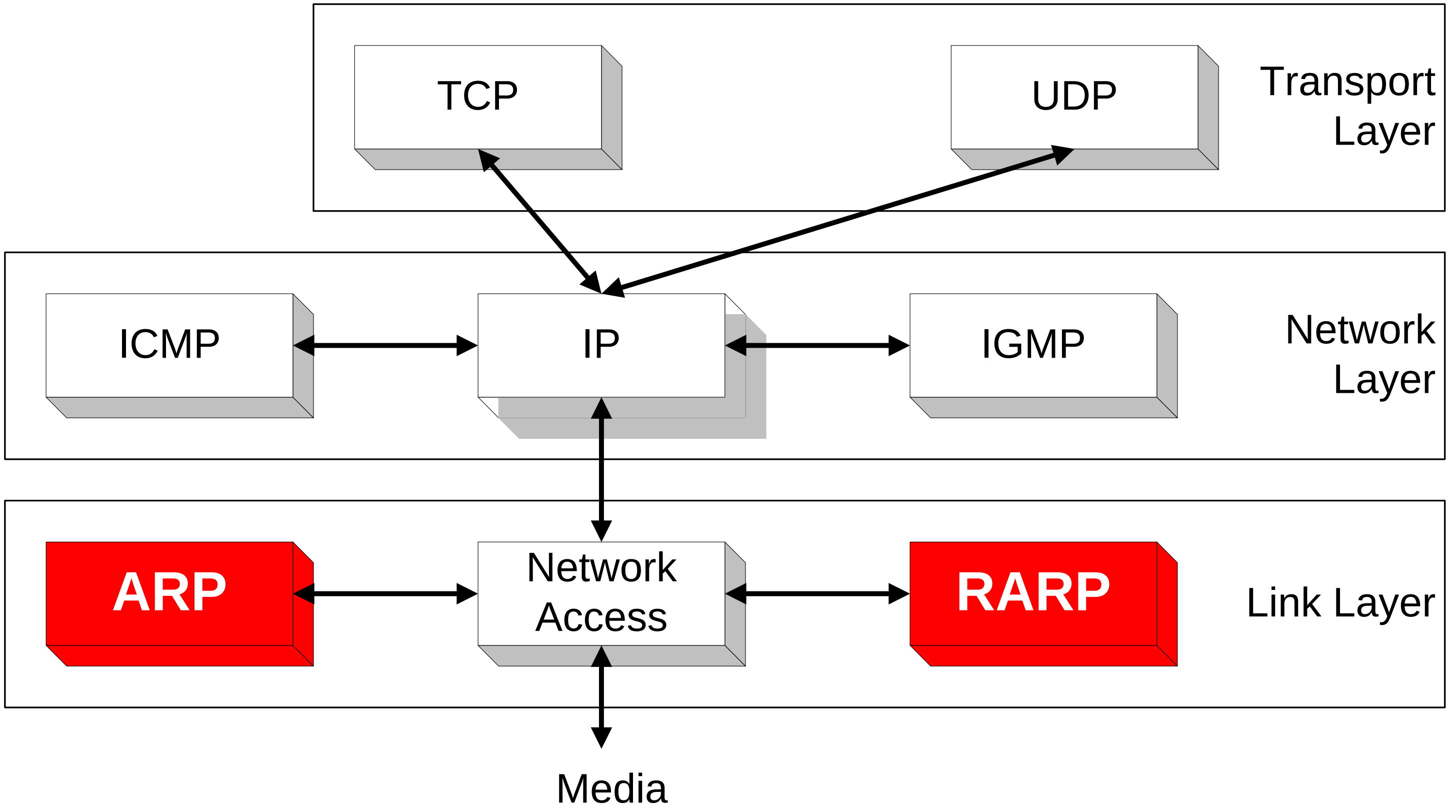
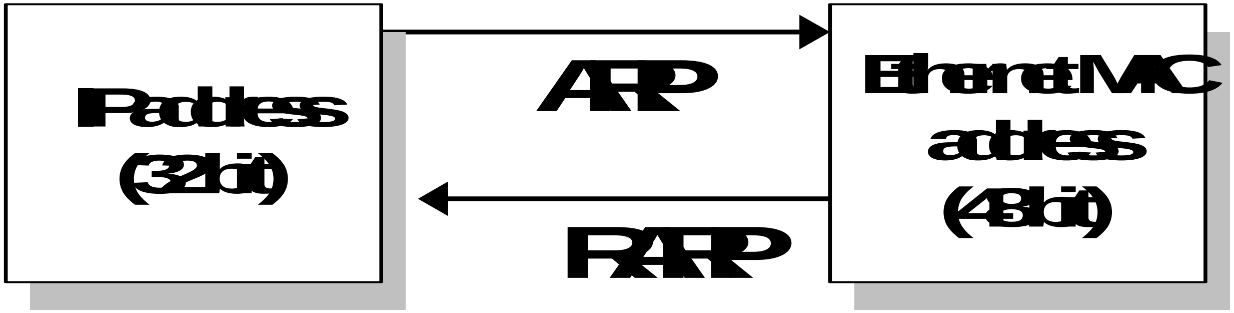
Address Resolution Protocol (ARP)

Overview


TCP
UDP
Transport Layer
ICMP
IP
IGMP
Network Layer
ARP
Network Access
RARP
Link Layer
Media

ARP-RARP



IP address (32 bit)

ARP

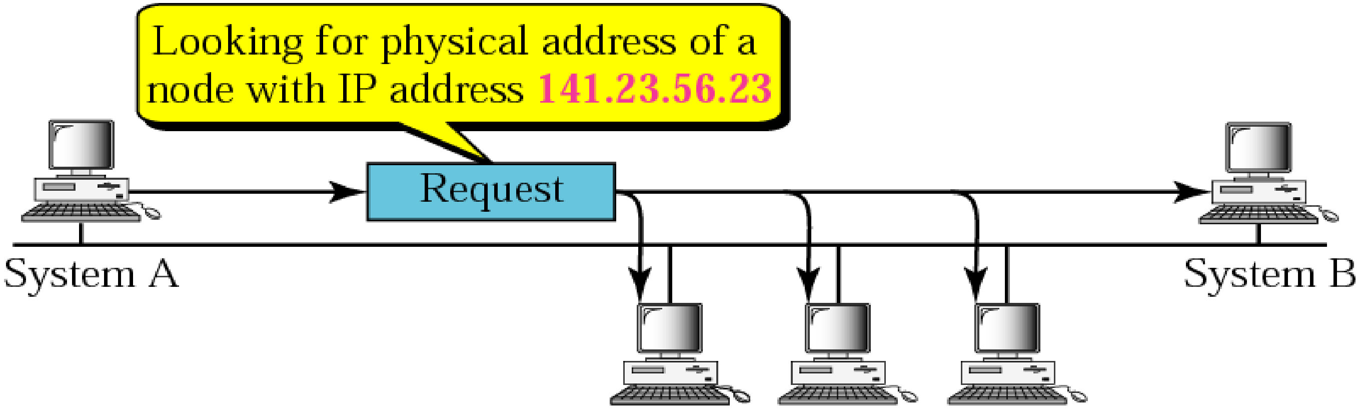
RARP

Ethernet MAC address (48 bit)

ARP

- ARP associates an IP address with its physical address
- On a LAN, each device on a link is identified by a physical or station address that is usually imprinted on the NIC
- Logical address to physical address translation can be done
  - statically (not practical) or
  - dynamically (with ARP)
- ARP Specification: RFC 826

ARP Operation



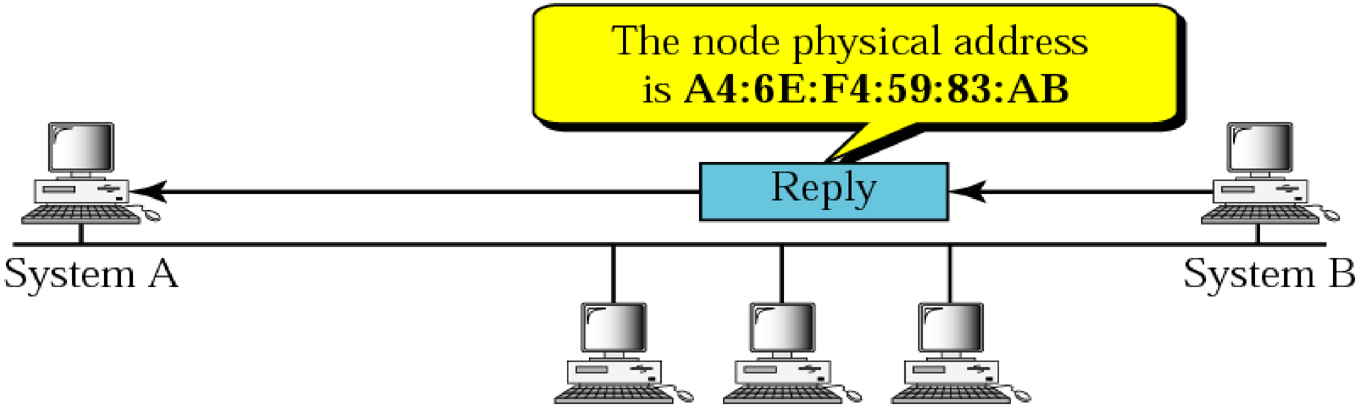
Looking for physical address of a node with IP address 141.23.56.23

Request

System A

System B

a. ARP request is broadcast



The node physical address is **A4:6E:F4:59:83:AB**

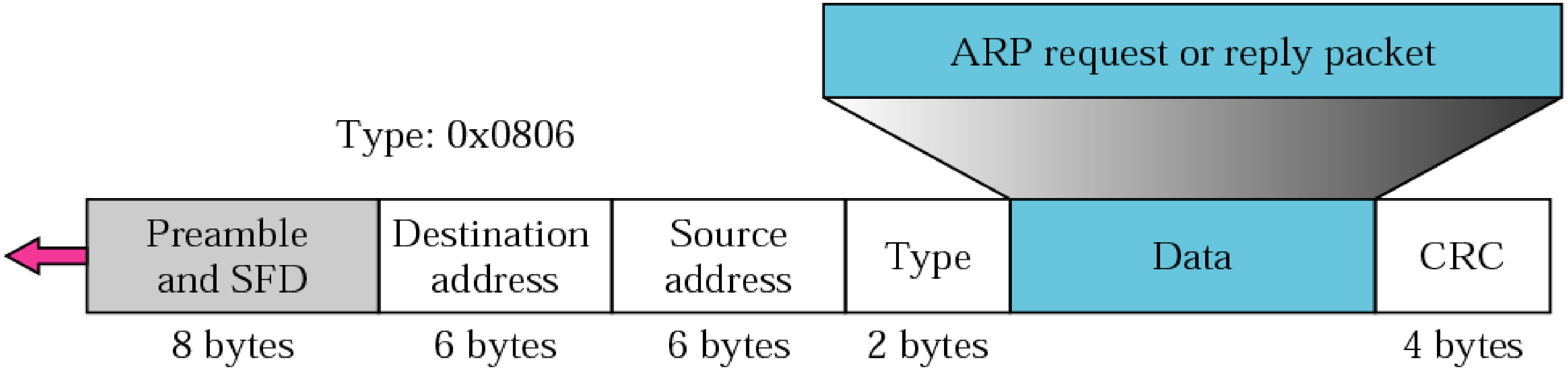
Reply

System A

System B

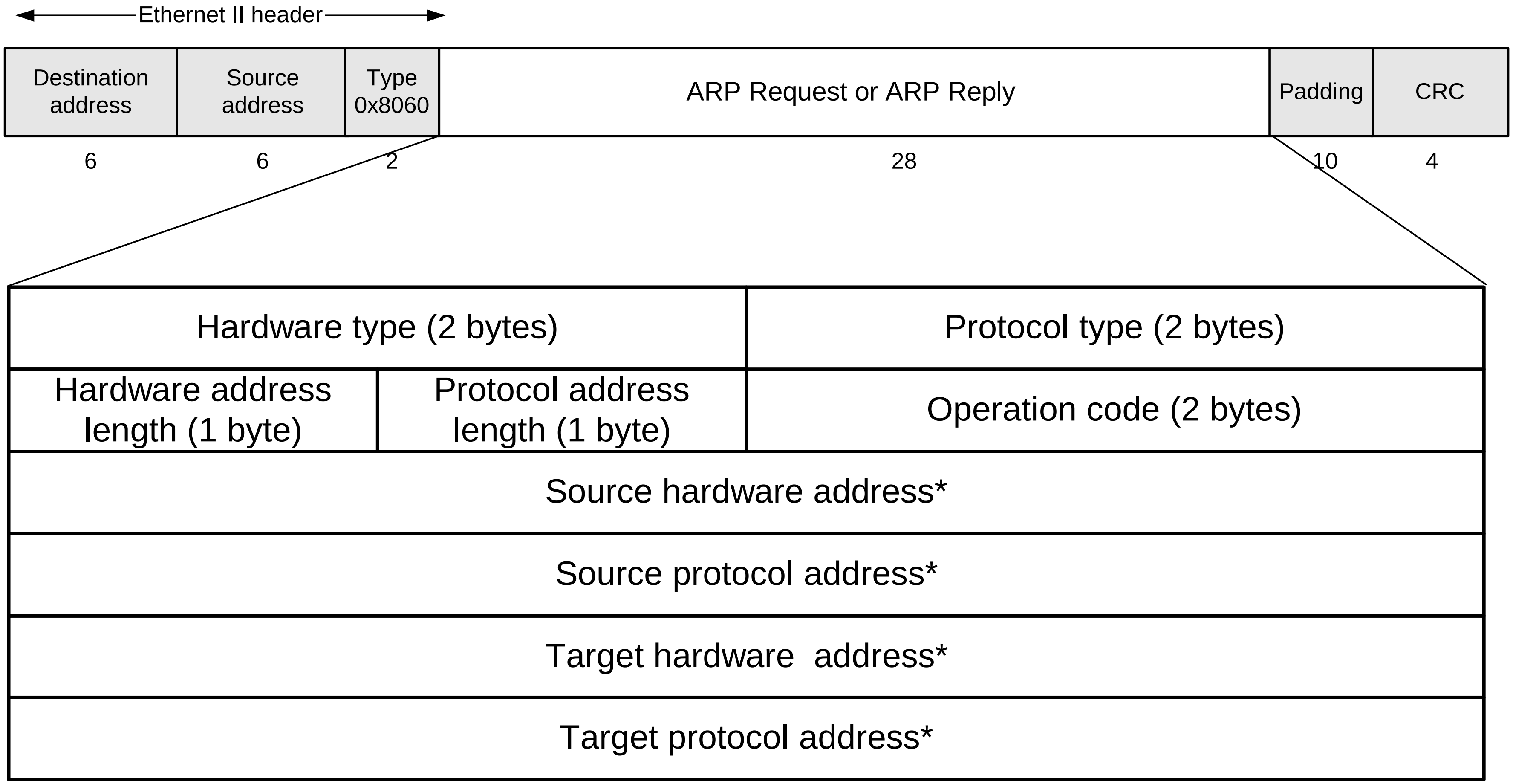
b. ARP reply is unicast

Encapsulation of ARP Packets



- The ARP packet is encapsulated within an Ethernet packet
- Frame Type field specifies the type of data that follows
  - For an ARP request or an ARP reply this field is 0x0806

ARP Packet Format



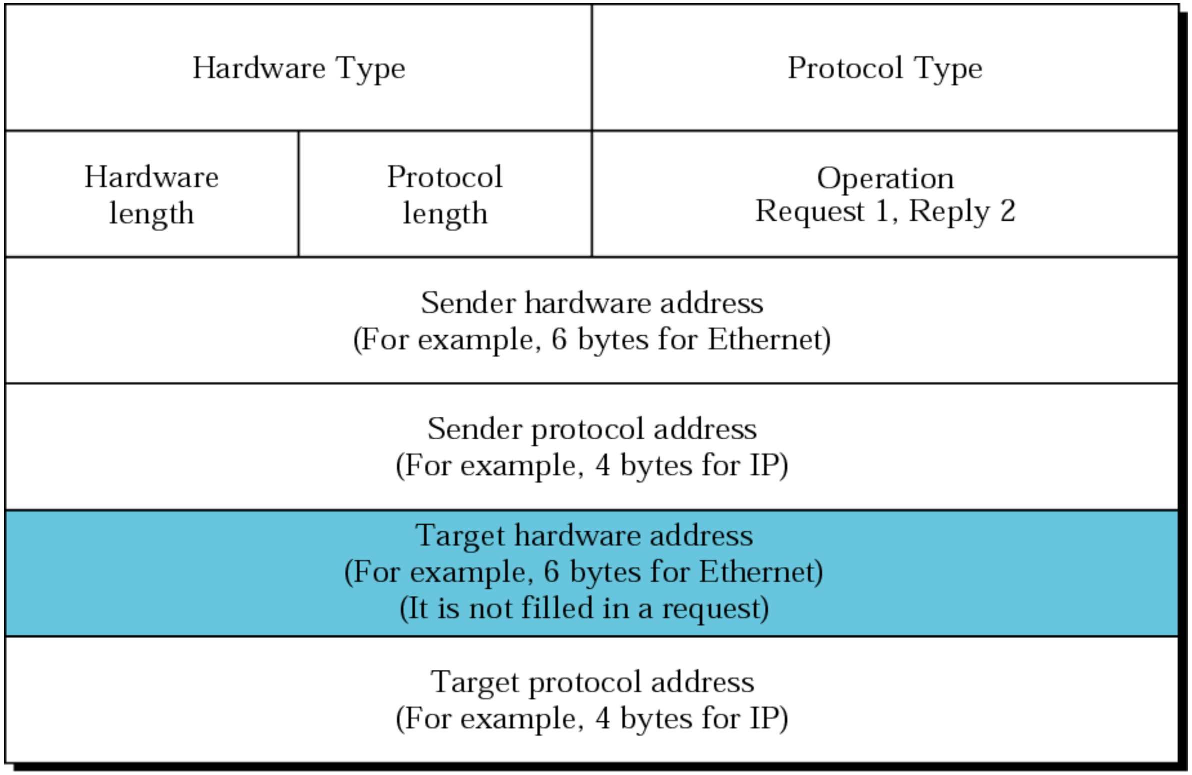
Ethernet II header

Destination address	Source address	Type 0x8060	ARP Request or ARP Reply	Padding	CRC
6	6	2	28	10	4

Hardware type (2 bytes)		Protocol type (2 bytes)
Hardware address length (1 byte)	Protocol address length (1 byte)	Operation code (2 bytes)
Source hardware address*		
Source protocol address*		
Target hardware address*		
Target protocol address*		

* Note: The length of the address fields is determined by the corresponding address length fields

ARP Packets

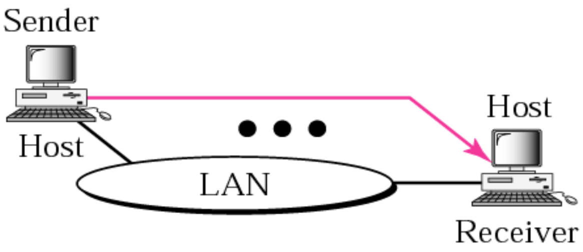


Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

- Hardware Type
  - Ethernet is type 1
- Protocol Type
  - IPv4=0x0800
- Hardware Length
  - Length of Ethernet Address (6)
- Protocol Length
  - Length of IPv4 address (4)
- Operation
  - ARP request – 1
  - ARP reply – 2
  - RARP request – 3
  - RARP reply - 4

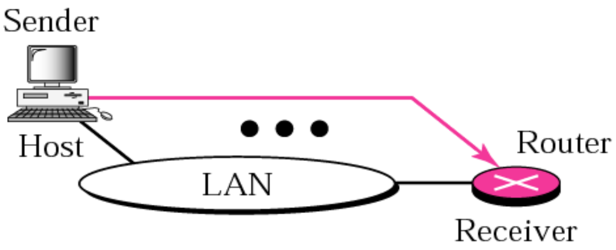
Four Cases using ARP

Target IP address:
Destination address in the IP datagram



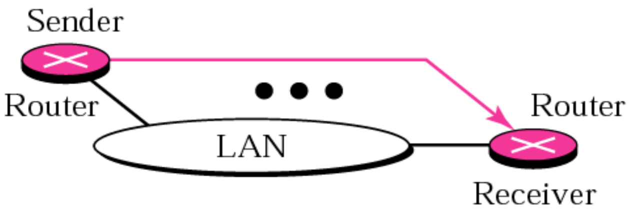
Case 1. A host has a packet to send to another host on the same network.

Target IP address:
IP address of a router



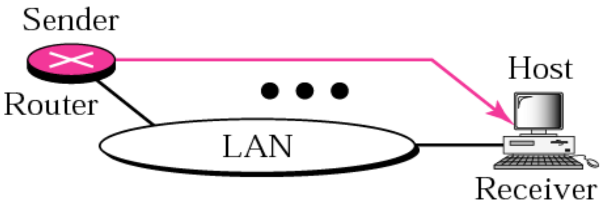
Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to a router.

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.

Target IP address:
Destination address in the IP datagram



Case 4. A router receives a packet to be sent to a host on the same network.

Example

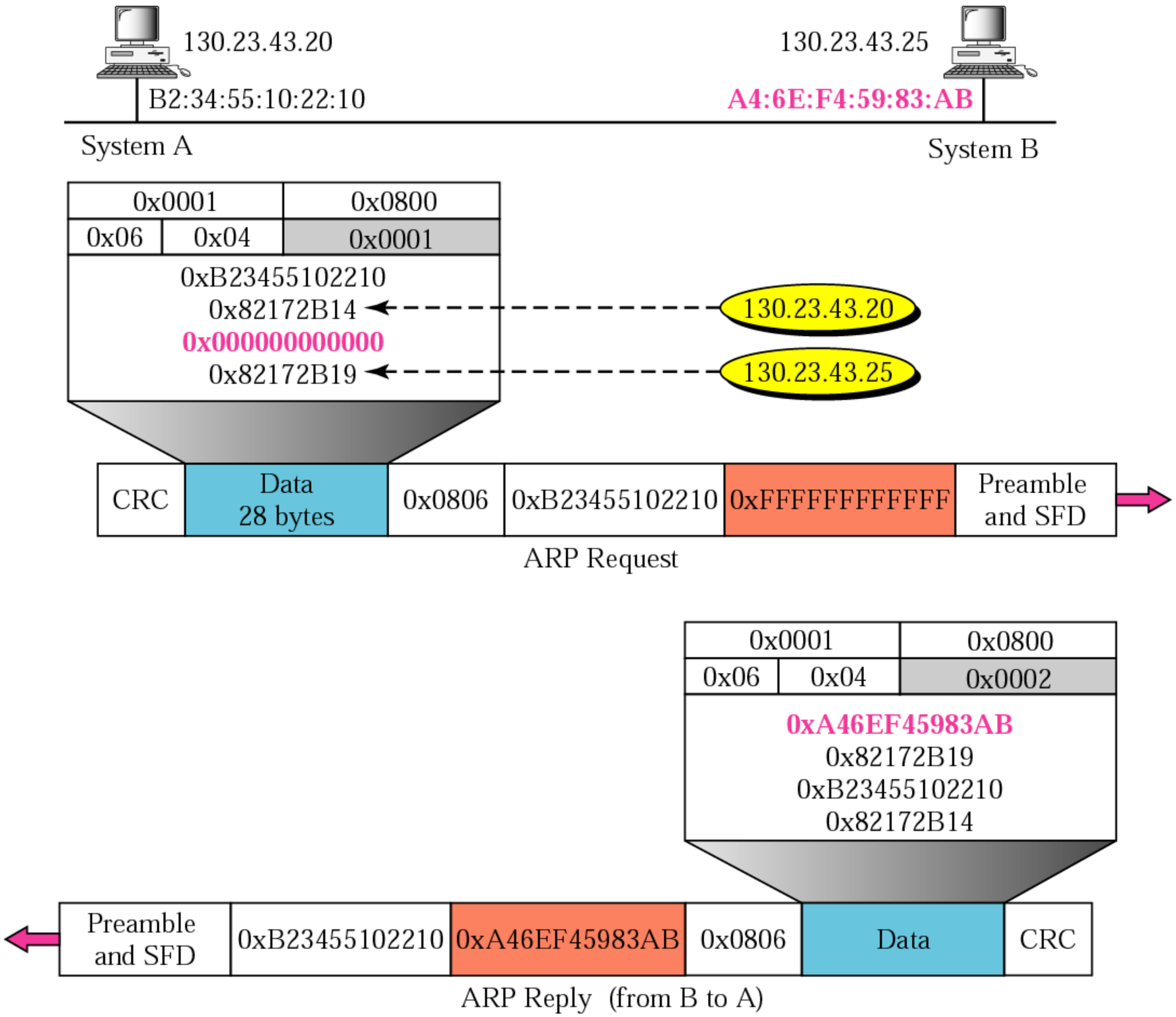
- *ARP Request from Argon:*

Source hardware address:	00:a0:24:71:e4:44
Source protocol address:	128.143.137.144
Target hardware address:	00:00:00:00:00:00
Target protocol address:	128.143.137.1

- *ARP Reply from Router137:*

Source hardware address:	00:e0:f9:23:a8:20
Source protocol address:	128.143.137.1
Target hardware address:	00:a0:24:71:e4:44
Target protocol address:	128.143.137.144

Example



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 (which is unknown to the first host)

The two hosts are on the same Ethernet network

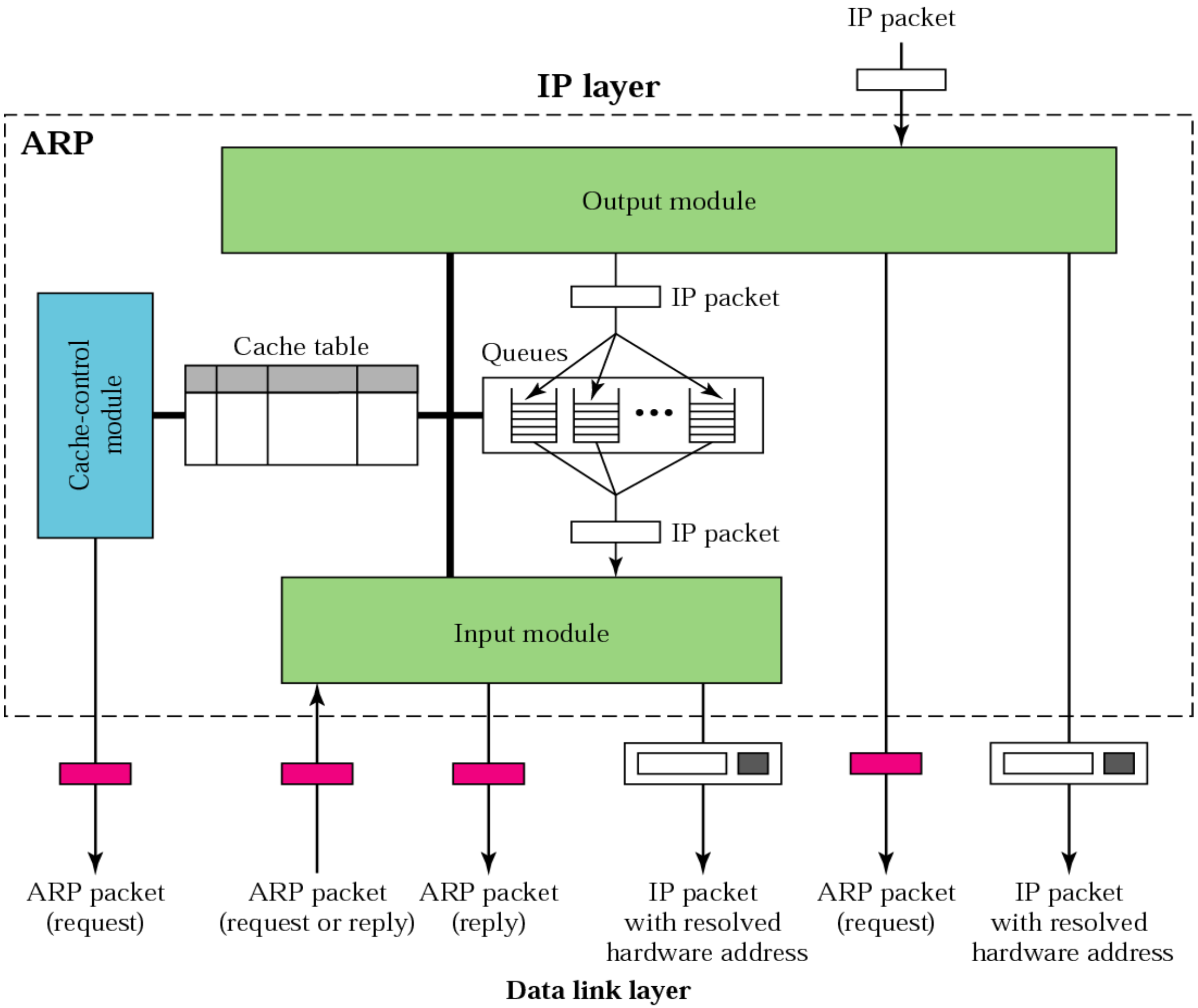
Show the ARP request and reply packets encapsulated in Ethernet frames

ARP Cache

- Essential to the efficient operation of ARP
- Maintained on each host
  - Contains most recent mappings
  - Normal expiration time of an entry is 20 minutes (for complete entry) and 3 minutes (for incomplete entry)
- `arp` command: The `–a` option displays all entries in the cache

```
[root@localhost ~]# arp -a
? (192.168.128.54) at 00:18:FE:A0:AE:57 [ether] on eth0
? (192.168.128.53) at 00:13:21:21:DA:A6 [ether] on eth0
? (192.168.147.139) at 00:19:D1:23:B3:19 [ether] on eth0
[root@localhost ~]#
```

ARP Components

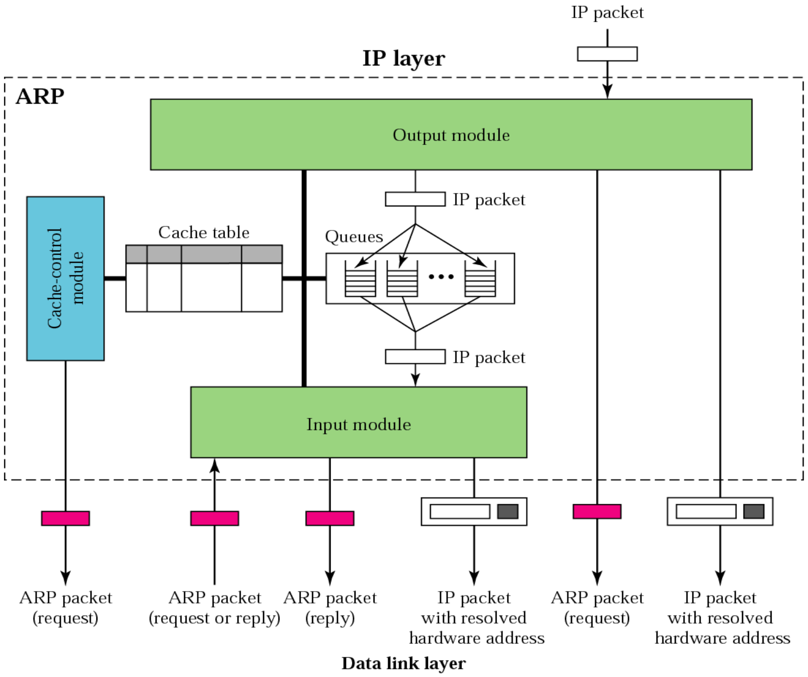


The Cache Table Contents

- State: FREE, PENDING, RESOLVED
- Hardware type: same as ARP field
- Protocol type: same as ARP field
- Hardware length: same as ARP field
- Protocol length: same as ARP field
- Interface number: port number (m0,m1, m2)

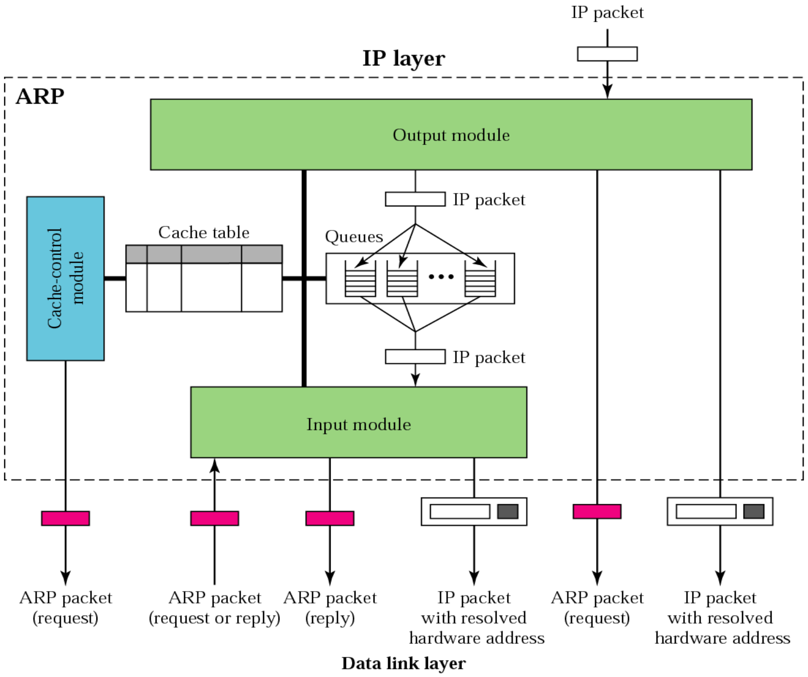
- Queue number: which queue the ARP request is sitting in
- Attempts: how many times have you tried to resolve this address?
- Time-out: how long until this address is tossed out (need the room in cache)
- Hardware address: destination hardware address
- Protocol address: destination IP address

How Does the Cache Work?



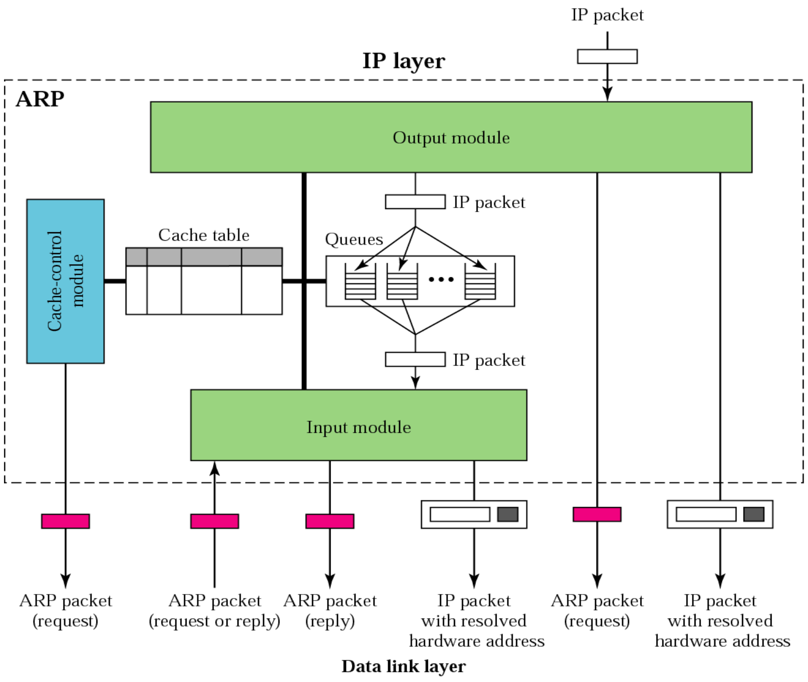
- The output module waits for an IP packet
- Checks the cache for an existing entry
  - If entry found and state RESOLVED, we already have this MAC address
    - Send the packet to the dest
  - If entry found and state PENDING, packet waits until dest hard addr found
  - If no entry found, output module places this request in queue, and a new entry is placed in cache with state PENDING and ATTEMPTS set to 1
    - An ARP request is then broadcast

How Does the Cache Work?



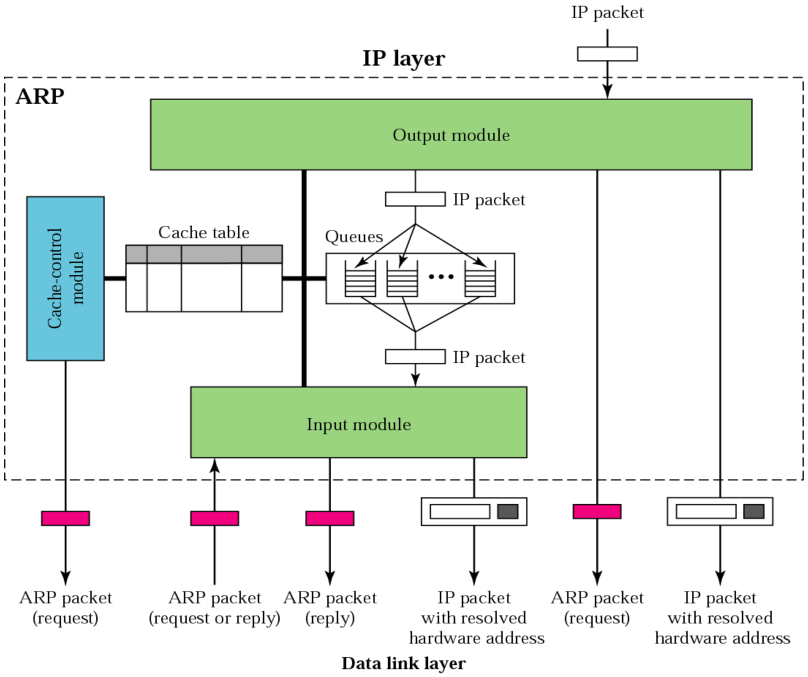
- The input module waits until an ARP request or reply arrives
- For an ARP reply packet -
  - Module checks the cache for this entry
  - If entry is found and state is PENDING, module updates entry’s target hardware address, changes state to RESOLVED, and sets the TIME-OUT value

How Does the Cache Work?



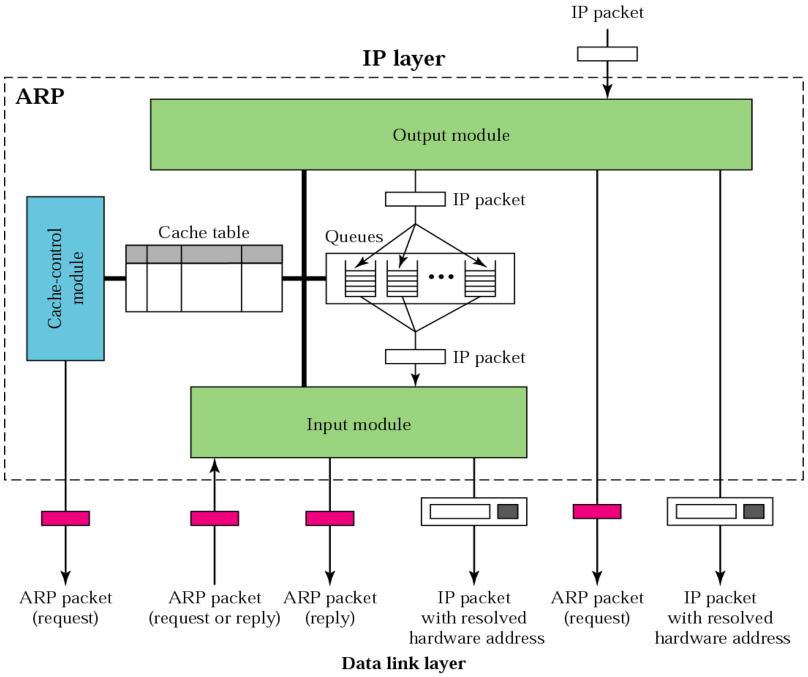
- If entry is found and state RESOLVED, module still updates the entry (target hard addr could have changed) and the TIME-OUT value reset
- If entry not found, module creates a new entry. State is set to RESOLVED and TIME-OUT is set

How Does the Cache Work?



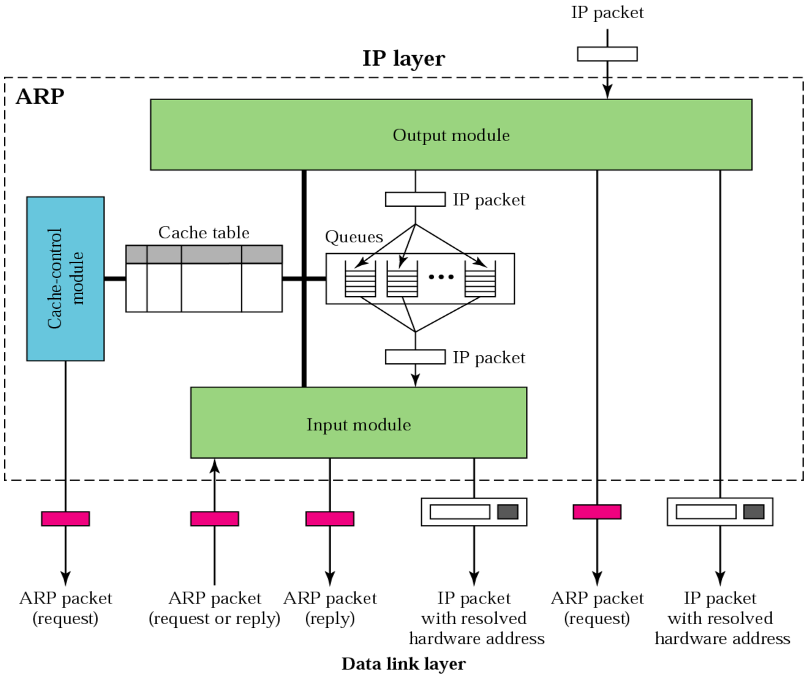
- If arrived ARP packet is a Request, the module immediately creates an ARP Reply message and sends it back to sender
  - The ARP reply packet is created by changing the operation field from Request to Reply
  - Filling in the target hardware address

How Does the Cache Work?



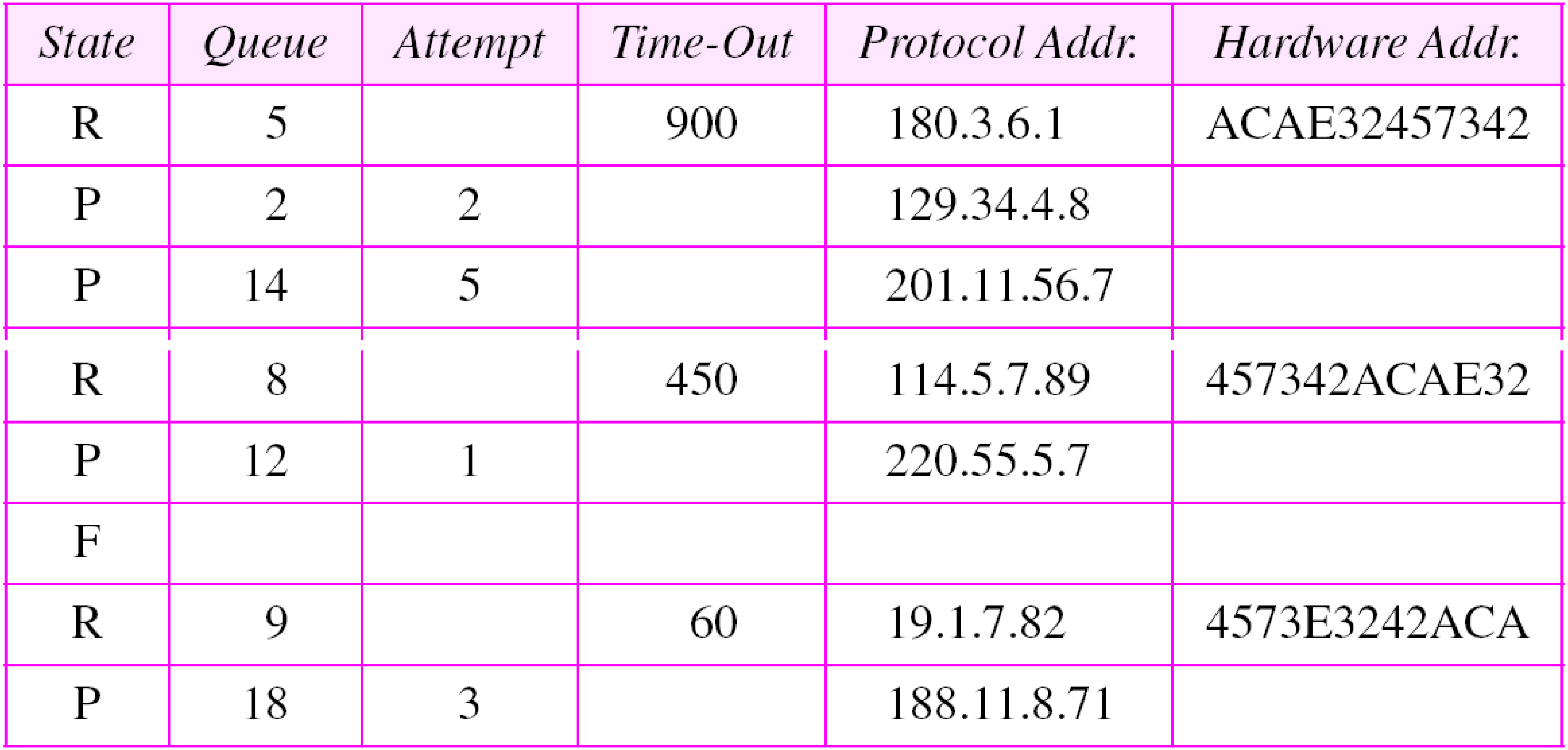
- The cache-control module periodically checks each cache entry
  - If entry's state is FREE, skips it
  - If entry's state is PENDING, Attempts field is incremented by 1
    - This value greater than max? Toss this entry (and mark entry as FREE)
    - Less than max? Send another ARP request

How Does the Cache Work?



- If state of entry is RESOLVED, module decrements value of Time-out field accordingly
  - If Time-out field < 0, then remove entry and set state to FREE

Original Cache Table

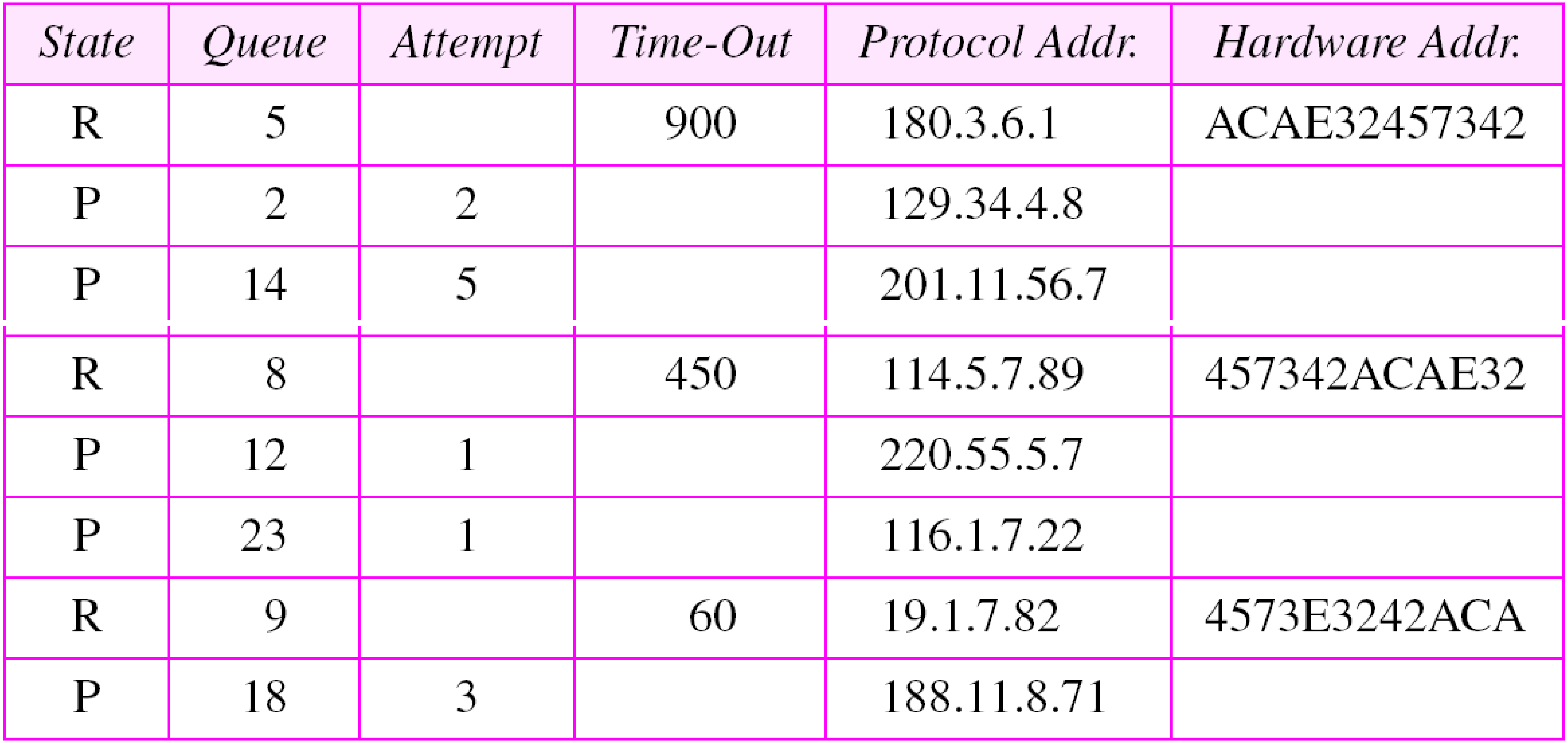


State	*Queue*	*Attempt*	*Time-Out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

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.

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.

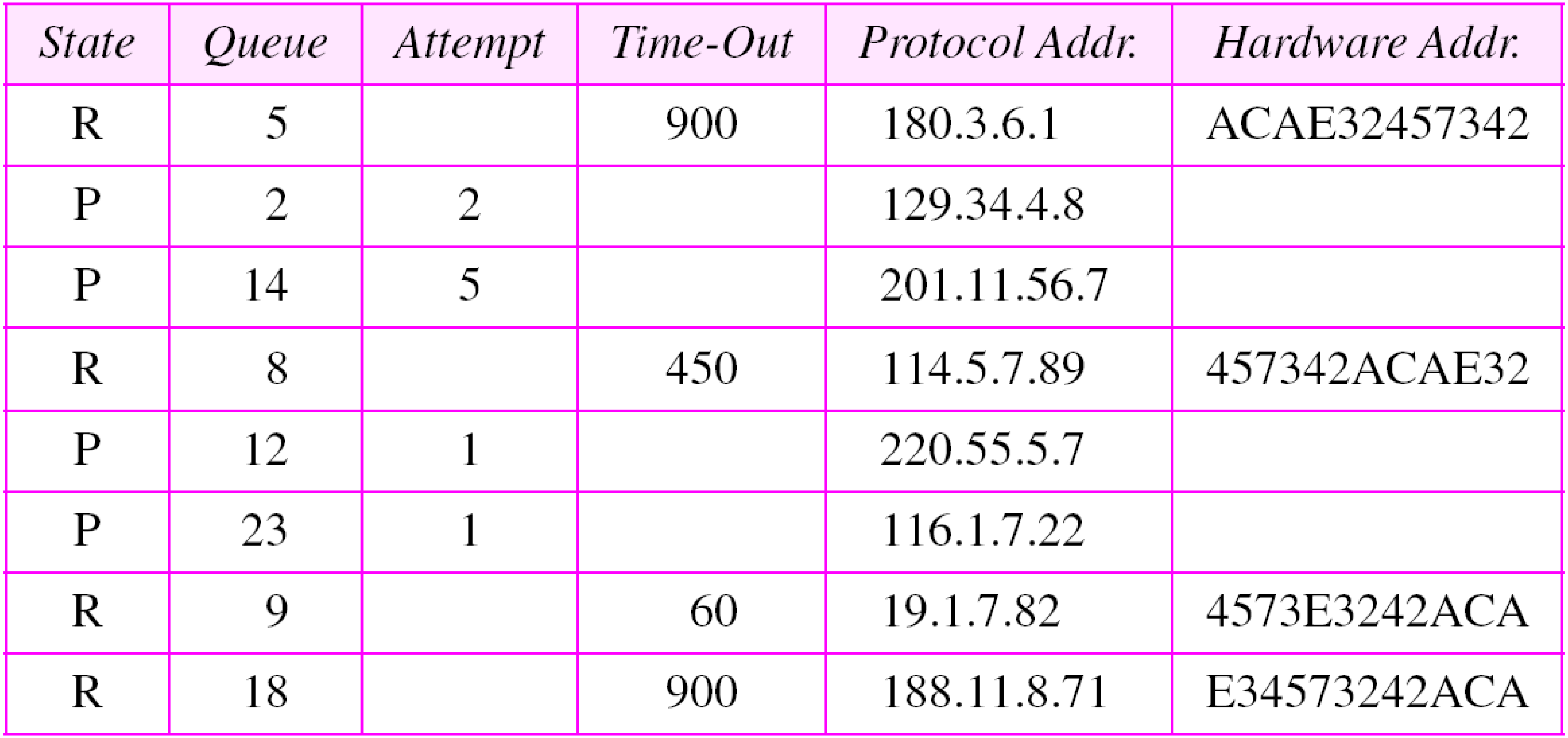
Updated Cache Table



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	

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.

Updated Cache Table



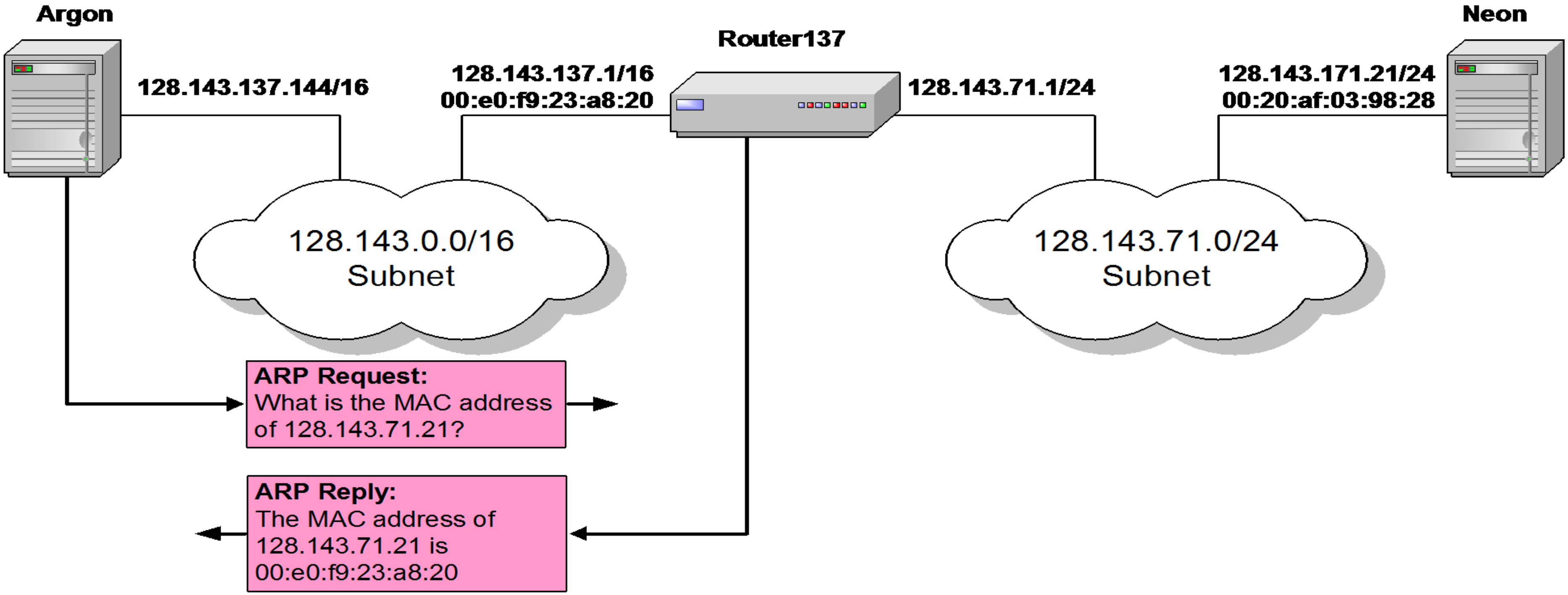
State	*Queue*	*Attempt*	*Time-Out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
R	18		900	188.11.8.71	E34573242ACA

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 increased by 1.

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

Proxy ARP

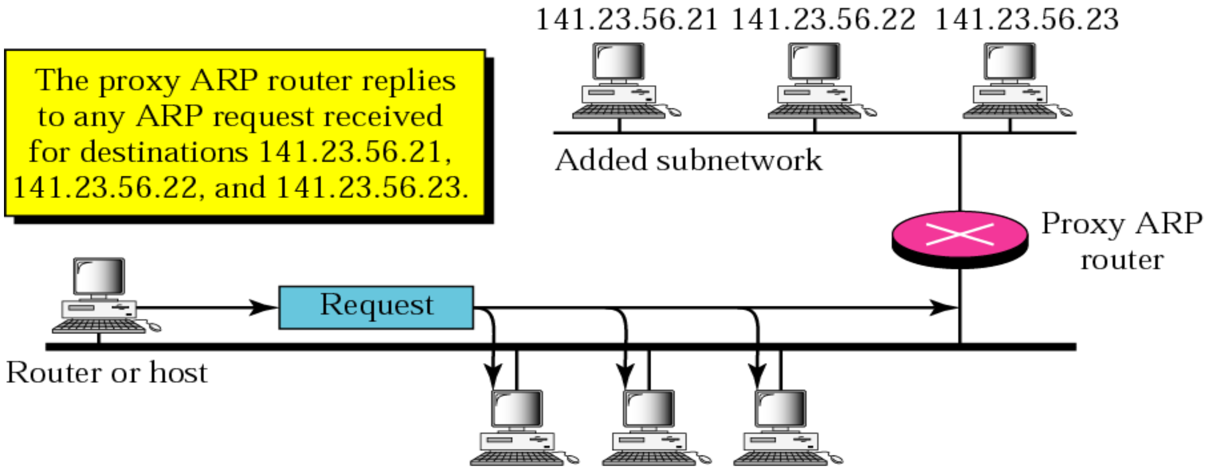
- **Proxy ARP:** Host or router responds to ARP Request that arrives from one of its connected networks for a host that is on another of its connected networks.



ARP Request:
What is the MAC address of 128.143.71.21?

ARP Reply:
The MAC address of 128.143.71.21 is 00:e0:f9:23:a8:20

Proxy ARP


141.23.56.21
141.23.56.22
141.23.56.23
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.
Added subnetwork
Proxy ARP router
Request
Router or host

- The proxy ARP replies with its own MAC address
- When the packet arrives, the router delivers it to the appropriate host
- Also called *promiscous ARP* or *ARP hack*
  - Can hide two physical networks from each other, with a router between the two

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.

- On some systems (including Linux) a host periodically sends ARP Requests for all addresses listed in the ARP cache. This refreshes the ARP cache content, but also introduces traffic.

- Gratuitous ARP Requests: A host sends an ARP request for its own IP address:
  - Useful for detecting if an IP address has already been assigned.

Vulnerabilities of ARP

1. Since ARP does not authenticate requests or replies, ARP Requests and Replies can be forged
2. ARP is stateless: ARP Replies can be sent without a corresponding ARP Request
3. According to the ARP protocol specification, a node receiving an ARP packet (Request or Reply) must update its local ARP cache with the information in the source fields, if the receiving node already has an entry for the IP address of the source in its ARP cache. (This applies for ARP Request packets and for ARP Reply packets)

Typical exploitation of these vulnerabilities:

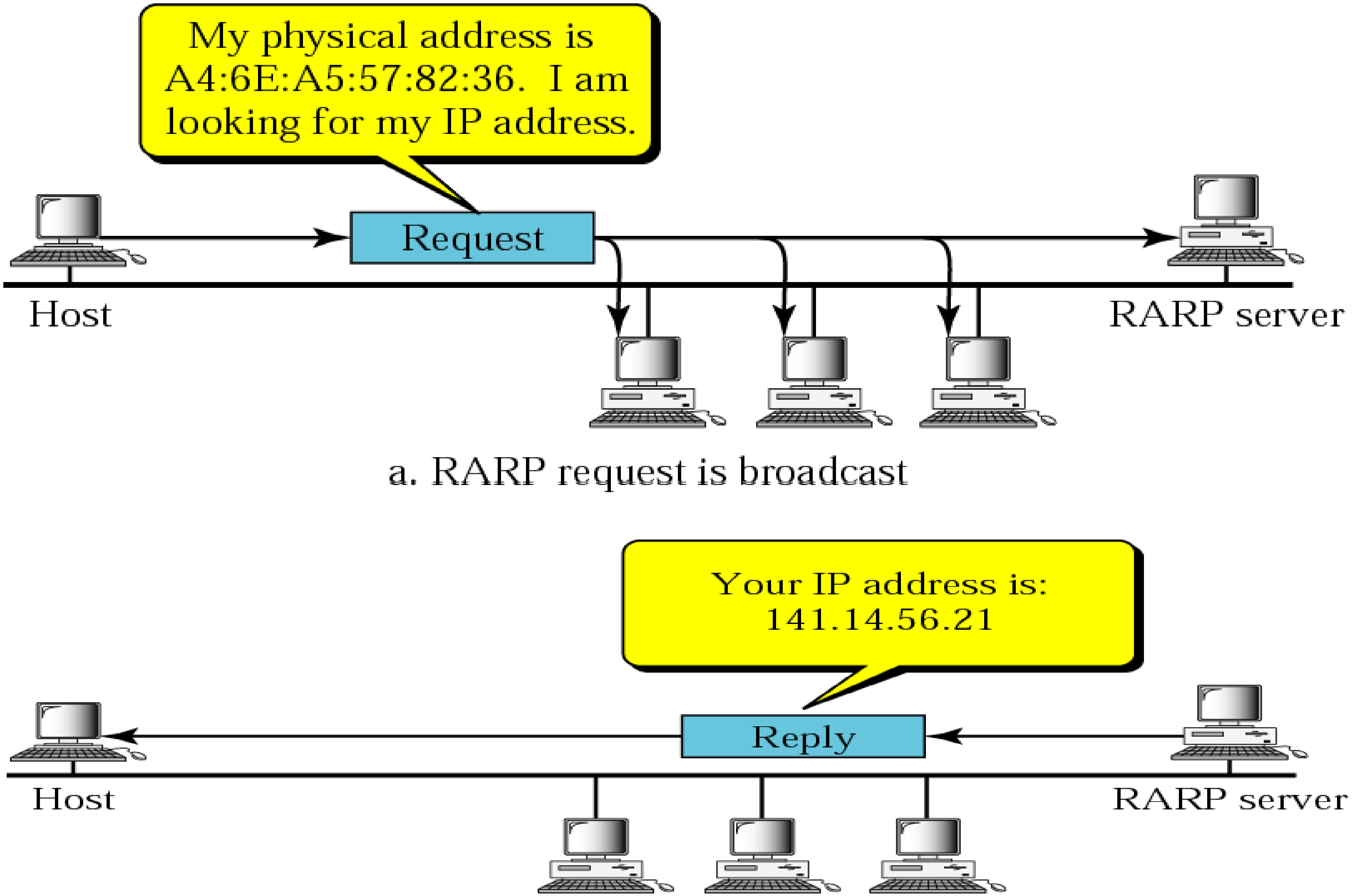
- A forged ARP Request or Reply can be used to update the ARP cache of a remote system with a forged entry (ARP Poisoning)
- This can be used to redirect IP traffic to other hosts

RARP

RARP finds the logical address for a machine that only knows its physical address

The RARP request packets are broadcast;
the RARP reply packets are unicast.

RARP



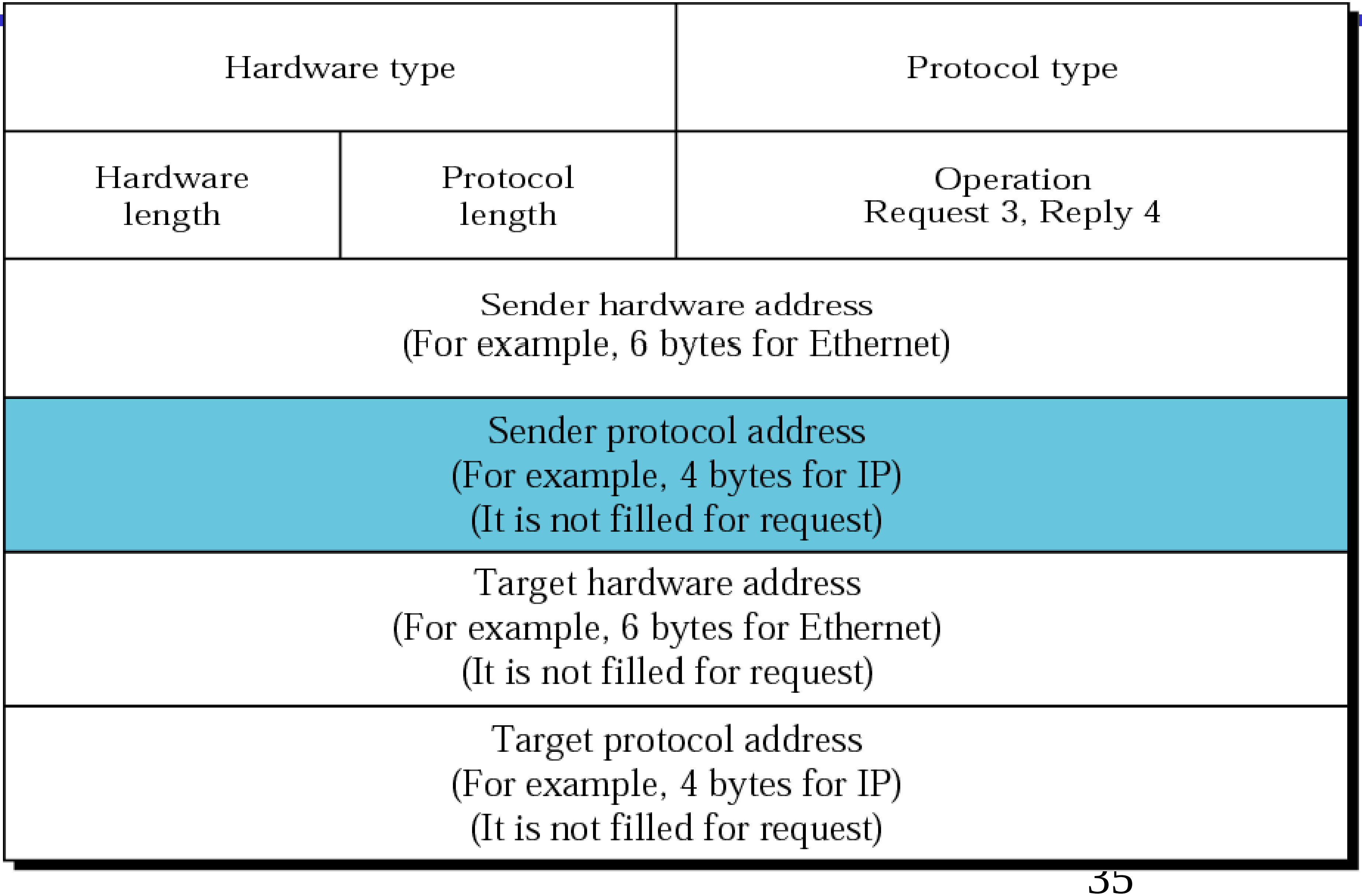
My physical address is A4:6E:A5:57:82:36. I am looking for my IP address.

a. RARP request is broadcast

Your IP address is: 141.14.56.21

b. RARP reply is unicast

RARP Packet



<table>
<tr><td colspan="2">Hardware type</td><td>Protocol type</td></tr>
<tr><td>Hardware length</td><td>Protocol length</td><td>Operation
Request 3, Reply 4</td></tr>
<tr><td colspan="3">Sender hardware address
(For example, 6 bytes for Ethernet)</td></tr>
<tr><td colspan="3">Sender protocol address
(For example, 4 bytes for IP)
(It is not filled for request)</td></tr>
<tr><td colspan="3">Target hardware address
(For example, 6 bytes for Ethernet)
(It is not filled for request)</td></tr>
<tr><td colspan="3">Target protocol address
(For example, 4 bytes for IP)
(It is not filled for request)</td></tr>
</table>