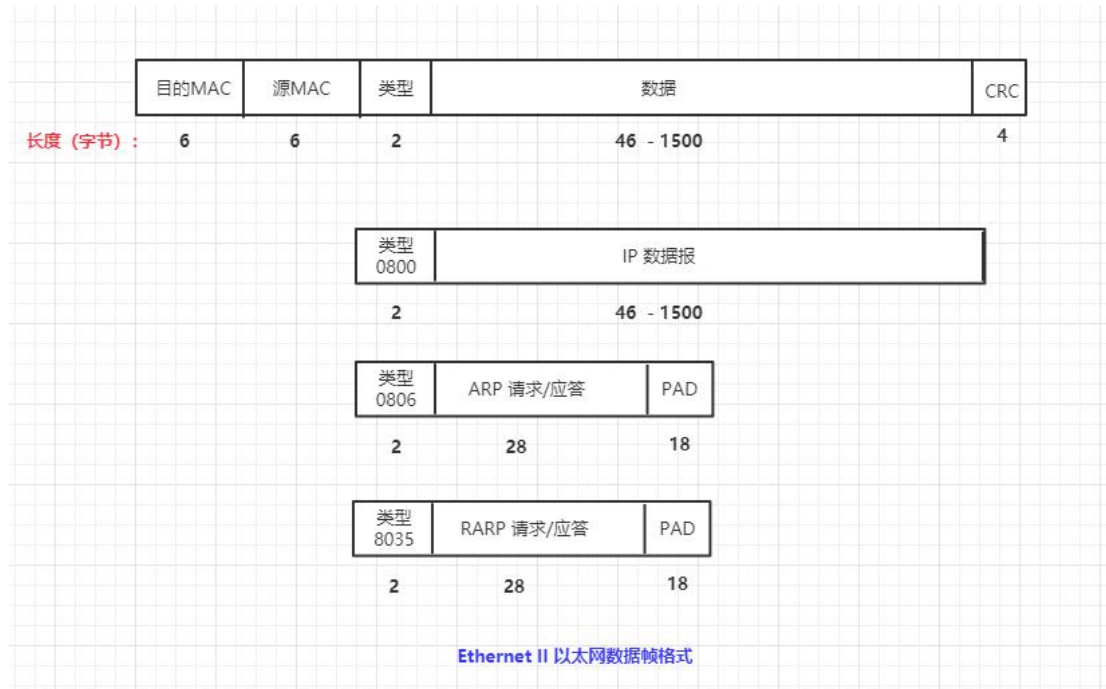


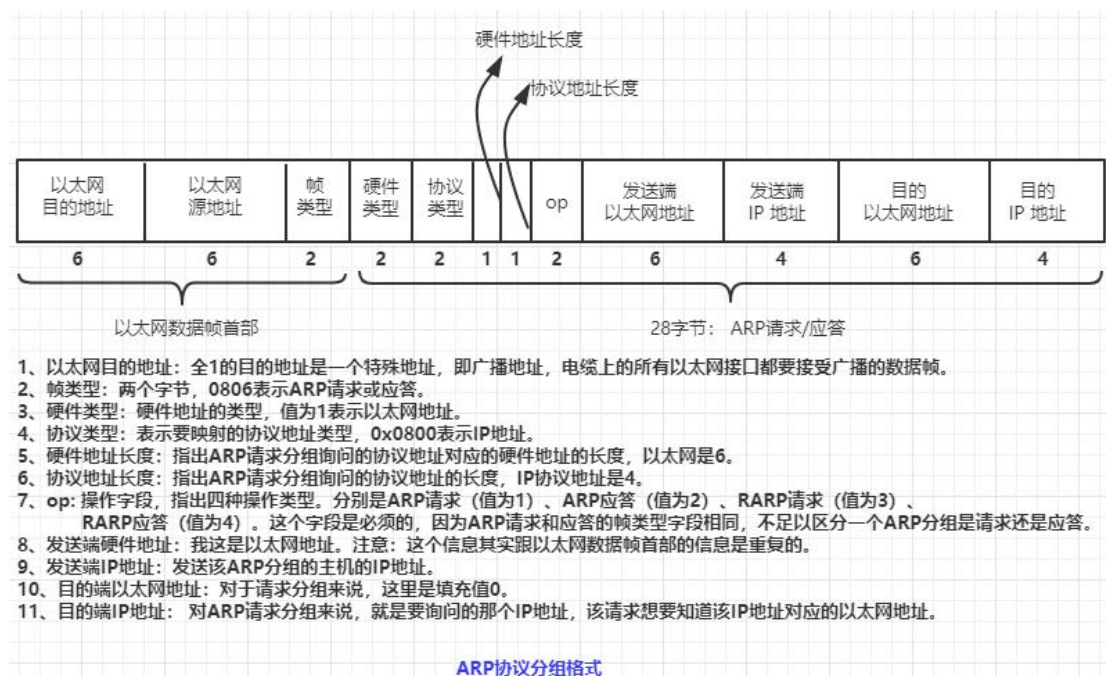
链路层

1、Ethernet II 以太网数据帧格式



网络层

1、ARP 协议



1.1 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。

1.2 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。

1.3 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。

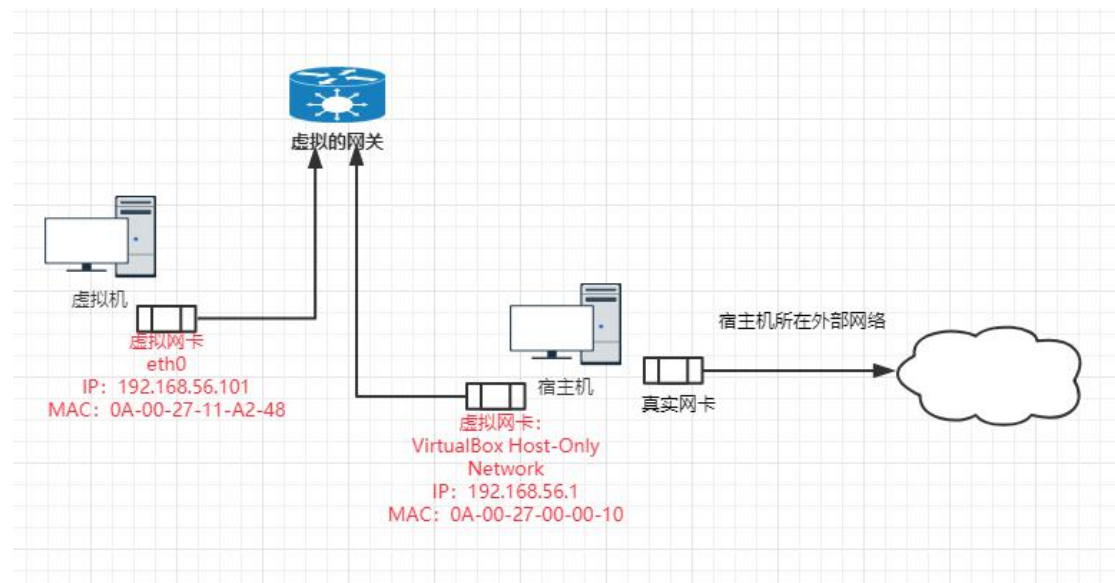
例如我在一台虚拟机启动后立即输入 `arp -a`:

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`:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_11:a2:48	Broadcast	ARP	60	Who has 192.168.56.1? Tell 192.168.56.101
2	0.000019	0a:00:27:00:00:10	PcsCompu_11:a2:48	ARP	42	192.168.56.1 is at 0a:00:27:00:00:10
13	4.565140	0a:00:27:00:00:10	PcsCompu_11:a2:48	ARP	42	Who has 192.168.56.101? Tell 192.168.56.1
14	4.565423	PcsCompu_11:a2:48	0a:00:27:00:00:10	ARP	60	192.168.56.101 is at 08:00:27:11:a2:48

>	Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
>	Ethernet II, Src: PcsCompu_11:a2:48 (08:00:27:11:a2:48), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
>	Address Resolution Protocol (request)
	Hardware type: Ethernet (1)
	Protocol type: IPv4 (0x0800)
	Hardware size: 6
	Protocol size: 4
	Opcode: request (1)
	Sender MAC address: PcsCompu_11:a2:48 (08:00:27:11:a2:48)
	Sender IP address: 192.168.56.101
	Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)

0000	ff ff ff ff ff 08 00 27 11 a2 48 08 06 00 01 '..H..
0010	08 00 06 04 00 01 08 00 27 11 a2 48 c0 a8 38 65 '..H..8e
0020	00 00 00 00 00 00 c0 a8 38 01 00 00 00 00 00 00 8.....
0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

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。

3、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。