

## 第5章 RARP：逆地址解析协议

### 5.1 引言

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），请求某个主机响应该无盘系统的 IP 地址（在 RARP 应答中）。

在概念上这个过程是很简单的，但是实现起来常常比 ARP 要困难，其原因在本章后面介绍。RARP 的正式规范是 RFC 903 [Finlayson et al. 1984]。

### 5.2 RARP 的分组格式

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

对应于 ARP，RARP 请求以广播方式传送，而 RARP 应答一般是单播(unicast)传送的。

### 5.3 RARP 举例

在互联网中，我们可以强制 sun 主机从网络上引导，而不是从本地磁盘引导。如果在主机 bsd1 上运行 RARP 服务程序和 tcpdump 命令，就可以得到如图 5-1 那样的输出。用 -e 参数使得 tcpdump 命令打印出硬件地址：

```
1  0.0                8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60:
                        rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
2  0.13 (0.13)        0:0:c0:6f:2d:40 8:0:20:3:f6:42 rarp 42:
                        rarp reply 8:0:20:3:f6:42 at sun
3  0.14 (0.01)        8:0:20:3:f6:42 0:0:c0:6f:2d:40 ip 65:
                        sun.26999 > bsd1.tftp: 23 RRQ "8CFC0D21.SUN4C"
```

图5-1 RARP请求和应答

RARP 请求是广播方式（第 1 行），而第 2 行的 RARP 应答是单播方式。第 2 行的输出中 at sun 表示 RARP 应答包含主机 sun 的 IP 地址（140.252.13.33）。

在第 3 行中，我们可以看到，一旦 sun 收到 IP 地址，它就发送一个 TFTP 读请求（RRQ）给文件 8CFC0D21.SUN4C（TFTP 表示简单文件传送协议。我们将在第 15 章详细介绍）。文件名中的 8 个十六进制数字表示主机 sun 的 IP 地址 140.252.13.33。这个 IP 地址在 RARP 应答中返回。文件名的后缀 SUN4C 表示被引导系统的类型。

tcpdump 在第 3 行中指出 IP 数据报的长度是 65 个字节，而不是一个 UDP 数据报（实际上是一个 UDP 数据报），因为我们运行 tcpdump 命令时带有 -e 参数，以查看硬件层的地址。在图 5-1 中

需要指出的另一点是, 第2行中的以太网数据帧长度比最小长度还要小(在4.5节中我们说过应该是60字节)。其原因是我们在发送该以太网数据帧的系统(bsd1)上运行tcpdump命令。应用程序rarpd写42字节到BSD分组过滤设备上(其中14字节为以太网数据帧的报头, 剩下的28字节是RARP应答), 这就是tcpdump收到的副本。但是以太网设备驱动程序要把这一短帧填充空白字符以达到最小传输长度(60)。如果我们在另一个系统上运行tcpdump命令, 其长度将会是60。

从这个例子可以看出, 当无盘系统从 RARP 应答中收到它的 IP 地址后, 它将发送 TFTP 请求来读取引导映像。在这一点上我们将不再进一步详细讨论无盘系统是如何引导的(第 16 章将描述无盘 X 终端利用 RARP、BOOTP 以及 TFTP 进行引导的过程)。

当网络上没有 RARP 服务器时, 其结果如图 5-2 所示。每个分组的目的地址都是以太网广播地址。在 who- 后面的以太网地址是目的硬件地址, 跟在 ell 后面的以太网地址是发送端的硬件地址。

请注意重发的频度。第一次重发是在 6.55 秒以后, 然后增加到 42.80 秒, 然后又减到 5.34 秒和 6.55 秒, 然后又回到 42.79 秒。这种不确定的情况一直继续下去。如果计算一下两次重发之间的时间间隔, 我们发现存在一种双倍的关系: 从 5.34 到 6.55 是 1.21 秒, 从 6.55 到 8.97 是 2.42 秒, 从 8.97 到 13.80 是 4.83 秒, 一直这样继续下去。当时间间隔达到某个阈值时(大于 42.80 秒), 它又重新置为 5.34 秒。

1	0.0	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
2	6.55 ( 6.55)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
3	15.52 ( 8.97)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
4	29.32 (13.80)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
5	52.78 (23.46)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
6	95.58 (42.80)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
7	100.92 ( 5.34)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
8	107.47 ( 6.55)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
9	116.44 ( 8.97)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
10	130.24 (13.80)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
11	153.70 (23.46)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42
12	196.49 (42.79)	8:0:20:3:f6:42 ff:ff:ff:ff:ff:ff rarp 60: rarp who-is 8:0:20:3:f6:42 tell 8:0:20:3:f6:42

图5-2 网络中没有RARP服务器的RARP请求

超时间隔采用这样的递增方法比每次都采用相同值的方法要好。在图 6-8 中, 我们将看到一种错误的超时重发方法, 以及在第 21 章中 will 看到 TCP 的超时重发机制。

## 5.4 RARP 服务器的设计

虽然 RARP 在概念上很简单, 但是一个 RARP 服务器的设计与系统相关而且比较复杂。相反, 提供一个 ARP 服务器很简单, 通常是 TCP/IP 在内核中实现的一部分。由于内核知道 IP 地

址和硬件地址，因此当它收到一个询问 IP地址的 ARP请求时，只需用相应的硬件地址来提供应答就可以了。

#### 5.4.1 作为用户进程的RARP服务器

RARP服务器的复杂性在于，服务器一般要为多个主机（网络上所有的无盘系统）提供硬件地址到 IP地址的映射。该映射包含在一个磁盘文件中（在 Unix系统中一般位于 /etc/ethers目录中）。由于内核一般不读取和分析磁盘文件，因此 RARP服务器的功能就由用户进程来提供，而不是作为内核的 TCP/IP实现的一部分。

更为复杂的是，RARP请求是作为一个特殊类型的以太网数据帧来传送的（帧类型字段值为 0x8035，如图 2-1 所示）。这说明 RARP服务器必须能够发送和接收这种类型的以太网数据帧。在附录 A 中，我们描述了 BSD 分组过滤器、Sun 的网络接口桩以及 SVR4 数据链路提供者接口都可用来接收这些数据帧。由于发送和接收这些数据帧与系统有关，因此 RARP服务器的实现是与系统捆绑在一起的。

#### 5.4.2 每个网络有多个RARP服务器

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，如图 5-2 所示。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。

### 5.5 小结

RARP协议是许多无盘系统在引导时用来获取 IP地址的。RARP分组格式基本上与 ARP分组一致。一个 RARP请求在网络上进行广播，它在分组中标明发送端的硬件地址，以请求相应 IP地址的响应。应答通常是单播传送的。

RARP带来的问题包括使用链路层广播，这样就阻止大多数路由器转发 RARP请求，只返回很少信息：只是系统的 IP地址。在第 16 章中，我们将看到 BOOTP 在无盘系统引导时会返回更多的信息：IP地址和引导主机的名字等。

虽然 RARP在概念上很简单，但是 RARP服务器的实现却与系统相关。因此，并不是所有的 TCP/IP实现都提供 RARP服务器。

### 习题

5.1 RARP需要不同的帧类型字段吗？ARP和RARP都使用相同的值 0x0806 吗？

5.2 在一个有多个 RARP服务器的网络上，如何防止它们的响应发生冲突？