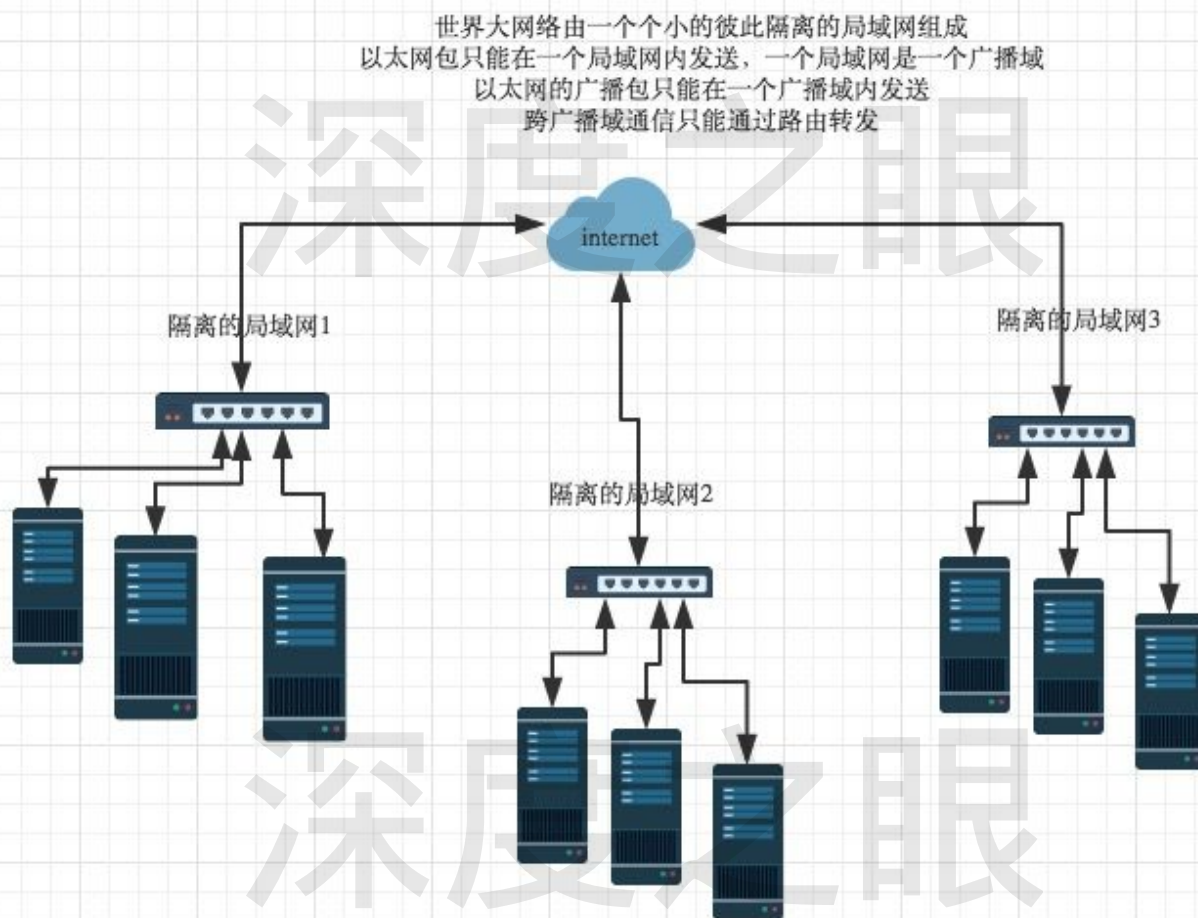
号)

这样做的目的就是要保证每一个mac地址是全世界独一无二的。

有使用过Linux系统的同学，可以看到计算机的网卡文件名（Linux一切皆文件）叫做eth0，eth1，eth2，这个eth就是以太网前三个字母。

计算机的通信方式是广播，通俗的理解就是 吼，计算机都是瞎的，比如唐僧和孙悟空通信，就喊一声，“悟空，你在哪，你快来救我啊”，从理论上讲，只要喊的声音足够大，孙悟空就能听到，根据称呼语 悟空，孙悟空就能知道这是再叫他，收到的数据内容是 去救他师傅。唐僧被妖怪抓了吼一嗓子，八戒被抓了也吼一嗓子，沙僧被抓了也吼一嗓子，全世界的人都在吼，你能否想象一下这个数据量有多大。这个数据量甚至会比 **广播风暴** 还要恐怖。

从理论上讲，这种广播的通信方式可以实现，但是并不现实，一方面因为数据量太大，另外一方面范围太广，所以计算机的通信还需要局域网才能完成。我们的一个办公室可以理解一个局域网，一个局域网可以理解为一个广播域。我在这个办公室吼一下，隔壁的办公室听不到，这个就是一个广播域，数据量不会很大。所以全世界的网络并不是我们之前想象的那样是一张大网，而是一个个的小网组成成的。



假如现在微信的客户端的机器就是我的手机，微信的服务端的机器就在我所在的办公室，那么我们基于以太网的工作方式通过mac地址就能实现网络通信，但问题是这不可能。如果两台机器不在同一个局域网内，我们必须首先通过一种方式定位到他所在的局域网，再来根据以太网协议根据mac地址找到这台机器，这种方式就不是数据链路层所能做到的了，下一层我们进入网络层。

3. 网络层

网络层有一个IP协议，我们常说的IPV4就是IP协议的第四个版本，IPV6就是IP协议的第六个版本。IP协议数据报规定：

ip数据报也分为head和data部分，无须为ip报定义单独的栏位，直接放入以太网报的data部分

head：长度为20到60字节（源ip地址，目标ip地址）

data：最长为65,515字节。

IP地址用来定外在那个局域网内，mac地址用来定位局域网的哪一台机器，
所以ip地址+mac地址就能定位到全世界范围内独一无二的一台机器

现在能够定位到了一套机器，再来想一下我们本质上要做的事情，是一台机器上的软件，产生了数据，再来由这个软件来通过操作系统调硬件，最后由机器上的网卡通过我们上面讲的协议发送出去。对方接收的时候则相反 对方网卡接收==>操作系统解包==>服务端软件拿到数据，所以本质的通信其实就是应用软件与应用软件之间的通信。

两个软件要想实现通信其实是依靠的计算机这个载体才能完成的，但是计算机可不只有一个软件，我们必须要有有一种方式标识一个软件具体在哪里，这时候以太网协议和IP协议很明显都解决不了，这就进入了我们的下一层：传输层

4. 传输层

传输层有一个TCP协议和UDP协议，这两个协议都是基于端口工作的协议。

端口就是一个数字，比如：现在你定位到我在中海国际中心这个写字楼办公，根据这个写字楼你要找到我，还需要一个门牌号，那么端口号指得就是具体的哪一个门牌号。

所以你的电脑上面所有的基于网络通信的软件都有一个端口号，手机也一样。

补充：客户端和服务端都要遵循以太网协议，ip协议和TCP/UDP协议，那么我们先想一下，什么是服务，你开一个饭店这个就是服务，饭店的地址肯定不能三天两头的换地方，否则你的客户就找不到你了，同理，服务端的地址也是一样的，需要有一个固定的地址。

TCP/UDP协议是我们下一章要讲的内容，现在我们就到了最后一层：应用层

5. 应用层

应用层就是应用软件，应用软件是你写的，这个标准可以由你来定，当然了你也可以遵循一些大家已经定制好了的应用层协议的标准，常见的有 http, mail, ftp, 这些应用层的协议，你写或者不写，他就在那里，不喜不悲。

总结：

发送过程：

应用层软件（有协议或无协议）=>传输层（TCP/UDP协议）=>网络层（ip协议）=>数据链路层（以太网协议）=>物理层==>电信号发送（100011001010110）

接收过程则相反

应用层有协议或者无协议无关紧要，到了传输层后就会在外面包一层TCP/UDP协议，这时候就是TCP/UDP协议暴露在外面，到了网络层包一层ip协议以此类推，解包的时候则相反，先从最外层解包。