

# 第十七章通信原理

## 一 服务器硬件基础

### 1. 配置备份

### 2. 双路电源

## 二 通信原理

### 1. 交换机mac地址学习

### 2. ARP协议寻找mac地址

### 3. 子网掩码

### 4. ARP协议工作原理

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器进行重写。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 服务器硬件基础

我们的客户端软件最后是安装在用户的手机或者电脑上的，服务端软件是部署在我们的服务器上，接下来就先给大家介绍一下服务器是什么？

### 1. 配置备份

服务器也是一个电脑，不过他是没有显示器鼠标键盘的，我们把我们的项目放在服务器上，目的就是当用户来访问的时候能够给用户提供一个他想要的数据库，所以才叫做服务器。正因为是这样，服务器在运行的时候最基本的要求就是稳定，那么怎么达到稳定呢？比如你花了10万买了一台超级牛逼的服务器，但是这个服务器突然有一天网卡坏了，你的服务器在贵都没用了，所以，要保证服务器的稳定最基本的就是同样的东西给他再来一份，两个网卡，两套硬盘，两套内存。除此之外，服务器运行的没有问题，突然国家电网给你断电了，你怎么办？

## 2. 双路电源

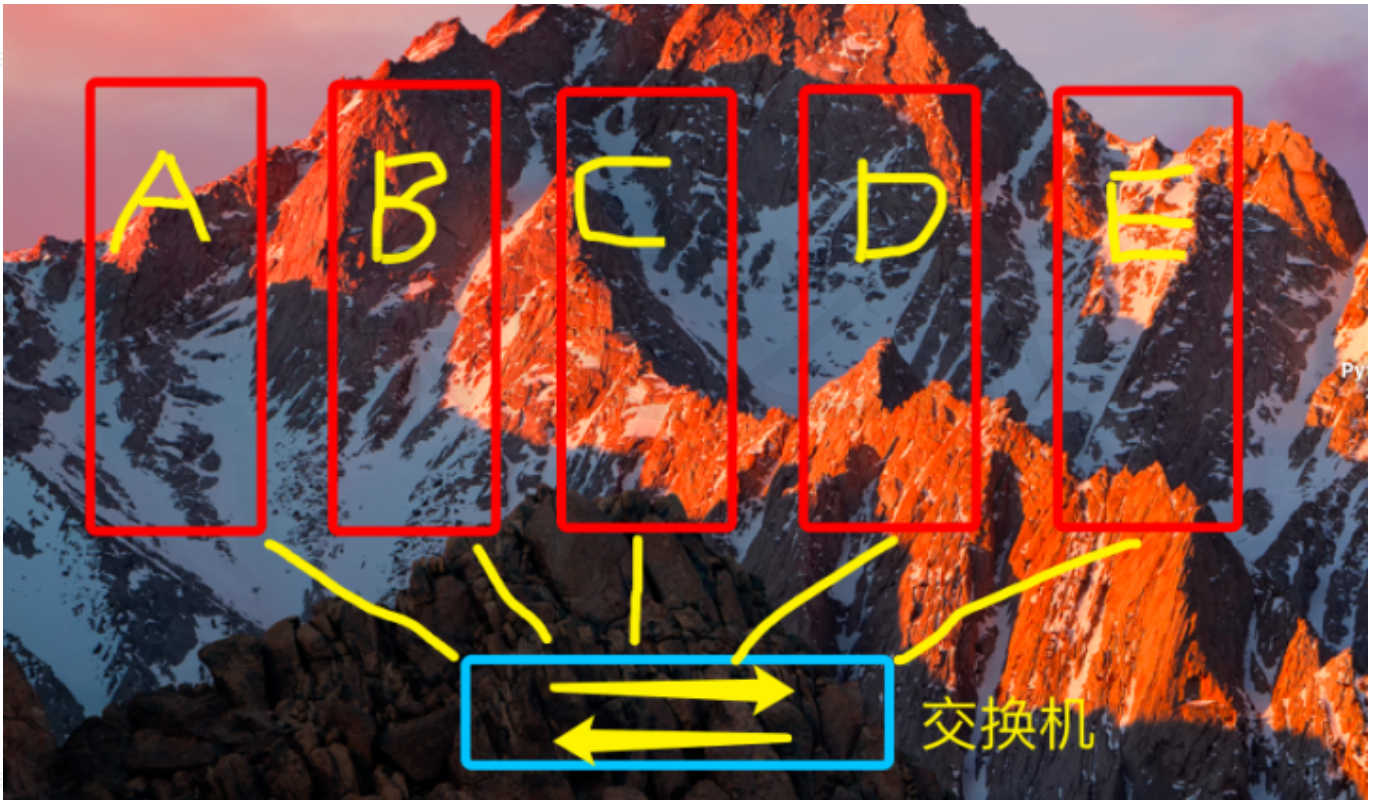
每当到了双十一的时候，网上就会流传一个段子，马云说今年双十一的成交额会突破多少亿，国家电网说我给你断电。其实这只是大家的一个设想，根本不会发生，但是如果真的发生了，难道马云就只能等待国家电网的宣判吗？

事实并不是这样的。为了保证服务器的运行，一般服务器会接双路电源，一路电源是我们正常使用的交流电，另外一路电源是备用电源，也是不间断供电，英文叫做UPS，由蓄电池和发电机组等一些设备直接供电的，一但发生断电，UPS会自动启动，以此来保证电量供应。服务器运行正常了，用户购物大部分的成交额都是通过移动端来完成了，所以如果双十一真的断电了，成交量可能会受影响，但是不可能像有些人想的那样。服务器的运行除了要有双路电源之外，对周边环境温度和湿度都是有要求的，因为服务器在工作中，时刻处于告诉运转的状态，声音非常大，所以我们自己创造一个服务器的运行条件，造价是比较昂贵的，因此大部分的中小企业都是用联通或者电信的IDC机房来托管服务器，一旦发生地震，火灾等险情你就找他赔钱的就可以了。

## 二 通信原理

### 1. 交换机mac地址学习

通过上面的学习，我们已经清楚了服务器是什么，那么接下来我们可以使用我们自己的一台电脑做服务器，另外一台电脑做客户端电脑，现在有这样一个场景，在我的办公室内（同一个局域网），电脑A与电脑D通信。



计算机通信靠的是广播，计算机A发一个包给所有的计算机，计算机D一看这个包发现目标地址是自己，那么就打开这个包，读取里面的数据，这个时候计算机D还想给A回一个包，并不按照我们原来设想的再广播一下给所有计算机都发一个包，交换机有一个mac地址学习的功能，这个交换机上有一个mac地址表，记录着一个网口（也有人叫端口，注意和传输层的端口不是一回事）对应一个mac地址，是这么一个记录表，刚开始把这几台计算机和交换机连接上的时候，他的记录为空。计算机A第一次发包广播的时候，会有一个源mac地址，目标mac地址，刚开始的记录值为空（可以把它想象成一个字典的形式），假如源地址的网口是18号口，交换机就会记录一个18号口的mac地址是 xx-xx-xx-xx-xx-xx，目标地址是计算机D，假如他的网口是21，那么交换机也会记录一个21号口的mac地址是xx-xx-xx-xx-xx-xx。假如你的18号口的计算机换了，那么这个mac地址也就换了，交换就会重新学习，更改18号口的mac地址（就像是更新字典里面以18号口为键的value）。如果交换机里面没有某个端口的mac地址的记录，他就会学习记录这个地址（就像是第六章项目，购物车中没有这个key，我们新添加一个key）。

计算机通信的时候如果发现交换机的mac地址记录表中有，那么他就不会在进行广播了，也因为交换机具备mac地址学习的功能，随着发包的机器越来越多，所有的机器都参与进来了，以后在进行通信的时候并不需要广播，而是在刚开始通信的时候，必须要广播，这时候交换机会把网口上所有的mac地址全都学到，或者某一网口上计算机换了才会进行广播，这时交换机也会重新学习 更新自己的mac地址表。

## 2. ARP协议寻找mac地址

上文我们所讲的是基于一个假设：我们知道对方的mac地址，但其实，我们只能知道自己的mac地址，而不能直接知道对方的mac地址，那么我们如何来找到对方的mac地址呢？

我们会通过ip地址和ARP协议来找到对方的mac地址，ARP协议是地址解析协议，ARP协议就是把ip地址解析成mac地址，把mac地址反解成ip地址。每一台电脑上有一个固定的mac地址之外，还有一个可以变

的ip地址（局域网内ip地址不能一样），在通信之前，对方的mac地址是不能获取的，但是一定要拿到对方的ip地址。在同一个子网（局域网也叫子网）内，要实现通信我们需要的是mac地址，那么我们怎么根据ip地址拿到这个mac地址？

ARP协议功能：广播的方式发送数据包，获取目标主机的mac地址。

实现原理：

假如我的计算机（mac地址：5c:f9:38:aa:e6:ce，ip地址：192.168.0.106）现在要发送给目标机器一份数据（mac地址：未知，ip地址：192.168.0.105）

在发包的时候会先把目标mac地址按照FF:FF:FF:FF:FF:FF（12个大写的F）来发送，也就是像下面这样发出来：

```
(5c:f9:38:aa:e6:ce, FF:FF:FF:FF:FF:FF) (192.168.0.106, 192.168.0.105)
```

源mac地址 暂定目标mac 源ip地址 目标ip地址

交换机mac地址表

```
1:5c:f9:38:aa:e6:ce
2:空
3:空
4:空
```

发给交换机之后，交换机首先学习到的是源mac地址，把它加入到mac地址表，另外它要根据目标mac地址，这个这个包给目标机器，当交换机一看到目标mac是十二个F，这不是一个具体的mac地址，他就明白源机器想要找到ip地址是 192.168.0.105 的一台机器的mac地址，所以这十二个F代表一个广播的mac地址，所以这个包应该群发，接下来每一台机器都会收到一份，作为接收者就会明白这是想要它的mac地址，如果不是这个包的接收者肯定不会给他返回自己的mac地址，但如果刚好一个接受者发现这个ip地址就是自己的ip地址，那么他就会把自己的mac地址返回给发送这个包的机器，这个时候交换机也就学习到了这个mac地址，这个时候包的发送就反过来了。

```
(原来接收者的mac地址, 5c:f9:38:aa:e6:ce, ) (192.168.0.105, 192.168.0.106)
```

源mac地址 接收者mac 源ip地址 目标ip地址

交换机mac地址表

```
1:5c:f9:38:aa:e6:ce
2:新学习的mac地址
```

3:空

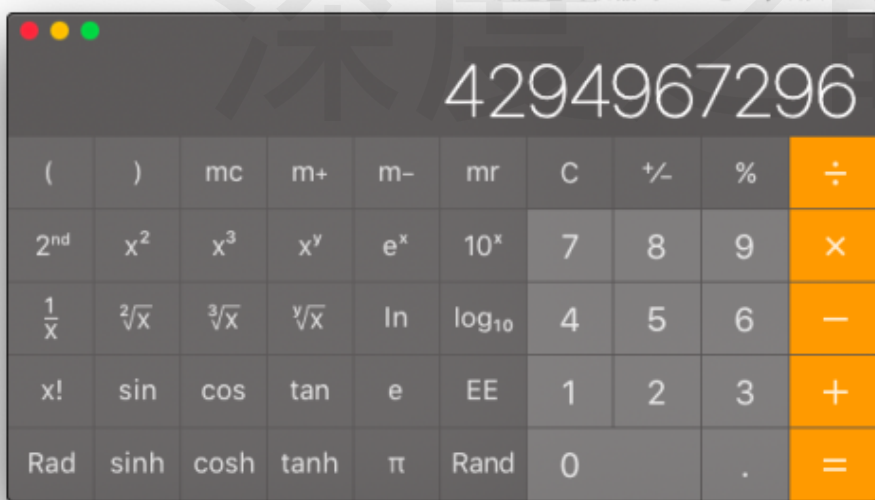
4:空

每台机器上面都会有一个ARP解析记录，这个解析记录就会记录着这台机器的mac地址和与之发过包的机器的mac地址，所以短时间如果再发包的话就不需要再走这个过程了。

### 3. 子网掩码

刚才我们假设是服务端机器和客户端的机器都在一个子网内，他们相互之间的通信就需要mac地址来完成，我们可以ip地址加ARP协议解析出mac地址，但是真实的场景，服务端的机器和客户端的机器不可能在一个子网内，通过广播包（广播的通信方式仅限于在同一个子网内，这是上一章的内容）这种形式我们不能拿到对方的mac地址，就算是拿到了，广播的这种通信方式出了局域网也不能通信，mac地址只能在局域网内使用。所以，出了局域网，我要想实现通信就要靠ip地址。

ip地址是网络层的规范，网络层有一个ip协议，它规定了ip地址是点分十进制的，以点为分割，分成四部分，每部分的都是十进制，每个部分的取值范围是0~255，也就是说ip地址最小是：0.0.0.0，最大是255.255.255.255，每个位置有256种可能，一共有4个位置，那么一共就是 $256^4=2^{32}$ 种可能，结果是下图：



这个数字大概是43亿不到，它的范围非常有限，现在已经有点不太够用了，所以现在正在推广IPV6，这又是另外一种ip地址的标识方式。我们现在只需要知道大部分在用的是IPV4就可以了，中国互联网的巨头像阿里和腾讯会已经在做全线互联网生态链的升级了，会全部都改成IPV6，你不用担心这个活你干不了，因为根部轮不到你干。接下来我们研究还是先以IPV4为对象进行研究。

单纯的ip地址是没有意义的，他还会配一个子网掩码，子网掩码也是点分十进制，比如说我们现在有一个IP地址是：172.16.10.1，子网掩码是：255.255.255.0，我们把子网掩码翻译成二进制就是11111111.11111111.11111111.00000000，从左到右数，一共有24个1，所以我们可以这样表示ip地址和子网掩码：172.16.10.1/24，子网掩码和ip地址的格式是一样的，那么它的功能是做什么呢？



子网掩码和ip地址他们两个合起来是为了标识一个网络地址，或者叫局域网的地址，子网地址。假如我们访问百度的网站，一定要先找到它的服务器所在的子网，进而找到里面的主机。我们的子网掩码可以翻译成二进制的形式，ip地址自然也可以翻译成二进制的形式：

ip地址翻译成二进制

172.16.10.1 : 10101100.00010000.00001010.00000001

子网掩码翻译成二进制

255.255.255.0 : 11111111.11111111.11111111.00000000

我们把ip地址和子网掩码做一个暗位语运算，两个都是1，最后的结果才为1

10101100.00010000.00001010.00000001

11111111.11111111.11111111.00000000

运算结果

10101100.00010000.00001010.00000000

拿到运算的结果后，我们再把这个结果翻译成十进制就是他的局域网地址：172.16.10.0

再来看一个例子

172.16.10.2 : 10101100.00010000.00001010.00000001

255.255.255.0 : 11111111.11111111.11111111.00000000

运算结果

10101100.00010000.00001010.00000000

翻译成十进制

172.16.10.0

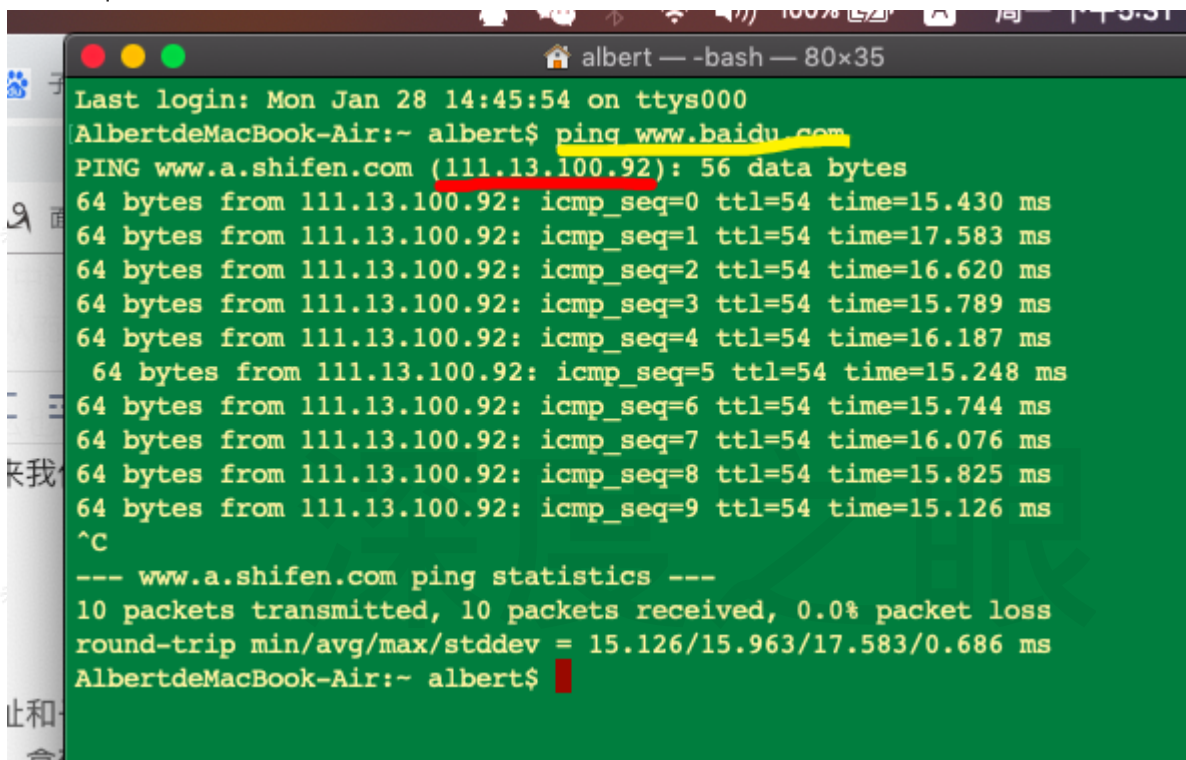
注意：在第一章的内容我们就学习过了，计算机最喜欢看到的就是二进制，在计算机内存中，根据ip地址和子网掩码可以得到局域网地址，使用子网掩码和二进制的形式，就是为了能够掩盖局域网，让你不能一眼看出他的局域网。使用二进制进行计算的这种形式并不少见，有学过C语言的同学可能会知道，在C语言中计算整数减法会用两个数字的补码相加来计算，这么做的目的就是用加法来代替减法，从而达到简化硬件电路的目的。

通过比较你会发现这两个子网地址是相同的，那么也就是说他们是在同一个局域网内，那么接下来的通信就是按照我们上面所讲过的了。接下来我们再来讲一下ARP协议的工作原理。

## 4. ARP协议工作原理

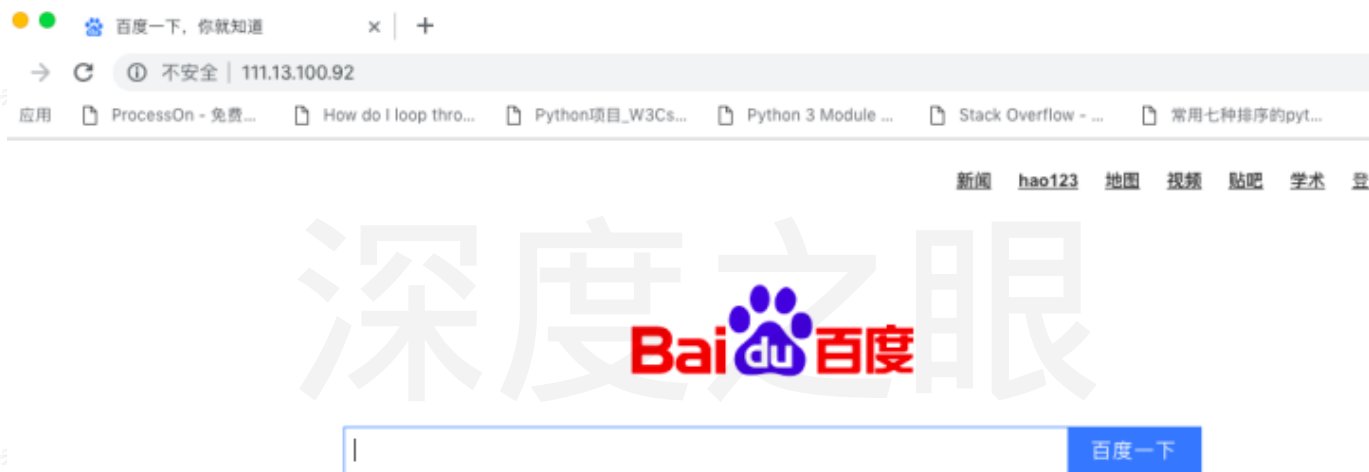
要想实现通信我们首先要有自己的机器的ip地址和子网掩码（每台电脑上面都有，可以自己查看），还要有对方机器的ip地址和子网掩码，拿到以后ARP协议就开始自动工作了，在通讯之前ARP协议会把你的ip地址和子网掩码全部换算成点分二进制，进行暗位语运算得到一个网络地址，这个就是源主机的局域网地址，同理，对方的ip地址和子网掩码也可以通过运算得到一个局域网地址，ARP协议会把这两个子网地址相比较，如果相同，那就是按照上面我们讲的那样来进行通信，但大多数情况，客户端机器和服务端机器是不可能在一个局域网内的，那么跨局域网他们是怎么实现通信的呢？

以百度为例，它的服务端机器肯定不会和我的电脑在一个局域网内，我们可以使用 ping 这个指令来查看百度的ip地址



```
albert ~ -bash - 80x35
Last login: Mon Jan 28 14:45:54 on ttys000
AlbertdeMacBook-Air:~ albert$ ping www.baidu.com
PING www.a.shifen.com (111.13.100.92): 56 data bytes
64 bytes from 111.13.100.92: icmp_seq=0 ttl=54 time=15.430 ms
64 bytes from 111.13.100.92: icmp_seq=1 ttl=54 time=17.583 ms
64 bytes from 111.13.100.92: icmp_seq=2 ttl=54 time=16.620 ms
64 bytes from 111.13.100.92: icmp_seq=3 ttl=54 time=15.789 ms
64 bytes from 111.13.100.92: icmp_seq=4 ttl=54 time=16.187 ms
64 bytes from 111.13.100.92: icmp_seq=5 ttl=54 time=15.248 ms
64 bytes from 111.13.100.92: icmp_seq=6 ttl=54 time=15.744 ms
64 bytes from 111.13.100.92: icmp_seq=7 ttl=54 time=16.076 ms
64 bytes from 111.13.100.92: icmp_seq=8 ttl=54 time=15.825 ms
64 bytes from 111.13.100.92: icmp_seq=9 ttl=54 time=15.126 ms
^C
--- www.a.shifen.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 15.126/15.963/17.583/0.686 ms
AlbertdeMacBook-Air:~ albert$
```

黄色的是我输入的指定：ping > [www.baidu.com](http://www.baidu.com)，红色的表示百度的IP地址，我们可以使用这个ip地址直接访问百度的网站



假如我现在在一个子网内，隔壁的办公室又在另外一个子网内，我们知道的广播的通信方式只能局限于同一个局域网内，现在有一个解决方案，每个办公室门口站一个人，这个专门负责和其他办公室门口的人通信，他们之间建立联系就可以了。一旦通过子网地址发现，源地址和目标地址不在同一个子网，这个包就不能在子网里面转悠，要把这个包给门口那个人，门口那个人再把它分发给他自己的子网。门口站的那个人就叫做网关，网关通常会配在路由器上



每一个子网有一个网关，它的ip地址通常会配置成 1 或者255结尾，不同网关之间通过路由协议来进行通信，这又是一个非常庞大的领域，通常是由网络工程师来完成的。一个子网内的网关是在他自己的局域网内的，他们之间的通信是通过mac地址来完成的。

总结：

当源地址和目标地址不在同一个局域网内的时候，会把这个包给网关，网关与网关之间通信走的是路由协议，找到了对应的网关，网关再通过以太网协议把这个包给相应的主机。

注意：

ARP协议是由操作系统自动执行的，上一章我们总结了ip地址加mac地址就能定位到全世界独一无二的一台机器，现在ARP协议可以帮助我们自动获取mac地址，那么我们可以说ip地址加子网掩码就可以定位到全世界独一无二的一台机器。

深度之眼