Chapter 16
BOOTP: Bootstrap Protocol

Introduction

- **There are two problems with RARP:**
  - the only thing returned is IP address
  - since RARP uses a link-layer broadcast, RARP requests are not forwarded by routers.
- **BOOTP uses UDP and normally works in conjunction with TFTP.**

BOOTP Packet Format

- BOOTP requests and replies are encapsulated in UDP datagrams.

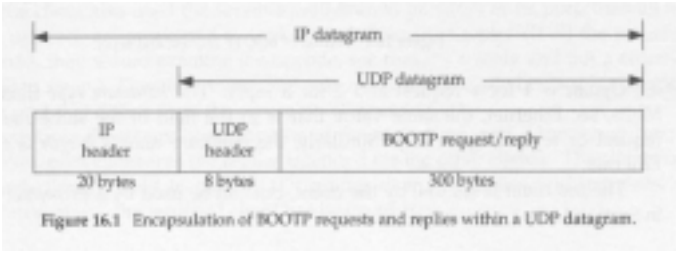
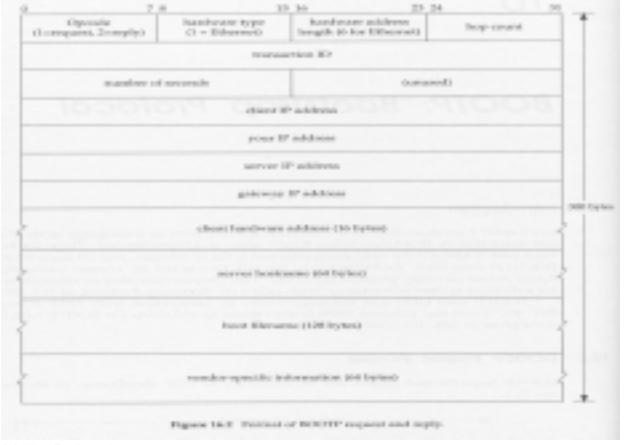


Figure 16.1 Encapsulation of BOOTP requests and replies within a UDP datagram.

BOOTP Packet Format (Cont.)

- Format of BOOTP request and reply



BOOTP Packet Format (Cont.)

- *opcode*:1=request ; 2=reply
- *hardware type*: the same value that is in the field of the same name in an ARP req. or reply. 1=10Mbits/sec Ethernet
- *hardware address length*: 6 bytes for an Ethernet
- *hop count*: set to 0 by the client, but can be used by a proxy server.
- *transaction ID*: a 32-bit integer set by the client and returned by the server.
- *number of seconds*: be set by the client to the time since it started trying to bootstrap.
- *client IP address, your IP address, server IP address*, and *gateway IP address*
- *server hostname:* a null terminated string that is optionally filled in by the server.

BOOTP Packet Format (Cont.)

- *boot filename*: the fully qualified null terminated pathname of a file to bootstrap from.
- *vendor-specific area*: is used for various extensions to BOOTP.

- **When a client is bootstrapping, the request is:**
  - Link layer broadcast
  - Destination IP address: 255.255.255.255
  - Source IP address: 0.0.0.0
- **Port numbers**
  - There are two well-known ports for BOOTP: 67 for the server and 68 for the client.
  - Client does not choose an unused ephemeral port
  - Multiple clients are bootstrapping at the same time, and if the server broadcasts the replies: transaction ID field to match replies with requests or returned client hardware address.

An Example

- **Scenario**
  - An X terminal is bootstrapped.
  - The client's name is *proteus* and the server's name is *mercury*.

An Example (Cont.)

- **The tcpdump output was obtained on a different network from the one.**

```
1   0.0                 0.0.0.0.68 > 255.255.255.255.bootp:
                        secs:180 ether 0:0:a7:0:62:7c
2   0.355446 (0.3554)   mercury.bootp > proteus.68: secs:180 Y:proteus
                        S:mercury G:mercury ether 8:0:a7:8:62:7c
                        file "/local/var/bootfiles/XNo05Pc"
3   0.355847 (0.0004)   arp who-has proteus tell 0.0.0.0
4   0.851388 (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:180 ether 0:0:a7:0:62:7c
7   1.871038 (0.0078)   mercury.bootp > proteus.68: secs:180 Y:proteus
                        S:mercury G:mercury ether 8:0:a7:8:62:7c
                        file "/local/var/bootfiles/XNo05Pc"
8   1.871038 (0.0000)   proteus.68 > 255.255.255.255.bootp:
                        secs:180 ether 0:0:a7:0:62:7c
9   1.878853 (0.0078)   mercury.bootp > proteus.68: secs:180 Y:proteus
                        S:mercury G:mercury ether 8:0:a7:8:62:7c
                        file "/local/var/bootfiles/XNo05Pc"
10  5.925786 (4.0469)   arp who-has mercury tell proteus
11  5.929682 (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/XNo05Pc"
13  5.936096 (0.0064)   mercury.2352 > proteus.tftp: 516 DATA block 1
14  6.000000 (0.0639)   proteus.tftp > mercury.2352: 4 ACK
                        many lines deleted here
33  14.983472 (8.9835)  mercury.2352 > proteus.tftp: 516 DATA block 2463
34  14.984376 (0.0039)  proteus.tftp > mercury.2352: 4 ACK
37  14.984597 (0.0002)  mercury.2352 > proteus.tftp: 208 DATA block 2464
38  14.984378 (0.0000)  proteus.tftp > mercury.2352: 4 ACK
```

Figure 16.5 Example of BOOTP being used to bootstrap an X terminal.

BOOTP Server Design

- **The server reads UDP datagrams from its well-known port (67). Nothing special is required.**
- **"Chicken and egg" issue: How can the server send a response directly back to the client? There are two solutions:**
  - Used by Unix servers, is for the server to issue an ioctl(2) request to the kernel, to place an entry into the ARP cache for this client.
  - For the server to broadcast the BOOTP reply, instead of sending it directly to the client.

BOOTP Through a Router

- **BOOTP can be used through a router, if supported by the router.**
- **What happens is that router (also called the "BOOTP relay agent") listens for BOOTP requests on the server's well-known port (67).**
  - When a request is received, the relay agent places its IP address into the gateway IP address field in the BOOTP request, and sends the request to the real BOOTP server.
  - The relay agent also increments the hops field by one.
  - Since the outgoing request is a unicast datagram, it can follow any route to the real BOOTP server, passing through other routers.
  - The real server gets the request, forms the BOOTP reply, and sends it back to the relay agent.
  - The relay agent receives the reply and sends it to the client.

Vendor-Specific Information

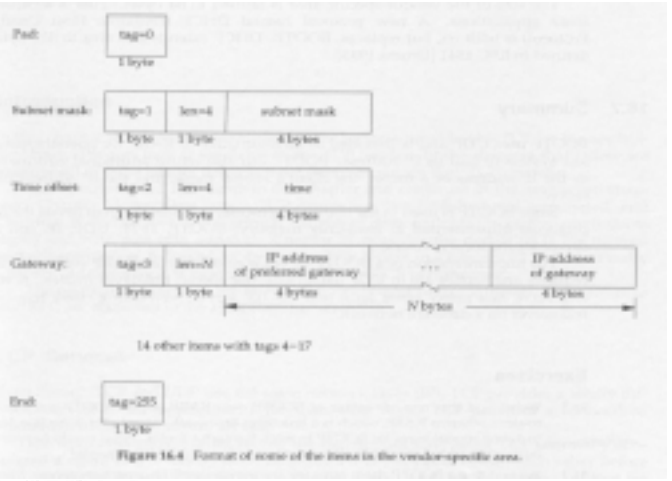


Figure 16.4 Format of some of the items in the vendor-specific area.

Vendor-Specific Information (Cont.)

- Magic cookie: if information is provided, the first 4 bytes of this area are set to the IP address 99.130.83.99
- A system obtains its subnet mask using BOOTP, not ICMP.
- The size of the vendor-specific area is limited to 64 bytes.
- A new protocol named DHCP (*Dynamic Host Configuration Protocol*) is built on, but replaces, BOOTP. DHCP extends this area to 312 bytes and is defined in RFC 1541.

Summary

- BOOTP uses UDP and is intended as an alternative to RARP for bootstrapping a diskless system to find its IP address.
- BOOTP can also return additional information, such as the IP address of router, the client's subnet mask, and the IP address of a name server.
- The implementation of a BOOTP server is easier than an RARP server.
- A router can also serve as a proxy agent for a real BOOTP server, forwarding client requests to the real server on a different network.