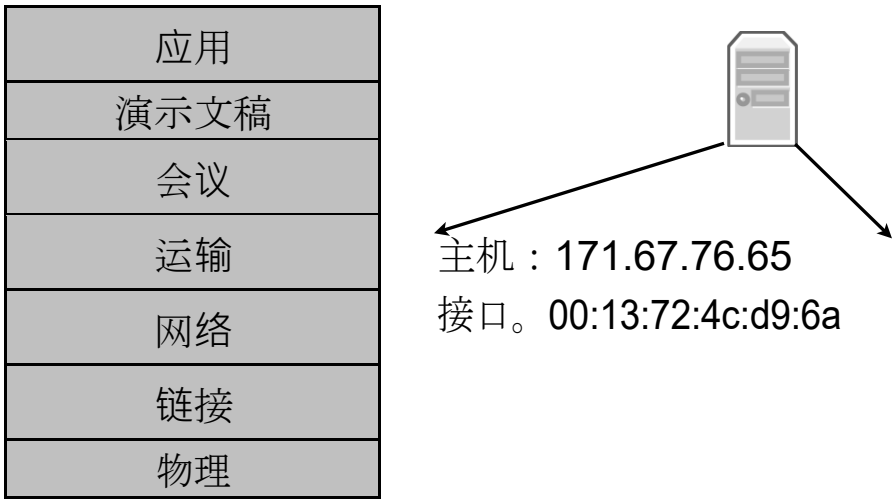
地址解析协议 (ARP)

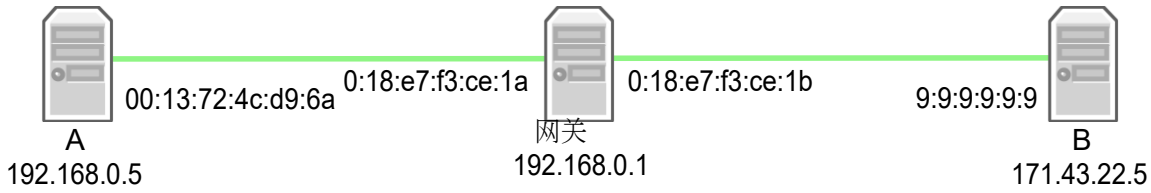
地址解析协议，即ARP，是网络层发现与它直接连接的网络地址相关的链接地址的机制。换句话说，它是一个设备如何获得问题的答案。"我有一个IP数据包，它的下一跳是这个地址--我应该把它发送到哪个链路地址？"

地址的层数



之所以需要ARP，是因为每个协议层都有自己的名称和地址。IP地址是一个网络层的地址。它描述了一个主机，一个在网络层的唯一目的地。相反，一个链路地址描述了一个特定的网卡，一个发送和接收链路层帧的独特设备。例如，以太网有48位地址。每当你买了一块以太网卡，它就已经被预设了一个独特的以太网地址。因此，一个IP地址说的是 "这个主机"，而一个以太网地址说的是 "这个以太网卡"。

解决问题

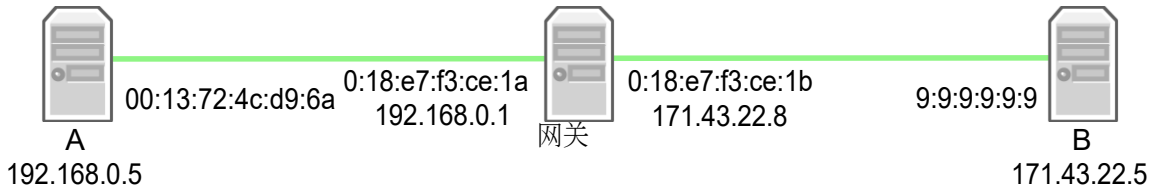


48位以太网地址通常写成以冒号为界的6个八位数，以十六进制书写，如源地址为0:13:72:4c:d9:6a，目的地址为9:9:9:9:9:9。

有一件事可能会让人困惑，那就是虽然这些链路层和网络层的地址在协议层方面是完全脱钩的，但在分配和管理方面，它们可能不是这样。例如，一台主机拥有多个IP地址是很常见的，每个接口都有一个。因为有了网络掩码的概念，它需要这样做。例如，请看这个假设的设置。网关在中间，有一个单一的IP地址。192.168.0.1.它有两个网卡，一个连接到目的地171.43.22.5，一个连接到源192.168.0.5。

192.168.0.1这个地址实际上只能在这些网络中的一个，即源网络。192.168.0.1与171.43.22.5在同一个网络中所需要的网络掩码是128.0.0.0，或者说只有一个比特的网络掩码!但不可能所有第一位为1的IP地址都与171.43.22.5在同一个网络中 --例如，192.168.0.5需要通过网关到达。

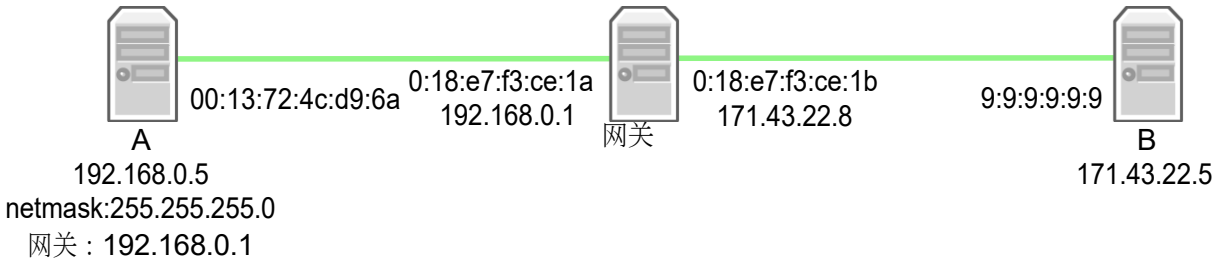
寻址实例



因此，我们经常看到这样的设置，即网关或路由器有多个接口，每个接口都有自己的链路层地址来识别该卡，同时也有自己的网络层地址来识别该卡所在网络中的主机。对于网关来说，左边的接口有IP地址192.168.0.1，而右边的接口有IP地址171.43.22.8。

链接层和网络层地址在逻辑上是解耦的，但在实践中是耦合的，这在某种程度上是一个历史遗留物。当互联网开始时，有许多链接层，它希望能够在所有的链接层之上运行。这些链接层不会突然开始使用IP地址而不是他们自己的寻址方案。此外，事实证明，在很多情况下，拥有一个独立的链接层地址是非常有价值的。例如，当我在斯坦福大学的网络中注册一台计算机时，我注册了它的链接层地址
--网卡的地址。

封装

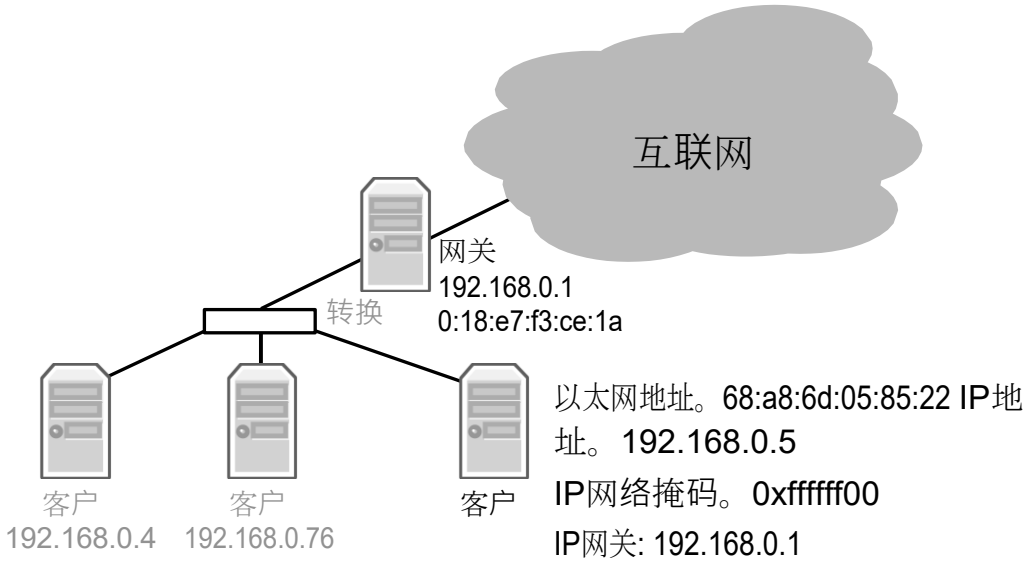


那么，这在实践中是什么意思呢？假设左边的节点A想向右边的节点B发送一个数据包。它将生成一个源地址为192.168.0.5、目的地址为171.43.22.5的IP数据包。

节点A检查目标地址是否在同一网络中。网络掩码告诉它，目的地址是在不同的网络中：255.255.255.0。这意味着节点A需要通过网关，即192.168.0.1发送数据包。为此，它发送一个数据包，其网络层目的地是171.43.22.5，但其链接层目的地是网关的链接层地址。所以这个数据包的网络层目的地是171.43.22.5，链接层目的地是0:18:e7:f3:ce:1a。网络层源是192.168.0.5，链接层源是0:13:72:4c:d9:6a。

所以我们有一个从A到B的IP包，被封装在一个从A到左边网关接口的链接层帧内。当数据包到达网关时，网关查找下一跳，决定是节点B，并把IP数据包放在到B的链接层帧内。

问题实例



因此，在这里我们得到了ARP解决的问题。我的客户知道它需要通过IP地址为192.168.0.1的网关发送一个数据包。然而，要做到这一点，它需要有与192.168.0.1相关的链接层地址。它如何获得这个地址呢？我们需要以某种方式将第三层（网络层）地址映射到其相应的第二层（链路层）地址。我们用一个叫做ARP的协议来做这件事，也就是地址解析协议。

地址解析协议

- 生成第二层和第三层地址之间的映射关系
  - 节点缓存映射，缓存条目过期
- 简单的请求-回复协议
  - "谁有网络地址X？"
  - "我有网络地址X"。
- 向链接层广播地址发送请求
- 向请求地址发送回复（非广播）。
- 数据包格式包括多余的数据
  请求有足够的信息来生成一个映射
  - 使得调试工作更加简单
- 没有 "分享 "国家：坏的国家最终会死亡

ARP是一个简单的请求-回复协议。每个节点都保留了一个从其网络上的IP地址到链路层地址的映射的缓存。如果一个节点需要向它没有映射的IP地址发送一个数据包，它就会发出一个请求。"谁有网络地址X？"拥有该网络地址的节点回应说："我有网络地址X。"响应包括链接层地址。收到响应后，请求者可以生成映射并发送数据包。

为了使每个节点都能听到请求，一个节点向一个链路层广播地址发送请求。网络中的每个节点都会听到这个数据包。

此外，ARP的结构使其包含多余的数据。请求包含请求者的网络和链接层地址。这样，当节点听到请求时（因为是广播），它们可以在它们的缓存中插入或刷新一个映射。节点*只对自己的请求作出回应。这意味着，假设没有人错误地生成数据包，你可以为另一个节点生成映射的唯一方法是响应该节点发送的数据包。因此，如果该节点崩溃或断开连接，其状态将不可避免地离开网络，当所有的缓存映射过期。这使得ARP的调试和故障排除更加容易。

那么，这些动态发现的映射会持续多久？这取决于设备：例如，某些版本的Mac OSX将它们保持20分钟，而某些思科设备使用4小时的超时。我们的假设是，这些映射不会经常变化。

ARP数据包格式（RFC826）

灰色区域是可变长度的，由长度区域决定

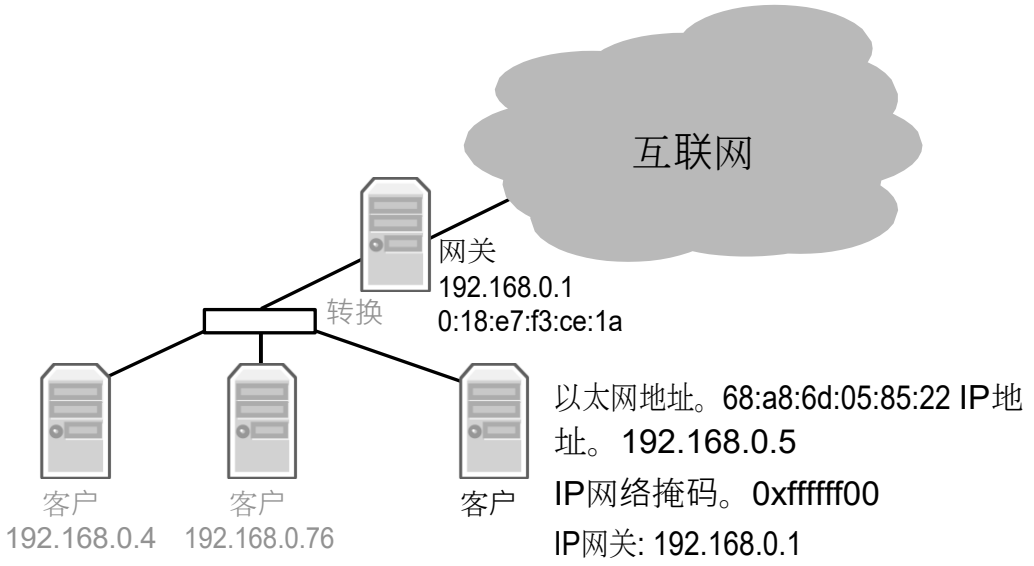
这就是ARP数据包的实际样子。它有10个字段。硬件字段说明这个请求或响应是针对哪个链路层的。协议字段说明这个请求或响应是针对什么网络协议的。长度字段规定了链路层和网络层地址的字节数。操作码指定该数据包是请求还是响应。

四个地址栏是用来请求和指定映射的。

记住，所有这些字段都是按网络顺序存储的，或者说是大恩典。因此，如果我的操作码是15，它将在操作码字段中被存储为0x000f。

ARP的全部细节在IETF Request for Comments, RFC, 826中。我只讲一个简单的请求/应答交换。

ARP请求

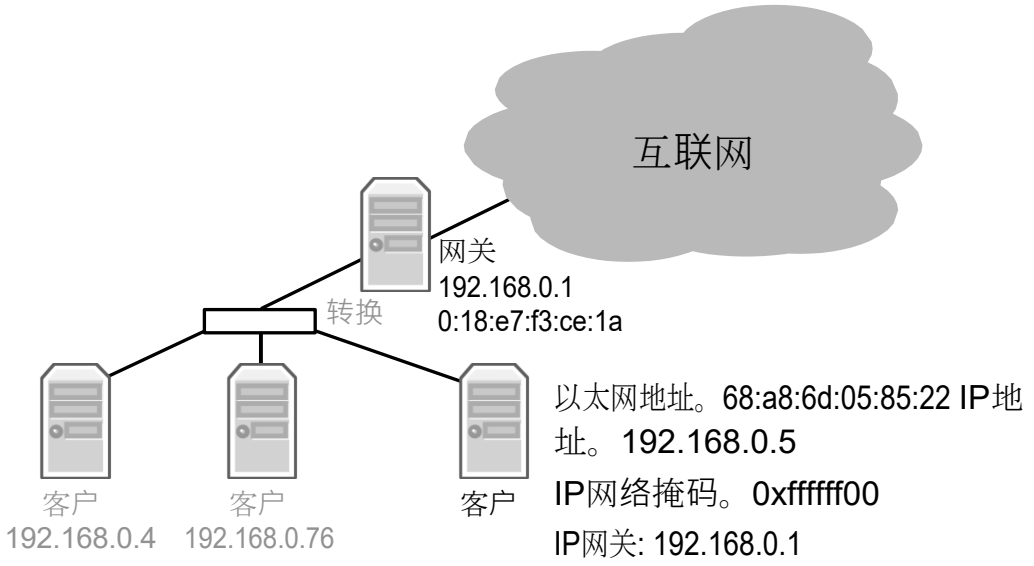


因此，假设我们的客户想通过其网关向更广泛的互联网发送一个数据包。但它没有该网关的以太网地址。

客户端将产生一个ARP请求包，其链路层源地址为其地址。68:a8:6d:05:82:22。目的链路层地址是广播地址：ff:ff:ff:ff:ff:ff:ff，都是1。

ARP请求将指定硬件是以太网，值为1，协议是IP，值为0x0800，硬件地址长度为6，协议长度为4。 操作码将是请求，其值为1。 ARP源硬件字段将是请求者的以太网地址，68：a8：6d：05：85：22。源协议字段是请求者的IP地址。192.168.0.5.目的硬件地址可以设置为任何内容--它是数据包试图发现的内容。目的协议地址是客户试图找到一个映射的地址。192.168.0.1.客户端在以太网上发送这个帧。网络中的每个节点都会收到它，并刷新其链路地址68:a8:6d:05:85:22和网络地址192.168.0.5之间的映射，如果它没有映射，则插入一个映射。

ARP回复



网关看到该请求是针对其IP地址的，因此产生了一个回复。

与请求一样，ARP回复将指定硬件是以太网，其值为1，协议是IP，其值为0x0800，硬件地址长度为6，协议长度为4。 操作码将是回复，其值为2。
ARP源硬件字段将是回复者的以太网地址，0:18:e7:f3:ce:1a。源协议字段是答案。192.168.0.1.目标硬件地址是请求的源硬件地址：68：a8：6d：05：85：22。目标协议地址是请求的源协议地址：192.168.0.5。

这是一个开放的问题，你把响应发送到什么链路层地址。最初的ARP规范说，回复者应该把它发送到请求者的链路层地址，所以是单播。然而，现在普遍的做法是广播，因为如果映射需要改变，这样做可以更积极地替换缓存条目。节点也可以发送所谓的无偿ARP数据包，请求不存在的映射，以便在网络上宣传自己。

因此，我们已经看到，为了路由数据包，人们需要能够将网络层地址映射到链接层地址。地址解析协议，即ARP，通过一个简单的请求-回复交换来提供这种服务。如果一个节点需要向或通过一个它没有的链接层地址的IP地址发送数据包，它可以通过ARP请求该地址。