Boot(ing) protocols

From (R)ARP to BSDP

Karst Koymans
Jeroen van der Ham & Jaap van Ginkel

Informatics Institute
University of Amsterdam

Tuesday, October 11, 2011

Table of Contents

ARP

- Address Resolution Protocol
  - RFC 826 (published in 1982)
  - Translates IP addresses to Ethernet addresses
  - Uses ethernet type (0x0806)
  - Is not a boot(ing) protocol
  - Uses requests (id=1) and replies (id=2)

RARP

- Reverse Address Resolution Protocol
  - RFC 903 (published in 1984)
  - $\neq$ Inverse Address Resolution Protocol (other ether $\Rightarrow$ IP)
  - Translates own Ethernet address to own IP address
  - Uses its own ethernet type (0x0835)
- Requests (id=3) use
  - (Ethernet) broadcast address as destination
  - Own hardware address as source
- Replies (id=4) fill in the looked up IP address
  - For instance using /etc/ethers file

ARP and Multicast DNS (Zeroconf)

- Zeroconfiguration Networking
  - Address assignment — ARP & LL
  - Hostname Resolution — Multicast DNS
  - Service Discovery — Multicast DNS
- RFC 3927 & mDNS draft

BOOTP

- Bootstrap protocol
  - RFC 951 (published in 1985)
  - with clarifications in RFC 1532 (published in 1993)
- BOOTP Vendor Information Extensions
  - RFC 1048, 1084, 1395 (published in 1988; obsolete)
  - RFC 1497, 1533 (published in 1993; obsolete)
  - RFC 2132 (with DHCP options)
  - updated by RFC 3442, 3942, 4361, 4833

BOOTP packet format

Operation Code	Hardware Type	Hardware Length	Hops
Transaction Identifier			
Seconds Elapsed		(Unused)	
Client IP Address			
Your IP Address			
Server IP Address			
Relay IP Address			
⋮			

BOOTP packet format (continued)

⋮
Client Hardware Address ⋮ (16 bytes)
Server Hostname ⋮ (64 bytes)
Boot Filename ⋮ (128 bytes)
Vendor Specific Area ⋮ (64 bytes)

BOOTP fields

BOOTP fields

Operation Code	1 (request), 2 (reply)
Hardware Type	1 (ethernet)
Hardware Length	6
Hops	0 (for client), increased by relay
Transaction Identifier	number to match request and reply
Seconds Elapsed	number of seconds since boot

BOOTP fields (continued)

BOOTP fields

Client IP Address	filled in by client (if known)
Your IP Address	provided by boot server
Server IP Address	address of boot server
Relay IP Address	address of relay server

BOOTP fields (continued)

BOOTP fields

Client Hardware Address	ethernet address of client
Server Hostname (sname)	optional name of wanted server set by client (and by server)
Boot Filename (file)	server supplied full path boot file possibly hinted by client on request
Vendor Specific Area	used for various extensions to BOOTP

BOOTP versus RARP

- RARP is link layer, needs kernel modification
- RARP only supplies IP address
- BOOTP is UDP/IP based
  - Chicken/egg problems (client cannot reply to ARP)
- BOOTP supplies more information

DHCP

- Dynamic Host Configuration protocol
  - RFC 2131 (published in 1997)
- DHCP Options and BOOTP Vendor Extensions
  - RFC 2132 (published in 1997)

DHCP packet format

Operation Code	Hardware Type	Hardware Length	Hops
Transaction Identifier			
Seconds Elapsed		Flags	
Client IP Address			
Your IP Address			
Server IP Address			
Relay IP Address			
⋮			

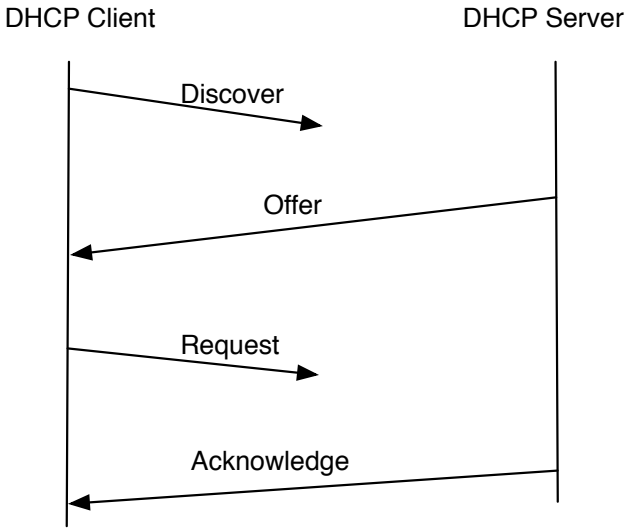
DHCP packet format (continued)

⋮
Client Hardware Address ⋮ (16 bytes)
Server Hostname ⋮ (64 bytes)
Boot Filename ⋮ (128 bytes)
Options ⋮ (variable number of bytes)

DHCP versus BOOTP

- DHCP has more space for options
- DHCP can overload sname and file fields
- DHCP can configure unknown clients
  - combines static/dynamic assignments
- DHCP has many more states, uses leases

DHCP Exchange



DHCP options

- DHCP specific, like
  - Requested IP Address
  - IP Address Lease Time
  - Option Overload
  - DHCP Message Type (DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, DHCPACK, DHCPNAK, DHCPDECLINE, DHCPRELEASE, DHCPINFORM)

DHCP options (continued)

- Generic, IP, TCP, Application server/service options, for instance
  - Domain Name Server Option
  - Interface MTU Option
  - TCP Keepalive Interval Option
  - Simple Mail Transport Protocol (SMTP) Server Option

TFTP

- Trivial File Transfer Protocol
  - RFC 1350 (published in 1992)
  - Uses lock-step protocol
  - Revision 2
    - fixes a bug called “Sorcerer’s Apprentice Syndrome”

Network Boot

- New system install
- Emergency boot (recovery)
- Network boot (thin client)

PXE

- Preboot eXecution Environment
  - Developed by Intel 1999
  - Part of WFM Wired for Management framework
  - Uses (slightly modified) DHCP
  - Uses (possibly multicast) TFTP
  - Defines own boot protocol using DHCP packet format
    - Has support for multiple TFTP servers

BIS

- Boot Integrity Services
  - Developed by Intel 1998
  - Integrity from BIOS handover to high level OS
  - Certificate based verification of boot file
  - Not widely deployed
  - Security of network booting still very tricky

iPXE

- Open source PXE implementation
  - Further development of gPXE
  - Supports booting from iSCSI, HTTP, Infiniband and others
  - Development started on IPv6 support
  - Reflash boot code on network card
  - or chainload from PXE

BSDP

- Boot Service Discovery Protocol
  - Apple developed 1999
  - Open source
  - Works within DHCP context and uses
    - Request: Vendor class identifier (option 60)
    - Response: Encapsulated vendor-specific options (option 43)

RIPL

- Remote Initial Program Load
  - IBM developed 1991
  - Used by Novell, Microsoft and others
  - Mostly IPX based