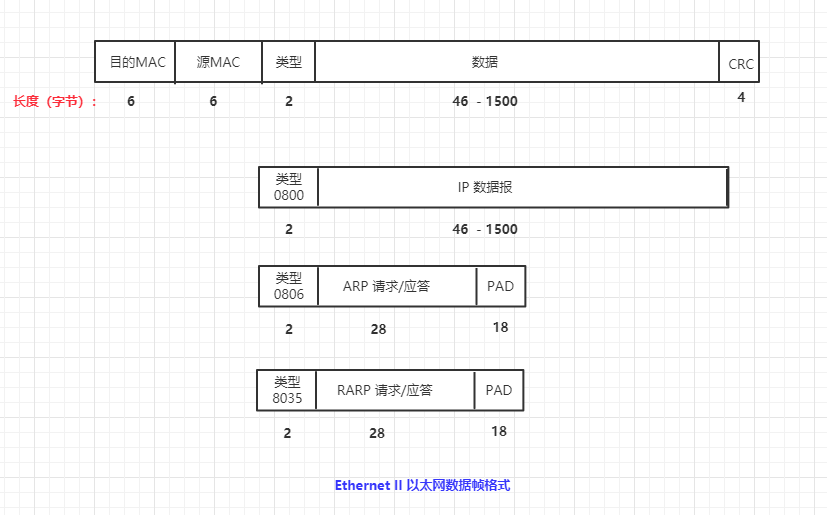
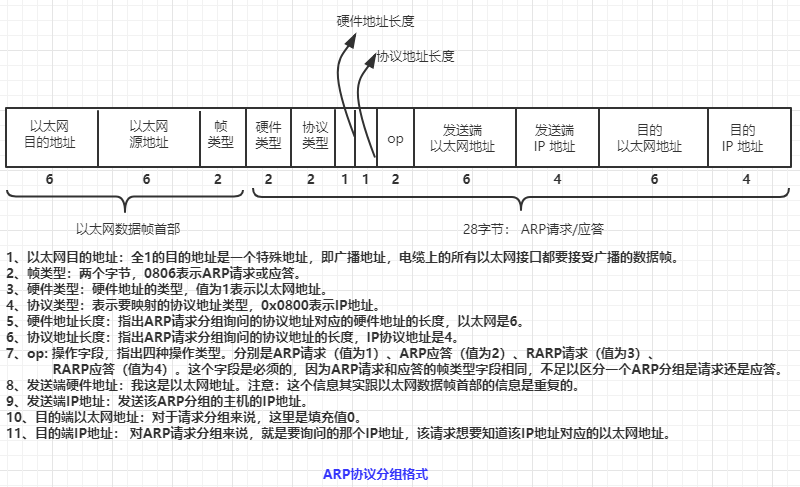
链路层

1. Ethernet II 以太网数据帧格式



网络层

1. ARP协议



* 1. 数据链路层寻址

数据链路层协议（比如上面的Ethernet II 以太网协议）有自己的寻址机制（通常是48 bit地址，比如Ethernet II中的MAC地址），任何使用链路层的网络层都必须遵从。

当一台使用TCP/IP协议的主机把以太网数据帧发送到位于同一局域网上的另一台主机时（不论发送方最终的目的主机是该主机，还是想让该主机负责转发），是根据48 bit的以太网地址来确定目的接口的。设备驱动程序从不检查IP数据报中的目的IP地址。

* 1. ARP协议： 地址映射

ARP协议就是为IP地址到对应的任何链路层使用的硬件地址之间提供动态映射。之所以用动态这个词是因为这个过程是自动完成的，一般应用程序用户或系统管理员不必关心。

* 1. ARP 高速缓存

每台主机上都有一个ARP高速缓存。

这个高速缓存存放了最近的Internet地址到硬件地址之间的映射记录。高速缓存中每一项的生存时间一般为20分钟，起始时间从被创建时开始算起。这个高速缓存是ARP高效运行的关键。

用命令arp -a可以查看目前ARP高速缓存中所有的记录。

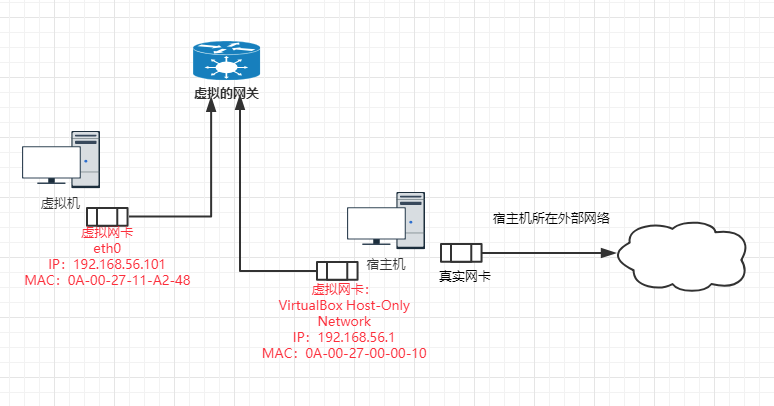
例如我在一台虚拟机启动后立即输入arp -a:

发现该主机的ARP告诉缓存是空的。此时尝试执行ping命令。

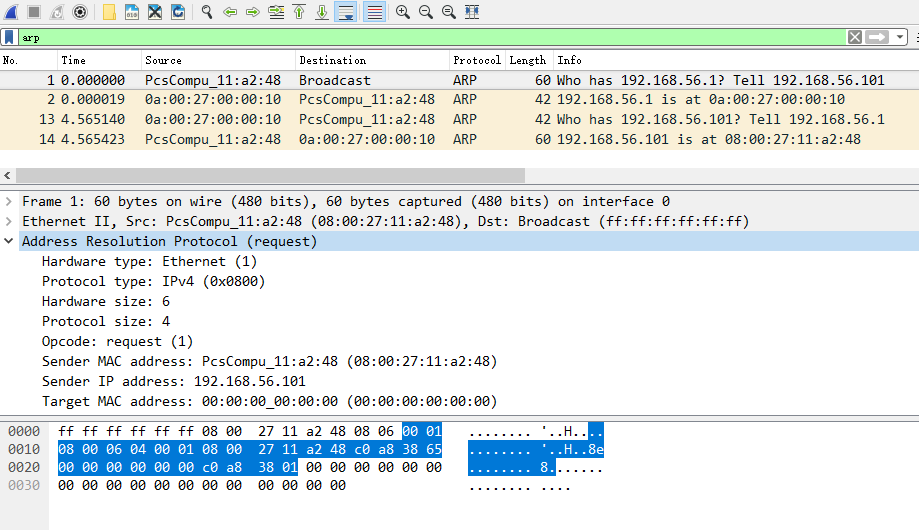
ping会发送一份ICMP回显请求报文给对方主机，并且等待回显应答。而ICMP报文是在IP数据报中传输的，它通常也被认为处于网络层。

也就是说，ping一台主机时，也会通过ARP协议获取对方的链路层地址。

我这里是在一个由一台VirtualBox中的虚拟机和宿主机组成的局域网中实验。拓扑图如下：



在虚拟机192.168.56.101中，ping 192.168.56.1，在宿主机中的wireshark中，对虚拟网卡进行抓包，并且过滤arp：



发现，总共抓到4个包：

1. 虚拟机192.168.56.101发了ARP请求。
2. 宿主机接受到一个以太网数据帧。

2.1 查看以太网数据帧的前6个字节，也就是目的地址，发现是全1。也就是说这是一个特殊地址：广播地址。电缆上的所有以太网接口都要接受广播的数据帧。目的地址后面的6个字节是数据帧源地址：08:00:27:11:a2:48。

2.2 紧接着，源地址后面的两个字节是类型0806，表示这是一个ARP请求或应答分组。

2.3 再后面两个字节，即ARP协议的硬件类型字段，其值为0x0001，表示这个ARP分组询问的硬件地址是以太网地址。

2.4 再后面两个字节，即ARP协议的协议类型字段，其值为0x0800，表示要映射的协议地址类型是IP地址。这个值与包含IP数据报的以太网数据帧中的类型字段的值相同，这是有意设计的。

2.5 接下来的两个字节，分别是ARP协议的硬件地址长度和协议地址长度这两个字段，分别指出硬件地址和协议地址的长度，以字节为单位。对于以太网上IP地址的ARP请求或应答来说，它们的值分别为6和4。Wireshark抓包数据中，这两个字节数据确实分别是0x06和

0x04。

2.6 接下来的两个字节，是ARP协议的op操作字段，值为0x0001，表示这是一个ARP请求分组。

2.7 接下来的6个字节是发送端的硬件地址，这里是以太网地址：08:00:27:11:a2:48。注意，这里有一些重复信息：在以太网的数据帧报头中和ARP请求分组数据中都有发送端的硬件地址。

2.8 接下来的4个字节是发送端的IP地址0xc0a83865。转换为点分十进制，就是192.168.56.101（http://www.ab126.com/system/2859.html这里可以转换）。

2.9 接下来的6个字节是目的端的硬件地址，这里可以看到，被填充成0，因为这是一个ARP请求分组。

2.10 接下来的四个字节是目的端IP地址，值为0xc0a83801，转换为点分十进制，就是192.168.56.1。

至此，这个ARP请求分组就解析完，剩下后面的全部是以太网数据帧的padding数据。

1. 宿主机解析完ARP请求分组，发现该分组内的目的IP地址就是自己的IP地址，那么宿主机就要回复一个ARP应答。

看wireshark抓的第二个数据帧，发现宿主机确实回了一个ARP应答，并填上了自己的硬件地址和IP地址。

**并且：从抓包数据中，还能看出，宿主机发送了ARP应答之后，也发送了一个ARP请求给之前的源主机。**