本篇文章是本系列网络层篇的最后一篇，上篇我们学习了VPN技术，并且初步引入了NAT技术，NAT技术帮助我们大大降低了IPv4地址枯竭带来的影响，本篇文章深入学习下NAT技术，这是一项极其常用和重要的技术，在工作生活中将经常碰到。

一、动态NAT概述

私有地址只能用于一个机构的内部通信，而不能用于和因特网上的主机通信，因此，私有的源IP地址应该被公有地址代替，这就是NAT的主要原理。

NAT有两种不同类型，动态NAT和静态NAT。

- 动态NAT：将多个私有地址与一个公有地址相关联。因此，仅使用一个公有地址就可以连接多台机器，这样就节约了 IP 地址。
- 静态NAT：每个私有地址都和一个公有地址相关联。因此，并没有节约 IP 地址。

目前静态NAT已经很少被使用了，因为它并没有解决IP地址短缺的问题，因此只讨论动态NAT。

让我们以你家中的局域网为例，家中上网设备如电脑想接入互联网，是以上网盒子作为与互联网沟通的桥梁的。（上网盒子，俗称猫，现在一般都是光纤入户，所以都叫做光纤猫或者直接叫做光猫）。

上网盒子就相当于一个本地的路由器。当然了，现在一般的上网盒子严格来说是调制解调器（Modem，俗称“猫”）和路由器（Router）的融合。可以同时完成调制解调器、路由器交换和wifi的功能，省得需要多个设备仪器工作了，这样确实方便了宽带用户。

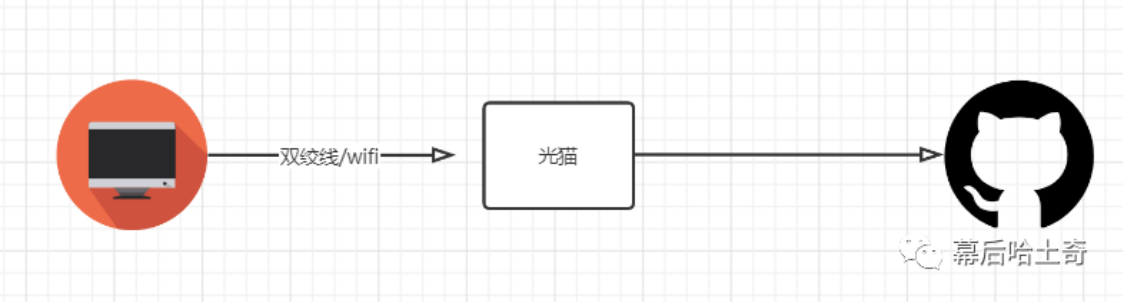
> 但是实际上，上网盒子自带的路由器功能即wifi的功能很不稳定，建议还是通过网线再外接一个稳定的路由器，为家人上网护航！

比如你购买了电信宽带，装维人员上门装宽带时，会提供一个光猫，一端连入户光纤，通过配置光猫后（你的电信账号），我们家庭上网设备通过光猫就可以接入到互联网。

- 猫的一端通过多种方式连接你的互联网服务提供商的基础设施，比如连接光纤；
- 另一端则可通过双绞线（就是网线）的方式，连接你的任何一台路由器（或一台电脑），然后就可以上网了（如果连接路由器，则一般是路由器共享 Wi-Fi 给各个设备上网，或者把设备用有线方式接入到路由器的以太网接口上；如果是电脑，则可以上网了）。

此时，本地路由器在互联网端具有一个合法的公有IP地址，一切准备就绪，我们现在唯一的需求就是能上网。

假设我们电脑的IP地址为192.168.0.1（稍微提一句：实际上192.168.0.1会优先分配给路由器，而不是我们的电脑，这里就暂且忽略此细节），向ip地址是13.250.177.223的github请求。



第一步离开我的电脑上还是一样：

DST 地址（接收方 MAC）	SRC 地址（发送方 MAC）	第 3 层使用的协议	...	源 IP 地址 192.168.0.1	目标 IP 地址 www.github.com 是 13.250.177.223	CRC

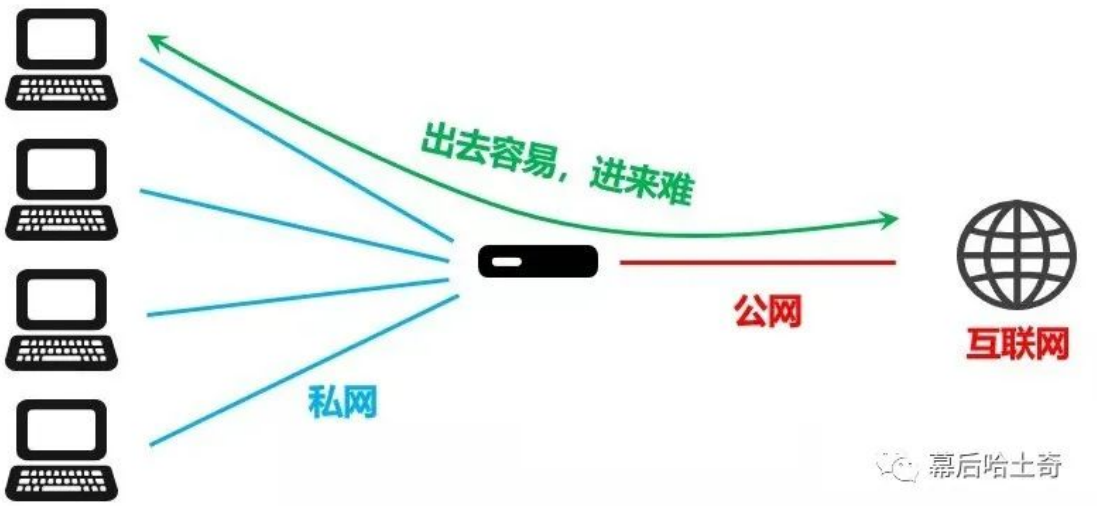
但是请求到了本地路由器的时候，这个时候就会运用NAT技术，将私有地址192.168.0.1转换为运营商为我们分配的公网地址117.148.64.237：

互联网上的某个路由器的 MAC 地址	本地路由器的 MAC 地址	第 3 层使用的协议	...	源 IP 地址 117.148.64.237	目标 IP 地址 www.github.com 是 13.250.177.223	CRC

这样，我们的数据包将可以到达Github网站的服务器，它将向本地路由器的IP地址117.148.64.237发送响应。

如果本地路由器下只有我们一台机器，那么流程很简单，直接把响应报文转发给我们这个电脑即可，大功告成！

但是等等，有个问题：家里人多个电脑同时请求github时，此外，我们电脑一般会同时运行多个程序，比如QQ和微信，我们先来看看出去的方向，依然是毫无问题的，但是回来时响应报文该发给谁呢？按照目前所学，本地路由器真的不知道该发给谁才好，所以我们会发现，NAT技术对于出去是简单的，不过回来嘛，就没那么容易了。



二、内网机器主动访问互联网的服务

家里某一台电脑要上网，我出去容易，但是回来的时候稍微有点麻烦，有点区分不开发给我这个电脑还是发给我的手机还是发给我的平板？

也就是说，光靠一个IP还不够，那么加个端口呢？

> 端口是传输层的概念，不过我相信读者朋友们都熟悉此概念，我们无论访问什么网站，输入的URL中必然要带着端口，如果没显式写出来，那么默认的是80端口而已。端口的概念很简单，比如我们电脑上可能会同时运行QQ、微信等软件，实际上要占用不同的端口来区分这些服务，不然QQ的消息串到了微信里，不就闹了笑话？

比如我家里有两台上网设备：笔记本电脑和平板电脑，直接连本地路由器访问github。

我的电脑会提供一个源端口用来区分，假设源端口号是 12345，目标端口号是 80，那么我的离开我电脑的帧结构为：

本地路由器的 MAC 地址	192.168.0.1 的 MAC 地址	第 3 层的协议	...	源 IP 地址 192.168.0.1	目标 IP 地址 www.github.com，是 13.250.177.223	源端口号 12345	目标端口号 80	CRC

本地路由器将收到该帧，并能够记录源 IP 地址192.168.0.1与使用的源端口12345的对应关系。

不过我们不能忘记，nat之后是变成了公网ip，所以实际上记录的是一对信息：

源 IP 地址，目标 IP 地址，源端口号，目标端口号	源 IP 地址，目标 IP 地址，源端口号，目标端口号
192.168.0.1，13.250.177.223，12345，80	117.148.64.237，13.250.177.223，12345，80

这样，当本地路由器收到了来自github的回应报文后，就可以在这张表中去查找12345端口对应的信息，发现需要以相反的方向来使用NAT（即公网转私网），将帧发送给192.168.0.1这台机器，就是你的机器。

不过，等等，好像有问题。

假设我的电脑跟我的平台不小心同时使用相同的源端口号向github发出请求呢？情况开始变得混乱起来。

源 IP 地址，目标 IP 地址，源端口号，目标端口号	源 IP 地址，目标 IP 地址，源端口号，目标端口号
192.168.0.1，13.250.177.223，**12345，80**	117.148.64.237，13.250.177.223，**12345，80**
192.168.0.2，13.250.177.223，**12345，80**	117.148.64.237，13.250.177.223，**12345，80**

可以看到，忙活半天又回到解放前了，还是不能有效区分谁是谁！不过有办法！

上面，我们的本地路由器只是将私网ip修改为了公网ip，那么能不能再顺便修改下端口号呢？这样不就不会冲突了吗？形如：

源 IP 地址，目标 IP 地址，源端口号，目标端口号	源 IP 地址，目标 IP 地址，源端口号，目标端口号
192.168.0.1，13.250.177.223，**12345，80**	117.148.64.237，13.250.177.223，**5678，80**
192.168.0.2，13.250.177.223，**12345，80**	117.148.64.237，13.250.177.223，**5679，80**

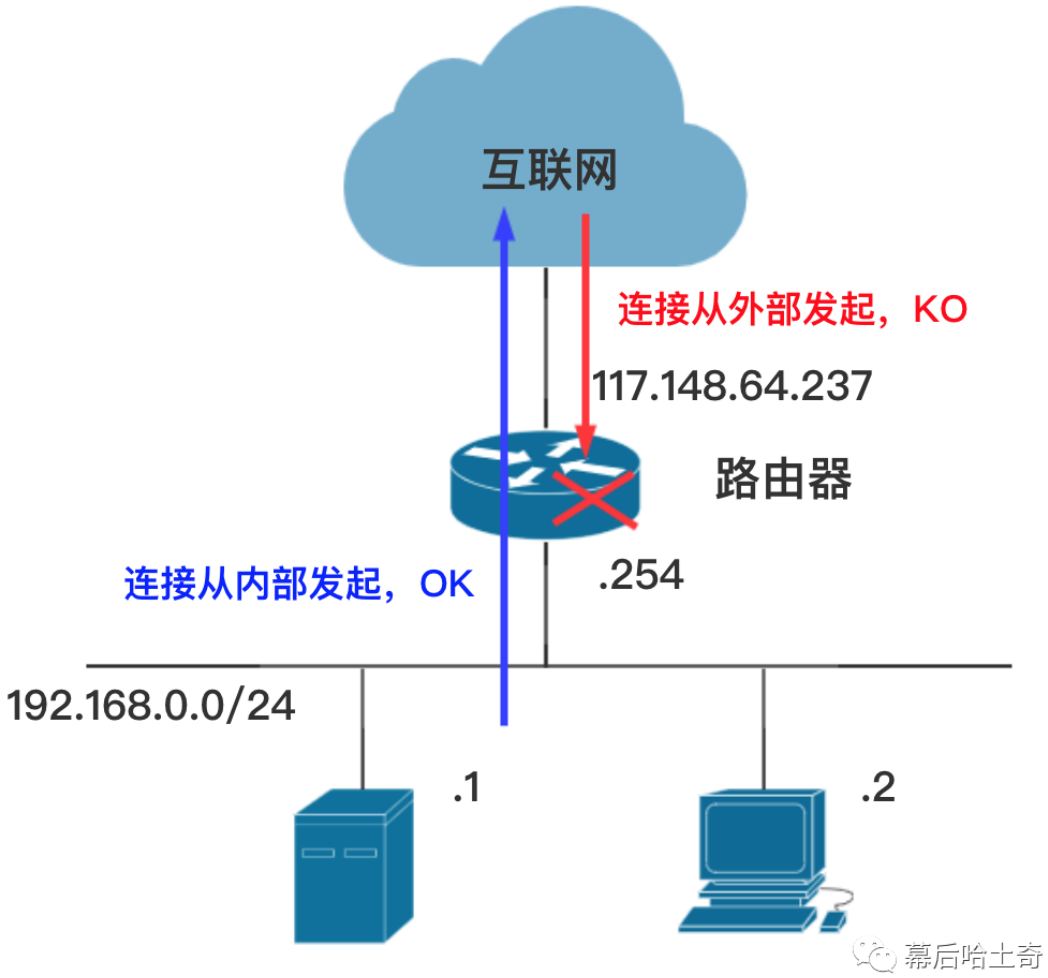
这样，当帧以目标端口号5678返回时，本地路由器就知道应该将其发往192.168.0.1这台机器；而当帧以目标端口号5679返回时，就将其发往192.168.0.2这台机器。

由于是路由器自己选择源端口，因此我们可以确定永远不会存在两个相同的源端口！

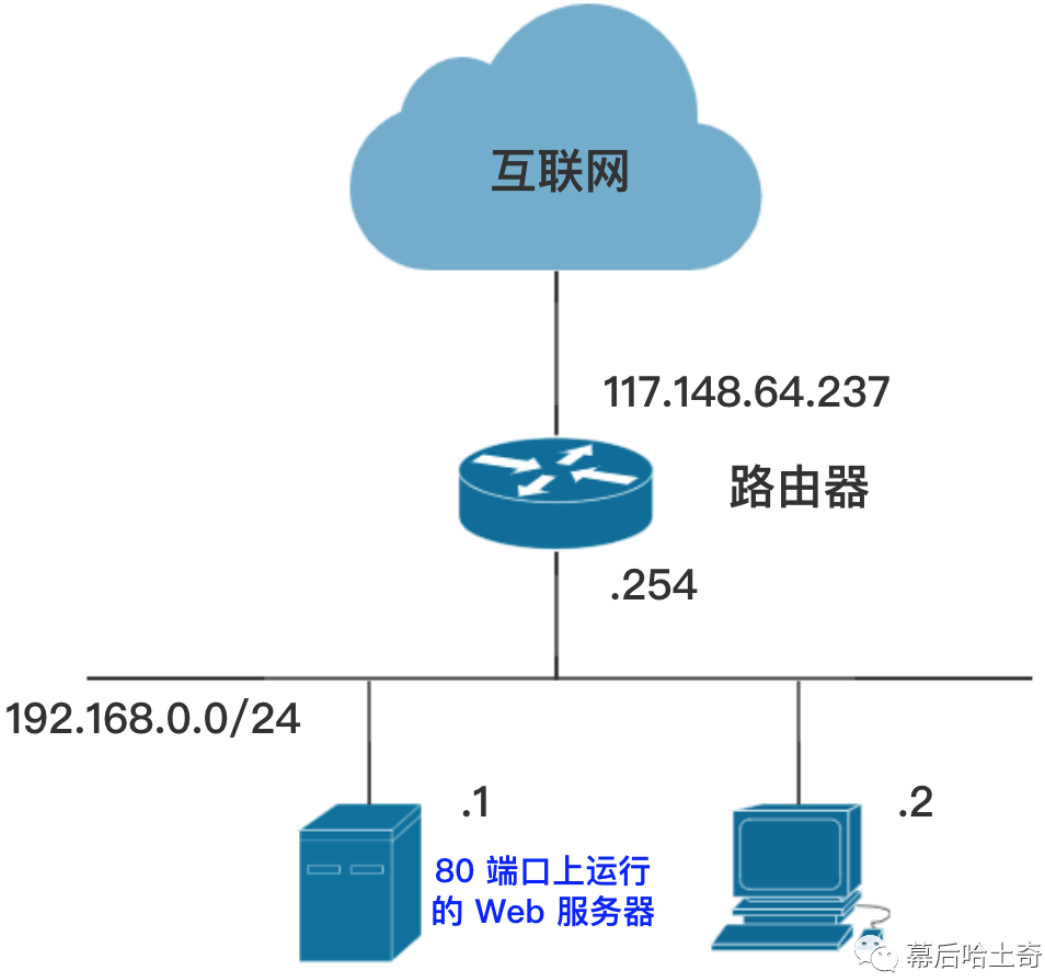
三、互联网访问咱们内网机器上的服务

我们内网的机器能访问互联网了，当然很棒了，不过，要是能被互联网访问则可能更有意思，比如我们作为服务端被访问。

当连接是由局域网内部的机器首先发起时，路由器可以用NAT表记下连接的信息，从而在返回时将响应帧发给正确的接收方机器。但是，当连接的第一个帧是先从外部到达时，路由器将无法知道这个帧是要发给局域网中的哪一台机器，因为此时路由器中未记录对应信息。



假设，我们在局域网的机器上运行一个服务程序。例如，在机器192.168.0.1的端口80上运行Web服务器。



这个时候，就会很尴尬了，因为外面访问我们只能通过本地路由器的公有 IP 地址 117.148.64.237，要访问80端口，有两种情况：

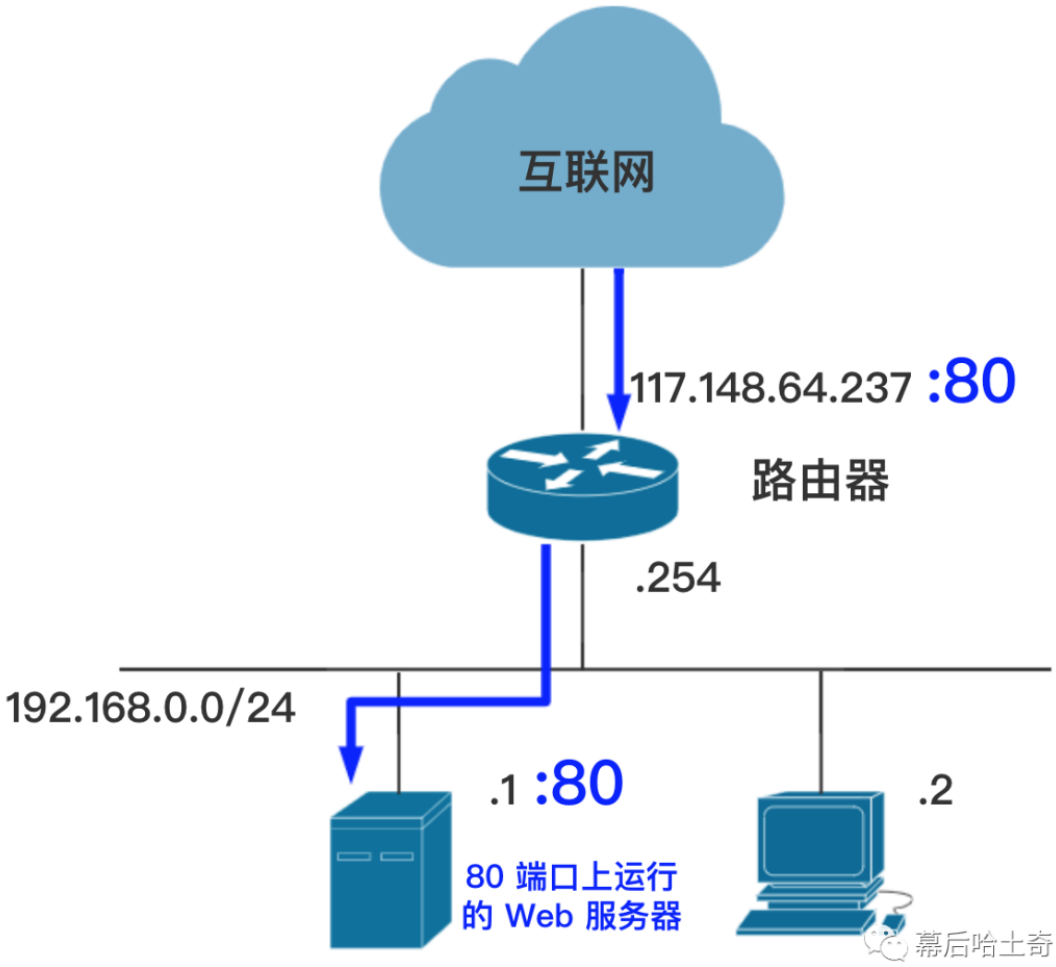
- 本地路由器上有一个 Web 服务器，此人将访问它；
- 本地路由器上没有 Web 服务器，路由器将返回设置了 RST 标志位的 TCP 报文段。

也就是说，**这个时候根本没有办法到达实际的本地机器192.168.0.1上运行的网站**。这个问题如何解决呢？答案是端口转发。即让本地路由器根据请求的端口将请求专门重定向到目标机器。

因此，**我们将告诉本地路由器，到达其 80 端口的任何数据包都必须被重定向到 IP 地址为192.168.0.1的机器的 80 端口上**。端口转发也有一个表，被称为端口转发表。我们目前的端口转发表如下所示：

协议	外部 IP 地址	外部端口号	内部 IP 地址	内部端口号
TCP	117.148.64.237	80	192.168.0.1	80

这样，任何对本地路由器的公有 IP 地址的 80 端口的访问，都将被自动重定向到局域网中的192.168.0.1这台机器上的 Web 服务器。



转口转发还有额外的好处。如果把电脑比喻为一个房子，房子的门窗就相当于端口，门窗越多，小偷进家的可能性就越大，就越不安全。我们使用NAT 和端口转发的主要好处之一是仅使必要的内容可被访问。在我们的例子中，我们只为 80 端口设置了端口转发，使得局域网的机器上的 Web 服务器可以被互联网访问，但机器上的其他端口却不能被互联网访问。

可以看到，除了解决 IP 地址短缺的问题之外，NAT 和端口转发通过仅使必需的端口可被访问，还带来了网络安全的巨大改进。

四、动态NAT总结

NAT 技术的重点提炼：

1. 动态 NAT 允许局域网上拥有私有 IP 地址的机器借助路由器的公有 IP 地址访问互联网，本地路由器在局域网和互联网之间建立了连接。
2. 这样可以节省大量的 IP 地址，因为它仅使用一个公有 IP 地址，却可以服务私有网络上的所有机器。
3. 路由器上有一张表来映射内网的ip、端口与公网ip、端口，从而可以让内网主动访问互联网的回复报文顺利知道是谁发出的。

4. 利用端口转发可以实现互联网访问内网某个机器上的服务。
5. 端口转发还可以提高网络安全性。

本篇文章只说明了NAT的功能和优点，但是它的缺点同样受到很多人的诟病，比如它隐藏了发送报文的主机的有关信息，使得外部网络难于对它们进行管理，例如，若内部网络中某台主机在Internet上违反安全规则，Internet就无法查处“元凶”。

感兴趣的读者朋友可以去详细了解下其若干缺陷和可能产生的问题。

五、网络层结语

亲爱的读者朋友们，网络层篇就要暂告一段落了，回望TCP/IP五层模型，我们已经逐一攻占了物理层、数据链路层以及网络层,接下来的旅程即将就是大名鼎鼎的传输层，也是跟我们最终应用层最近的一层，这一层中将来实现可靠传输，就是伟大的TCP协议。

朋友们，接下来咱们传输层见！