PART V

MAPPING INTERNET ADDRESSES TO PHYSICAL ADDRESSES (ARP)
Motivation

- Must use hardware (physical) addresses to communicate over network
- Applications only use Internet addresses
Example

- Computers A and B on same network
- Application on A generates packet for application on B
- Protocol software on A must use B’s hardware address when sending a packet
UNICAST FROM A → B

1. A B'CAST TO ALL

   I AM IPA HWA
   I AM LOOKING FOR HW@ OF IPA

2. ALL IGNORE EXCEPT B

3. B SENDS I AM IPB HWB
   SENDING TO IPA HWA

ARP

ARP REQ

ARP REPLY
Consequence

- Protocol software needs a mechanism that maps an IP address to equivalent hardware address
- Known as *address resolution* problem
Address Resolution

- Performed at each step along path through Internet
- Two basic algorithms
  - Direct mapping
  - Dynamic binding
- Choice depends on type of hardware
Direct Mapping

- Easy to understand
- Efficient
- Only works when hardware address is small
- Technique: assign computer an IP address that encodes the hardware address
Example Of Direct Mapping

- Hardware: proNet ring network
- Hardware address: 8 bits
- Assume IP address 192.5.48.0 (24-bit prefix)
- Assign computer with hardware address $K$ an IP address 192.5.48.$K$
- Resolving an IP address means extracting the hardware address from low-order 8 bits
Example Of Direct Mapping

- Hardware: proNet ring network
- Hardware address: 8 bits
- Assume IP address 192.5.48.0 (24-bit prefix)
- Assign computer with hardware address $K$ an IP address 192.5.48.$K$
- Resolving an IP address means extracting the hardware address from low-order 8 bits
Dynamic Binding

- Needed when hardware addresses are large (e.g., Ethernet)
- Allows computer A to find computer B’s hardware address
  - A starts with B’s IP address
  - A knows B is on the local network
- Technique: broadcast query and obtain response
- Note: dynamic binding only used across one network at a time
Internet Address Resolution Protocol (ARP)

- Standard for dynamic address resolution in the Internet
- Requires hardware broadcast
- Intended for LAN
- Important idea: ARP only used to map addresses within a single physical network, never across multiple networks
ARP

- Machine A broadcasts ARP request with B’s IP address
- All machines on local net receive broadcast
- Machine B replies with its physical address
- Machine A adds B’s address information to its table
- Machine A delivers packet directly to B
Illustration Of ARP Request And Reply Messages

A broadcasts request for B
(across local net only)

B replies to request
# ARP Packet Format When Used With Ethernet

<table>
<thead>
<tr>
<th>0</th>
<th>8</th>
<th>16</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHERNET ADDRESS TYPE (1)</td>
<td>IP ADDRESS TYPE (0800)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ETH ADDR LEN (6)</td>
<td>IP ADDR LEN (4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SENDER'S ETH ADDR (first 4 octets)</td>
<td>SENDER'S ETH ADDR (first 4 octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SENDER'S IP ADDR (last 2 octets)</td>
<td>SENDER'S IP ADDR (first 2 octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET'S ETH ADDR (all 4 octets)</td>
<td>TARGET'S ETH ADDR (all 4 octets)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**
- **Address Type:** Ethernet (16909) and IPv4 (0800)
- **Lengths:** 6 octets for Ethernet and 4 octets for IP
- **Senders:**
  - 4 octets for first (sender's Ethernet)
  - 2 octets for last (sender's IP)
- **Targets:**
  - 4 octets for all (target's Ethernet)

**Diagram:**
- **ARP Packet Format:**
  - 0-4: Ethernet Address Type (1)
  - 5-8: IP Address Type (0800)
  - 9-12: Ethernet Address Length (6)
  - 13-16: IP Address Length (4)
  - 17-21: Sender's Ethernet Address (4 octets)
  - 22-25: Sender's IP Address (2 octets)
  - 26-30: Target's Ethernet Address (4 octets)
  - 31: Target's IP Address (all 4 octets)
### ARP Packet Format When Used With Ethernet

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETH ADDR LEN (6)</td>
<td>Length of Ethernet Address field</td>
</tr>
<tr>
<td>IP ADDR LEN (4)</td>
<td>Length of IP Address field</td>
</tr>
<tr>
<td>SRC ETH ADDR (first 4 octets)</td>
<td>Source Ethernet Address</td>
</tr>
<tr>
<td>SRC IP ADDR (last 2 octets)</td>
<td>Source IP Address</td>
</tr>
<tr>
<td>SRC MAC ADDR (last 2 octets)</td>
<td>Source MAC Address</td>
</tr>
<tr>
<td>SRC MAC ADDR (first 2 octets)</td>
<td>Source MAC Address</td>
</tr>
<tr>
<td>TARGET ETH ADDR (last 4 octets)</td>
<td>Target Ethernet Address</td>
</tr>
<tr>
<td>TARGET MAC ADDR (all 4 octets)</td>
<td>Target MAC Address</td>
</tr>
</tbody>
</table>

**Notes:**
- **ETH ADDR LEN (6)**: Length of Etherent Address field
- **IP ADDR LEN (4)**: Length of IP Address field
- **AMD**: Address Resolution Protocol
Observations About Packet Format

- General: can be used with
  - Arbitrary hardware address
  - Arbitrary protocol address (not just IP)
- Variable length fields (depends on type of addresses)
- Length fields allow parsing of packet by computer that does not understand the two address types
Retention Of Bindings

- Cannot afford to send ARP request for each packet
- Solution
  - Maintain a table of bindings
- Effect
  - Use ARP one time, place results in table, and then send many packets
ARP Caching

- ARP table is a cache
- Entries time out and are removed
- Avoids stale bindings
- Typical timeout: 20 minutes
Algorithm For Processing ARP Requests

- Extract sender’s pair, (IA, EA) and update local ARP table if it exists
- If this is a request and the target is “me”
  - Add sender’s pair to ARP table if not present
  - Fill in target hardware address
  - Exchange sender and target entries
  - Set operation to reply
  - Send reply back to requester
Algorithm Features

- If A ARPs B, B keeps A’s information
  - B will probably send a packet to A soon
- If A ARPs B, other machines do not keep A’s information
  - Avoids clogging ARP caches needlessly
Conceptual Purpose Of ARP

- Isolates hardware address at low level
- Allows application programs to use IP addresses
ARP Encapsulation

- ARP message travels in data portion of network frame
- We say ARP message is *encapsulated*
Illustration Of ARP Encapsulation
Ethernet Encapsulation

- ARP message placed in frame data area
- Data area padded with zeroes if ARP message is shorter than minimum Ethernet frame
- Ethernet type 0x0806 used for ARP
Reverse Address Resolution Protocol

- Maps Ethernet address to IP address
- Same packet format as ARP
- Intended for bootstrap
  - Computer sends its Ethernet address
  - RARP server responds by sending computer’s IP address
- Seldom used (replaced by DHCP)
Summary

- Computer’s IP address independent of computer’s hardware address
- Applications use IP addresses
- Hardware only understands hardware addresses
- Must map from IP address to hardware address for transmission
- Two types
  - Direct mapping
  - Dynamic mapping
Summary (continued)

- Address Resolution Protocol (ARP) used for dynamic address mapping
- Important for Ethernet
- Sender broadcasts ARP request, and target sends ARP reply
- ARP bindings are cached
- Reverse ARP was originally used for bootstrap