

## 第16章 BOOTP：引导程序协议

### 16.1 引言

在第5章我们介绍了一个无盘系统，它在不知道自身 IP地址的情况下，在进行系统引导时能够通过RARP来获取它的IP地址。然而使用RARP有两个问题：（1）IP地址是返回的唯一结果；（2）既然RARP使用链路层广播，RARP请求就不会被路由器转发（迫使每个实际网络设置一个RARP 服务器）。本章将介绍一种用于无盘系统进行系统引导的替代方法，又称为引导程序协议，或BOOTP。

BOOTP使用UDP，且通常需与TFTP（参见第15章）协同工作。RFC 951 [Croft and Gilmore 1985]是BOOTP的正式规范，RFC 1542 [Wimer 1993]则对它作了说明。

### 16.2 BOOTP 的分组格式

BOOTP 请求和应答均被封装在UDP数据报中，如图16-1所示。

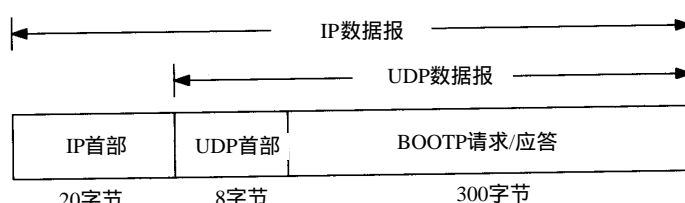


图16-1 BOOTP 请求和应答封装在一个UDP数据报内

图16-2显示了长度为300字节的BOOTP请求和应答的格式。

“操作码”字段为1表示请求，为2表示应答。硬件类型字段为1表示10 Mb/s的以太网，这和ARP请求或应答（图4-3）中同名字段表示的含义相同。类似地，对于以太网，硬件地址长度字段为6字节。

“跳数”字段由客户设置为0，但也能被一个代理服务器设置（参见16.5节）。

“事务标识”字段是一个由客户设置并由服务器返回的32 bit整数。客户用它对请求和应答进行匹配。对每个请求，客户应该将该字段设置为一个随机数。

客户开始进行引导时，将“秒数”字段设置为一个时间值。服务器能够看到这个时间值，备用服务器在等待时间超过这个时间值后才会响应客户的请求，这意味着主服务器没有启动。

如果该客户已经知道自身的IP地址，它将写入“客户IP地址”字段。否则，它将该字段设置为0。对于后面这种情况，服务器用该客户的IP地址写入“你的IP地址”字段。“服务器IP地址”字段则由服务器填写。如果使用了某个代理服务器（见16.5节），则该代理服务器就填写“网关IP地址”字段。

客户必须设置它的“客户硬件地址”字段。尽管这个值与以太网数据帧头中的值相同，UDP数据报中也设置这个字段，但任何接收这个数据报的用户进程能很容易地获得它（例如

一个BOOTP 服务器)。一个进程通过查看 UDP数据报来确定以太网帧首部中的该字段通常是很难的 (或者说是不可可能的)。

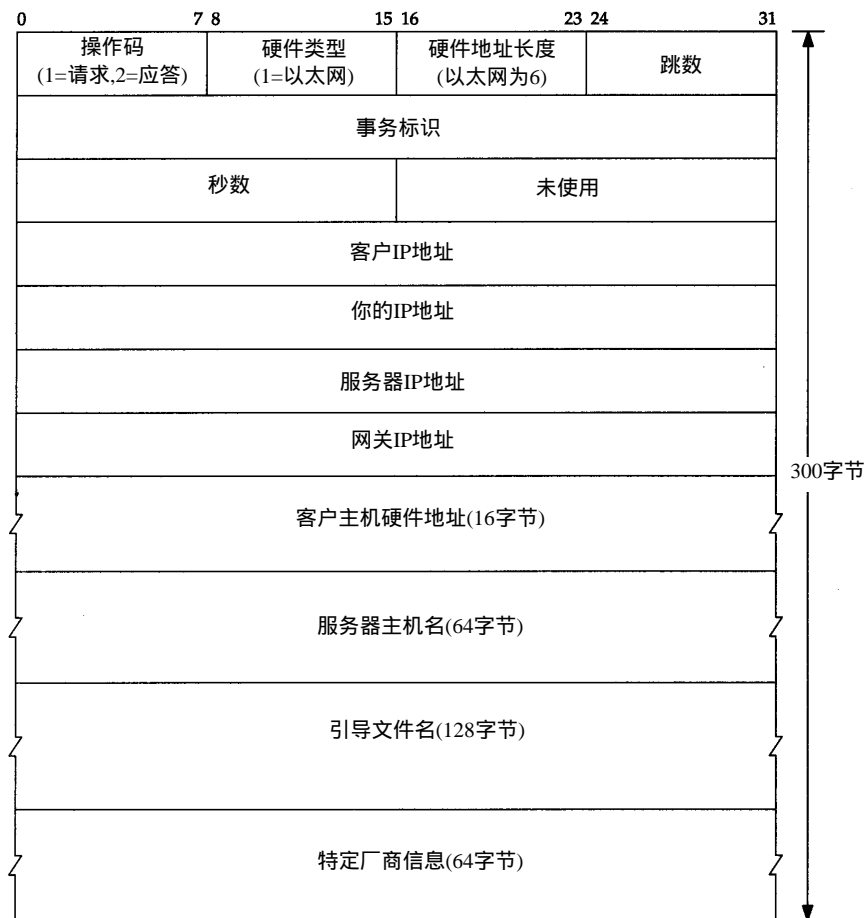


图16-2 BOOTP请求和应答的格式

“服务器主机名”字段是一个空值终止串, 由服务器填写。服务器还将在“引导文件名”填入包括用于系统引导的文件名及其所在位置的路径全名。

“特定厂商区域”字段用于对BOOTP进行不同的扩展。16.6节将介绍这些扩展中的一些。

当一个客户使用BOOTP (操作码为1) 进行系统引导时, 引导请求通常是采用链路层广播, IP首部中的目的IP地址为255.255.255.255 (受限的广播, 12.2节)。源IP地址通常是0.0.0.0, 因为此时客户还不知道它本身的IP地址。回顾图3-9, 在系统进行自引导时, 0.0.0.0是一个有效的IP地址。

## 端口号

BOOTP有两个熟知端口: BOOTP 服务器为67, BOOTP 客户为68。这意味着BOOTP 客户不会选择未用的临时端口, 而只用端口68。选择两个端口而不是仅选择一个端口为BOOTP 服务器用的原因是: 服务器的应答可以进行广播 (但通常是不用广播的)。

如果服务器的应答是通过广播传送的, 同时客户又选择未用的临时端口, 那么这些广播

也能被其他的主机中碰巧使用相同临时端口的应用进程接收到。因此，采用随机端口（即临时端口）对广播来说是一个不好的选择。

如果客户也使用服务器的知名端口（67）作为它的端口，那么网络内的所有服务器会被唤醒来查看每个广播应答（如果所有的服务器都被唤醒，它们将检查操作码，如果是一个应答而不是请求，就不作处理）。因此可以让所有的客户使用与服务器知名端口不同的同一知名端口。

如果多个客户同时进行系统引导，并且服务器广播所有应答，这样每个客户都会收到其他客户的应答。客户可以通过 BOOTP 首部中的事务标识字段来确认应答是否与请求匹配，或者可以通过检查返回的客户硬件地址加以区分。

### 16.3 一个例子

让我们看一个用 BOOTP 引导一个 X 终端的例子。图 16-3 显示了 tcpdump 的输出结果（例中客户名为 proteus，服务器名为 mercury。这个 tcpdump 的输出是在不同的网络上获得的，这个应用程序是其他例子中一直使用的）。

```

1  0.0                0.0.0.0.68 > 255.255.255.255.bootp:
                             secs:100 ether 0:0:a7:0:62:7c
2  0.355446 (0.3554)  mercury.bootp > proteus.68: secs:100 Y:proteus
                             S:mercury G:mercury ether 0:0:a7:0:62:7c
                             file "/local/var/bootfiles/Xncd19r"
3  0.355447 (0.0000)  arp who-has proteus tell 0.0.0.0
4  0.851508 (0.4961)  arp who-has proteus tell 0.0.0.0
5  1.371070 (0.5196)  arp who-has proteus tell proteus
6  1.863226 (0.4922)  proteus.68 > 255.255.255.255.bootp:
                             secs:100 ether 0:0:a7:0:62:7c
7  1.871038 (0.0078)  mercury.bootp > proteus.68: secs:100 Y:proteus
                             S:mercury G:mercury ether 0:0:a7:0:62:7c
                             file "/local/var/bootfiles/Xncd19r"
8  3.871038 (2.0000)  proteus.68 > 255.255.255.255.bootp:
                             secs:100 ether 0:0:a7:0:62:7c
9  3.878850 (0.0078)  mercury.bootp > proteus.68: secs:100 Y:proteus
                             S:mercury G:mercury ether 0:0:a7:0:62:7c
                             file "/local/var/bootfiles/Xncd19r"
10 5.925786 (2.0469)  arp who-has mercury tell proteus
11 5.929692 (0.0039)  arp reply mercury is-at 8:0:2b:28:eb:1d
12 5.929694 (0.0000)  proteus.tftp > mercury.tftp: 37 RRQ
                             "/local/var/bootfiles/Xncd19r"
13 5.996094 (0.0664)  mercury.2352 > proteus.tftp: 516 DATA block 1
14 6.000000 (0.0039)  proteus.tftp > mercury.2352: 4 ACK

                             这里删除了许多行
15 14.980472 (8.9805)  mercury.2352 > proteus.tftp: 516 DATA block 2463
16 14.984376 (0.0039)  proteus.tftp > mercury.2352: 4 ACK
17 14.984377 (0.0000)  mercury.2352 > proteus.tftp: 228 DATA block 2464
18 14.984378 (0.0000)  proteus.tftp > mercury.2352: 4 ACK

```

图16-3 用BOOTP引导一个X终端的例子

在第1行中，我们看到客户请求来自 0.0.0.0.68，发送目的站是 255.255.255.255.67。该客户已经填写的字段是秒数和自身的以太网地址。我们看到客户通常将秒数设置为 100。tcpdump 没有显示跳数和事务标识，因为它们均为 0（事务标识为 0 表示该客户忽略这个字段，

因为如果打算对返回响应进行验证, 它将把这个字段设置为一个随机数值)。

第2行是服务器返回的应答。由服务器填写的字段是该客户的 IP 地址 (tcpdump 显示为名字 proteus)、服务器的 IP 地址 (显示为名字 mercury)、网关的 IP 地址 (显示为名字 mercury) 和引导文件名。

在收到 BOOTP 应答后, 该客户立即发送一个 ARP 请求来了解网络中其他主机是否有 IP 地址。跟在 who-has 后的名字 proteus 对应目的 IP 地址 (图 4-3), 发送者的 IP 地址被设置为 0.0.0.0。它在 0.5 秒后再发一个相同的 ARP 请求, 之后再过 0.5 秒又发一个。在第 3 个 ARP 请求 (第 5 行) 中, 它将发送者的 IP 地址改变为它自己的 IP 地址。这是一个没有意义的 ARP 请求 (见 4.7 节)。

第 6 行显示该客户在等待另一个 0.5 秒后, 广播另一个 BOOTP 请求。这个请求与第 1 行的唯一不同是此时客户将它的 IP 地址写入 IP 首部中。它收到来自同一个服务器的相同应答 (第 7 行)。该客户在等待 2 秒后, 又广播一个 BOOTP 请求 (第 8 行), 同样收到来自同一服务器的相同应答。

该客户等待 2 秒后, 向它的服务器 mercury 发送一个 ARP 请求 (第 10 行)。收到这个 ARP 应答后, 它立即发送一个 TFTP 读请求, 请求读取它的引导文件 (第 12 行)。文件传送过程包括 2464 个 TFTP 数据分组和确认, 传送的数据量为  $512 \times 2463 + 224 = 1\,261\,280$  字节。这将操作系统调入 X 终端。我们已在图 16-3 中删除了大多数 TFTP 行。

当和图 15-2 比较 TFTP 的数据交换过程时, 要注意的是这儿的客户在整个传输过程中使用 TFTP 的知名端口 (69)。既然通信双方中的一方使用了端口 69, tcpdump 就知道这些分组是 TFTP 报文, 因此它能用 TFTP 协议来解释每个分组。这就是为什么图 16-3 能指明哪些包含有数据, 哪些包含有确认, 以及每个分组的块编号。在图 15-2 中我们并不能获得这些额外的信息, 因为通信双方均没有使用 TFTP 的知名端口进行数据传送。由于 TFTP 服务器作为一个多用户系统, 且使用 TFTP 的知名端口, 因此通常 TFTP 客户不能使用那个端口。但这里的系统处于正被引导的过程中, 无法提供一个 TFTP 服务器, 因此允许该客户在传输期间使用 TFTP 的知名端口。这也暗示在 mercury 上的 TFTP 服务器并不关心客户的端口号是什么——它只将数据传送到客户的端口上, 而不管发生了什么。

从图 16-3 可以看出在 9 秒内共传送了 1 261 280 字节。数据速率大约为 140 000 bps。这比大多数以 FTP 文件传送形式访问一个以太网要慢, 但对于一个简单的停止等待协议如 TFTP 来说已经很好了。

X 终端系统引导后, 还需使用 TFTP 传送终端的字体文件、某些 DNS 名字服务器查询, 然后进行 X 协议的初始化。图 16-3 中的所有步骤大概需要 15 秒钟, 其余的步骤需要 6 秒钟, 这样无盘 X 终端系统引导的总时间是 21 秒。

## 16.4 BOOTP 服务器的设计

BOOTP 客户通常固化在无盘系统只读存储器中, 因此了解 BOOTP 服务器的实现将更有意义。

首先, BOOTP 服务器将从它的熟知端口 (67) 读取 UDP 数据报。这没有特别的地方。它不同于 RARP 服务器 (5.4 节), 它必须读取类型字段为 “RARP 请求” 的以太网帧。BOOTP 协议通过将客户的硬件地址放入 BOOTP 分组中, 使得服务器很容易获取客户的硬件地址 (图 16-2)。

这里出现了一个有趣的问题：TFTP 服务器如何能将一个响应直接送回 BOOTP 客户？这个响应是一个 UDP 数据报，而服务器知道该客户的 IP 地址（可能通过读取服务器上的配置文件）。但如果这个客户向那个 IP 地址发送一个 UDP 数据报（正常情况下会处理 UDP 的输出），BOOTP 服务器的主机就可能向那个 IP 地址发送一个 ARP 请求。但这个客户不能响应这个 ARP 请求，因为它还不知道它自己的 IP 地址！（这就是在 RFC951 中被称作“鸡和蛋”的问题。）

有两种解决办法：第一种，通常被 Unix 服务器采用，是服务器发一个 `ioctl(2)` 请求给内核，为该客户在 ARP 高速缓存中设置一个条目（这就是命令 `arp-s` 所做的工作，见 4.8 节）。服务器能一直这么做直到它知道客户的硬件地址和 IP 地址。这意味着当服务器发送 UDP 数据报（即 BOOTP 应答）时，服务器的 ARP 将在 ARP 高速缓存中找到该客户的 IP 地址。

另一种可选的解决办法是服务器广播这个 BOOTP 应答而不直接将应答发回该客户。既然通常期望网络广播越少越好，因此这种解决方案应该只在服务器无法在它的 ARP 高速缓存设置一个条目的情况下使用。通常只有拥有超级用户权限才能在 ARP 高速缓存设置一个条目，如果没有这种权限就只能广播 BOOTP 应答。

## 16.5 BOOTP 穿越路由器

我们在 5.4 节中提到 RARP 的一个缺点就是它使用链路层广播，这种广播通常不会由路由器转发。这就需要在每个物理网络内设置一个 RARP 服务器。如果路由器支持 BOOTP 协议，那么 BOOTP 能够由路由器转发（绝大多数路由器厂商的产品都支持这个功能）。

这个功能主要用于无盘路由器，因为如果在磁盘的多用户系统被用作路由器，它就能够自己运行 BOOTP 服务器。此外，常用的 Unix BOOTP 服务器（附录 F）支持这种中继模式（relay mode）。但如果在这个物理网络内运行一个 BOOTP 服务器，通常没有必要将 BOOTP 请求转发到在另外网络中的另一个服务器。

研究一下当路由器（也称作“BOOTP 中继代理”）在服务器的熟知端口（67）接收到 BOOTP 请求时将会发生什么。当收到一个 BOOTP 请求时，中继代理将它的 IP 地址填入收到 BOOTP 请求中的“网关 IP 地址字段”，然后将该请求发送到真正的 BOOTP 服务器（由中继代理填入网关字段的地址是收到的 BOOTP 请求接口的 IP 地址）。该代理中继还将跳数字段值加 1（这是为防止请求被无限地在网络内转发。RFC 951 认为如果跳数值到达 3 就可以丢弃该请求）。既然发出的请求是一个单播的数据报（与发起的客户的请求是广播的相反），它能按照一定的路由通过其他的路由器到达真正的 BOOTP 服务器。真正的 BOOTP 服务器收到这个请求后，产生 BOOTP 应答，并将它发回中继代理，而不是请求的客户。既然请求网关字段不为零，真正的 BOOTP 服务器知道这个请求是经过转发的。中继代理收到应答后将它发给请求的客户。

## 16.6 特定厂商信息

在图 16-2 中我们看到了 64 字节的“特定厂商区域”。RFC 1533 [Alexander and Droms 1993] 定义了这个区域的格式。这个区域含有服务器返回客户的可选信息。

如果有信息要提供，这个区域的前 4 个字节被设置为 IP 地址 99.130.83.99。这可称作魔术甜饼 (magic cookie)，表示该区域内包含信息。

这个区域的其余部分是一个条目表。每个条目的开始是 1 字节标志字段。其中的两个条目仅有标志字段：标志为 0 的条目作为填充字节（为使后面的条目有更好的字节边界），标志为

255的条目表示结尾条目。第一个结尾条目后剩余的字节都应设置为这个数值 ( 255 )。

除了这两个1字节的条目, 其他的条目还包含一个单字节的长度字段, 后面是相应的信息。图16-4显示了厂商说明区域中一些条目的格式。

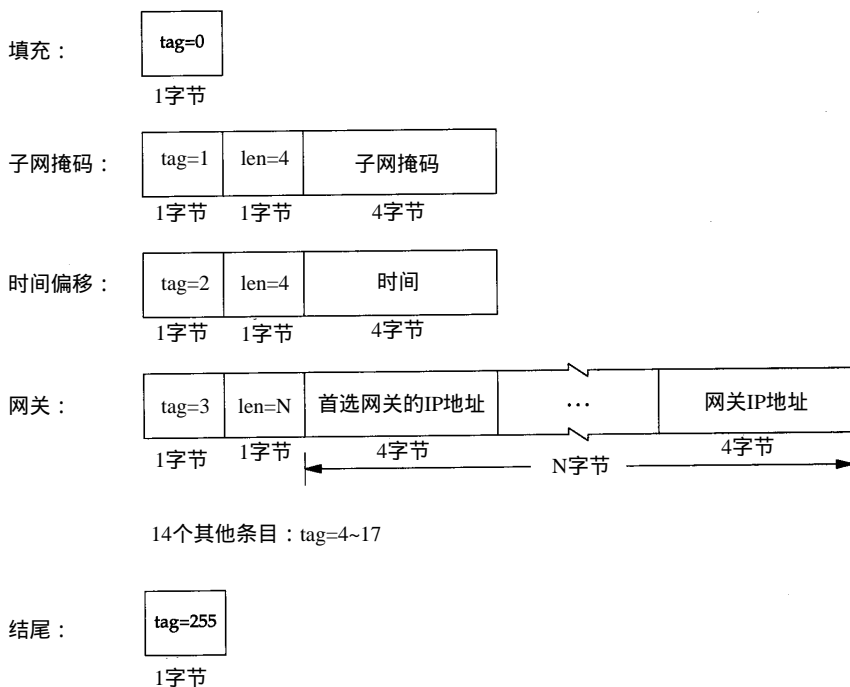


图16-4 厂商说明区域中一些条目的格式

子网掩码条目和时间值条目都是定长条目, 因为它们的值总是占 4 个字节。时间偏移值是从1900年1月1日0时以来的秒数 ( UTC )。

网关条目是变长条目。长度通常是 4 的 倍数, 这个值是一个或多个供客户使用的网关 ( 路由器 ) 的IP地址。返回的第一个必须是首选的网关。

RFC 1533还定义了其他 14个条目。其中最重要的可能是 DNS名字服务器的IP地址条目, 条目的标志为6。其他的条目包括打印服务器、时间服务器等的 IP地址。详细情况可参考 RFC文档。

回到在图16-3中的例子, 我们从未看到客户广播一个 ICMP地址掩码请求 ( 6.3节 ) 来获取它的子网掩码。尽管 tcpdump不能显示出来, 但我们可认为客户所在网络的子网掩码在返回的 BOOTP应答的厂商说明区域内。

Host Requirements RFC文档推荐一个系统使用 BOOTP来获悉它的子网掩码, 而不是采用 ICMP。

厂商说明区域的大小被限制为 64字节。这对某些应用是个约束。一个新的称为动态主机配置协议DHCP ( Dynamic Host Configuration Protocol ) 已经出现, 但它不是替代 BOOTP的。DHCP将这个区域的长度扩展到 312字节, 它在RFC 1541 [Droms 1993] 中定义。

## 16.7 小结

BOOTP使用UDP, 它为引导无盘系统获得它的 IP地址提供了除RARP外的另外一种选择。

BOOTP还能返回其他的信息，如路由器的 IP 地址、客户的子网掩码和名字服务器的 IP 地址。

既然 BOOTP 用于系统引导过程，一个无盘系统需要下列协议才能在只读存储器中完成：BOOTP、TFTP、UDP、IP 和一个局域网的驱动程序。

BOOTP 服务器比 RARP 服务器更易于实现，因为 BOOTP 请求和应答是在 UDP 数据报中，而不是特殊的数据链路层帧。一个路由器还能作为真正 BOOTP 服务器的代理，向位于不同网络的真正 BOOTP 服务器转发客户的 BOOTP 请求。

## 习题

- 16.1 我们说 BOOTP 优于 RARP 的一个方面是 BOOTP 能穿越路由器，而 RARP 由于使用链路层广播则不能。在 16.5 节为使 BOOTP 穿越路由器，我们必须定义特殊的方式。如果在路由器中增加允许转发 RARP 请求的功能会发生什么？
- 16.2 我们说过，当有多个客户程序同时向一个服务器发出引导请求时，因为服务器要广播多个 BOOTP 应答，BOOTP 客户就必须使用事务标识来使响应与请求相匹配。但在图 16-3 中，事务标识为 0，表示这个客户不考虑事务标识。你认为这个客户将如何将这些响应与其请求匹配。