Chapter 7

ARP and RARP
Outline

- ARP
- ARP Package
- RARP
Logical Addresses

- The hosts and routers are recognized at the network level by their *logical addresses*
  - A *logical address* is an internet address
  - Called a *logical* address because it is usually implemented in software
  - The logical addresses in the TCP/IP are called **IP address** and are 32 bits long
Physical Address

- However, hosts/routers are recognized at the physical layer by their *physical address*
  - A *physical address* is an local address
  - Called a *physical* address because it is usually implemented in hardware

- Examples
  - 48-bit MAC addresses in Ethernet
Translation

- We need both the physical address and the logical address for packet delivery
- Thus, we need to be able to map a logical address to its corresponding physical address and vice versa

Solutions
- Static mapping
- Dynamic mapping
Static Mapping

- Create a table that associates a logical address with a physical address and store in each machine.
- However, physical addresses may change:
  - A machine could change its NIC resulting in a new physical address.
  - In some LANs, such as LocalTalk, the physical address changes every time the computer is turned on.
  - A mobile station can move from one physical network to another, resulting in a change in its physical address.
Dynamic Mapping

- Use a protocol to find another address

- **ARP**: Address Resolution Protocol
  - Map a logical address to a physical address

- **RARP**: Reverse Address Resolution Protocol
  - Map a physical address to a logical address
**ARP and RARP**

![Diagram showing ARP and RARP](image-url)
Position of ARP and RARP in TCP/IP Protocol Suite
ARP Operation

- To find the physical address of another host or router on its network
  - Send an ARP request message

- ARP request message
  - The physical address of the sender
  - The IP address of the sender
  - The physical address of the receiver is 0s
  - The IP address of the receiver
ARP Operation (Cont.)

- Then, ARP request message is broadcast by the physical layer
  - For example: in Ethernet, MAC header’s destination address is all 1s (broadcast address)
  - Received by every station on the physical network
- The intended recipient send back an ARP reply message
  - ARP reply message packet is *unicast*
Figure 7-3

**ARP Operation**

Looking for physical address of a node with IP address **141.23.56.23**

a. ARP request is broadcast

The node physical address is **A46EF45983AB**

b. ARP reply is unicast
## ARP Packet

<table>
<thead>
<tr>
<th>Hardware Type</th>
<th>Protocol Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware length</td>
<td>Protocol length</td>
</tr>
</tbody>
</table>

- **Sender hardware address**
  
  (For example, 6 bytes for Ethernet)

- **Sender protocol address**
  
  (For example, 4 bytes for IP)

- **Target hardware address**
  
  (For example, 6 bytes for Ethernet)
  
  (It is not filled in a request)

- **Target protocol address**
  
  (For example, 4 bytes for IP)
Packet Format

- **HTYPE (Hardware type)**
  - 16-bit field defining the underlying type of the network
  - Ethernet is given the type 1
  - ARP can be used on any physical network

- **PTYPE (Protocol type)**
  - 16-bit field defining the protocol
  - IPv4 is $0800_{16}$
  - ARP can be used with any higher-level protocol
Packet Format (Cont.)

- **HLEN (Hardware length)**
  - 8-bit field defining the length of the physical address in bytes
  - Ethernet has the value of 6

- **PLEN (Protocol length)**
  - 8-bit field defining the length of the logical address in bytes
  - IPv4 has the value of 4

- **OPER (Operation)**
  - 16-bit field defining the type of packet
  - (1) = ARP request, (2) = ARP reply
Packet Format (Cont.)

- **SHA (Sender hardware address)**
  - A variable-length field defining the physical address of the sender

- **SPA (Sender protocol address)**
  - A variable-length field defining the logical address of the sender
Packet Format (Cont.)

- **THA (Target hardware address)**
  - A variable-length field defining the physical address of the target
  - For an ARP request operation packet
    - This field is all 0s

- **TPA (Target protocol address)**
  - A variable-length field defining the logical address of the target
An ARP packet is encapsulated directly into a data link frame.

Type field indicates that the data carried by the frame is an ARP packet.
Operations

- The sender knows the target’s IP address
- IP asks ARP to create an ARP request message
  - The sender physical address
  - The sender IP address
  - The target physical address field is filled with 0s
  - The target IP address
- The message is passed to the data link layer to encapsulate in a data link frame
  - Physical destination address is broadcast address
Operations (Cont.)

- Every host or routers receives the frame and since the destination address is broadcast, pass it to the ARP
  - All machines’ ARP except the one targeted drop the packet
- The target reply with an ARP reply message that contains its physical address and is unicast
- The sender receives the reply message and knows the target’s physical address
Four Cases to Use ARP

- **Case 1:** The sender is a host and wants to send a packet to another host on the same network
  - Use ARP to find another host’s physical address

- **Case 2:** The sender is a host and wants to send a packet to another host on another network
  - Sender looks at its routing table
  - Find the IP address of the next hop (router) for this destination
  - Use ARP to find the router’s physical address
Four Cases Using ARP: Case 1

Case 1. A host has a packet to send to another host on the same network.

Target IP address:
Destination address in the IP datagram
Four Cases Using ARP: Case 2

Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.
Four Cases to Use ARP (Cont.)

- **Case 3:** the sender is a router and received a datagram destined for a host on another network
  - Router check its routing table
  - Find the IP address of the next router
  - Use ARP to find the next router’s physical address

- **Case 4:** the sender is a router that has received a datagram destined for a host in the same network
  - Use ARP to find this host’s physical address
Four Cases Using ARP: Case 3

Target IP address:
IP address of the appropriate router found in the routing table

Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.
Four Cases Using ARP: Case 4

Case 4. A router receives a packet to be sent to a host on the same network.
An ARP request is broadcast; an ARP reply is unicast.
Example 1

- A host with IP address 130.23.43.20 and physical address 0xB23455102210
- Another host with IP address 130.23.43.25 and physical address 0xA46EF45983AB.
- The two hosts are on the same Ethernet network
- Show the ARP request and reply packets encapsulated in Ethernet frames
Solution

- Figure 7.7 shows the ARP request and reply packets
- Note that
  - The IP addresses are shown in hexadecimal
Example 1

System A

<table>
<thead>
<tr>
<th>0x0001</th>
<th>0x0800</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x06</td>
<td>0x04</td>
</tr>
<tr>
<td>0xB23455102210</td>
<td></td>
</tr>
<tr>
<td>0x82172B14</td>
<td></td>
</tr>
<tr>
<td>0x00000000000000</td>
<td></td>
</tr>
<tr>
<td>0x82172B19</td>
<td></td>
</tr>
</tbody>
</table>

System B

130.23.43.20

xB23455102210

130.23.43.25

x46EF45983AB

ARP Request

CRC Data 28 bytes 0x0806 0xB23455102210 0xFFFFFFFF Preamble and SFD
Example 1 (Continued)
Proxy ARP

- Used to create a subnetting effect
- A router running a proxy ARP
  - Its ARP acts on behalf of a set of hosts
  - If it receives an ARP request message looking for the address of one of these hosts
    - The router sends an ARP reply announcing its own hardware (physical) address
  - After the router receives the actual IP packet
    - It sends the packet to the appropriate host or router
Example

- Administrator need to create a subnet without changing the whole system

- Add a router running a proxy ARP
Proxy ARP

The proxy ARP router replies to any ARP request received for destinations 141.23.56.21, 141.23.56.22, and 141.23.56.23.
7.2

ARP PACKAGE
ARP Package

- Five components in an ARP package
  - A cache table
  - Queues
  - An output module
  - An input module
  - A cache-control module
**ARP Components**

**IP layer**

**ARP**

- **Output module**
  - Queues
  - IP packet

- **Input module**

**Data link layer**

- ARP packet (request)
- ARP packet (request or reply)
- ARP packet (reply)
- IP packet with resolved hardware address
- ARP packet (request)
- IP packet with resolved hardware address

Cache Table

- Inefficient to use ARP to each datagram destined for the same host or router
  - Introduce the cache table

- Cache table: an array of entries that contains the following’s entries
Content of a Cache Table Entry

- State:
  - FREE: the time-to-live for this entry has expired
  - PENDING: a request for this entry has been sent, but the reply has not yet been received
  - RESOLVED: the entry is complete and valid

- Hardware type
- Protocol type
- Hardware length
- Protocol length
  - Above fields are all the same as in the ARP packet
Content of a Cache Table Entry (Cont.)

- Interface number
- Queue number: ARP uses numbered queues to enqueue the packet waiting for address resolution
- Attempts: the number of times an ARP request is sent out for this entry
- Time-out: the lifetime of an entry in seconds
- Hardware address: the destination hardware address
- Protocol address: the destination IP address
Queues

- ARP package maintains a set of queues to hold the IP packets while ARP tries to resolve the hardware address.
- Packets for the same destination are usually enqueued in the same queue.
- The output module sends unsolved packets into the queue.
- The input module removes a packet from the queue and sends it, with the resolved physical address, to data link layer for transmission.
Output Module

- Wait until an IP packet from the IP software
- Check the cache table if receiving a IP packet
  - If found and state = RESOLVED
    - Passed to the data link layer for transmission
  - If found and state = PENDING
    - Send packet to this queue and wait
  - If not found
    - Create an entry with state = PENDING
    - Create a queue and enqueue this packet
    - Send an ARP request
Input Module

- Wait until an ARP packet (request or reply) arrives and check the cache table
  - If found state = PENDING
    - Copy the target hardware address in the packet
    - Change the state to RESOLVED
    - Set the value of TIME-OUT for this entry
    - Dequeue the packets from the corresponding queue and set them to the data link layer
Input Module (Cont.)

- If found and state = RESOLVED
  - Copy the target hardware address in the packet
  - Set the value of TIME-OUT for this entry
  - This is because the target hardware address could have been changed

- If not found
  - Create a new entry and adds it to the table

- If the packet is a request
  - Send an ARP reply
Cache-Control Module

- Maintain the cache table by periodically check the cache table, entry by entry
- If state is PENDING
  - Increment the value of attempts by 1
  - If (attempts greater than maximum)
    - Change the state to FREE and Destroy the corresponding queue
  - Else
    - Send an ARP request
Cache-Control Module (Cont.)

- If state is RESOLVED
  - Decrement the value of time-out by the value of elapsed time
  - If (time-out <= 0)
    - Change the state to FREE
    - Destroy the corresponding queue
- If state is FREE
  - Continue to the next entry
# Original Cache Table

<table>
<thead>
<tr>
<th>State</th>
<th>Queue</th>
<th>Attempt</th>
<th>Time-out</th>
<th>Protocol</th>
<th>Addr</th>
<th>Hardware Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>5</td>
<td>900</td>
<td>180.3.6.1</td>
<td></td>
<td></td>
<td>ACAE32457342</td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td>2</td>
<td>129.34.4.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>14</td>
<td>5</td>
<td>201.11.56.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>8</td>
<td>450</td>
<td>114.5.7.89</td>
<td></td>
<td></td>
<td>457342ACAE32</td>
</tr>
<tr>
<td>P</td>
<td>12</td>
<td>1</td>
<td>220.55.5.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>9</td>
<td>60</td>
<td>19.1.7.82</td>
<td></td>
<td></td>
<td>4573E3242ACA</td>
</tr>
<tr>
<td>P</td>
<td>18</td>
<td>3</td>
<td>188.11.8.71</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The ARP output module receives an IP datagram from the IP layer with the destination address 114.5.7.89. It checks the cache table and finds that an entry exists for this destination with the RESOLVED state. It extracts the hardware address, which is 457342ACAE32, and sends the packet and the address to the data link layer.
Twenty seconds later, the ARP output module receives an IP datagram from the IP layer with the destination address 116.1.7.22.

It checks the cache table and does not find this destination in the table.

The module adds an entry to the table with the state PENDING and the Attempt value 1.

It also creates a new queue for this destination and enqueues the packet.

It then sends an ARP request to the data link layer for this destination.
## Cache table for Example 3

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>5</td>
<td>900</td>
<td>180.3.6.1</td>
<td></td>
<td>ACAE32457342</td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td>2</td>
<td>129.34.4.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>14</td>
<td>5</td>
<td>201.11.56.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>8</td>
<td>450</td>
<td>114.5.7.89</td>
<td>457342ACAE32</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>12</td>
<td>1</td>
<td>220.55.5.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>23</td>
<td>1</td>
<td>116.1.7.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>9</td>
<td>60</td>
<td>19.1.7.82</td>
<td>4573E3242ACA</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>18</td>
<td>3</td>
<td>188.11.8.71</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fifteen seconds later, the ARP input module receives an ARP packet with target protocol (IP) address 188.11.8.71. The module checks the table and finds this address. It changes the state of the entry to RESOLVED and sets the time-out value to 900. The module then adds the target hardware address (E34573242ACA) to the entry. Now it accesses queue 18 and sends all the packets in this queue, one by one, to the data link layer.
## Cache table for Example 4

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>5</td>
<td>900</td>
<td>180.3.6.1</td>
<td></td>
<td>ACAE32457342</td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td>2</td>
<td>129.34.4.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>14</td>
<td>5</td>
<td>201.11.56.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>8</td>
<td>450</td>
<td>114.5.7.89</td>
<td>457342ACAE32</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>12</td>
<td>1</td>
<td>220.55.5.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>23</td>
<td>1</td>
<td>116.1.7.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>9</td>
<td>60</td>
<td>19.1.7.82</td>
<td>4573E3242ACA</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>18</td>
<td>900</td>
<td>188.11.8.71</td>
<td>E34573242ACA</td>
<td></td>
</tr>
</tbody>
</table>
Example 5

- Twenty-five seconds later, the cache-control module wakes up
- The time-out values for the first three resolved entries are decremented by 60
- The time-out value for the last resolved entry is decremented by 25
- The state of the next-to-the last entry is changed to FREE because the time-out is zero
Example 5 (Cont.)

- For each of the three pending entries, the value of the attempts field is incremented by one.
- Then, the attempts value for one entry (the one with IP protocol address 201.11.56.7) is more than the maximum:
  - the state is changed to FREE, the queue is deleted.
  - An ICMP message is sent to the original destination.
## Cache Table for Example 5

<table>
<thead>
<tr>
<th>State</th>
<th>Queue</th>
<th>Attempt</th>
<th>Time-out</th>
<th>Protocol</th>
<th>Addr</th>
<th>Hardware Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>5</td>
<td>840</td>
<td>180.3.6.1</td>
<td></td>
<td>ACAE32457342</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td>3</td>
<td>129.34.4.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>8</td>
<td>390</td>
<td>114.5.7.89</td>
<td></td>
<td>457342ACAE32</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>12</td>
<td>2</td>
<td>220.55.5.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>23</td>
<td>2</td>
<td>116.1.7.22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>18</td>
<td>875</td>
<td>188.11.8.71</td>
<td></td>
<td>E34573242ACA</td>
<td></td>
</tr>
</tbody>
</table>

RARP
RARP

- A diskless machine is usually booted from ROM
- It cannot include the IP address
  - IP address are assigned by the network administrator
- Obtain its logical address by the physical address using the RARP protocol
RARP Operation

My physical address is A46EA4578236. I am looking for my IP address.

a. RARP request is broadcast

Your IP address is: 141.14.56.21

b. RARP reply is unicast
The RARP request packets are broadcast; the RARP reply packets are unicast.
Packet Format

- The format of the RARP packet is the same as the ARP packet
- Except that the operation field is
  - Three for RARP request message
  - Four for RARP reply message
# RARP Packet

<table>
<thead>
<tr>
<th>Hardware type</th>
<th>Protocol type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware length</td>
<td>Protocol length</td>
</tr>
</tbody>
</table>

**Operation**

- Request 3, Reply 4

- **Sender hardware address**
  (For example, 6 bytes for Ethernet)

- **Sender protocol address**
  (For example, 4 bytes for IP)
  (It is not filled for request)

- **Target hardware address**
  (For example, 6 bytes for Ethernet)
  (It is not filled for request)

- **Target protocol address**
  (For example, 4 bytes for IP)
  (It is not filled for request)
Encapsulation of RARP Packet

Type: 0x8035

- Preamble and SFD: 8 bytes
- Destination address: 6 bytes
- Source address: 6 bytes
- Type: 2 bytes
- Data: 4 bytes
- CRC: 4 bytes

RARP request or reply packet
Alternative Solutions to RARP

- When a diskless computer is booted, it needs more information in addition to its IP address:
  - The subnet mask
  - The IP address of a router
  - The IP address of a name server

- RARP cannot provide this extra information

- Two protocols, BOOTP and DHCP, can be used instead of RARP