Address resolution

The first thing to perform in the network

Mapping Between IP and Physical Addresses

- Today's topics:
  - How does a machine learn the physical address for a computer for which it knows the IP address?
    - Example: transmitting a message over a physical network
  - How does a machine learn the IP address for a computer for which it knows the physical address?
    - Example: a booting machine learning its own IP address

Fitting into OSI

- ARP has always been a Layer 2 protocol. The reason: The highest layer addresses carried within ARP are Layer2 MAC addresses for typical ARP operation. The IP addresses in the ARP packets are protocol payload, no addressing information of the ARP packet itself.
- ARP is a protocol that does not fit too well into the 7 layer OSI model. These models were defined for end user applications like HTTP or FTP and they still define, how traffic is sent from application to application through a network stack (L3+L4) and a network interface (L2 + L1) down on the wire.
- ARP is a service protocol that glues together layer 2 and layer 3 protocols. It solves the problem that you need to add a layer 2 (MAC) destination address over a shared media like Ethernet or Wireless LAN using IP packets. But ARP is a separate process with separate packets. You will find no ARP protocol information within an IP packet. *This is the reason, why ARP is definitely not a layer 2.5 protocol.*

Mapping IP Addresses to Physical Address

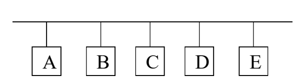
- Internetwork
  - Each host is assigned one (or more) 32-bit IP address
  - Behaves like a virtual network, using only IP addresses when sending and receiving packets
- Physical Network
  - Two machines can communicate only if they know each other's physical network addresses

The Address Resolution Problem

- Want to hide details of the physical network
- Application programs should use IP addresses only
- Ultimately, communication must be carried out by physical networks
- The *address resolution problem* - need to map IP address to physical address

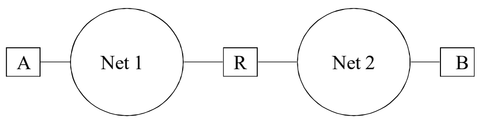
The Address Resolution Problem

- Hosts *A* and *B* are on the same physical network
- *B* wants to communicate with *A* but only knows *A*'s IP address



The Address Resolution Problem

- Hosts *A* and *B* are on different physical networks connected by router *R*
- *B* wants to communicate with *A* and only knows *A*'s and *R*'s IP address



The Address Resolution Problem

- Two types of physical addresses
  - Ethernet: large, fixed addresses
  - ProNET: small, configurable addresses
    - Uses small integers (0-255) for physical addresses
    - Allows the administrator to choose an address when installing the interface board

Address Resolution Through Direct Mapping

- Assign the *hostid* field of the IP address to match the machine's physical address
- Physical address can be extracted trivially from the IP address
- Example:
  - Physical address 1, IP address 192.5.48.1
  - Physical address 2, IP address 192.5.48.2
  - Physical address 3, IP address 192.5.48.3

Resolution Through Dynamic Binding

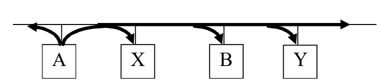
- Can't assign the *hostid* field of the IP address to match the machine's physical address
- Don't want to maintain a centralized database of mappings
- Want to bind addresses dynamically

The Address Resolution Protocol (ARP)

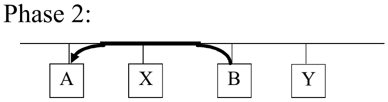
- Host *A* wants to resolve the IP address I_B
- Host *A* broadcasts a special (ARP) packet that asks the host with IP address I_B to respond with its physical address
- All hosts receive the request
- Host B recognizes its IP address
- Host B sends a reply containing its physical address

ARP

- Phase 1:



- Phase 2:



ARP forwarding

- When a device makes an ARP request, it sends a broadcast and includes the its source MAC address, so the reply can be unicast to the requester. The device that owns the requested address (or a device acting as its proxy) sees the broadcast and unicasts the response.

  Exception: A **Gratuitous ARP** reply gets sent as a broadcast in order to refresh neighbor ARP caches, detect duplicate IP addresses in the layer 2 domain or populate upstream switch tables to help prevent unknown MAC destination address flooding.

Gratuitous ARP

- The gratuitous ARP packet has the following characteristics:
  - Both source and destination IP in the packet are the IP of the host issuing the gratuitous ARP
  - The destination MAC address is the broadcast MAC address (ff:ff:ff:ff:ff:ff)
  - This means the packet will be flooded to all ports on a switch
  - No reply is expected
- Gratuitous ARP is used for some reasons:
  - Update ARP tables after a MAC address for an IP changes (failover, new NIC, etc.)
  - Update MAC address tables on L2 devices (switches) that a MAC address is now on a different port
  - Send gratuitous ARP when interface goes up to notify other hosts about new MAC/IP bindings in advance so that they don't have to use ARP requests to find out
  - When a reply to a gratuitous ARP request is received you know that you have an IP address conflict in your network

ARP Refinements

- Each host maintains a soft state cache of recently-used mappings
  - Information in the cache expires after a set time has elapsed
- When sending an ARP request a host includes its IP-to-physical address binding
- All machines on a physical network "snoop" ARP packets for addresses

ARP Encapsulation

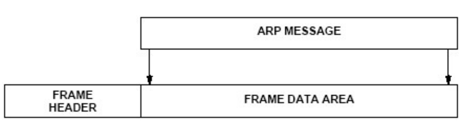
- ARP message must travel from one machine to another inside physical frames
- Recall Ethernet frames:

Preamble	Destination Address	Source Address	Frame Type	Frame Data	CRC
8 octets	6 octets	6 octets	2 octets	46–1500 octets	4 octets

- The value 0806_{16} in the type field indicates a frame carries an ARP message in its data field

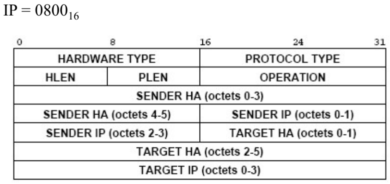
ARP Encapsulation

- The ARP message is encapsulated in the physical frame:



ARP Packet Format

- Hardware type – the type of hardware addresses used
  - Ethernet = 1
- Protocol type – the type of protocol addresses used
  - IP = 0800_{16}



ARP Packet Format (cont)

- Hlen – the length of the hardware addresses in octets
  - Ethernet = ?
- Plen – the length of the protocol addresses in octets
  - IP = ?

0	8	16	24 31
HARDWARE TYPE		PROTOCOL TYPE	
HLEN	PLEN	OPERATION	
SENDER HA (octets 0-3)			
SENDER HA (octets 4-5)		SENDER IP (octets 0-1)	
SENDER IP (octets 2-3)		TARGET HA (octets 0-1)	
TARGET HA (octets 2-5)			
TARGET IP (octets 0-3)			

ARP Packet Format (cont)

- Operation:
  - ARP request = 1
  - ARP reply = 2
  - RARP request = 3
  - RARP reply = 4

0	8	16	24 31
HARDWARE TYPE		PROTOCOL TYPE	
HLEN	PLEN	OPERATION	
SENDER HA (octets 0-3)			
SENDER HA (octets 4-5)		SENDER IP (octets 0-1)	
SENDER IP (octets 2-3)		TARGET HA (octets 0-1)	
TARGET HA (octets 2-5)			
TARGET IP (octets 0-3)			

A Sample ARP Request

- Machine A knows machine B's IP address and wants to know B's physical address
  - A's IP = 134.126.24.120
  - A's physical address = 00:06:5B:75:95:76
  - B's IP = 134.126.20.50

0	8	16	24 31
HARDWARE TYPE		PROTOCOL TYPE	
HLEN	PLEN	OPERATION	
SENDER HA (octets 0-3)			
SENDER HA (octets 4-5)		SENDER IP (octets 0-1)	
SENDER IP (octets 2-3)		TARGET HA (octets 0-1)	
TARGET HA (octets 2-5)			
TARGET IP (octets 0-3)			

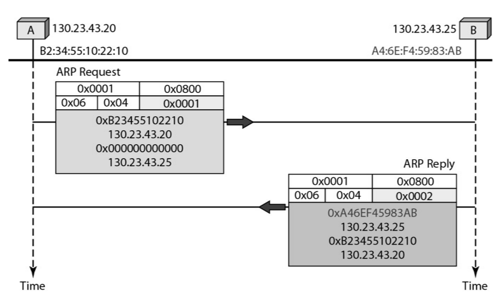
A Sample ARP Reply

- Machine B responds with its physical address
  - B's physical address = F0:15:EC:83:17:76

0	8	16	24 31
HARDWARE TYPE		PROTOCOL TYPE	
HLEN	PLEN	OPERATION	
SENDER HA (octets 0-3)			
SENDER HA (octets 4-5)		SENDER IP (octets 0-1)	
SENDER IP (octets 2-3)		TARGET HA (octets 0-1)	
TARGET HA (octets 2-5)			
TARGET IP (octets 0-3)			

A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB. The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution
Next figure shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses.



Mapping Physical Addresses to IP Address

- Determining an IP Address at startup
  - Hosts with secondary storage read their IP address from disk at boot time
  - How does a diskless host get its IP address at boot time?
  - Answer: must resort to physical network addressing temporarily

The Reverse Address Resolution Protocol (RARP)

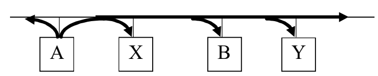
- Host *A* is booting and needs to know its IP address
- Broadcasts a RARP request on the physical network to which it attaches
  - Specifies itself as both the sender and target
  - Supplies its physical network address

RARP Servers

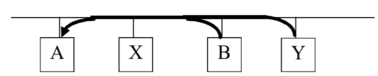
- Receive RARP requests broadcast on a physical network
- Has access to disk where it keeps a database of physical to IP address mappings
- Sends a reply containing the IP address using the physical network

RARP

- Phase 1:



- Phase 2:



Multiple RARP Servers

- RARP requests are subject to loss or corruption
- RARP servers can be down or overloaded

Summary

- The Address Resolution Protocol (ARP):
  - Performs dynamic address resolution
  - Communicates over a physical network
  - Asks the machine with a given IP address to respond with its physical address
- The Reverse Address Resolution Protocol (RARP):
  - Enables a machine to learn it IP address during startup
  - Communicates over a physical network
  - Asks a RARP server to reply with the IP address corresponding to the machine's physical address

Commands

- arp –a IP_ADDR (works in unix + win)
- nbstat –a IP_ADDR (Win)
- arp –a (shows ARP cache of the system)

Get IP address of a client in JAVA

```
public String getIpAddr(HttpServletRequest request) {
    String ip = request.getHeader("x-forwarded-for");
    if(ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getHeader("Proxy-Client-IP");
    }
    if(ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getHeader("WL-Proxy-Client-IP");
    }
    if(ip == null || ip.length() == 0 || "unknown".equalsIgnoreCase(ip)) {
            ip = request.getRemoteAddr();
    }
    return ip;
}
```

Finding MAC of a client in JAVA

```
public String getMACAddress(String ip){
  String str = "";
  String macAddress = "";
  try {
     Process p = Runtime.getRuntime().exec("nbtstat -A " + ip);  //or, arp –A to support Unix as well
     InputStreamReader ir = new InputStreamReader(p.getInputStream());
     LineNumberReader input = new LineNumberReader(ir);
     for (int i = 1; i <100; i++) {
            str = input.readLine();
            if (str != null) {
                    if (str.indexOf("MAC Address") > 1) {
                            macAddress = str.substring(str.indexOf("MAC Address") + 14,
     str.length());
                            break;
                    }
            }
     }
  }
  catch (IOException e) { e.printStackTrace(System.out); }
  return macAddress;
}
```

RARP disadvantage

- RARP had the disadvantage that it was a hardware link level protocol (not IP/UDP based). This means that RARP could only be implemented on hosts containing **special kernel or driver modifications to access these 'raw' packets**.
- Since there are many network kernels existent now, with each source maintained by different organizations, a boot protocol that does not require kernel modifications is a decided advantage.

BOOTP

Bootstrap Protocol
(RFC 951)

DHCP

Dynamic Host Configuration Protocol
(RFC 2131)

BOOTP

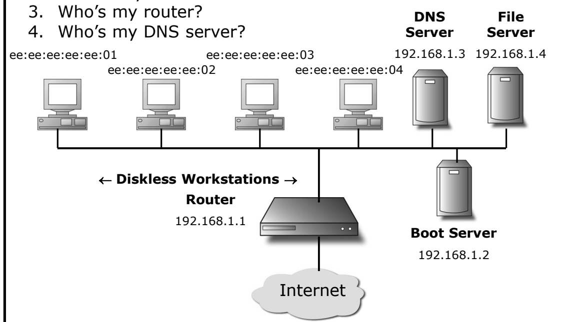
- The **Bootstrap Protocol** (**BOOTP**) is a computer networking protocol used in Internet Protocol networks to automatically assign an IP address to network devices from a configuration server. The BOOTP was originally defined in RFC 951.
- When a computer that is connected to a network is powered up and boots its operating system, the system software broadcasts BOOTP messages onto the network to request an IP address assignment. A BOOTP configuration server assigns an IP address based on the request from a pool of addresses configured by an administrator.
- BOOTP is implemented using the User Datagram Protocol (UDP) as transport protocol, port number 67 is used by the server to receive client requests and port number 68 is used by the client to receive server responses. BOOTP operates only on IPv4 networks.

BOOTP contd.

- Historically, BOOTP has also been used for Unix-like diskless workstations to obtain the network location of their boot image, in addition to the IP address assignment. Enterprises used it to roll out a pre-configured client (e.g., Windows) installation to newly installed PCs.
- Originally requiring the use of a boot floppy disk to establish the initial network connection, manufacturers of network cards later embedded the protocol in the BIOS of the interface cards as well as system boards with on-board network adapters, thus allowing direct network booting.
- While some parts of BOOTP have been effectively superseded by the Dynamic Host Configuration Protocol (DHCP), which adds the feature of leases, parts of BOOTP are used to provide service to the DHCP protocol. DHCP servers also provide the legacy BOOTP functionality.

Why BOOTP?

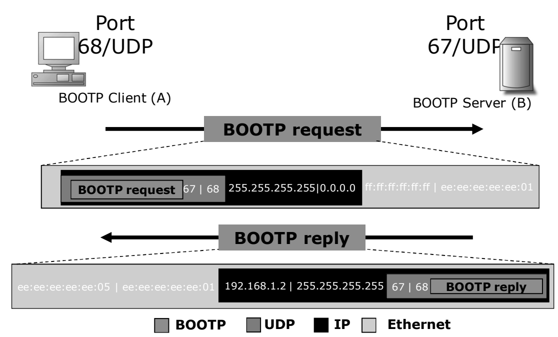
1. What’s my IP address?
2. What’s my subnet mask?
3. Who’s my router?
4. Who’s my DNS server?



BOOTP: Bootstrap Protocol

- RFC 951
- Designed for diskless workstations
- Supplies static configuration:
  - IP address
  - Subnet mask
  - Router IP address
  - Name server IP address
  - Boot image

BOOTP Operation



BOOTP PDU Format

Operation Code	Hardware Type	Hardware Length	Hop Count
Transaction ID			
Number of seconds		Unused	
Client IP address			
Your IP address			
Server IP address			
Gateway IP address			
Client hardware address (16 bytes)			
Server name (64 bytes)			
Boot file name (128 bytes)			
Options (up to 64 bytes)			

$\underbrace{\hspace{10em}}_{\text{4 bytes}}$

Sun Advanced Lights-Out Management (ALOM) card attempting to obtain DHCP service will construct a *DHCP Client Identifier* which begins with 00 53 55 4E 57 2C 53 43 3D . Similar 16 byte CHADDR is used in BOOTP. Its not MAC.

BOOTP PDU Format

```
FIELD   BYTES   DESCRIPTION
-----   -----   -----------

  op      1       packet op code / message type.
                  1 = BOOTREQUEST, 2 = BOOTREPLY

  htype   1       hardware address type,
                  see ARP section in "Assigned Numbers" RFC.
                  '1' = 10mb ethernet

  hlen    1       hardware address length
                  (eg '6' for 10mb ethernet).

  hops    1       client sets to zero,
                  optionally used by gateways
                  in cross-gateway booting.

  xid     4       transaction ID, a random number,
                  used to match this boot request with the
                  responses it generates.

  secs    2       filled in by client, seconds elapsed since
                  client started trying to boot.

  ciaddr  4       client IP address;
                  filled in by client in bootrequest if known.

  yiaddr  4       'your' (client) IP address;
                  filled by server if client doesn't
                  know its own address (ciaddr was 0).

  siaddr  4       server IP address;
                  returned in bootreply by server.

  giaddr  4       gateway IP address,
                  used in optional cross-gateway booting.

  chaddr  16      client hardware address,
                  filled in by client.

  sname   64      optional server host name,
                  null terminated string.

  file    128     boot file name, null terminated string;
                  'generic' name or null in bootrequest,
                  fully qualified directory-path
                  name in bootreply.

  vend    64      optional vendor-specific area,
                  e.g. could be hardware type/serial on request,
                  or 'capability' / remote file system handle
                  on reply.  This info may be set aside for use
                  by a third phase bootstrap or kernel.
```

Chicken / Egg Issues

- If the client knows its own IP address ('ciaddr' field is nonzero), then the IP can be sent 'as normal', since the client will respond to ARPs.
- If the client does not yet know its IP address (ciaddr zero), then the client cannot respond to ARPs sent by the transmitter of the bootreply. There are two options:
  - If the transmitter has the necessary kernel or driver hooks to 'manually' construct an ARP address cache entry, then it can fill in an entry using the 'chaddr' and 'yiaddr' fields. Of course, this entry should have a timeout on it, just like any other entry made by the normal ARP code itself. The transmitter of the bootreply can then simply send the bootreply to the client's IP address. UNIX (4.2 BSD) has this capability.
  - If the transmitter lacks these kernel hooks, it can simply send the bootreply to the IP broadcast address on the appropriate interface. This is only one additional broadcast over the previous case.

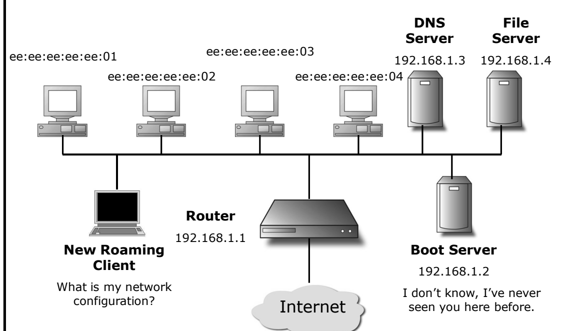
TFTP (RFC 1350)

- **TFTP**, or Trivial File Transfer Protocol, is a simple high-level protocol for transferring data servers use to boot diskless workstations, X-terminals, and routers by using User Data Protocol (UDP).
- TFTP is a simple protocol for transferring files, implemented on top of the UDP/IP protocols using well-known port number 69.
- TFTP was designed to be small and easy to implement, and therefore it lacks most of the advanced features offered by more robust file transfer protocols.
- TFTP only reads and writes files from or to a remote server. It cannot list, delete, or rename files or directories and it has no provisions for user authentication.
- Today TFTP is generally only used on local area networks (LAN)

Client's use of BOOTP

- The client PROM must contain a simple implementation of ARP, e.g. the address cache could be just one entry in size. This will allow a second-phase-only boot (TFTP) to be performed when the client knows the IP addresses and bootfile name.
- Any time the client is expecting to receive a TFTP or BOOTP reply, it should be prepared to answer an ARP request for its own IP to hardware address mapping (if known).
- Since the bootreply will contain (in the hardware encapsulation) the hardware source address of the server/gateway, the client MAY be able to avoid sending an ARP request for the server/gateway IP address to be used in the following TFTP phase. However this should be treated only as a special case, since it is desirable to still allow a second-phase-only boot.

BOOTP Problem

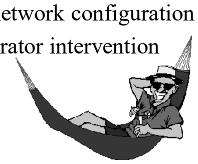


BOOTP Limitations

- Static configuration
- Does not dynamically allocate IP addresses
- Manual administrator intervention to add/remove clients

DHCP Motivations

- Automatic network configuration for clients
- No administrator intervention

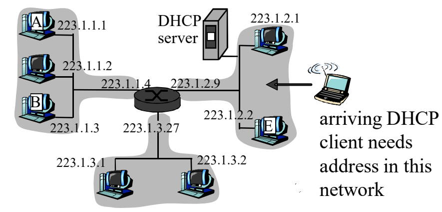


- Effective allocation of limited addresses
- Support for transient/roaming systems

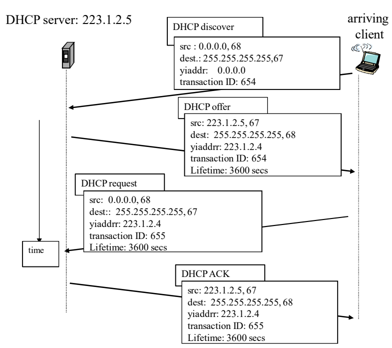
DHCP Evolution

- DHCP is an extension of Bootstrap Protocol
- Uses same basic PDU format for backwards compatibility
- Introduces pool of IP addresses for dynamic assignment
- Concept of temporary leased addresses

DHCP client-server scenario

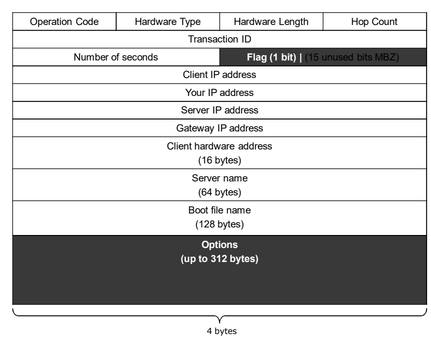


DHCP client-server scenario



DHCP uses UDP on port 67, 68

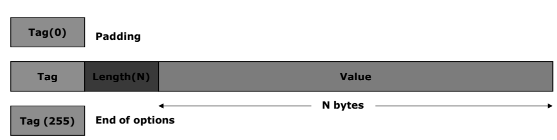
DHCP PDU Format



DHCP PDU Format

- Broadcast bit is to inform server if it can respond with unicast IP PDUs or if it must instead broadcast the reply to the entire network.
- DHCP PDU has 312 bytes for options versus 64 bytes in BOOTP PDU
- DHCP messages carried in options portion of the PDU

Typical Options

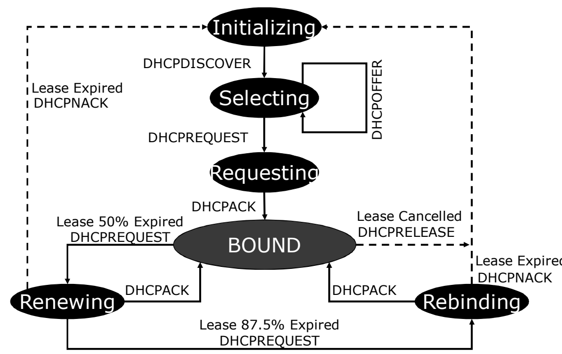


Tag ID	Function	Tag ID	Function
1	Subnet Mask	13	Boot File size
37	TCP Default TTL	72	WWW Server
69	SMTP Server	61	Client Identifier
54	Server Identifier	66	TFTP Server
3	Time server	**53**	**DHCP Message**
4	DNS name server	55	Parameter Request List

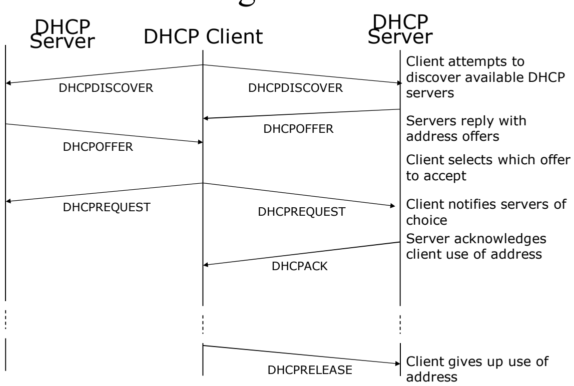
Message Types

- Type identified by value field of option with tag 53:
  - DHCPDISCOVER (1)
  - DHCPOFFER (2)
  - DHCPREQUEST (3)
  - DHCPDECLINE (4)
  - DHCPACK (5)
  - DHCPNACK (6)
  - DHCPRELEASE (7)
  - DHCPINFORM (8)

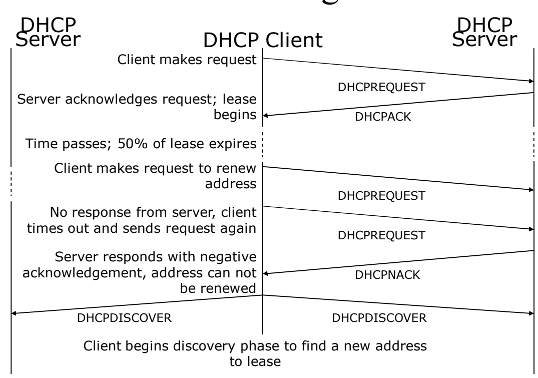
DHCP Client State Diagram



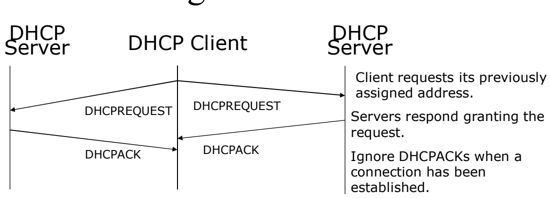
Allocating New Address



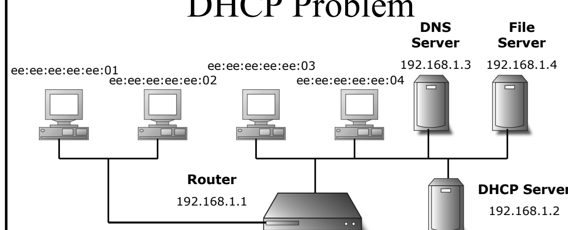
Address Renewing Scenario



Renewing a Previous Address



DHCP Problem



- What is the problem here?
- Routers do not forward IP broadcast PDUs

BOOTP was not so bad. At least for internet.

DHCP Infrastructure

- Use relay agents to transmit DHCP messages between physical networks
- Prohibitive/costly to have DHCP server on each physical LAN segment

DHCP Security Considerations

- Hostile environments with open physical access to network
- Rouge DHCP server on network
- Denial of service by exhausting address pool
- Authentication introduced in RFC 3118 but not implemented