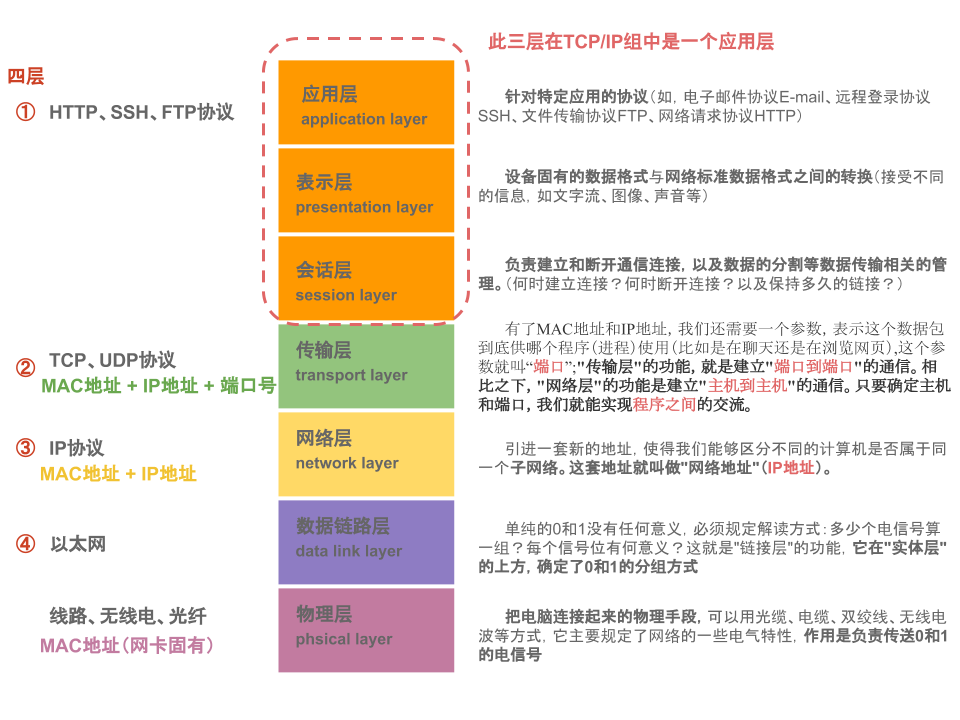
自定义控件（尚学谷）

一.七层？四层？

[OSI模型](https://link.jianshu.com?t=https:/zh.wikipedia.org/wiki/OSI%E6%A8%A1%E5%9E%8B)（Open System Interconnection Reference Model，缩写为OSI）,全名**“开放式系统互联通信参考模型”**，是一个试图使各种计算机在全世界范围内互联为网络的标准框架。1983年，国际标准组织（ISO）发布了著名的ISO/IEC 7498标准，它定义了网络互联的7层框架，也就是开放式系统互联参考模型。（链接需自搭梯子）



OSI七层模型.png

1.为什么需要协议？

什么是协议（protocol）？**通俗的来讲，协议是一种双方都明白或者必须遵守的（事先）约定**，比如说长城上放狼烟，是因为人们已经预先设定好狼烟这个物理信号代表了“敌人入侵”这一抽象信号。这样一个“狼烟=敌人入侵”就是一个简单的协议。协议可以更复杂，比如摩尔斯码(Morse Code)，使用短信号和长信号的组合，来代表不同的英文字母。  
  同样，计算机之间的通信也要遵循不同层次的协议，来实现计算机的通信。早期的计算机网络，都是由各厂商自己规定一套协议，IBM，Apple，和MicroSoft都有自己的网络协议，比如MicroSoft的两台电脑用网线连起来，互相说话能听懂。但是MicroSoft和Apple的电脑连接起来说话就听不懂了，想想你和我微信聊天，我是MicroSoft电脑，你是Apple电脑，你发送的消息到我这里显示不了或者解析成另一个意思，这样通讯就不能进行了（通过上面的图我们可以看到，表示层就是**消除不同设备之间固有数据格式差异的**）。  
  为了把全世界的所有不同类型的计算机都连接起来，就必须规定一套全球通用的协议，为了实现这个目标，**互联网协议簇**（Internet Protocol Suite）就成为了通用协议标准。互联网协议包含了上百种协议，但是最重要的两个协议是TCP和IP协议，而我们**通常把基于TCP和IP协议的所有协议统称为”TCP/IP协议（蔟）**”。

2.OSI七层模型是干什么的？

**互联网的实现，分成好几层，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**。我们在上图中已经大致标出了每一层的功能。OSI模型就是这样的一个分层，它是一个由国际标准化组织􏰁提出的概念模型,试图提供一个**使各种不同的计算机和网络在世界范围内实现互联的标准框架**。它将计算机网络体系结构划分为七层，每层都可以提供抽象良好的接口。

3.TCP/IP四层（参考）模型

TCP/IP和OSI模型组并不能精确的匹配，但是我们可以尽可能的参考OSI模型并在其中找到TCP/IP的对应位置。如上图所示，我们已经标出了TCP/IP对应的四层位置所在。**通常人们认为OSI模型最上面三层（应用层、表示层、会话层）在TCP/IP中是一个应用层**。由于TCP/IP有一个相对比较弱的会话层，由TCP和RTP下的打开和关闭连接组成，并在TCP/UDP下的各种应用提供不同的端口号，这些功能被单个的应用程序添加。

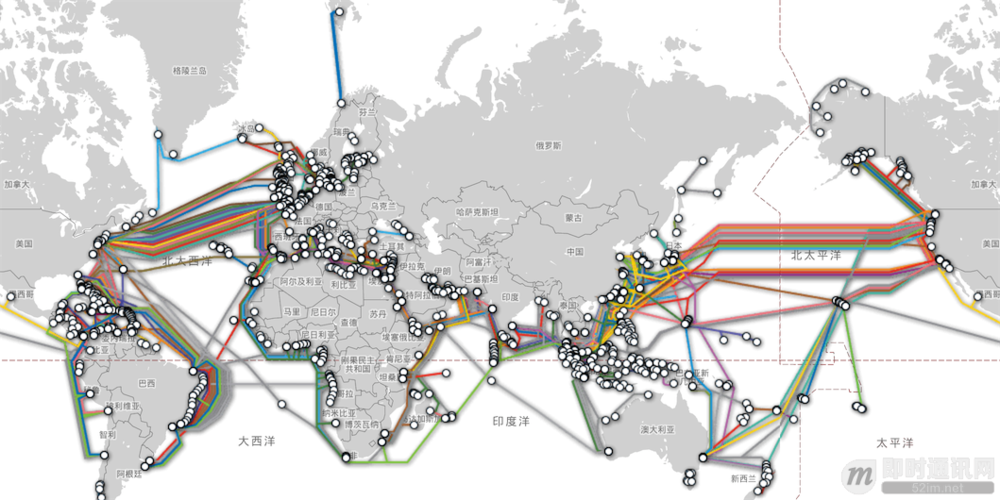
4.TCP/IP（参考）模型与OSI七层模型有什么异同？

前面我们说过，TCP/IP协议是**互联网协议（簇）**的统称，他是互联网标准通信的基础，它**提供点对点的链接机制，将数据应该如何封装、定址、传输、路由以及在目的地如何接收，都加以标准化**。而OSI模型是**开放式系统互联通信参考模型**——笔者的理解是：  
  OSI是一个完整的、完善的宏观模型，他包括了硬件层（物理层），当然也包含了很多上面途中没有列出的协议(比如DNS解析协议等)；而TCP/IP（参考）模型，更加侧重的是互联网通信核心（也是就是围绕TCP/IP协议展开的一系列通信协议）的分层，因此它不包括物理层，以及其他一些不想干的协议；其次，之所以说他是参考模型，是因为他本身也是OSI模型中的一部分，因此参考OSI模型对其分层。

二.自底向上的网络分层

1.物理层

电脑要组网，第一件事要干什么？当然是先把电脑连起来，可以用光缆、电缆、双绞线、**无线电波（WiFi）**等方式。**物理层的作用就是通过物理手段把电脑连接起来，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。**  
  这里说一下，通过**物理手段**将设备连接起来组网，物理手段就是光缆、电缆、双绞线、**无线电波（WiFi）**等，比如中美之间的网络通信是通过海底光缆；两个不同的局域网（电信的网络和移动的网络）通讯，嗯，稍微麻烦点，我的电信手机先连上电信的服务器，你的移动手机连移动服务器，他们两个ISP（Internet Service Provider 互联网服务提供商）之间是通过物理手段链接的，这样我们就能够间接的实现通讯了。下图是2015年全球互联网跨国通信光缆的连接情况：



2015年全球互联网跨国通信光缆的连接情况.png

可以看到，图中各个密密麻麻的线就是各个国家的链接情况，所以互联网可以说就是**用物理设备将各个“局域网”相连组成的更大的“局域网“，更大局域网层层相连，最终就组成了”互联网“**，比如小的互联网就是你家和我家的WiFi,一个省的每户家庭的WiFi组成这个省的局域网，各个省的局域网组成中国的局域网，各个国家之间的局域网通过物理手段互联就组成了横跨世界的**”互联网“**（当然天朝还有一堵墙）。

2.数据链路层

①.定义

物理层就是传输电路的0和1信号的，但是单纯的0和1没有意义，必须规定解读方式：多少个0和1算一组？每个信号有什么意义？——**这就是链路层的意义，它在物理层的上方，确定了0和1的分组方式。**

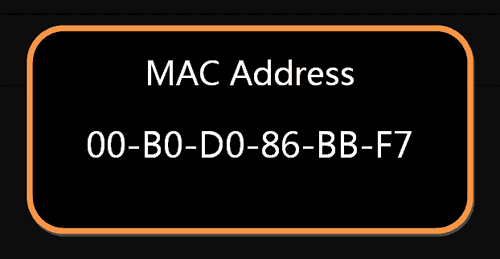
②.以太网协议

早些时候，各个公司都有自己的电信号分组方式，后来出现了**“以太网”**这种协议逐渐占据了主导的地位。**“以太网”**规定，**一组电信号构成一个数据包，叫做“帧（Frame）”；****每一帧分成两个个部分：标头（Head）和数据（Data）。**  
  因此，数据链路层链路层的数据包就叫“以太网数据包”，他由“标头”和“数据”两部分组成——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。**

③.MAC地址

上面我们提到，以太网数据包的“标头”包含了发送者和接受者的信息，那么，发送者和接受者是如何标识的呢？

* **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受地址，也叫MAC地址。**
* **每块网卡出厂的时候，都有全世界独一无二的MAC地址，长度是48位的二进制，通常用12个十六进制数表示。**
* **前6个十六进制是厂商编号，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。



MAC地址.png

④.广播

定义地址只是第一步，后面还有更多步骤——首先，一块网卡怎么只带另一块网卡的MAC地址？回答是**有一种ARP协议**，可以解决这个问题。这个留到后面介绍，这里只需要知道，**以太网数据包必须知道接收方的MAC地址，然后才能发送。**  
  其次，就算有了MAC地址，系统怎样才能把数据包准确送到接收方？回答是以太网采用了一种很"原始"的广播式的方式，**它不是把数据包准确送到接收方**，**而是向本网络（局域网）内所有计算机发送，让每台计算机自己判断，是否为接收方**。  
  一台计算机向本局域网内的所有电脑均发送相同的数据包，其他计算机收到这个数据包之后，会读取这个数据包的“标头”，找到其中接收方（目标方）的MAC地址，然后与自身的MAC地址进行比对，如果两者相同，说明就是要发给自己的，然后接受这个包并做出进一步的处理，否则丢弃这个包。**这种发送方式就叫“广播”，主要通过分组交换机或者网络交换机进行。**

以上总结：

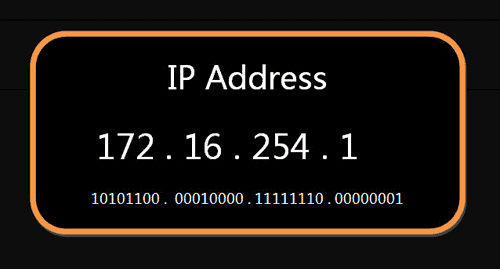
1. 确定接受方和接收方地址：以太网协议规定，在网络上中的设备都必须有网卡，每个 又有唯一的Mac地址（48位二进制）。
2. 保证接受方确定收到（广播）：在局域网中，向所有设备发送数据包，让设备自己确定是否为接受方。接收方收到数据包后，读取标头中的Mac地址，如果一直则处理。------通过分组交换机或

3.网络层

根据上面的讲解，理论上依靠MAC地址和广播技术，上海的网卡发出的数据包就可以找到洛杉矶网卡了——但是如果全世界的计算机都这么干，那么每一台计算机发出的数据包都同步广播到全世界其他电脑，再一一比对判断，这样显然是低效、不现实的。  
  因此，上面我们强调，广播是在发送者所在的局域网内广播的，不同也就是说，如果两台计算机没有在同一个子网（局域网）内，是无法通过广播直接传过去的。前面我们说过，互联网是由一个个子网组成的更大的子网，一级一级组网，最终构成的互联网。  
  因此我们必须找到一种方法，区分哪些MAC地址属于同一个子网。如果是同一个子网就采用广播的形式，如果不是，则采用“路由”的方式（后面会讲）发送——这就导致了**网络层的出现，他的作用是引入一套新的地址，使我们能够区分哪些计算机属于同一个子网，这个套机制就叫做“网络地址”，也就是“IP地址”**。

①.IP协议

规定网络地址的协议，叫IP协议。他定义的地址，就叫做“IP地址”。IP地址目前有IPV4(Internet Protocol version 4，IPv4)和IPV6(Internet Protocol version 4，IPv6)两版，又称“互联网通信协议第四/六版”。2011年，IANA IPv4 pool地址完全用尽时，IPv6仍处在部署的初期，**因此IPV4地址也是目前最为广泛的IP地址——这个版本规定，网络地址由32个二进制位组成,习惯上，我们分成四段十进制数表示IPV4地址，从0.0.0.0到255.255.255.255。**



IPV4.png

**互联网上的每一台计算机，都会被分配到一个IP地址，这个地址由两部分组成，前一部分代表网络，后一部分代表主机（又称终端系统，end system）。**比如，IP地址172.16.254.1，这是一个32位的地址，假定它的网络部分是前24位（172.16.254），那么主机部分就是后8位（最后的那个1）。**处于同一个子网络的电脑，它们IP地址的网络部分必定是相同的，也就是说172.16.254.2应该与172.16.254.1处在同一个子网络，**而后面的“2”与“1”则是同一子网内两台不同电脑（主机）的编号。

②.子网掩码

问题在于单单从IP地址，我们无法判断网络部分。还是以172.16.254.1为例，它的网络部分，到底是前24位，还是前16位，甚至前28位，从IP地址上是看不出来的。那么，怎样才能从IP地址，判断两台计算机是否属于同一个子网络呢？这就要用到另一个参数**"子网掩码"（subnet mask）。**  
  **所谓"子网掩码"，就是表示子网络特征的一个参数。它在形式上等同于IP地址，也是一个32位二进制数字，它的网络部分全部为1，主机部分全部为0。**比如，IP地址172.16.254.1，如果已知网络部分是前24位，主机部分是后8位，那么子网络掩码就是11111111.11111111.11111111.00000000，写成十进制就是255.255.255.0。  
  知道"子网掩码"，我们就能判断，任意两个IP地址是否处在同一个子网络。方法是**将两个IP地址与子网掩码分别进行AND运算（两个数位都为1，运算结果为1，否则为0），然后比较结果是否相同，如果是的话，就表明它们在同一个子网络中，否则就不是。**

③.路由/路由器/网关/交换机

上面我们已经确定了两台计算机是否遭同一个子网中，如果在，则采用广播+MAC寻址的的方式发送数据包，如果不是，则要采用“路由”的方式了，那么什么是“路由”呢？

[路由（routing）](https://link.jianshu.com?t=https:/zh.wikipedia.org/wiki/%E8%B7%AF%E7%94%B1)

**就是通过互联的网络把信息从源地址传输到目的地址的活动。路由引导分组转送，经过一些中间的节点后，到它们最后的目的地。**

从“路由”的定义中可以看到，“路由”是一种活动，一种动作，一种行为，作用是是将信息从原地址传输到目的地址，**比较特殊的是，原地址和目标地址是在两个不同的子网中的**。那么如何传输呢？路由定义一条路径，经过因特网发送包到另一网络上的地址，但路由不定义完全路径，只定义从主机到可以将包转发到目的地的网关（子网）间的路径段（或从一个子网到另一个子网）。

[路由器（英语：Router）](https://link.jianshu.com?t=https:/zh.wikipedia.org/wiki/%E8%B7%AF%E7%94%B1%E5%99%A8)

**简单理解就是实现路由功能的机器。路由器连接两个或多个网络并提供路由功能。**

前面我们说过，网卡是计算机的一个硬件，它在接收到网路信息之后，将信息交给计算机。当计算机需要发送信息的时候，也要通过网卡发送。**一台计算机可以有不只一个网卡，比如笔记本就有一个以太网卡和一个WiFi网卡。**计算机在接收或者发送信息的时候，要先决定想要通过哪个网卡。**路由器(router)可以通俗理解为一台配备有多个网卡的专用电脑，它让网卡接入到不同的网络中。**

[网关（英语：Gateway）](https://link.jianshu.com?t=https:/zh.wikipedia.org/wiki/%E7%BD%91%E5%85%B3)

是路由器的一种，通常我们把**网络层使用的路由器称为网关**，路由器可以在网络接口级或物理级路由；网关是在网络层上路由（个人感觉应该是一种概念，即在网络层连节两个子网的概念，并不存在实体，真正实现路由功能还是得靠路由器）。  
  说的再通俗一点，路由器上面有MAC地址和MAC地址对应的IP，而网关由于是网络层的概念因此只有IP地址。在今天很多局域网采用都是路由来接入网络，因此**现在通常指的网关就是路由器的IP。**  
  另外，需要强调一点，虽然路由器上面有MAC地址和IP地址，但**它并不能通过MAC地址工作，必须通过IP寻址。因此它是工作在网络层的设备。**

[网络交换机（英语：Network switch）](https://link.jianshu.com?t=https:/zh.wikipedia.org/wiki/%E7%B6%B2%E8%B7%AF%E4%BA%A4%E6%8F%9B%E5%99%A8)

是一个扩大网络的器材，能为子网中提供更多的连接端口，以便连接更多的电脑。交换机与路由器的区别：

* **工作层次不同**  
  交换机主要工作在数据链路层（第二层）  
  路由器工作在网络层（第三层）。
* **转发依据不同**  
  交换机转发所依据的对象是：MAC地址。（物理地址）  
  路由转发所依据的对象是：IP地址。（网络地址）
* **主要功能不同**  
  交换机主要用于组建局域网，连接同属于一个(广播域)子网的所有设备，负责子网内部通信（广播）。  
  路由主要功能是将由交换机组好的局域网相互连接起来，或者将他们接入Internet。  
  交换机能做的，路由都能做。  
  交换机不能分割广播域（子网），路由可以。  
  路由还可以提供防火墙的功能。  
  路由配置比交换机复杂。

这里我们还是需要说明一点：

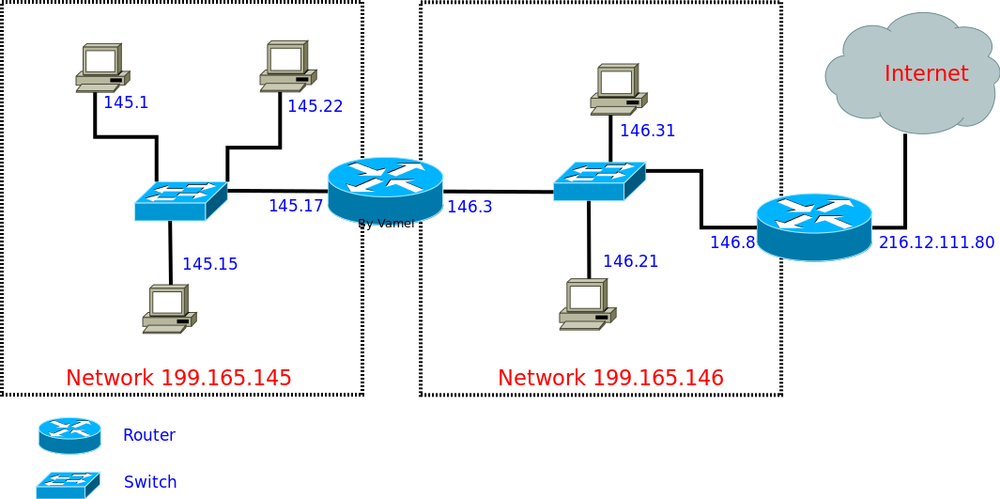
* 交换机虽然主要依靠MAC地址查找工作在数据链路层，但是他也可以有IP地址，这样就可以进行远程登录等操作了。
* **LAN**，全称Local Area Network，中文名叫做**局域网**；**WAN**，全称Wide Area Network，中文名叫做**广域网**。WAN是一种跨越大的、地域性的计算机网络的集合。通常跨越省、市，甚至一个国家。广域网包括大大小小不同的子网，子网可以是局域网，也可以是小型的广域网；**WLAN**，全称Wireless LAN, **无线局域网**，通俗点讲就是WiFi。
* 家用的路由器，一般包括了交换机和路由器，因此他有两个接口——WAN端口用于连接至Internet；LAN端口用于连接至局域网设备。



LAN、WAN、WLAN的区别.png

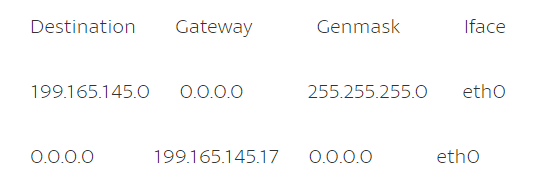
举个栗子

上面说了这么多了，我们来举个例子——比如下图中位于中间位置的路由器有两个接口IP，地址分别为199.165.145.15和199.165.146.3。它们分别接入到两个网络：199.165.145和199.165.146。



网关.png

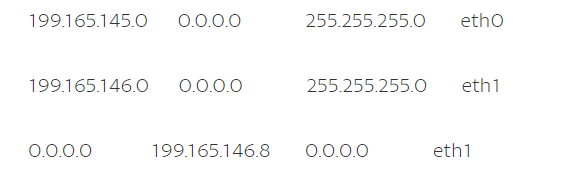
**显然，199.165.145和199.165.146是两个不同的子网，他们通过中间的路由器连接节，这个路由器有两个网卡——199.165.145.17和199.165.146.3。**  
  现在考虑一种情况，我们从主机145.17生成发送到146.21的IP包:  
**第一步：**  
  先写好数据包的标头，即**写清楚发送者的IP地址（199.165.145.15）和接受者的IP地址（199.165.146.21）**，145.15会参照自己的路由表（routing table），里面有两行记录（当然实际的路由表肯定远远超过两条记录）：



主机路由表.png

第一行表示，如果IP目的地是199.165.145.0这个网络中的主机，那么说明是在同一个子网中，只需要用自己在eth0上的网卡（MAC地址）通过交换机直接传送，不需要前往router(Gateway 0.0.0.0 = “本地送信”)。  
  第二行表示所有不符合第一行的IP目的地，则应该送往送往Gateway 199.165.145.17这个主机，也就是中间router接入在eth0的网卡IP地址。  
  我们的IP包目的地为199.165.146.21，不符合第一行，所以按照第二行，发送到中间的router。主机145.15会在数据包的头部写上199.165.145.17对应的MAC地址，这样，就在199.165.145这个局域网中通过交换机（通过广播MAC地址）广播到199.165.145.17对应的主机（路由器）。

第二步：  
  中间的router在收到IP包之后(，提取目的地IP地址，然后对照自己的routing table：



路由器路由表.png

从前两行我们看到，由于router横跨eth0和eth1两个网络，它可以直接通过eth0和eth1上的网卡直接传送IP包。第三行表示，如果是前面两行之外的IP地址，则需要通过eth1，送往199.165.146.8(右边的router接口IP)。我们的目的地符合第二行，所以将IP放入一个新的帧中，在帧的头部写上199.165.146.21的MAC地址，通过199.165.146网中的交换机广播发往主机146.21。

IP包可以进一步接力，到达更远的主机。IP包从主机出发，根据沿途路由器的routing table指导，在router间接力。**IP包最终到达某个router，这个router与目标主机位于一个局域网中，可以直接建立数据链路层的（广播）通信**。最后，IP包被送到目标主机。这样一个过程叫做routing(我们就叫IP包接力好了，路由这个词实在是混合了太多的意思)。  
  整个过程中，IP包不断被主机和路由封装入帧(信封)并拆开，然后借助连接层，在局域网的各个网卡之间传送帧。整个过程中，我们的IP包的内容保持完整，没有发生变化。最终的效果是一个IP包从一个主机传送到另一个主机。**利用IP包，我们不需要去操心底层(比如数据链路层)发生了什么**。

④.ARP协议

在上面的过程中，我们实际上假设了，每一台主机和路由都能了解局域网内的IP地址和MAC地址的对应关系，这是实现IP包封装(encapsulation)到帧的基本条件。I**P地址与MAC地址的对应是通过ARP协议传播到局域网的每个主机和路由。每一台主机或路由中都有一个ARP cache，用以存储局域网内IP地址和MAC地址如何对应**。

ARP协议(**ARP介于数据链路层和网络层之间，ARP包需要包裹在一个帧中**)的工作方式如下：**主机发出一个ARP包，该ARP包中包含有自己的IP地址和MAC地址。通过ARP包，主机以广播的形式询问局域网上所有的主机和路由：我是IP地址xxxx，我的MAC地址是xxxx，有人知道199.165.146.4的MAC地址吗？拥有该IP地址的主机会回复发出请求的主机：哦，我知道，这个IP地址属于我的一个NIC（网卡），它的MAC地址是xxxxxx。由于发送ARP请求的主机采取的是广播形式，并附带有自己的IP地址和MAC地址，其他的主机和路由会同时检查自己的ARP cache，如果不符合，则更新自己的ARP cache。**

这样，经过几次ARP请求之后，ARP cache会达到稳定。如果局域网上设备发生变动，ARP重复上面过程。**ARP协议只用于IPv4。IPv6使用Neighbor Discovery Protocol来替代ARP的功能。**

4.传输层

①.端口号

  有了MAC地址和IP地址，我们已经可以在互联网上的任意两台电脑之间建立通信了。接下来的问题是，同一台主机上许多程序（进程）都需要用到网络，比如你一遍浏览网页一遍聊天。**当一个数据包从网上发送过来的时候，我们需要一个参数来区分，他到底是提供哪个进程使用的——这个参数就叫做“端口号”，他其实就是每一个使用网卡的程序的编号。**每个数据包发送到主机特定的端口，所以不同的程序就能取到自己想要的数据包。

**端口是0到65535之间的一个整数，正好16个二进制。0~1023的端口被系统占用，用户只能使用大于1023的端口。不管是浏览网页还是聊天，应用程序都会随机选用一个端口，然后与服务器简历相应的端口关系。这里需要补充一点，HTTP协议默认使用80端口，8080是用来访问代理服务的。**

**“传输层”的功能，就是建立“端口到端口”之间的通信。相比之下，“网络层”的功能是建立“主机到主机"的通信。**只要确定主机和端口号，我们就能实现程序之间的交流。

②.Socket

  上面我们已经说了，传输层是建立“端口到端口”之间的通信，更具体一点，也就是程序和程序之间的通信，或者**“进程间通信”**。嗯，挺唬人的一个概念。  
  进程间通信分为两种——一种是**主机内部（或终端内部）进程间通信**，这个由终端或主机上的操作系统决定，比如在Android系统上面进程间通信就是**AIDL**；另一种是**跨主机进程间通信或者网络进程间通信**，也叫**“socket通信”**。我们可以先笼统的理解——Unix系统把主机+端口，叫做"套接字"（socket），当然，这样说是有失偏颇的。

**那么，Socket是什么？**

* **从编程语言的角度，socket是一个无符号整型变量，用来标识一个通信进程**。两个进程通信，总要知道这几个信息：双方的ip地址和端口号，通信所采用的协议栈。socket就是和这些东西绑定的，实现socket可以使用unix提供的接口，也可以使用wIndows提供的winSock。
* **socket本质是编程接口(API)，对TCP/IP的封装。TCP/IP只是一个协议栈，必须要具体实现，同时还要提供对外的操作接口（API），这就是Socket接口**。通过Socket,我们才能使用TCP/IP协议，因此有了一系列我们知道的函数接口——connect、accept、send、read、write等。

  JDK的java.net包下有两个类：Socket和ServerSocket，在Client和Server建立连接成功后，两端都会产生一个Socket实例，操作这个实例，完成所需的会话，而程序员就通过这些API进行网络编程。 Socket连接过程分为三个步骤：服务器监听，客户端请求，连接确认。

③.UDP/TCP协议

  UDP和TCP协议都是传输层的协议，他们的主要作用就是在应用层的数据包标头加上端口号（或者在IP协议的数据包中插入端口号）。

* 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。
* TCP协议可以近似认为是有确认机制的UDP协议。每发出一个数据包都要求确认。如果有一个数据包遗失，就收不到确认，发出方就知道有必要重发这个数据包了。

  TCP协议主要的确认机制是**"三次握手，四次挥手"**,由于这个协议非常复杂，我们会另起一篇文章详细讲解。

5.应用层

  应用程序收到"传输层"的数据，接下来就要进行解读。由于互联网是开放架构，数据来源五花八门，必须事先规定好格式，否则根本无法解读。**"应用层"的作用，就是规定应用程序的数据格式。**举例来说，TCP协议可以为各种各样的程序传递数据，比如Email、WWW、FTP等等。那么，必须有不同协议规定电子邮件、网页、FTP数据的格式，这些应用程序协议就构成了"应用层"。这是最高的一层，直接面对用户。

  这里的引用层是文章开头的OSI七层模型的最上面三层的综合，因为是直接面向用户，因此它的主要作用是**“消除设备固有数据格式和网络标准数据格式直接的差异”**，因为在网络流中，数据的格式是标准化的，但是具体到不同得设备，不同的操作系统上，他的要求数据呈现格式是不同的，因此需要转化成统一的、用户能够感知的声音、图片、文字等信息，这就是应用层做的事情。

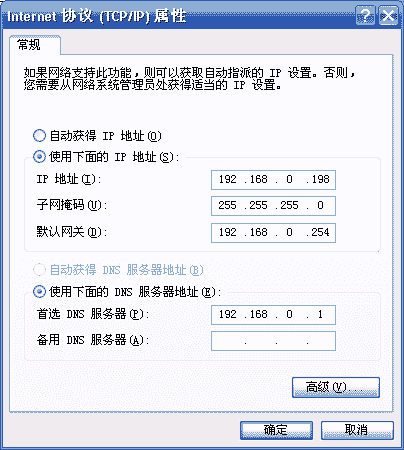
  具体的例子，也就是我们熟知的Android编程，也就是应用层编程（嗯，笔者是做Android的）。在Android系统中，应用层网络编程类是**HttpURLConnection**和**HttpClient**（后者已经被google弃用）。具体可参照笔者的另一篇文章[HTTP协议详解与Android相关基础网络编程](https://www.jianshu.com/p/3ea1bf1ba915" \t "_blank)。

①.用户的上网设置

  你买了一台新电脑，插上网线，开机，这时电脑能够上网吗？通常你必须做一些设置。有时，管理员（或者ISP）会告诉你下面四个参数，你把它们填入操作系统，计算机就能连上网了：

* **本机的IP地址**
* **子网掩码**
* **网关的IP地址**
* **DNS的IP地址**

下图是Windows系统的设置窗口：



四个必要参数.png

根据上面的讲解，我们应该知道这四个参数的必要性，只有设置了这些我们才能上网。

②.DNS解析

[网域名称系统（英文：Domain Name System，缩写：DNS）](https://link.jianshu.com?t=https:/zh.wikipedia.org/wiki/%E5%9F%9F%E5%90%8D%E7%B3%BB%E7%BB%9F)，端口53，是互联网的一项服务。**它作为将域名和IP地址相互映射的一个分布式数据库，能够使人更方便地访问互联网**。  
  什么意思？我们上面已经说了，网络中的数据包，是通过**“端口号+IP地址+MAC地址”**来识别目的地址的，也就是说定位一个主机的时候，我们是定位的是他的IP地址，然后我们通过浏览器访问的时候呢？比如我们要访问谷歌，就在浏览器中输入他的**"域名"：[www.google.com](https://link.jianshu.com?t=http:/www.google.com" \t "_blank)**，却不是谷歌服务器的IP地址——这个时候，DNS协议就起作用了：  
  我们输入[www.google.com](https://link.jianshu.com?t=http:/www.google.com)并按下回车的时候，本机服务器先是请求DNS服务器，DNS服务器根据我们发送的域名，根据DNS协议，解析成该域名对应的IP地址并返回给本机，这样，我们就可以进行下面几层的地址封装了。

三.自顶向下的数据包结构

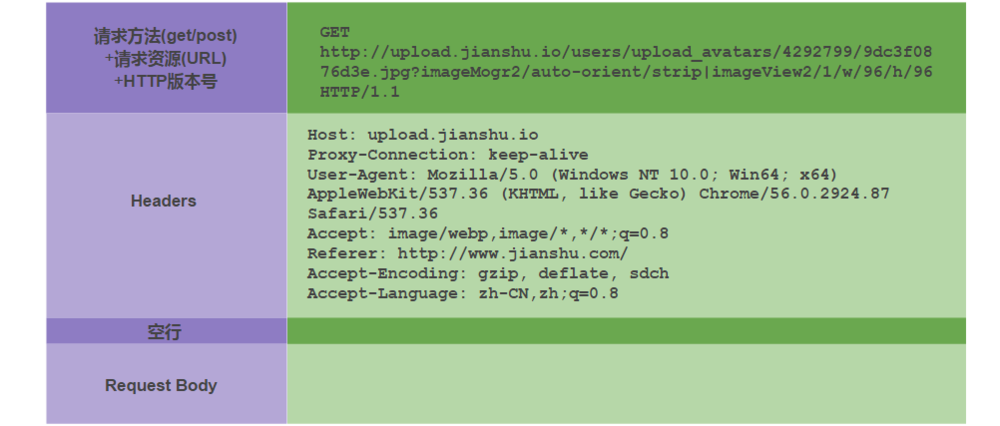
  现在我们从一个用户的角度来自顶向下的过一遍，一个网络数据包的过程。首先我们设置了本机参数：

* **本机的IP地址：192.168.1.100**
* **子网掩码：255.255.255.0**
* **网关的IP地址：192.168.1.1**
* **DNS的IP地址：8.8.8.8**

然后打开浏览器访问谷歌的网址：[www.google.com](https://link.jianshu.com?t=http:/www.google.com" \t "_blank)，按下回车。这意味着，浏览器要向Google发送一个网页请求的数据包。  
**第一步：**  
  主机会像DNS服务器发送请求，已知DNS服务器为8.8.8.8，于是我们向这个地址发送一个DNS数据包（53端口）；然后，DNS服务器做出响应，告诉我们Google的IP地址是172.194.72.105。于是，我们知道了对方的IP地址。  
**第二步：**  
  接下来，我们要判断，这个IP地址是不是在同一个子网络，这就要用到子网掩码。已知子网掩码是255.255.255.0，本机用它对自己的IP地址192.168.1.100，做一个二进制的AND运算（两个数位都为1，结果为1，否则为0），计算结果为192.168.1.0；然后对Google的IP地址172.194.72.105也做一个AND运算，计算结果为172.194.72.0。这两个结果不相等，所以结论是，Google与本机不在同一个子网络。  
**第三步：**  
  因此，我们要向Google发送数据包，必须通过网关192.168.1.1转发，也就是说，接收方的MAC地址将是网关的MAC地址。至此发送的各种必要参数已经基本确定了，数据包也可以发送了。

1.应用层数据包

  首先，根据笔者[HTTP协议详解与Android相关基础网络编程](https://www.jianshu.com/p/3ea1bf1ba915)这篇文章中的讲解，我们可以知道应用层HTTP协议的报文结构：



Http请求.png

  这里是请求简书的一个GET请求报文，请求谷歌的报文结构也是相同的，只是域名等内容不同罢了。我们假定这个部分的长度为4960字节，此时的数据包结构如下：

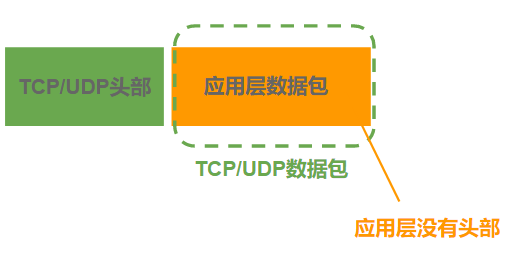


应用层数据包.png

就是一个单纯的数据包，没有头部，数据部分就是上面的报文。

2.传输层数据包（TCP/UDP数据包）

  TCP/UDP数据包需要设置端口，接收方（Google）的HTTP端口默认是80，发送方（本机）的端口是一个随机生成的1024-65535之间的整数，假定为51775。  
  **TCP数据包的标头长度为20字节**，加上嵌入HTTP的数据包，总长度变为4980字节。

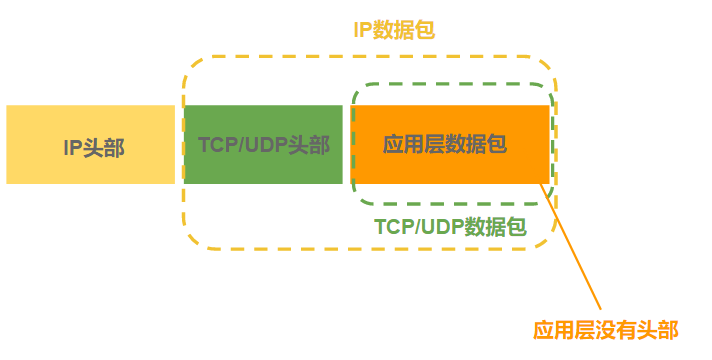


TCP或UDP数据包.png

可以看到，TCP/UDP数据包就是在应用层数据包前面加上端口号的等必要的寻址信息作为头部。

3.网络层数据包（IP数据包）

  下面就到了网络层，TCP数据包再嵌入IP数据包。IP数据包需要设置双方的IP地址，这是已知的，发送方是192.168.1.100（本机），接收方是172.194.72.105（Google）。  
  **IP数据包的标头长度为20字节**，加上嵌入的TCP数据包，总长度变为5000字节。

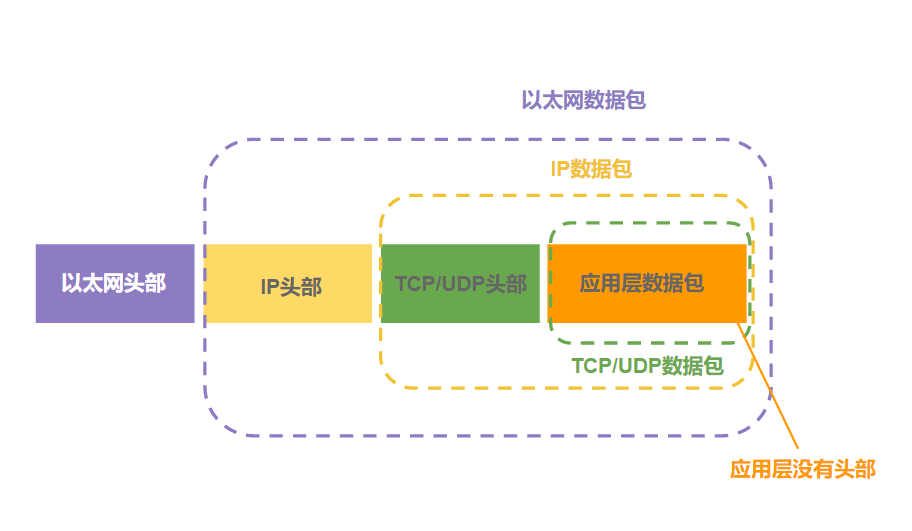


IP数据包.png

4.数据链路层数据包（以太网数据包）

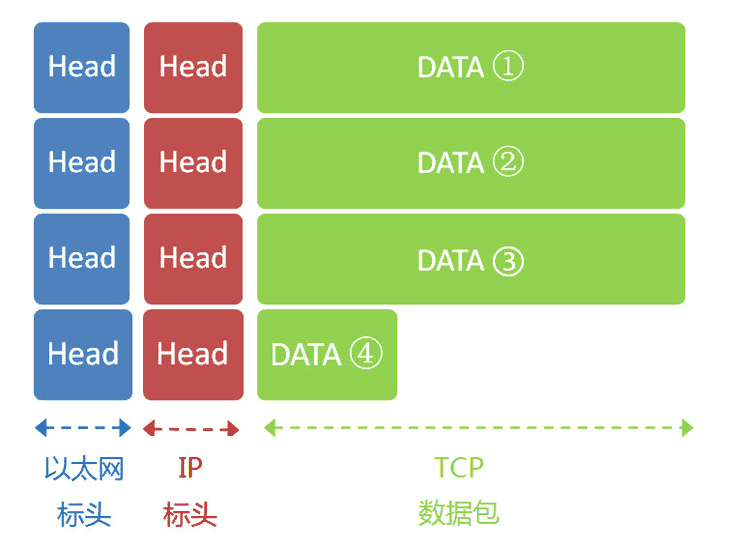
  最后，IP数据包嵌入以太网数据包。以太网数据包需要设置双方的MAC地址，发送方为本机的网卡MAC地址，接收方为网关192.168.1.1的MAC地址（通过ARP协议得到）。

**以太网数据包的数据部分，最大长度为1500字节**，而现在的IP数据包长度为5000字节。因此，IP数据包必须分割成四个包。因为每个包都有自己的IP标头（20字节），所以四个包的IP数据包的长度分别为1500、1500、1500、560。



以太网数据包.png

分割成四个数据包，分割只能分割数据部分，每个数据包都要具有相同的标头，不然找不到目的地址：

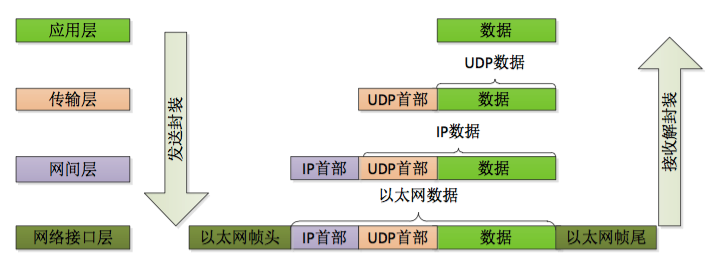


分割成四个.png

服务端响应：

  经过多个网关的转发，Google的服务器172.194.72.105，收到了这四个以太网数据包。  
  根据IP标头的序号，Google将四个包拼起来，取出完整的TCP数据包，然后读出里面的"HTTP请求"，接着做出"HTTP响应"，再用TCP协议发回来。  
  本机收到HTTP响应以后，就可以将网页显示出来，完成一次网络通信。  
  这个例子就到此为止，虽然经过了简化，但它大致上反映了互联网协议的整个通信过程。

我们再总结一下整个过程中数据包的结构包装变化（图片来源于网络）：



数据包变化.png

站在巨人的肩膀上摘苹果：

[互联网协议入门（一）阮一峰](https://link.jianshu.com?t=http:/www.ruanyifeng.com/blog/2012/05/internet_protocol_suite_part_i.html)  
[互联网协议入门（二）阮一峰](https://link.jianshu.com?t=http:/www.ruanyifeng.com/blog/2012/06/internet_protocol_suite_part_ii.html)  
《图解TCP/IP》  
《计算机网络：自顶向下方法》等

作者：游骑兵810  
链接：https://www.jianshu.com/p/c793a279f698  
來源：简书  
简书著作权归作者所有，任何形式的转载都请联系作者获得授权并注明出处。