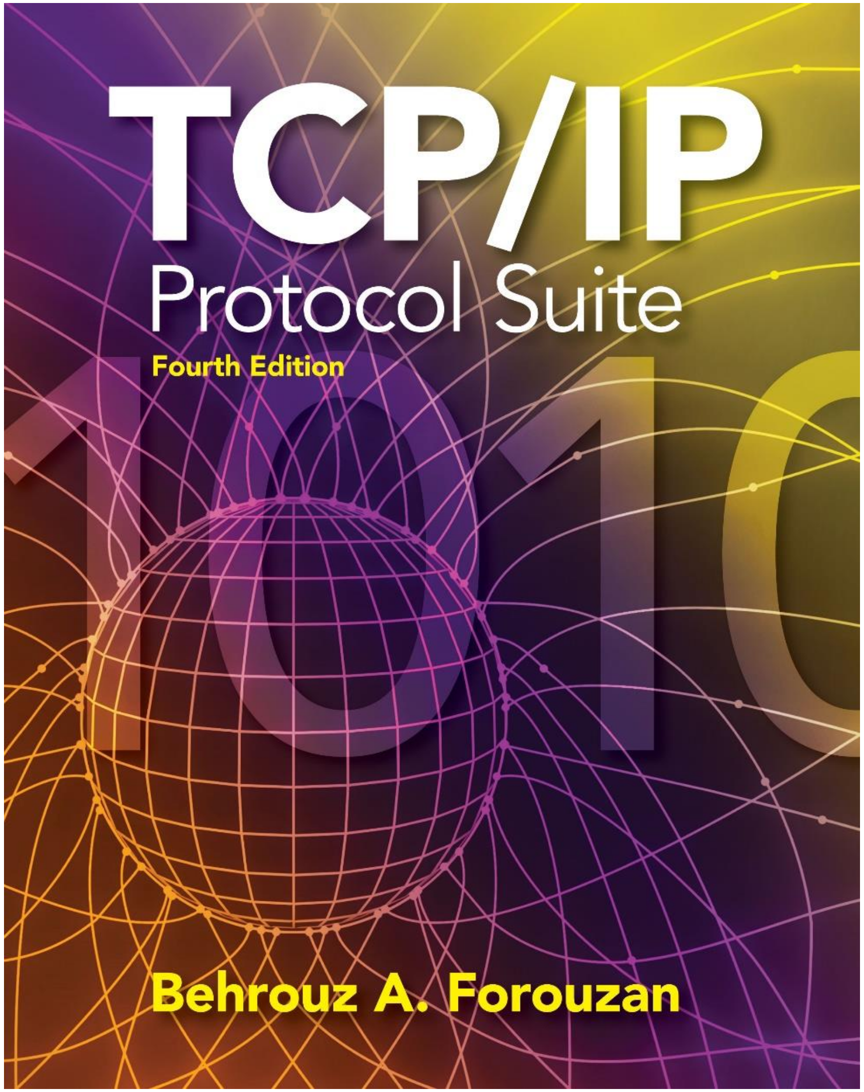
Chapter 8

Address Resolution Protocol (ARP)



OBJECTIVES:

- To make a distinction between logical address (IP address) and physical address (MAC address).
- To describe how the mapping of a logical address to a physical address can be static or dynamic.
- To show how the address resolution protocol (ARP) is used to dynamically map a logical address to a physical address.
- To show that the proxy ARP can be used to create a subnetting effect.
- To show that an ARP software package can be made of five components.
- To show the pseudocode for each module used in the ARP software package.

Chapter Outline

8.1 Address Mapping

8.2 The ARP Protocol

8-1 ADDRESS MAPPING

An internet is made of a combination of physical networks connected together by internetworking devices such as routers. A packet starting from a source host may pass through several different physical networks before finally reaching the destination host.

8-1 ADDRESS MAPPING

The delivery of a packet to a host or a router requires two levels of addressing: *logical* and *physical*. We need to be able to map a logical address to its corresponding physical address and vice versa. These can be done using either *static* or *dynamic* mapping.

- The hosts and routers are recognized at the network level by their **logical addresses.** A logical address is an internetwork address. Its jurisdiction is **universal**.
- At the physical level, the hosts and routers are recognized by their **physical addresses**. A physical address is a local address. Its jurisdiction is a **local network**. It should be unique locally. **but not necessarily universally**

Topics Discussed in the Section

- ✓ **Static Mapping**
- ✓ **Dynamic Mapping**

8-1 ADDRESS MAPPING

Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver. So, a mapping corresponds a logical address to a physical address we need .

8-1 ADDRESS MAPPING

Static Mapping: it means creating a table that associates a logical address with a physical address. This table is stored in each machine on the network. This has some limitations because physical addresses may change in the following ways:

1. A machine could change its NIC, resulting in a new physical address.
2. In some LANs, such as LocalTalk, the physical address changes every time the computer is turned on.
3. A mobile computer can move from one physical network to another, resulting in a change in its physical address.

Dynamic mapping: each time a machine knows the logical address of another machine, it can use a protocol to find the physical address. Two protocols have been designed to perform dynamic mapping: Address Resolution Protocol (**ARP**) and Reverse Address Resolution Protocol (**RARP**). ARP maps a logical address to a physical address; RARP maps a physical address to a logical address

Topics Discussed in the Section

- ✓ **Packet Format**
- ✓ **Encapsulation**
- ✓ **Operation**
- ✓ **Proxy ARP**

Figure 8.1 *Position of ARP in TCP/IP protocol suite*

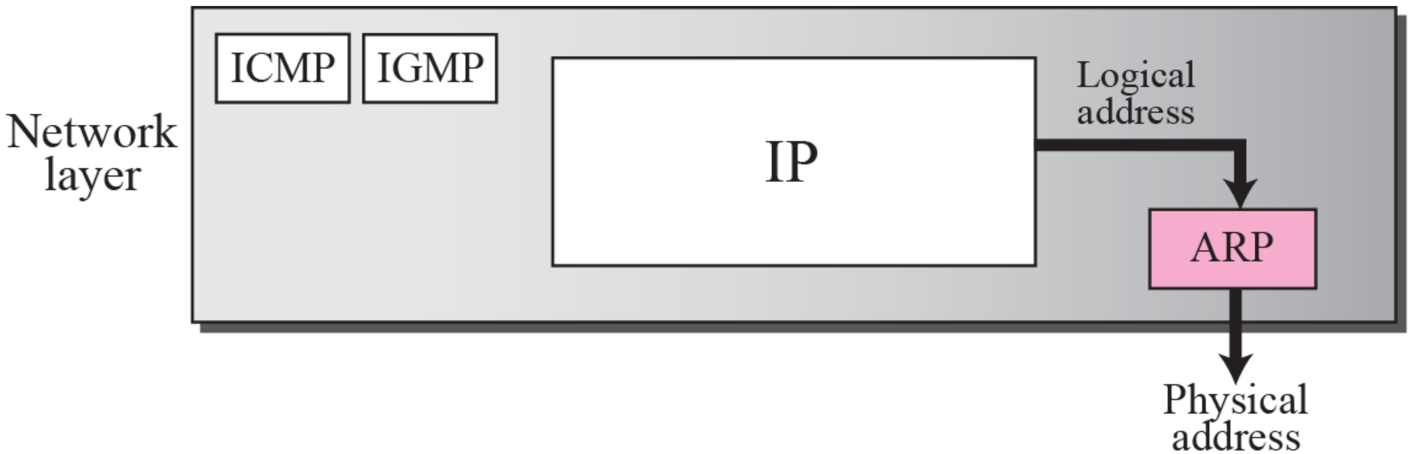
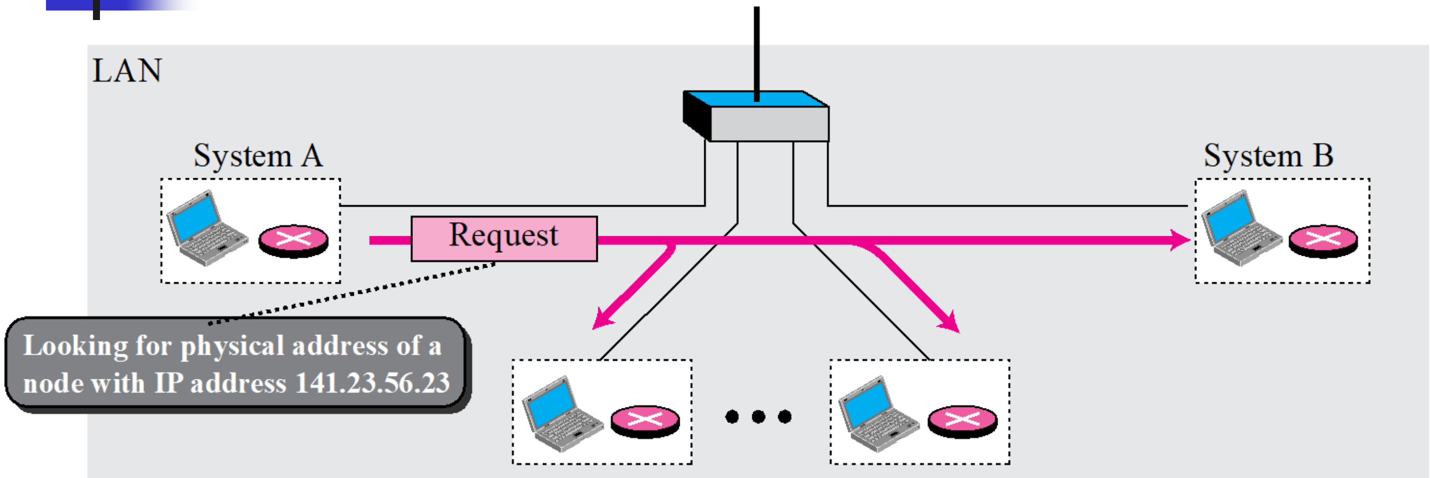
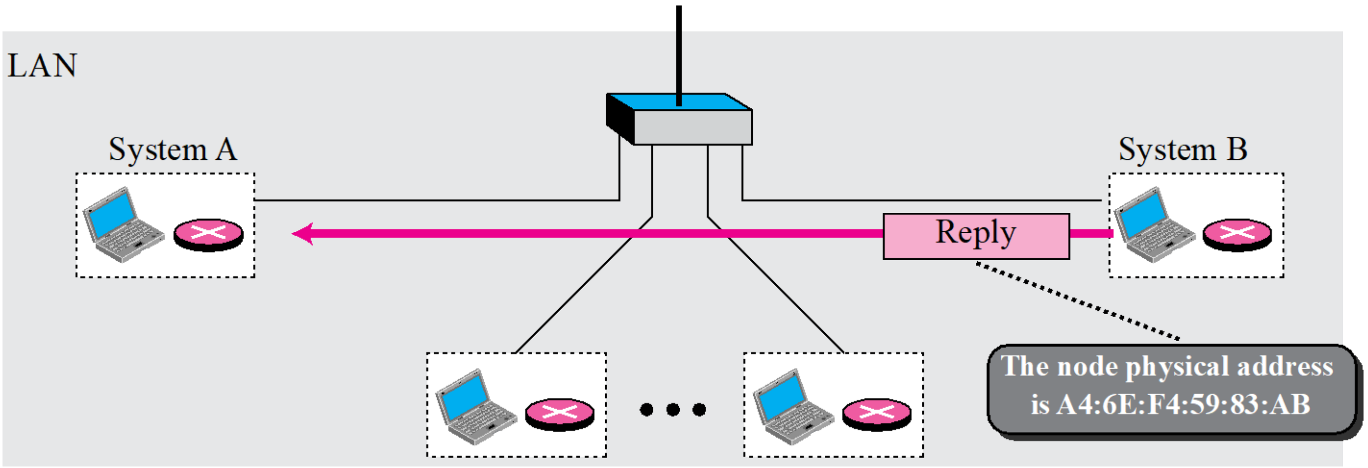


Figure 8.2 *ARP operation*



a. ARP request is broadcast



b. ARP reply is unicast

Figure 8.3 *ARP packet*

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
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)		

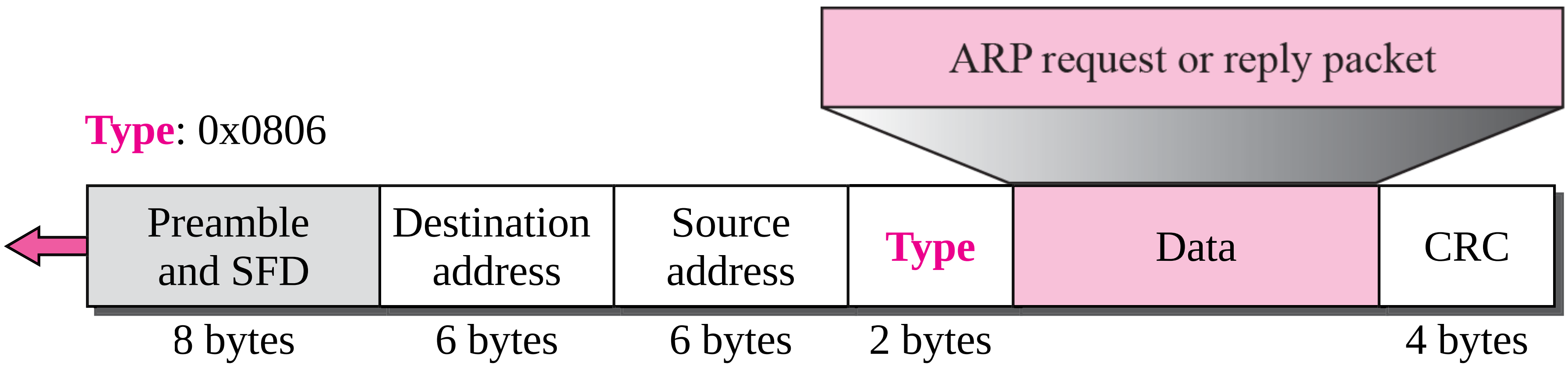
- **Hardware type.** This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given the type 1
- **Protocol type.** This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 0800_{16}. ARP can be used with any higher-level protocol.
- **Hardware length.** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.
- **Protocol length.** This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.

Figure 8.3 *ARP packet*

- **Operation.** This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1), ARP reply (2).
- **Sender hardware address.** This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.
- **Sender protocol address.** This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.
- **Target hardware address**. This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is all 0s because the sender does not know the physical address of the target.
- **Target protocol address.** This is a variable-length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long.

Figure 8.4 *Encapsulation of ARP packet*

An ARP packet is encapsulated directly into a data link frame. For example, in Figure 8.4 an ARP packet is encapsulated in an Ethernet frame. Note that the type field indicates that the data carried by the frame is an ARP packet.



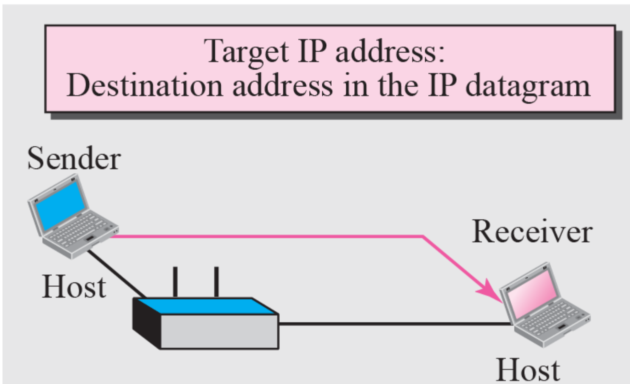
Note

An ARP request is broadcast; an ARP reply is unicast.

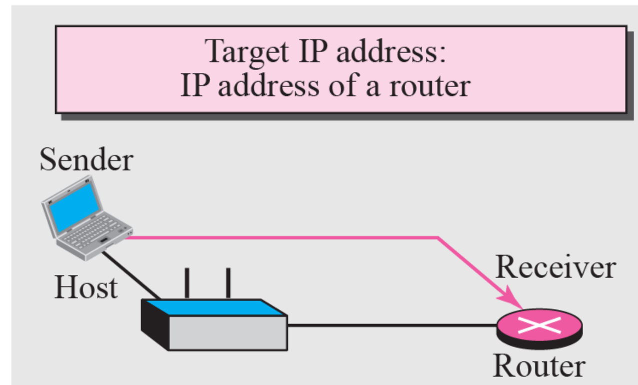
Figure 8.5 *Four cases using ARP*

The four cases in which a host or router needs to use ARP.

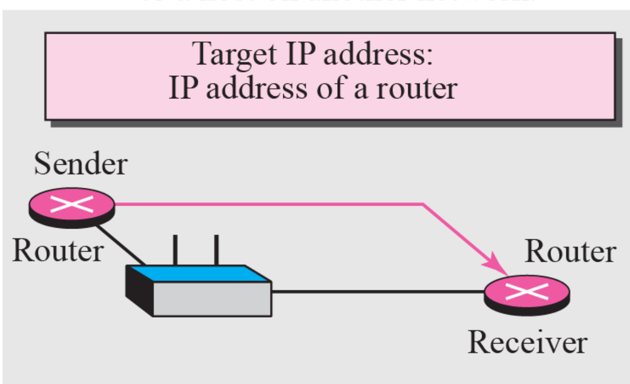
Case 1: A host has a packet to send to a host on the same network.



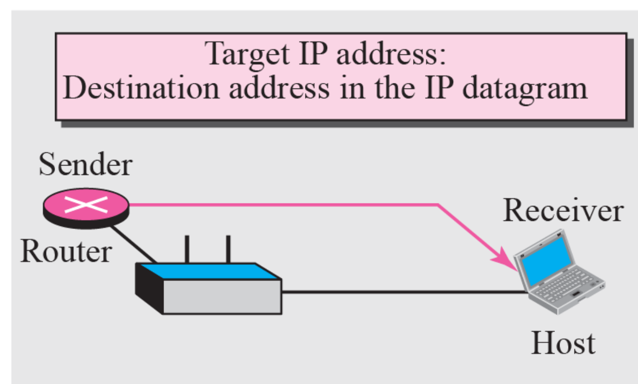
Case 2: A host has a packet to send to a host on another network.



Case 3: A router has a packet to send to a host on another network.



Case 4: A router has a packet to send to a host on the same network.



These are seven steps involved in an ARP process:

1-The sender knows the IP address of the target. We will see how the sender obtains this shortly.

2-IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address. The target physical address field is filled with 0s.

3-The message is passed to the data link layer where it is encapsulated in a frame using the physical address of the sender as the source address and the physical broadcast address as the destination address.

4- Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP. All machines except the one targeted drop the packet. The target machine recognizes the IP address.

5-The target machine replies with an ARP reply message that contains its physical address. The message is unicast.

6-The sender receives the reply message. It now knows the physical address of the target machine.

7-The IP datagram, which carries data for the target machine, is now encapsulated in a frame and is unicast to the destination.

Example 8.1

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

Figure 8.6 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. Also note that the IP addresses are shown in hexadecimal.

Figure 8.6 *Example 8.1*

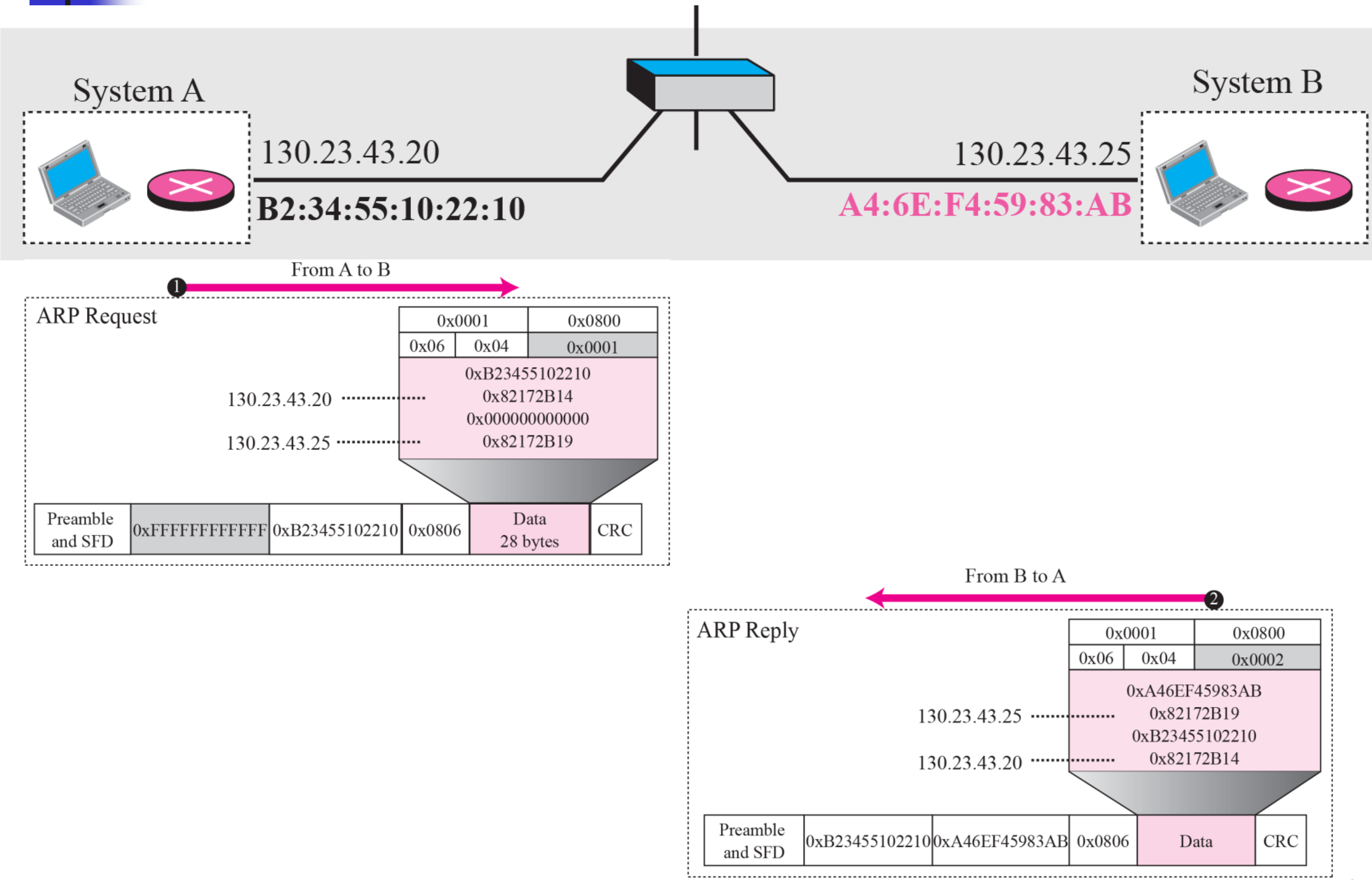
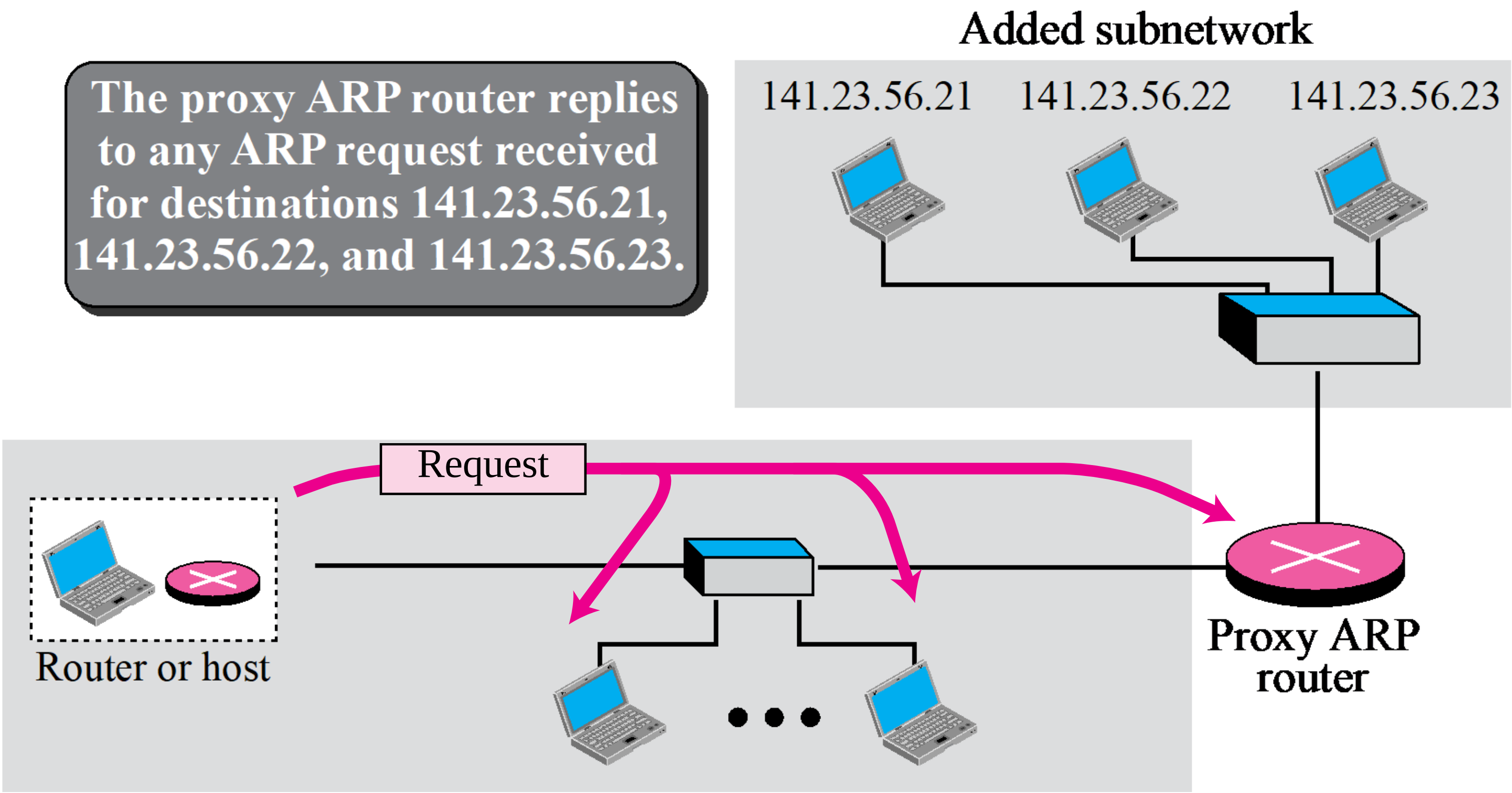


Figure 8.7 *Proxy ARP*

A proxy ARP is an ARP that acts on behalf of a set of hosts. Whenever a router running a proxy ARP receives an ARP request looking for the IP 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.



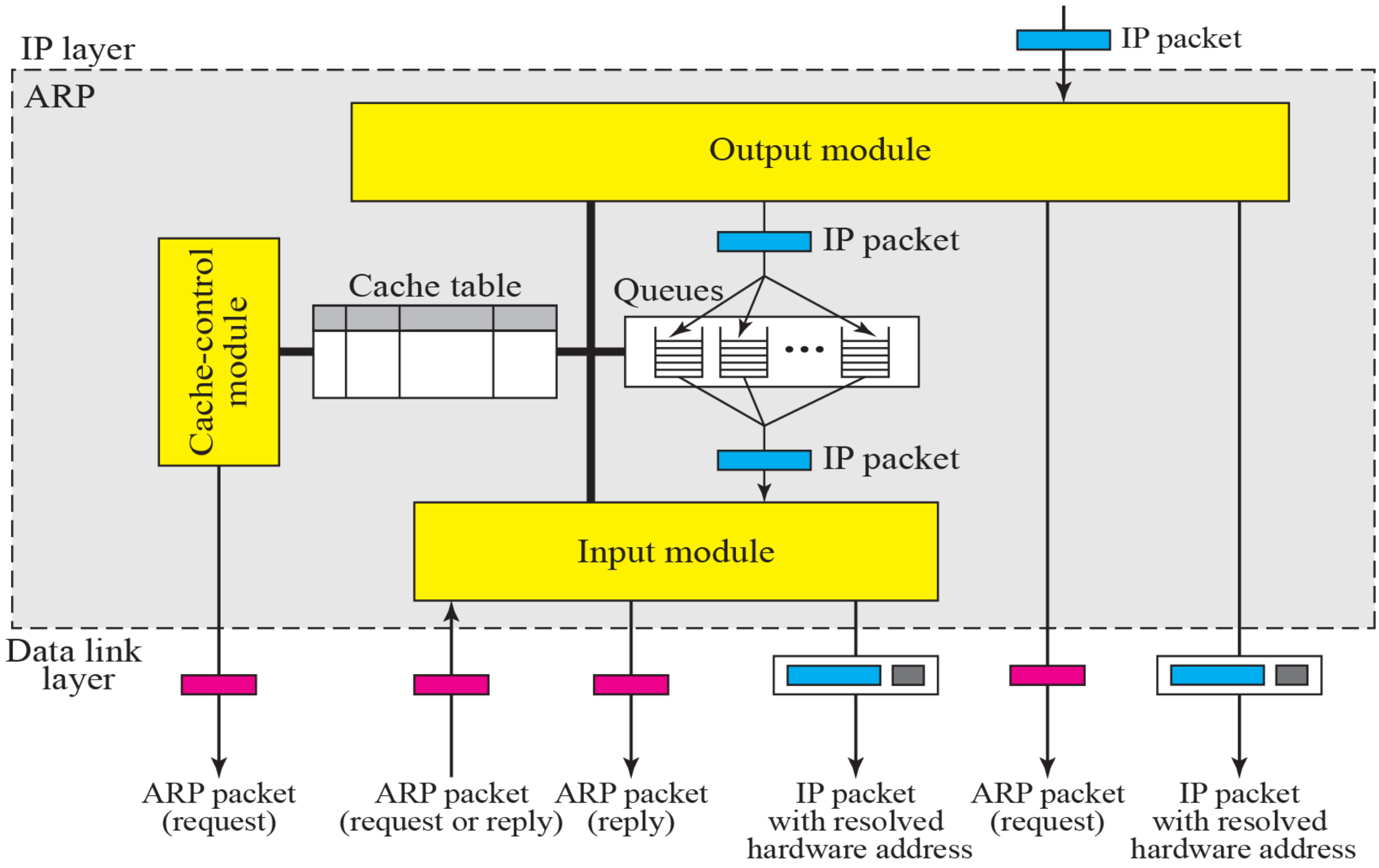
8-4 ARP PACKAGE

In this section, we give an example of a simplified ARP software package. The purpose is to show the components of a hypothetical ARP package and the relationships between the components. Figure 8.13 shows these components and their interactions. We can say that this ARP package involves five components: a cache table, queues, an output module, an input module, and a cache-control module.

Topics Discussed in the Section

- ✓ **Cache Table**
- ✓ **Queues**
- ✓ **Output Module**
- ✓ **Input Module**
- ✓ **Cache-Control Module**

Figure 8.13 *ARP components*



ARP software package can be made of five components:

Cache-Control Module: The cache-control module is responsible for maintaining the cache table. It periodically (for example, every 5 s) checks the cache table, entry by entry.

Input module: its waits until an ARP packet (request or reply) arrives. The input module checks the cache table to find an entry corresponding to this ARP packet. The target protocol address should match the protocol address of the entry.

Queues: Our ARP package maintains a set of queues, one for each destination, **to hold the IP packets while ARP tries to resolve the hardware address.**

Output module: it waits for an IP packet from the IP software. The output module checks the cache table to find an entry corresponding to the destination IP address of this packet. The destination IP address of the IP packet must match the protocol address of the entry.

Cache Table Fields

A sender usually has more than one IP datagram to send to the same destination. It is inefficient to use the ARP protocol for each datagram destined for the same host or router. The solution is the cache table. When a host or router receives the corresponding physical address for an IP datagram, the address can be saved in the cache table.

The Cache fields are:

- **State.** This column shows the state of the entry. It can have one of three values: FREE, PENDING, or RESOLVED.
- **Hardware type, Protocol type, Hardware length, Protocol length**: These column are the same as the corresponding field in the ARP packet.
- **Interface number:** A router (or a multihomed host) can be connected to different networks, each with a different interface number. Each network can have different.

Cache Table Fields

- **Queue number:** ARP uses numbered queues to enqueue the packets waiting for address resolution. Packets for the same destination are usually enqueued in the same queue.
- **Attempts:** This column shows the number of times an ARP request is sent out for this entry.
- **Time-out:** This column shows the lifetime of an entry in seconds.
- **Hardware address:** This column shows the destination hardware address. It remains empty until resolved by an ARP reply.
- Protocol address: This column shows the destination IP address.

Cache Table:

Table 8.5 *Original cache table used for examples*

State	Queue	Attempt	Time-Out	Protocol Addr.	Hardware Addr.
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

Example 8.2

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 (R in the table). It extracts the hardware address, which is 457342ACAE32, and sends the packet and the address to the data link layer for transmission. The cache table remains the same.

Example 8.3

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 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. The new cache table is shown in Table 8.6.

Cache Table:

Table 8.6 *Updated cache table for Example 8.3*

State	*Queue*	*Attempt*	*Time-Out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

Example 8.4

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. The new cache table is shown in Table 8.7.

Cache Table:

Table 8.7 *Updated cache table for Example 8.4*

State	Queue	Attempt	Time-Out	Protocol Addr.	Hardware Addr.
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
R	18		900	188.11.8.71	E34573242ACA

Example 8.5

Twenty-five seconds later, the cache-control module updates every entry. 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. For each of the three pending entries, the value of the attempts field is incremented by one. After incrementing, the attempts value for one entry (the one with IP address 201.11.56.7) is more than the maximum; the state is changed to FREE, the queue is deleted, and an ICMP message is sent to the original destination (see Chapter 9). See Table 8.8.

Cache Table:

Table 8.8 *Updated cache table for Example 8.5*

State	*Queue*	*Attempt*	*Time-Out*	*Protocol Addr.*	*Hardware Addr.*
R	5		840	180.3.6.1	ACAE32457342
P	2	3		129.34.4.8	
F					
R	8		390	114.5.7.89	457342ACAE32
P	12	2		220.55.5.7	
P	23	2		116.1.7.22	
F					
R	18		875	188.11.8.71	E34573242ACA