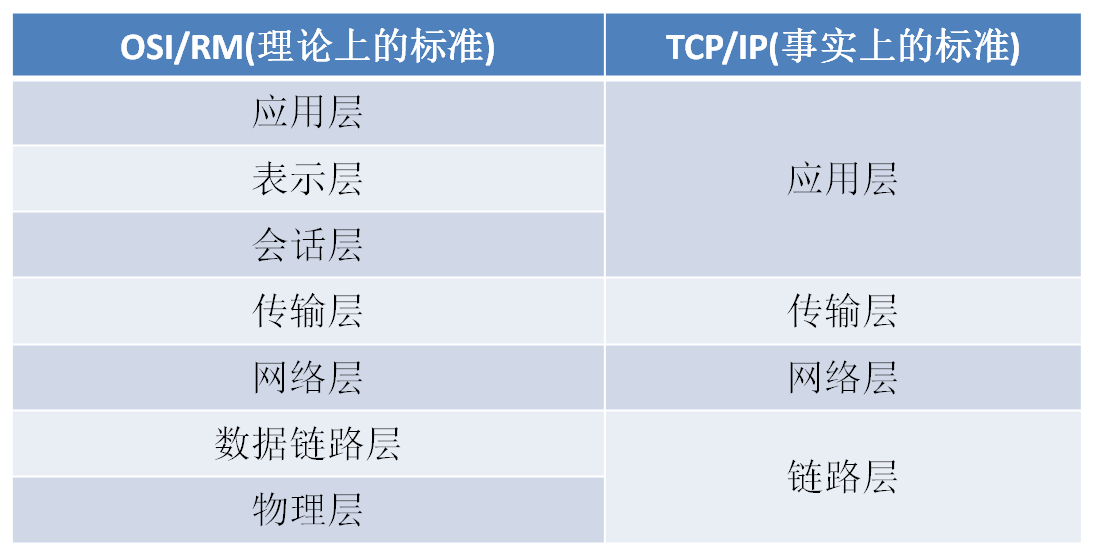
为了使各种不同的计算机之间可以互联，ARPANet指定了一套计算机通信协议，即TCP/IP协议( 族 )，它们对电脑如何连接和组网，做出了详尽的规定。理解了这些协议，就理解了网络的原理。

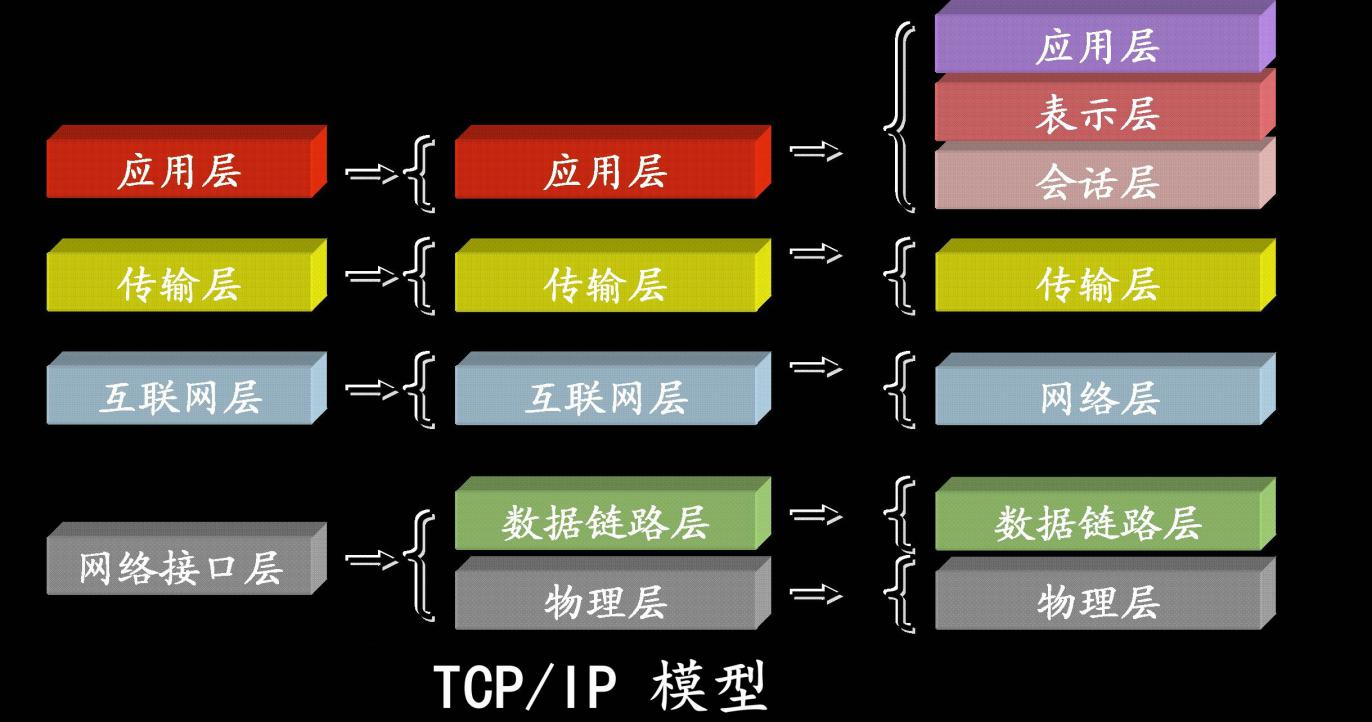
因为这些协议实在太复杂、太庞大，这里只是整理一个简洁的框架，帮助大家从总体上把握它们。

## 一、概述

### ****1.1 模型****

为了减少协议设计的复杂性，大多数网络模型均采用分层的方式来组织。如何分层有不同的模型，有的模型分七层( 不常用 )，有的分四层( 现在就是用这种 )，如下图：





### ****1.2 层与协议****

每一层都是为了完成一种功能。为了实现这些功能，就需要大家都遵守共同的规则。大家都遵守这规则，就叫做“协议”（protocol）。

网络的每一层，都定义了很多协议。这些协议的总称，叫“TCP/IP协议”。TCP/IP协议是一个大家族，不仅仅只有TCP和IP协议，它还包括其它的协议

## 二、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、无线电波等方式。



这就叫做“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

## 三、链接层

### ****3.1 定义****

单纯的 0 和 1 没有任何意义，必须规定解读方式：多少个电信号算一组？每个信号位有何意义？这就是“链接层”的功能，它在“物理层”的上方，确定了 0 和 1 的分组方式。

### ****3.2 以太网协议****

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）。

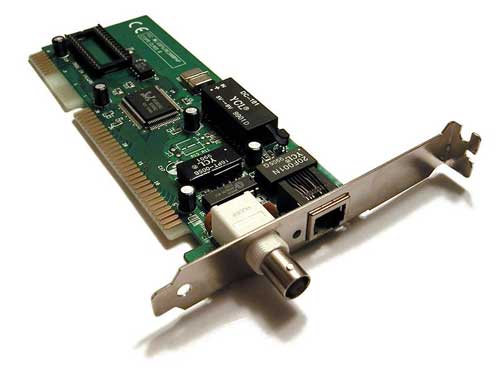


“标头”包含数据包的一些说明项，比如发送者、接受者、数据类型等等；"数据"则是数据包的具体内容。

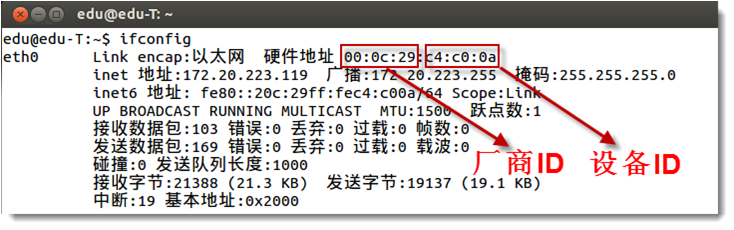
“标头”的长度，固定为 18 字节。"数据"的长度，最短为 46 字节，最长为 1500 字节。如果数据很长，就必须分割成多个帧进行发送。

### ****3.3 MAC 地址****

以太网规定，连入网络的所有设备，都必须具有“网卡”接口。每个网卡都拥有唯一的地址。数据包（帧）标头的发送地址和接收地址就是网卡的地址，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（理论上全球唯一），长度是 48 个二进制位，通常用 12 个十六进制数表示。



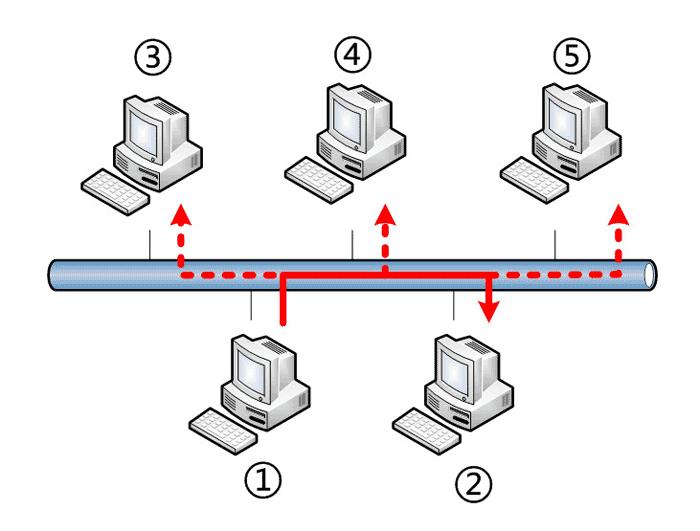
### ****3.4 广播****

首先，一块网卡怎么会知道另一块网卡的 MAC 地址？

回答：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：向本网络内所有计算机发送，让每台计算机与自身的 MAC 地址相比较判断，是否为接收方。这种发送方式就叫做“广播”（broadcasting）。



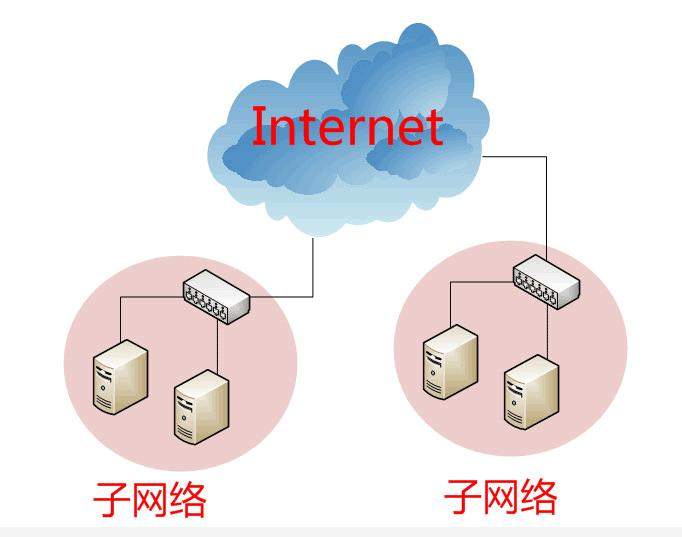
有了数据包的定义、网卡的 MAC 地址、广播的发送方式，“链接层”就可以在多台计算机之间传送数据

## 四、网络层

### ****4.1 网络层的由来****

以太网采用广播方式发送数据包，如果两台计算机不在同一个子网络，广播是传不过去的。

互联网是无数子网络共同组成的一个巨型网络



因此，必须找到一种方法，能够区分哪些 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。这套地址就叫做“网络地址”，简称“网址”。

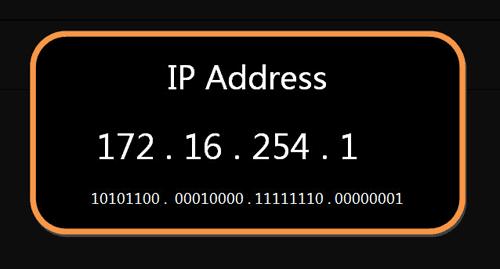
“网络层”出现以后，每台计算机有了两种地址，一种是 MAC 地址，另一种是网络地址。两种地址之间没有任何联系，MAC 地址是绑定在网卡上的，网络地址则是管理员分配的

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。

### ****4.2 IP 协议****

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。这个版本规定，网络地址由 32 个二进制位组成。



习惯上，我们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。

互联网上的每一台计算机，都会分配到一个 IP 地址。这个地址分成两个部分，前一部分代表网络，后一部分代表主机。

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。

单单从 IP 地址，我们无法判断网络部分。到底是前 24 位，还是前 16 位，甚至前 28 位，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。它的网络部分全部为1，主机部分全部为0，并且1和0分别连续。

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。

总结一下，IP 协议的作用主要有两个，一个是为每一台计算机分配 IP 地址，另一个是确定哪些地址在同一个子网络。

### ****4.3 IP 数据包****

根据 IP 协议发送的数据，就叫做 IP 数据包。不难想象，其中必定包括 IP 地址信息。

具体来说，IP 数据包也分为“标头”和“数据”两个部分。



Ip数据包传到链接层后，会被以太网数据包（帧）包裹。它放进以太网数据包后，以太网数据包就变成了下面这样。



IP 数据包的总长度最大为 65,535字节。而以太网数据包的“数据”部分，最长只有 1500 字节。因此，如果 IP 数据包超过了 1500 字节，它就需要分割成几个以太网数据包，分开发送了。

### ****4. 4 ARP 协议****

关于"网络层"，还有最后一点需要说明。

因为 IP 数据包是放在以太网数据包里发送的，所以我们必须同时知道两个地址，一个是对方的 MAC 地址，另一个是对方的 IP 地址。通常情况下，对方的 IP 地址是已知的（后文会解释），但是我们不知道它的 MAC 地址。

所以，我们需要一种机制，能够从 IP 地址得到 MAC 地址。

这里又可以分成两种情况。

第一种情况，如果两台主机不在同一个子网络，那么事实上没有办法得到对方的 MAC 地址，只能把数据包传送到两个子网络连接处的“网关”（gateway），让网关去处理。

第二种情况，如果两台主机在同一个子网络，那么我们可以用 ARP 协议，得到对方的 MAC 地址。ARP 协议也是发出一个数据包（包含在以太网数据包中），其中包含它所要查询主机的 IP 地址，在对方的 MAC 地址这一栏，填的是 FF:FF:FF:FF:FF:FF，表示这是一个“广播”地址。它所在子网络的每一台主机，都会收到这个数据包，从中取出 IP 地址，与自身的 IP 地址进行比较。如果两者相同，都做出回复，向对方报告自己的 MAC 地址，否则就丢弃这个包。

有了 ARP 协议之后，我们就可以得到同一个子网络内的主机 MAC 地址，可以把数据包发送到任意一台主机之上

## 五、传输层

### ****5.1 传输层的由来****

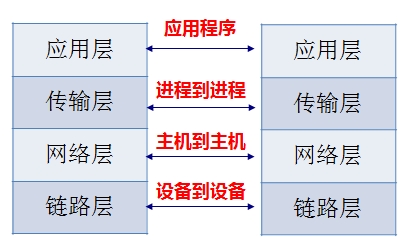
有了 MAC 地址和 IP 地址，我们已经可以在互联网上任意两台主机上建立通信。

接下来的问题是，同一台主机上有许多程序都需要用到网络，比如，你一边浏览网页，一边与朋友在线聊天。当一个数据包从互联网上发来的时候，你怎么知道，它是表示网页的内容，还是表示在线聊天的内容？

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）

“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。

“传输层”的功能，就是建立“端口到端口”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，我们就能实现程序之间的交流。因此，Unix 系统就把主机+端口，叫做“套接字”（socket）。有了它，就可以进行网络应用程序开发了。



### ****5. 2 UDP 协议****

现在，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

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。所以整个以太网数据包现在变成了下面这样：



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。

### ****5.3 TCP 协议****

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。它就是有确认机制的 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。

因此，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 数据包和 UDP 数据包一样，都是内嵌在 IP 数据包的“数据”部分。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

## 六、应用层

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。

这是最高的一层，直接面对用户。它的数据就放在 TCP 数据包的“数据”部分。因此，现在的以太网的数据包就变成下面这样。

