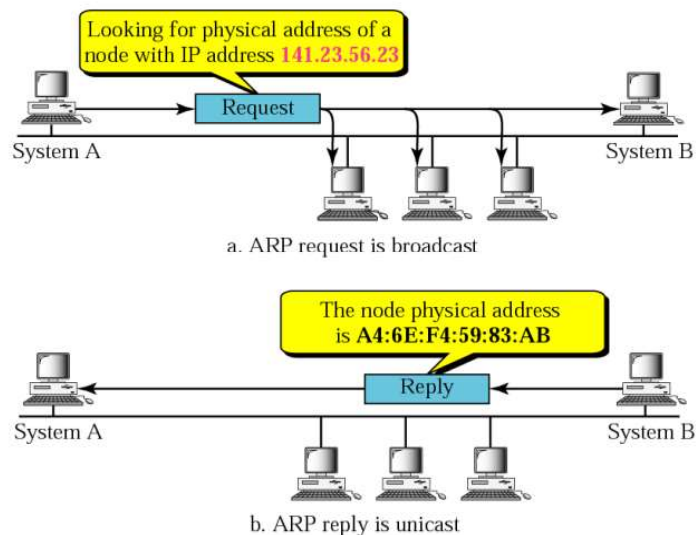


ARP associates an IP address with its physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address that is usually imprinted on the NIC.

Logical address to physical address translation can be done statically (not practical) or dynamically (with ARP).



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)		

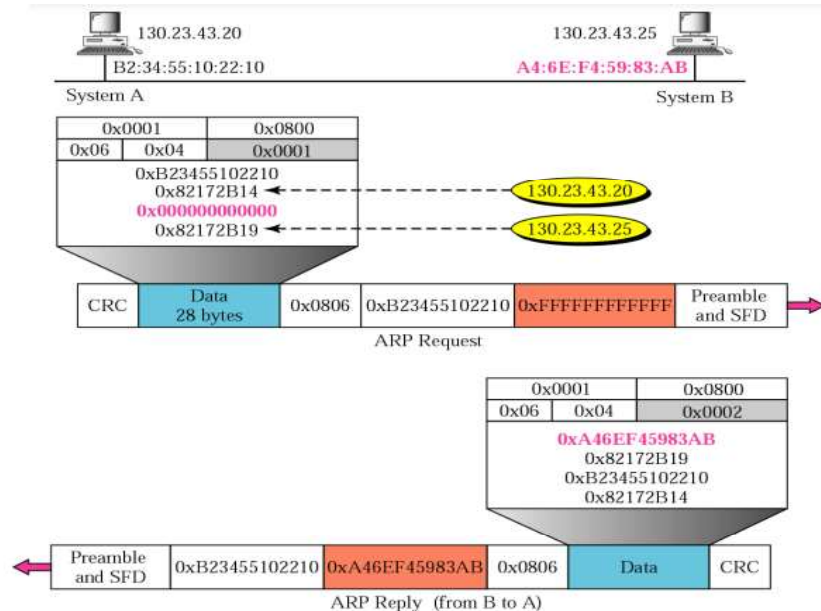
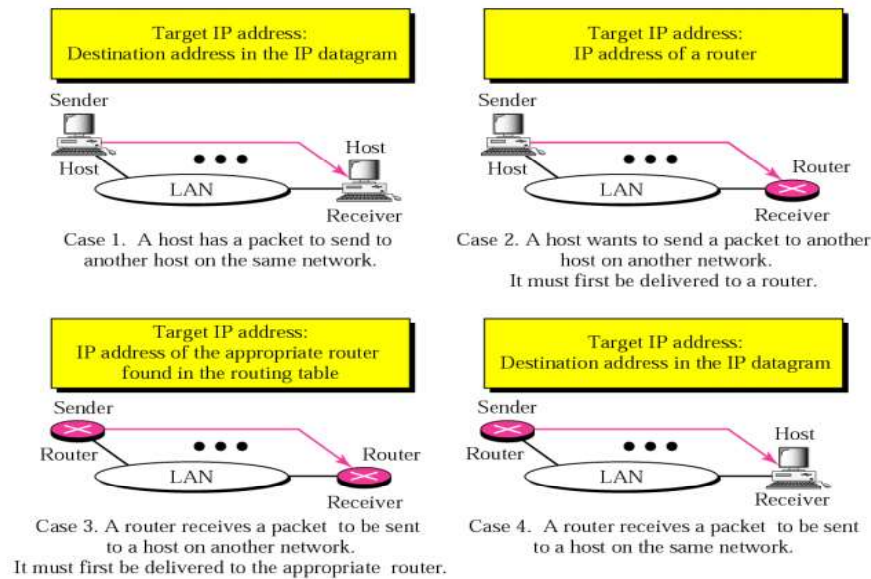
Hardware Type - Ethernet is type 1

Protocol Type- IPv4=x0800

Hardware Length: length of Ethernet Address (6)

Protocol Length: length of IPv4 address (4)

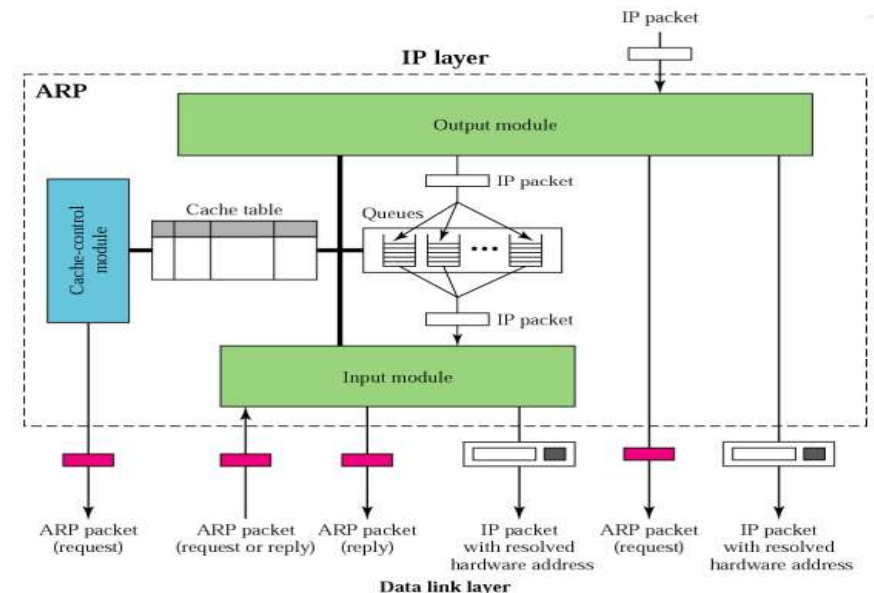
ARP usage examples



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 internal components



The Cache Table Contents

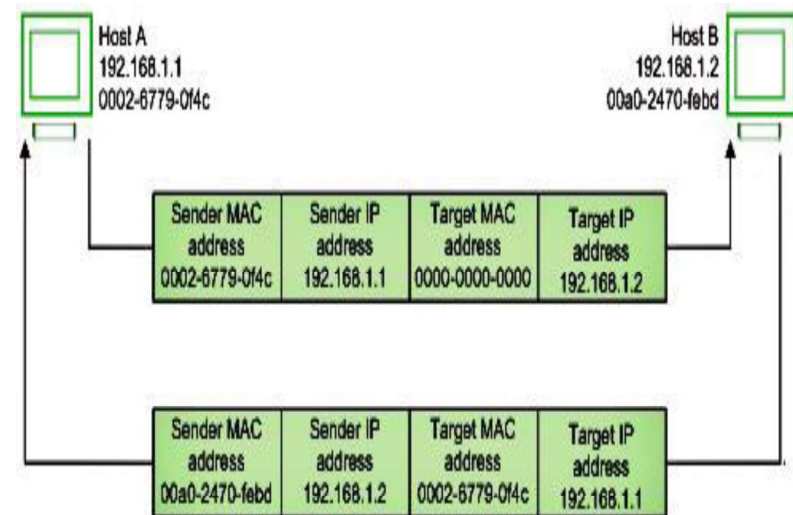
- State: FREE, PENDING, RESOLVED
- Hardware type: same as ARP field
- Protocol type: same as ARP field
- Hardware length: same as ARP field
- Protocol length: same as ARP field
- Interface number: port number (m0,m1, m2)

The Cache Table Contents

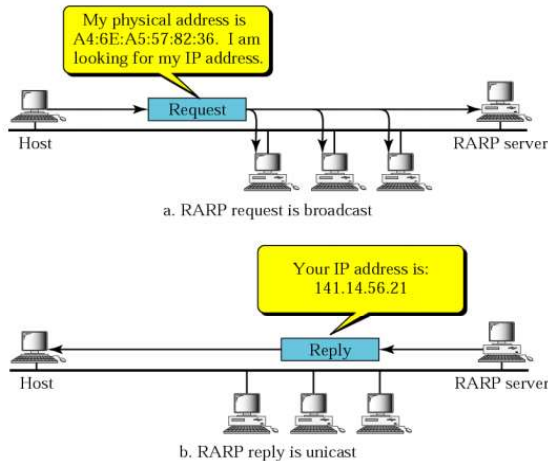
- Queue number: which queue the ARP request is sitting in
- Attempts: how many times have you tried to resolve this address?
- Time-out: how long until this address is tossed out (need the room in cache)
- Hardware address: destination hardware address
- Protocol address: destination IP address

How Does the Cache Work?

- The **output module** waits for an IP packet with a request
- Checks the cache for an existing entry
- If entry found and state RESOLVED, we already have this MAC address
- If entry found and state PENDING, packet waits until dest hard addr found



RARP operation



How Does the Cache Work?

- 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
- 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 hardware address 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?

- Now the module checks to see if arrived ARP packet is a Request. If it is, the module immediately creates an ARP Reply message and sends it back to sender.
- 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

RARP

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	

Hardware type		Protocol type
Hardware length	Protocol length	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)		