Overview

ARP and RARP
ARP and RARP

- Note:
  - The Internet is based on IP addresses
  - Data link protocols (Ethernet, FDDI, ATM) may have different (MAC) addresses
- The ARP and RARP protocols perform the translation between IP addresses and MAC layer addresses
- We will discuss ARP for broadcast LANs, particularly Ethernet LANs

IP address (32 bit) → ARP → Ethernet MAC address (48 bit)

RARP

Ethernet MAC address

Processing of IP packets by network drivers

- IP Input
  - Put on IP input queue
  - IP destination = multicast or broadcast?
  - No: get MAC address with ARP
  - Yes: put IP datagram
- Ethernet Driver
  - IP destination of packet = local IP address?
  - No: get MAC address with ARP
  - Yes: demultiplex Ethernet Frame
- Loopback Driver
  - Put on IP input queue
  - Yes: put IP datagram
  - Yes: put IP datagram
Address Translation

(1) HOST-A wants to send an IP datagram to HOST-B
(2) HOST-A broadcasts an ARP request to all stations on the network: “What is the hardware address of HOST-B?”
(3) HOST-B responds with an ARP Reply which contains its hardware address
(4) HOST-A transmits the IP datagram to HOST-B

ARP Packet Format

- Ethernet Destination/Source Address: ff:ff:ff:ff:ff:ff is broadcast
- frame type: “0x0806” for ARP request/reply
- hw type: “1” for Ethernet MAC addresses
- prot type: “0x0800” for IP addresses
- hw size, prot size: size of the respective address in bytes.
- Op field: 1 = ARP request, 2 = ARP reply, 3 = RARP request, 4 = RARP reply
Example with tcpdump

Indy2 does “ping indy 1”:
20:56:42 8:0:69:7:e3:b3 Broadcast
  arp 60: arp who-has 192.0.1.5 tell 192.0.1.6
  arp 60: arp reply 192.0.1.5 is-at 8:0:69:7:e3:ab
  ip 98: 192.0.1.6 > 192.0.1.5: icmp: echo request
  ip 98: 192.0.1.5 > 192.0.1.6: icmp: echo

ARP Cache

• Since sending an ARP request/reply for each IP datagram is inefficient, hosts maintain a cache (ARP Cache) of current entries. The entries expire after 20 minutes.

• ARP cache of indy2 (displayed with arp -a):

  sparc3.CS.VIRGINIA.EDU (192.0.1.4) at 8:0:20:e:28:2d
  INDY2.CS.VIRGINIA.EDU (192.0.1.6) at 8:0:69:7:e3:b3
  ? (192.0.1.50) at 0:60:8:1e:2a:3d
  ? (192.0.1.255) at ff:ff:ff:ff:ff:ff permanent
Things to know about ARP

- What happens if an ARP Request is made for a non-existing host?
  Several ARP requests are made with increasing time intervals between requests. Eventually, ARP gives up.

- What if a host sends an ARP request for its own IP address?
  The other machines respond (gratuitous ARP) as if it was a normal ARP request. This is useful for detecting if an IP address has already been assigned.

- Routers can respond to an ARP request for a host that is on a different subnet (Proxy ARP)
  We will revisit this later, when we talk about subnets

ARP commands in BSD Unix (selection)

- **arp -a**: Displays the content of the ARP table
- **arp -d -a**: Clears the table
- **arp -d inet_addr**: Deletes the entry with `inet_addr`
- **arp -s hostname MAC_Address**: Add an entry to the ARP table (the MAC address is entered as six hexadecimal bytes separated by colons)
ARP commands in Windows (Selection)

arp -a
Displays current ARP entries

arp -g
Same as -a.

arp -d inet_addr
Deletes the host specified by inet_addr.

arp -s inet_addr MAC_Address
Adds the host and associates the Internet address with the Physical address eth_addr. (The Physical address is given as 6 hexadecimal bytes separated by hyphens.)

RARP - Reverse ARP

• Reverse ARP (RARP) performs a translation from a physical (MAC) address into a logical (IP) address.

• When does one need RARP?
  Hosts without secondary storage (e.g., X-terminals) do not know their IP address when they are booted.

• Packet format is the same as in ARP:
  – frame type: "0x0806" for ARP request/reply
  – Op field: 3 = RARP request
  4 = RARP reply
Example from Textbook

- A diskless host with name “sun” boots up:
  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:f6:4 
     This is the broadcast RARP request 
  2. 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 
     This is the response which contains the complete IP address 
  3. 0.14 8:0:20:3:f6:42 0:0:c0:6f:2d:40 ip 42: 
     sun.26999 > bsdi.tftp: 23 RRQ “8CFC0D21.SUN4C” Request to setup an TFTP read-request for bootstrapping.
**DHCP**

- DHCP = Dynamic Host Configuration Protocol
- Client/Server-based approach for dynamic assignment of IP addresses, default gateway, and other host configurations
- DHCP server maintains a pool of IP addresses, and assigns them upon request of a host

**Finding the DHCP server**

- DHCP server address is part of the host configuration!
  (So, assuming knowledge of the DHCP server would defeat purpose of dynamic configuration)
- A host broadcasts a DHCP Discover message
- Server responds
Finding the DHCP server

- **Problem**: If DHCP server is not on the same IP network as the host
  - Why is this a problem? Because IP routing cannot be used!
- **Solution**: Each network has a relay agent
  - Relay agent only knows the IP address of the DHCP agent
  - Forwards DHCP discover packet (via unicast) to DHCP server
  - Waits for response, and forwards response

![DHCP Diagram](image)

What is the difference between a router and a DHCP relay agent?

Fun Facts about DHCP

- DHCP is based on BOOTP, an earlier protocol for dynamic assignment of IP addresses
- DHCP servers are configured with blocks of IP addresses for assignment to hosts
- DHCP allows IP addresses to be “leased” for some periods of time. When the lease is up, the server can reassign the address.
- DHCP packets are transmitted as UDP/IP packets
  - How do the DHCP request packets get to the server from the client?
  - How do the reply packets from the DHCP server get to the client?
### DHCP Packet format

<table>
<thead>
<tr>
<th>Operation</th>
<th>HType</th>
<th>HLen</th>
<th>Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td>ciaddr</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>yiaddr</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>siaddr</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>giaddr</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>chaddr (16 bytes)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>sname (64 bytes)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>file (128 bytes)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **DHCP Discover**
- **DHCP Offer**
- **DHCP Request**
- **DHCP Ack**