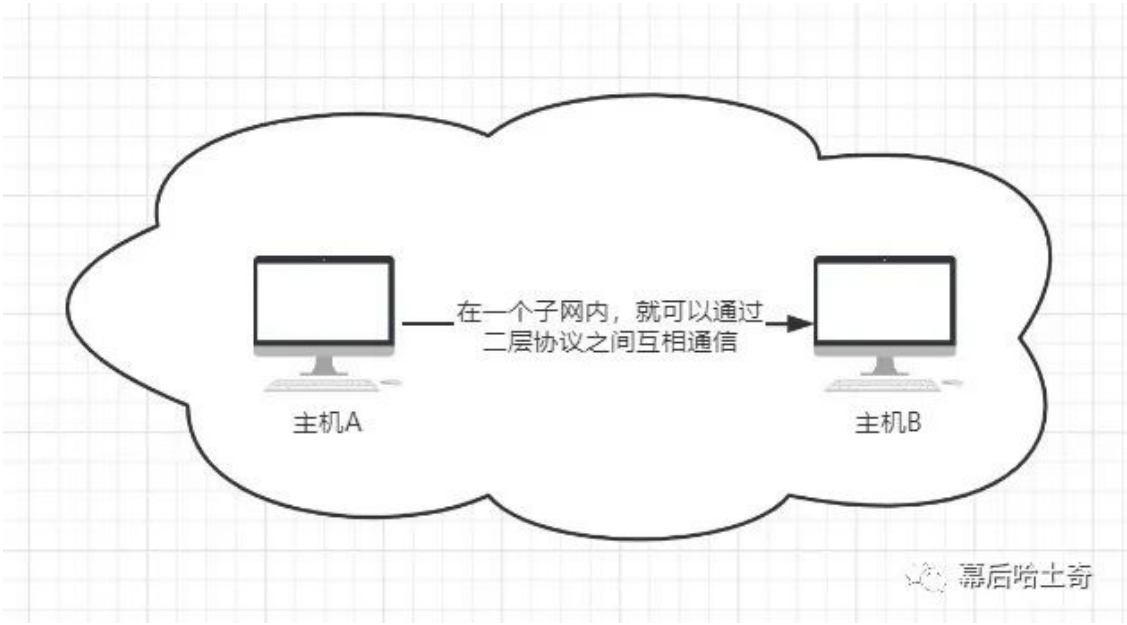
ARP协议是我们经常碰到的一个协议，无时无刻不在使用它。

我们先考虑最简单的场景，若A和B属于同一子网，如果A知道B的MAC地址，则在主机A的数据链路层把IP数据包封装成“帧”的形式，其中：

- 源MAC地址：A的MAC地址
- 目的MAC地址：B的MAC地址
- 源IP地址：A的IP地址
- 目的IP地址：B的IP地址

如果A不知道B的MAC地址，则使用ARP协议，发送一个数据包来获取B的MAC地址，获得之后，在使用同样的方法来对IP数据包进行封装。然后直接发给主机B，至此完成数据传输。



没错，我们A需要知道B的MAC地址，就需要通过ARP协议来解决。

一、ARP协议简介

ARP 是 Address Resolution Protocol 的缩写，表示“地址解析协议”。

那如何在不知道其 MAC 地址的情况下向我们所属的网络上的机器发送消息？

这个时候广播地址就开始起作用啦，多亏了广播地址，该消息将被发送给所有人，这被称为 ARP 请求 或 **ARP 广播**。

上图的主机A会通过广播将此信息广播出去：“XXXIP是谁的？告诉我你的MAC地址！”，这个时候主机B也会收到此广播，发现IP是自己的IP，因此回应自己的 MAC 地址给主机A。

因此，**ARP 是这样一种协议，它可以关联 OSI 第 2 层的 MAC 地址与第 3 层的 IP 地址**，因此本文的标题起成了处于2.5层的协议，不过非要将其分类，自然需要分类到网络层啦，毕竟这篇文章可没安排在数据链路层。

不过这里有个问题，如果每次你要发送信息时都用广播，难道网络不会饱和吗？当然会。所以我们得采用一个解决方案，就是ARP 表。

二、ARP表

到目前为止，我们已经接触了两个表，一个是交换机的CAM表，用来记录交换机某个接口和所连接机器的MAC映射关系，另外一个是路由表，记录的是要加入的网络和加入网络需要经过的网关地址之间的映射关系。

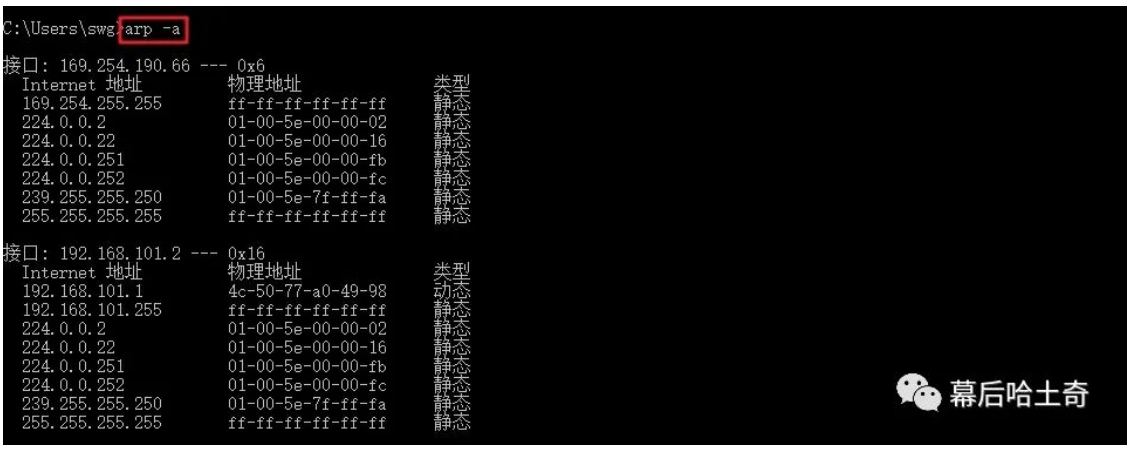
本篇文章迎来第三张表，为了避免每次要向机器发送信息时都必须发送 ARP 广播，我们用 ARP 表短时间内维持 IP 地址和 MAC 地址的对应关系，这也被称为 ARP 缓存。

> 将常用的信息缓存一段时间，也是软件系统为了提高处理速度的重要思路，用空间换时间。

因此，如果主机A将数据包发送给主机B，则将B的 MAC 地址记在 ARP 表中。下一次我想与之通信时，将不再需要在网络上发送 ARP 广播了。

借助 ARP 协议，我们可以根据 IP 地址获取 MAC 地址。还有一个 **RARP 协议**，是 Reverse Address Resolution Protocol 的缩写，表示“反向地址解析协议”，和 ARP 正相反，RARP 协议可以根据 MAC 地址获取 IP 地址。

在 windows 上可以用 arp -a 命令查看 ARP 缓存：



ARP 表也有TTL寿命，这个 TTL 的大小是不一定的，和操作系统有关，也可以设置。一般来说，如果与此地址在大约两分钟内没有通信，那么对应的记录就会从 ARP 表中被删除。因此，ARP 表是动态的，它会随着时间的流逝而变化，具体取决于与本机通信的机器。

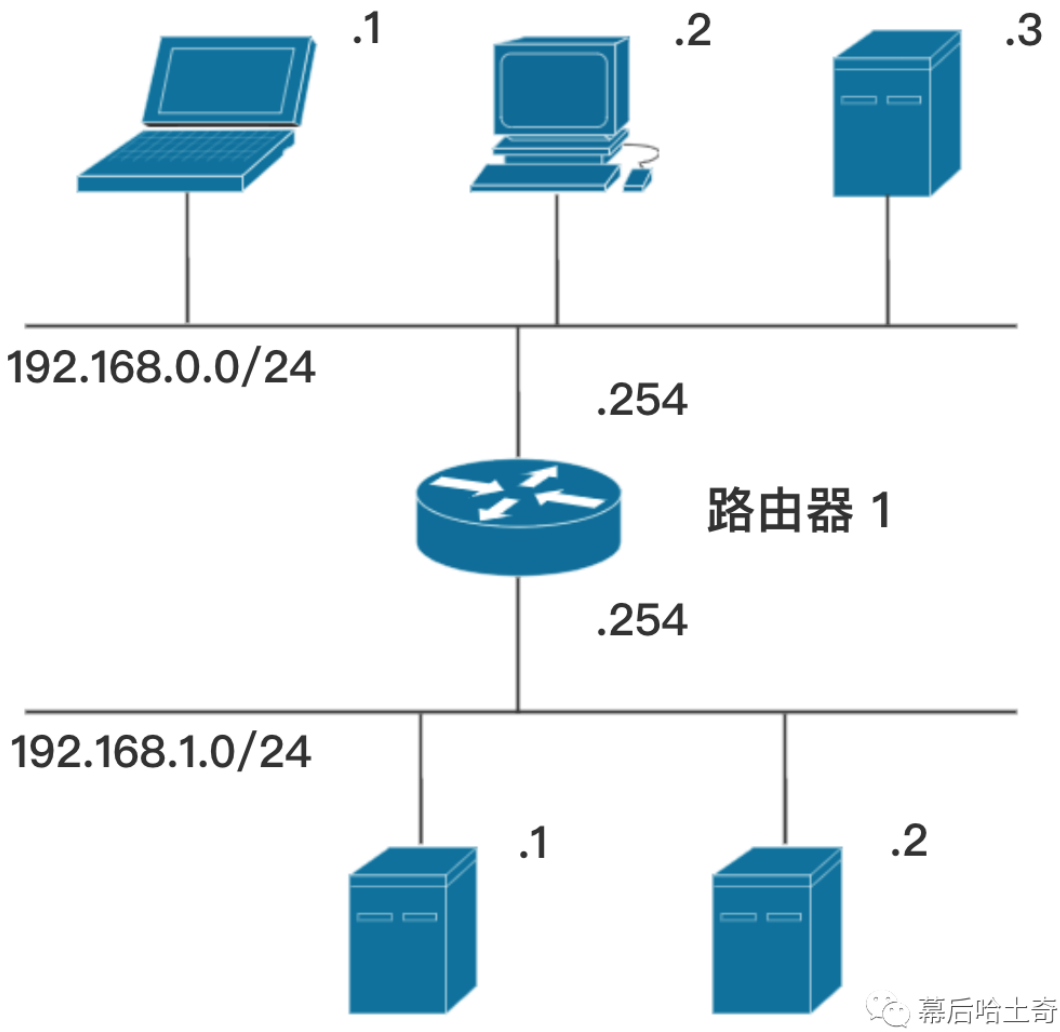
三、ARP请求过程

假设我们是机器A 192.168.0.1，想要向机器B 192.168.0.2 发送消息。此时还不知道机器B的 MAC 地址，这个时候 ARP 协议开始发挥作用了：

- 我们首先查看本地 ARP 表，看看表中是否存在 IP 地址 192.168.0.2 与它的 MAC 地址的关联记录；
- 如果表中有此关联记录的话，我们直接向 MAC 地址发送信息，完毕；
- 如果表中没有此关联记录，则通过网络发送 ARP 广播；
- 机器 192.168.0.2 将回复我们，并告诉我们它的 MAC 地址；
- 我们将在 ARP 表中写入 IP 地址 192.168.0.2 和其 MAC 地址的关联记录；
- 我们向 MAC 地址发送信息，完毕。

现实情况往往是两台机器不处于同一个局域网内，机器A如何知道机器B的MAC地址呢?

我们通过一个例子来重新梳理两个机器之间的通信过程：机器A 192.168.0.1 希望向机器B 192.168.1.2 发送消息。



显然，本地机器要经过交换机、路由器等设备才能到达目的机器，那么趁此机会我们把之前学习过的知识整体梳理一遍。

第一步：机器A本地处理

首先一条消息要通过 OSI 封装下来，从应用层到最后的物理层。当消息到达第三层的时候，机器A即可通过子网掩码来计算机器B不跟自己在一个子网。那么就需要通过路由器1 192.168.0.254 才能离开机器A的网络。

如果我们的机器发现ARP表中没有缓存路由器1的MAC地址，则我们的机器A发出 ARP 请求，先获取 192.168.0.254 的MAC地址。获取到之后，构建帧，帧的格式为：

192.168.0.254 的 MAC 地址（目标 MAC 地址）	192.168.0.1 的 MAC 地址（源 MAC 地址）	IP	???	192.168.0.1（源 IP 地址）	192.168.1.2（目标 IP 地址）	要发送的信息	CRC

可以看到，目的MAC地址是路由器1的MAC地址，目标IP地址是我们希望访问的机器B。

一般情况下，机器到路由器之间还会有个交换机，比如一层楼一个交换机，每层楼的电脑等终端都最终通过交换机连到到“出去”的路由器。

第二步：交换机

接收到此帧的第一台机器将是 192.168.0.0/24 这个网络的交换机 。交换机接收到了帧，读取帧中的目标 MAC 地址。交换机将查看自己的 CAM 表以了解表中是否有此 MAC 地址，以便了解应该将帧发送至哪个端口。如果交换机在自己的 CAM 表中找不到目标 MAC 地址，它就将帧发送到所有活动的端口！

因此，交换机现在可以将帧发送至输出端口，此输出端口连接到路由器，路由器就接收到了帧。

第三步：路由器

帧到达路由器，路由器的 OSI 第 2 层读取目标 MAC 地址。它发现目标 MAC 地址正是自己的 MAC 地址！因此，它就不再读取第 2 层的头部了，它将第 2 层以太网协议的头部移除，并将剩余的 IP 数据报发送给指定的第 3 层的协议：IP 协议。

OSI 第 3 层将读取整个第 3 层的头部，特别是目标 IP 地址。路由器看到目标 IP 地址不是自己的 IP 地址，因此它知道必须将此数据报发送到目标机器。

因此，它将在其路由表中查找，看看应该将数据包发送到哪个网关，以便到达机器 192.168.1.2。

路由器看到目标 IP 地址 192.168.1.2 是在自己所属的网络之一（192.168.1.0/24）中，因此能够直接将数据包发送给它。

但是，要通过网络发送帧给192.168.1.2，将需要 192.168.1.2 的 MAC 地址，因此路由器将发出 ARP 请求。一旦收到 192.168.1.2 的 MAC 地址后，它将能够构建帧并通过网络发送。此时，帧的格式变成了：

192.168.1.2 的 MAC 地址（目标 MAC 地址）	192.168.1.254 的 MAC 地址（源 MAC 地址）	IP	???	192.168.0.1（源 IP 地址）	192.168.1.2（目标 IP 地址）	要发送的信息	CRC

可以看到，与之前的帧的格式相比，仅修改了第 2 层的信息（目标 MAC 地址和源 MAC 地址）！

下面，帧将离开路由器继续前行。

第四步：交换机

帧将到达交换机，但是这次的交换机和第二步的交换机不一样，这次是网络 192.168.1.0/24 的交换机。

交换机查看目标 MAC 地址，看到是机器 192.168.1.2 的 MAC 地址，因此将帧发给机器 192.168.1.2。

第五步：机器B本地处理

机器B收到帧，OSI 第 2 层读取目标 MAC 地址。它看到目标 MAC 地址正是自己的 MAC 地址。因此，它将读取第二层头部的剩余部分，并将帧中包含的数据报发送给 OSI 第 3 层的协议，就是帧中指定的 IP 协议。

第 3 层的 IP 协议接收数据报并读取数据报的头部，它发现目标 IP 地址正是自己的 IP 地址，因此它将把信息发送到第 4 层（传输层），第 4 层将把信息发送到第 7 层（应用层）。