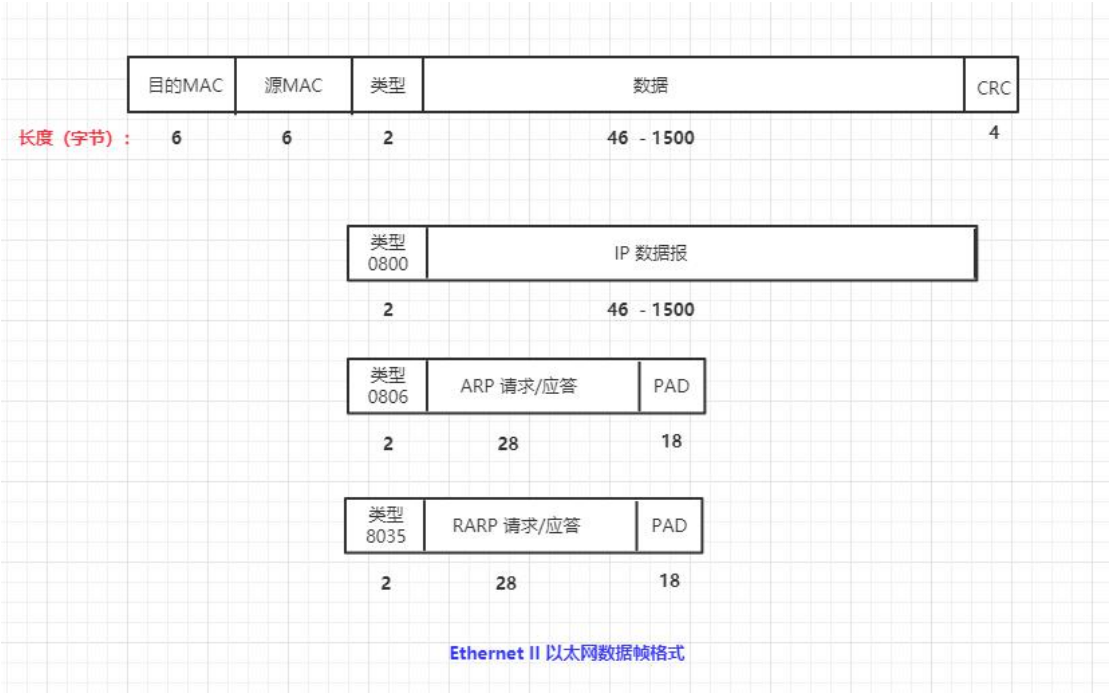
链路层

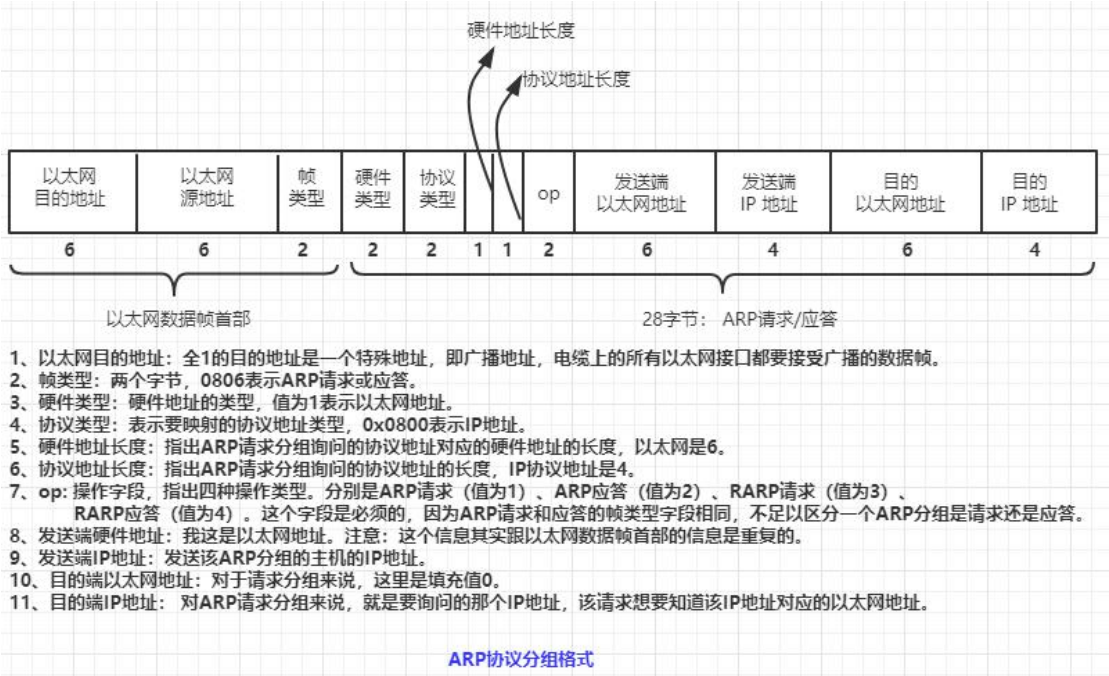
1、 Ethernet II 以太网数据帧格式



Ethernet II 以太网数据帧格式

网络层

1、ARP 协议



1、以太网目的地址：全1的目的地址是一个特殊地址，即广播地址，电缆上的所有以太网接口都要接受广播的数据帧。
2、帧类型：两个字节，0806表示ARP请求或应答。
3、硬件类型：硬件地址的类型，值为1表示以太网地址。
4、协议类型：表示要映射的协议地址类型，0x0800表示IP地址。
5、硬件地址长度：指出ARP请求分组询问的协议地址对应的硬件地址的长度，以太网是6。
6、协议地址长度：指出ARP请求分组询问的协议地址的长度，IP协议地址是4。
7、op：操作字段，指出四种操作类型。分别是ARP请求（值为1）、ARP应答（值为2）、RARP请求（值为3）、RARP应答（值为4）。这个字段是必须的，因为ARP请求和应答的帧类型字段相同，不足以区分一个ARP分组是请求还是应答。
8、发送端硬件地址：我这是以太网地址。注意：这个信息其实跟以太网数据帧首部的信息是重复的。
9、发送端IP地址：发送该ARP分组的主机的IP地址。
10、目的端以太网地址：对于请求分组来说，这里是填充值0。
11、目的端IP地址： 对ARP请求分组来说，就是要询问的那个IP地址，该请求想要知道该IP地址对应的以太网地址。

ARP协议分组格式

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。

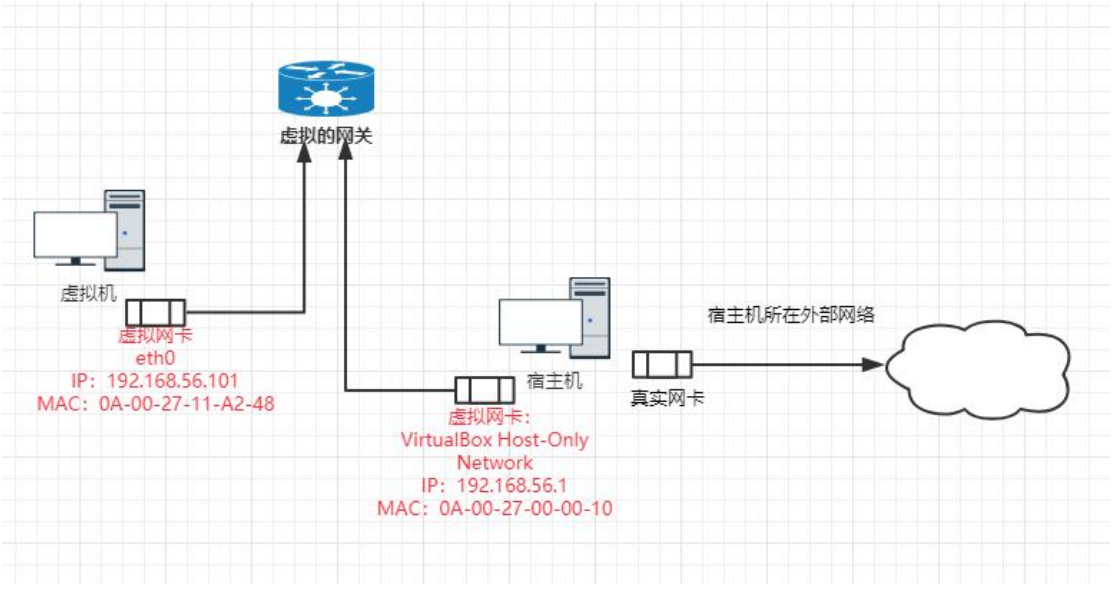
例如我在一台虚拟机启动后立即输入 arp -a:

发现该主机的 ARP 告诉缓存是空的。此时尝试执行 ping 命令。

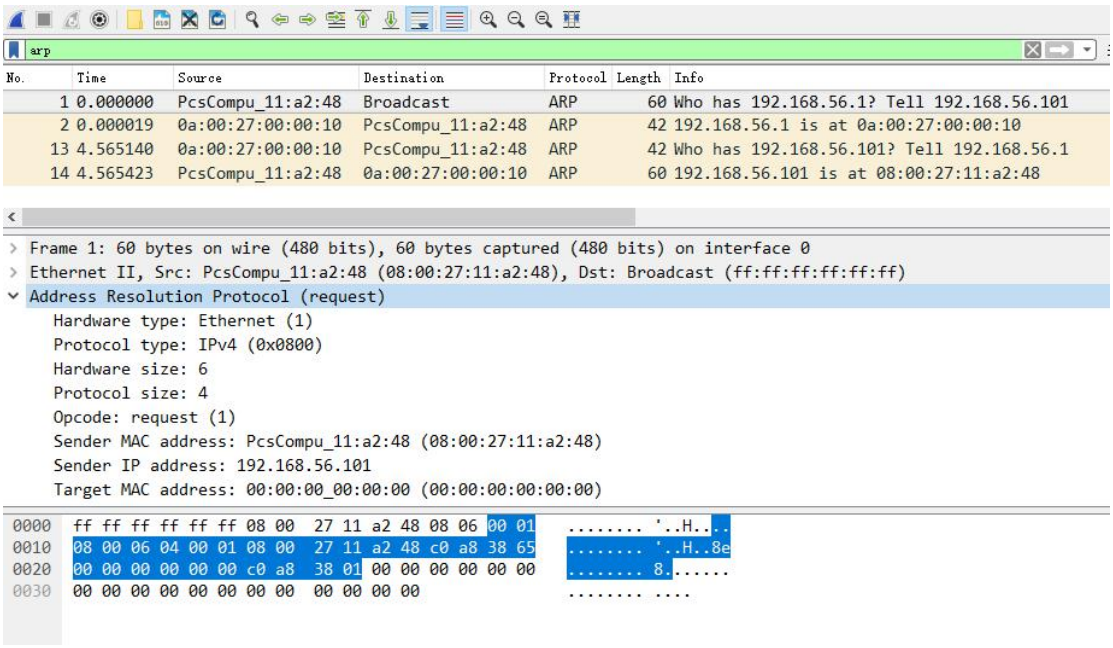
ping 会发送一份 ICMP 回显请求报文给对方主机，并且等待回显应答。而 ICMP 报文是在 IP 数据报中传输的，它通常也被认为处于网络层。

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：



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。

此时，再在虚拟机 192.168.56.101 中输入 arp -a，会发现系统 ARP 高速缓存已经多了一条记录：

？ 192.168.56.1 at 0a:00:27:00:00:10 [ether] on eth0

我们再尝试 ping 一台不存在当前局域网内的主机：ping 192.168.56.111

ping 程序会提示目标不可达。此时再次查看 ARP 高速缓存，会发现多了一条记录：

？ 192.168.56.111 at [incomplete] on eth0

可见，对不存在的主机（或者已经关机）发送 ARP 请求，ARP 高速缓存中会增加一条不完整的表项。

在 ARP 高速缓存中的表项一般都要设置超时值。从伯克利系统演变而来的系统一般对完整的表项设置超时值为 20 分钟，而对不完整的表项设置超时值为 3 分钟。

1.4 ARP 代理

如果 ARP 请求是从一个网络的主机发往另一个网络上的主机，那么连接这两个网络的路由器（具有 ARP 代理功能的路由器）就可以回答该请求，这个过程称作委托 ARP 或 ARP 代理(Proxy ARP)。这样可以欺骗发起 ARP 请求的发送端，使它误以为路由器就是目的主机，而事实上目的主机是在路由器的“另一边”。路由器的功能相当于目的主机的代理，把分组从其他主机转发给它。

这样那台源主机中的 ARP 高速缓存中就可能会有两条记录，其中的 IP 地址（记为 IP_a）IP_a 为另一个网络中的主机地址，IP_b 是本网络的 ARP 代理路由器地址，这两个 IP 地址都映射到了同一个硬件地址：ARP 代理路由器的地址。通常这是委托 ARP 的线索。

1.5 免费 ARP

另一个 ARP 特性称作免费 ARP(gratuitous ARP)。它是指主机发送 ARP 查找自己的 IP 地址。通常，它发生在系统引导期间进行接口配置的时候。

免费 ARP 可以有两个方面的作用：

1.5.1 一个主机可以通过它来确定另一个主机是否设置了相同的 IP 地址。主机如果收到一个回答，那么就会在终端日志上产生一个错误消息“以太网地址：a:b:c:d:e:f 发送来重复的 IP 地址”。这样就可以警告系统管理员，某个系统有不正确的设置。

1.5.2 如果发送免费 ARP 的主机正好改变了硬件地址（很可能是主机关机了，并换了一块接口卡，然后重新启动），那么这个分组就可以使其他主机高速缓存中旧的硬件地址进行相应的更新。一个比较著名的 ARP 协议事实[Plummer 1982]是，如果主机收到某个 IP 地址的 ARP 请求，而且它已经在接收者的高速缓存中，那么就要用 ARP 请求中的发送端硬件地址（如以太网地址）对高速缓存中相应的内容进行更新。主机接收到任何 ARP 请求都要完成这个操作（ARP 请求是在网上广播的，因此每次发送 ARP 请求时网络上的所有主机都要这样做）。

但是我这的 openSUSE 虚拟机启动时并没有发免费 ARP 请求。

2、RARP 协议 ： 获取自己的 IP 地址

具有本地磁盘的系统引导时，一般是从磁盘上的配置文件中读取 IP 地址。但是无盘机，如 X 终端或无盘工作站，则需要采用其他方法来获得 IP 地址。

网络上的每个系统都具有**唯一的硬件地址**，它是由网络接口生产厂家配置的。无盘系统的 RARP 实现过程是从接口卡上读取唯一的硬件地址，然后发送一份 RARP 请求（一帧在网络上广播的数据），请求某个主机（通常是一个 RARP 服务器）响应该无盘系统的 IP 地址（在 RARP 应答中）。

2.1 RARP 分组格式

RARP 分组的格式与 ARP 分组基本一致。它们之间主要的差别是 RARP 请求或应答的帧类型字段为 0x8035，而且 RARP 请求的操作代码为 3，应答操作代码为 4。

2.2 RARP 服务器设计

RARP 服务器实现的一个复杂因素是 RARP 请求是在硬件层上进行广播的。这意味着它们不经过路由器进行转发。为了让无盘系统在 RARP 服务器关机的状态下也能引导，通常在一个网络上（例如一根电缆）要提供多个 RARP 服务器。

当服务器的数目增加时（以提供冗余备份），网络流量也随之增加，因为每个服务器对每个 RARP 请求都要发送 RARP 应答。发送 RARP 请求的无盘系统一般采用最先收到的 RARP 应答（对于 ARP，我们从来没有遇到这种情况，因为只有一台主机发送 ARP 应答）。另外，还有一种可能发生的情况是每个 RARP 服务器同时应答，这样会增加以太网**发生冲突**的概率。

如何防止这种冲突？

2.2.1 每个 RARP 服务器在发送一个响应之前可以延迟一个小的随机时间。

2.2.2 优化：可以指定一个 RARP 服务器为主服务器，其他的为次服务器。主服务器发出响应不需要延迟，而次服务器发出响应则需要一个随机的延迟。

2.2.3 优化：指定一个主 RARP 服务器，其他为次服务器。次服务器只对在一个短时间段内发生的重复请求进行响应。这里假设出现重复请求的原因是由于主服务器停机了。